LEVERAGING MACHINE LEARNING TO IDENTIFY QUALITY ISSUES IN THE MEDICAID CLAIM ADJUDICATION PROCESS

A Dissertation

Presented to

The College of Graduate and Professional Studies

The College of Technology

Indiana State University

Terre Haute, Indiana

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Technology Management

---

by

Cyrus Hoseini

December 2020

Keywords: Machine Learning, Supervised Learning, Technology Management, Quality Management, Medicaid Claim Adjudication, Medicaid Management Information System

CURRICULUM VITAE

CYRUS HOSEINI

EDUCATION

- Ph.D. in Technology Management, 2020 Indiana State University | Terre Haute, IN
- Master of Business Administration, 2003 Sharif University of Technology | Tehran, Iran
- Bachelor of Science in Applied Mathematics (Operations Research), 1997 Sharif University of Technology | Tehran, Iran

PROFESSIONAL EXPERIENCE

- QualMine, Inc., V.P. Advisory Services, 02.2012–Present | El Dorado Hills, CA
- CGI Group, Inc., Director of Consulting (Quality Management Director for California Medicaid Management Information System), 01.2012–02.2019 | Sacramento, CA
- Doctoral Fellow, 11.2017–08.2018, Indiana State University | Terre Haute, IN
- Maximus Inc. Quality Assurance Director, 07.2008–01.2012 | Rancho Cordova, CA
- US Voting member, US TAG to ISO/TC 176
- Senior Member, American Society for Quality (ASQ), 2008-Present
- Technical Paper Reviewer, American Society for Quality (ASQ) World Conference in Quality and Improvement, 2008 through 2021

PUBLICATIONS AND PRESENTATIONS

- Hoseini, C. (2020), Augment your Quality Systems with Machine Learning, ASQ World Conference in Quality and Improvement, May 5, 2020
- Hoseini, C. (2018), Augmenting your Quality Systems using Big Data Analytics and Machine Learning: challenges for quality practitioners, ASQ Statistics Digest
- Hoseini, C. (2017). Quality Management Analytics: Apply Big Data Analytics to renovate your Quality Systems. ASQ European Quality Conference. Nov. 6 – 7, 2017

COMMITTEE MEMBERS

Committee Chair: Dr. M Affan Badar

Professor, Dept. of Applied Engineering and Technology Management

Indiana State University

Committee Member: Dr. A. Mehran Shahhosseini

Professor, Dept. of Applied Engineering and Technology Management

Indiana State University

Committee Member: Dr. Christopher Kluse

Assistant Professor, College of Technology, Architecture and Applied Engineering

Bowling Green State University

ABSTRACT

Medicaid is the largest health insurance in the U.S. It provides health coverage to over 68 million individuals, costs the nation over $600 billion a year, and subject to improper payments (fraud, waste, and abuse) or inaccurate payments (claim processed erroneously). Medicaid programs partially use Fee-For-Services (FFS) to provide coverage to beneficiaries by adjudicating claims and leveraging traditional inferential statistics to verify the quality of adjudicated claims. These quality methods only provide an interval estimate of the quality errors and are incapable of detecting most claim adjudication errors, potentially millions of dollar opportunity costs. This dissertation studied a method of applying supervised learning to detect erroneous payment in the entire population of adjudicated claims in each Medicaid Management Information System (MMIS), focusing on two specific claim types: inpatient and outpatient. A synthesized source of adjudicated claims generated by the Centers for Medicare & Medicaid Services (CMS) was used to create the original dataset. Quality reports from California FFS Medicaid were used to extract the underlying statistical pattern of claim adjudication errors in each Medicaid FFS and data labeling utilizing the goodness of fit and Anderson-Darling tests. Principle Component Analysis (PCA) and business knowledge were applied for dimensionality reduction resulting in the selection of sixteen (16) features for the outpatient and nineteen (19) features for the inpatient claims models. Ten (10) supervised learning algorithms were trained and tested on the labeled data: Decision tree with two configurations - Entropy and Gini, Random forests with two configurations - Entropy and Gini, Naïve Bayes, K Nearest Neighbor, Logistic Regression, Neural Network, Discriminant Analysis, and Gradient Boosting. Five (5) cross-validation and event-based sampling were applied during the training process (with oversampling using SMOTE method and stratification within oversampling). The prediction power (Gini importance) for the selected features were measured using the Mean Decrease in Impurity (MDI) method across three algorithms. A one-way ANOVA and Tukey and Fisher LSD pairwise comparisons were conducted. Results show that the Claim Payment Amount significantly outperforms the rest of the prediction power (highest Mean F-value for Gini importance at the $\alpha = 0.05$ significance) for both claim types. Finally, all algorithms' recall and F1-score were measured for both claim types (inpatient and outpatient) and with and without oversampling. A one-way ANOVA and Tukey and Fisher LSD pairwise comparisons were conducted. The results show a statistically significant difference in the algorithm's performance in detecting quality issues in the outpatient and inpatient claims. Gradient Boosting, Decision Tree (with various configurations and sampling strategies) outperform the rest of the algorithms in recall and F1-measure on both datasets. Logistic Regression showing better recall on the outpatient than inpatient data, and Naïve Bays performs considerably better from recall and F1-score on outpatient data. Medicaid FFS programs and consultants, Medicaid administrators, and researchers could use this study to develop machine learning models to detect quality issues in the Medicaid FFS claim datasets at scale, saving potentially millions of dollars.

DEDICATION

This dissertation is dedicated to my wife, Dr. Nusheen Reykandeh. This dissertation would not have been possible without her love, support, and sacrifices. She has been with me throughout my long Ph.D. journey and has made them the best years of my life.

ACKNOWLEDGMENTS

Firstly, I would like to express my sincere gratitude to my advisor Prof. M. Affan Badar, for the continuous support of my Ph.D. study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. Besides my advisor, I would like to thank the rest of my dissertation committee: Dr. A. Mehran Shahhosseini and Dr. Christopher Kluse, for their insightful inputs, detailed comments, and encouragement, which guided me throughout my journey. I also would like to thank the faculty and administrative personnel of the Ph.D. consortium program, especially Marti Mix, for their support.

I also want to thank Dr. Hisham Rana for his great insights, inputs, and supports. I would also like to thank Dr. Shehab Parnianchi for providing valuable information, Dr. Zeinab Kermansaravi, and Mahmood Vahedi for their tremendous help with Python codes, and Rani Ipong with his support in managing data and creating the data pipeline. I also want to thank Rich Davis, who challenged me to come up with a method to use data patterns and train models without using PHI.

Finally, I want to thank my deceased father and father-in-law, Kazem and Ray, for their life-long insights and advisements. I know both of you would consider this step just another milestone in my long journey, and you are right. I missed you, but I know you are in a better place and smiling as you know, I have a long way to go.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The purpose of chapter one is to clarify the importance of the problem at hand, offer some background information, and then explain the problem statement and research questions clearly. The chapter starts with an introduction and some background information about the importance of Medicaid, different Medicaid programs, and the typical quality challenges in the Medicaid claim processing. The problem statement will then be discussed related to leveraging machine learning as an innovative method to detect quality issues in the Medicaid claim adjudication process. After that, the research questions and hypotheses will be formulated, and the goal of this research will be discussed. Finally, a list of the assumptions and limitations of this research will be provided, research methodology will be introduced, and the chapter will be concluded with a list of terms and definitions.

Background

Medicaid is a subsidized healthcare program for low income or individuals with special needs (e.g., disable and senior citizens). Federal and State governments collaboratively fund Medicaid to provide health care services to over 67 million individuals in 50 states and the District of Colombia (Medicaid.gov, 2018). Medicaid cost taxpayers over $570 billion a year (The Henry J. Kaiser Family Foundation, 2018) and subject to improper payments estimated to be over $29 billion (*Testimony of Ann Maxwell Before the United States House of Representatives*, 2017). Medicaid programs are usually divided into two subcategories: Fee-For-

Services (FFS) and the Managed Care Organization (MCO). Both FFS and MCO models typically require a Medicaid claim to be processed either by a health plan or a Fiscal Intermediary (FI). As a result, it is crucial to ensure all Medicaid claims are processed correctly and free from payment errors. In an FFS model, a Medicaid beneficiary goes to a provider to receive a healthcare service. The provider then sends the bill (claim) to a FI contractor to be processed. FI contractors use an elaborate system consisting of various software, hardware, and processes called the Medicaid Management Information System (MMIS) to adjudicate millions of claims.

In the past few decades, FI contractors and MCO health plans have invented and implemented several methods to detect and prevent improper payments, fraud, waste, and abuse in Medicaid programs. They have also implemented quality systems augmented by conventional quality control reviews and process audits. Such programs are usually based on traditional inferential statistics and sampling processes. In this method, a relatively small sample of processed claims is reviewed to detect claim processing errors and extrapolate the sample statistics to estimate population parameters. If used correctly and effectively, these methods can provide a point or interval estimate of population parameters, including the average error rate of adjudicating claims. For example, a sample of a few hundred manually reviewed random claims could give us an interval estimate of the quality of millions of claims processed during a week with an acceptable confidence level. However, if the client wants to detect and identify all quality issues in the entire population (e.g., all claims processed erroneously), a standard acceptance sampling method cannot solve it. Quality reviews based on sampling and inferential statistical methods, trying to infer a “population parameter” using “sample statistics.” As a result, while these methods are appropriate to provide an overall estimation of the proportion of the population's quality issues, they only identify a few actual quality issues that happened to be in the reviewed samples.

Machine Learning (ML) and its application to identify quality issues in the Medicaid claim adjudication process

Machine learning is a collective name for a group of computer algorithms and statistical methods that use computational learning and artificial intelligence principles to enable computers to learn a pattern or make a prediction without being explicitly programmed (Simon, 2013) (Kohavi & Provost, 1998). Recent advancements in algorithm design and rapid increase in hardware processing power resulted in the popularity of machine learning methods in various subject matters. Machine learning algorithms are categorized into two major categories (in chapter 2, other categories are introduced in more details):

1. *Supervised learning*: the principal function of supervised learning algorithms is to infer a function from labeled training data. In this group of algorithms, the researcher uses historical data as a baseline (also called training data set) to produce an inferred function that predicts a pattern in the future datasets, commonly referred to as test data (Mohri et al., 2014). A regression model is an example of the supervised model in which a probabilistic model predicts the future behavior of a dependent variable (test data) based on existing data for independent variables (training data).

2. *Unsupervised learning*: These types of machine learning algorithms do not require pre-labeled training data and analyze a set of unlabeled data to find and model hidden structures in

the data set. Cluster analysis is an example of unsupervised methods in which an algorithm analyzes and divides a dataset into different clusters.

Detecting quality issues is a form of classification problem and could be formulated as a supervised learning problem if labeled data is available. ML methods (specifically supervised learning), in theory, can classify claims in scale if they are correctly configured. This is a significant advantage over conventional sampling methods currently used by FIs to detect quality issues in the claim adjudication process. At least on an abstract level, there is a method to detect all claim adjudication problems before FI sends the payment to a provider, translating to millions of dollars of saving for budget-constraints Medicaid programs. However, in practice, doing this requires a careful formulating of the problem, selecting proper algorithms, and extensive data prep. Many technological limitations may prevent us from using machine learning to solve this problem, including the processing power needed to handle massive computations. This dissertation aims to offer a practical approach to using supervised learning to detect most claim adjudication issues in the entire population of claims processed by an FI. The proposed model should use the claim attributes and suggest a specific ML algorithm (e.g., Decision Tree or Logistic Regression) to classify a particular Medicaid claim payment as "correctly adjudicated" or "erroneously adjudicated" using the training data. Examples of such attributes (also known as features of independent variables) could include claim attributes such as International Classification of Diseases (ICD) codes, payment data, and Current Procedural Terminology (CPT) codes, provider attributes, beneficiary attributes, claim adjudication team attributes, policy attributes, etc.

Statement of the Problem

Conventional quality methods used to detect quality issues in the Medicaid FFS claim adjudication process only provide a population proportion estimate of the claims processed erroneously. This study aims to apply supervised learning as a method to detect erroneously adjudicated claims in the entire population and find out the most important feature and most effective supervised learning algorithm in detecting if an inpatient or outpatient Medicaid FFS claim has been erroneously adjudicated.

Research Questions and Hypothesis

The following research questions will be answered, and hypotheses will be tested to complete this study.

Research Question 1 (RQ1): What supervised machine learning algorithms can be used to determine Medicaid claim payment issues?

Research Question 2 (RQ2): What are the most critical measures to compare the performance of different machine learning algorithms (resulted from RQ1) for our problem?

Research Question 3 (RQ3): What are the claim attributes that could predict if a given FFS claim has been adjudicated correctly or erroneously (also known as predictors or independent variables)?

Research Question 4 (RQ4): Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid outpatient claims?

- Hypothesis 4.1: there is no significant difference among the feature importance score (Gini importance) of the predictors in identifying erroneously processed Medicaid outpatient claims:
  - $H_{0_{4.1}}$: $\mu_{1.1} = \mu_{1.2} = \cdots = \mu_{1.n}$ where μ_{1i} is the feature importance score (Gini importance) of the i-th outpatient predictor.
  - $H_{1_{4.1}}$: There is at least one predictor (j) for which μ_{1j} is not equal to the rest.

Research Question 5 (RQ5): Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid inpatient claims?

- Hypothesis 5.1: there is no significant difference among the feature importance score (Gini importance) of the predictors in identifying erroneously processed Medicaid inpatient claim:
  - $H_{0_{5.1}}$: $\mu_{2.1} = \mu_{2.2} = \cdots = \mu_{2.n}$ where μ_{2i} is the feature importance score (Gini importance) of the i-th inpatient predictor.
  - $H_{1_{5.1}}$: There is at least one algorithm (j) for which μ_{2j} is not equal to the rest.

Research Question 6 (RQ6): Is there any statistically significant difference among the selected supervised learning algorithms (the result of RQ1) in identifying erroneously adjudicated Medicaid outpatient claims (as measured by the result of the RQ2)?

- Hypothesis 6.1: there is no significant difference among the average recall of selected supervised learning algorithms in identifying erroneously processed Medicaid outpatient claims:
  - $H_{0_{6.1}}$: $\mu_{3.1} = \mu_{3.2} = \cdots = \mu_{3.10}$ ($\mu_{3.i}$: the mean recall of algorithm *i*).

  - $H_{1_{6.1}}$: There is at least one algorithm (j) for which $\mu_{3.j}$ is not equal to the rest.

- Hypothesis 6.2: there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid outpatient claims:

  - $H_{0_{6.2}}$: $\mu_{41} = \mu_{42} = \cdots = \mu_{4.10}$ ($\mu_{4.i}$: the mean F1-score of algorithm *i*).
  - $H_{1_{6.2}}$: There is at least one algorithm (j) for which $\mu_{4.j}$ is not equal to the rest.

Research Question 7 (RQ7): Is there any statistically significant difference among the selected supervised learning algorithms (the result of RQ1) in identifying erroneously adjudicated Medicaid inpatient claims (as measured by the result of the RQ2)?

- Hypothesis 7.1: there is no significant difference among the average recall of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims:

  - $H_{0_{7.1}}$: $\mu_{5.1} = \mu_{5.2} = \cdots = \mu_{5.10}$ (μ_{5i}: the mean recall of algorithm *i*).
  - $H_{1_{7.1}}$: There is at least one algorithm (j) for which μ_{5j} is not equal to the rest.

- Hypothesis 7.2: there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims:

  - $H_{0_{7.2}}$: $\mu_{6.1} = \mu_{6.2} = \cdots = \mu_{6.10}$ (μ_{4i}: the mean F1-score of algorithm *i*)
  - $H_{1_{7.2}}$: There is at least one algorithm (j) for which μ_{4j} is not equal to the rest.

Research Question 8 (RQ8): Are the most powerful predictors different between outpatient and inpatient claims? Are the most accurate algorithms in detecting erroneously paid claims different between the two types of Medicaid claims studied?

Research Objectives

There are two specific objectives for this research:

1. Find the most important feature (predictor or independent variable) in detecting if an inpatient or outpatient Medicaid FFS claim has been erroneously adjudicated.
2. Find the most effective supervised learning algorithm in detecting if an inpatient or outpatient Medicaid FFS claim has been erroneously adjudicated.

Statement of the Need

The most significant portion of the expenditure budget in different states is allocated to Medicaid. All Medicaid programs are dealing with improper payments and erroneously adjudicated claims, especially claims processed by FIs. In 2015 the average nationwide improper payment for FFS Medicaid was 10.6%, but a similar MCO metric was estimated to be 0.1%. This means just the federal share of FFS improper payments was over $20 billion (The Centers for Medicare and Medicaid Services, 2015). Undetected Medicaid claim adjudicating issues are just one form of improper or erroneous payments. Many of these erroneous payments are not caused or controlled by FIs; some are caused by the way claims are adjudicated. FI contractors use sampling to “estimate” the error rates in the population of adjudicated claims. Still, such reviews barely result in a large-scale detection and correction of erroneously adjudicated claims. An innovative method to detect issues on the entire population of adjudicated FI claims could result in millions of dollars saving in taxpayers' money. This research allows the Medicaid administrators, researchers, FI managers, and Centers of Medicare and Medicaid Services (CMS) to leverage the power of machine learning to increase the chance of finding inaccurate payments in a Medicaid Management Information System (MMIS) claim adjudication process.

Statement of the Assumption

The study assumes no significant changes in the Medicaid payment policy would change the test supervised learning model's effectiveness. A substantial transformation in the Medicaid program (something like the impact of the Affordable Care Act) may have an unknown effect on the importance of claim features in predicting quality issues. Researchers are advised to review the potential impact of the significant Medicaid policy changes before generalizing this study's findings to other cases.

Statement of the Limitation

These are the limitations pertain to this research:

1. This research has used a set of synthesized data; a dataset created using the "characteristics" of real Medicaid data from scratch. Due to security and privacy concerns, the actual data could not be used. Using synthesized data adds some limitations to the generalization of the findings, as explained in more detail in chapter 3.
2. The scope of this study is limited to two types of Medicaid claims: outpatient and inpatient. Other researchers can continue this method and try to reproduce the result and modify the model to apply them to all different claim types (e.g., pharmacy, medical, dental, long-term care, crossover, etc.).

3. The focus of this research is on Medicaid claims processed by FI contractors. Medicaid health plans are part of the MCO model and usually have slightly different processes for claim adjudication. These differences may impact or change the predictive model (e.g., the predictors), thus require a fine-tuning in the model to make it customized to an MCO claim adjudication process.
4. The performance of various machine learning algorithms may be impacted by the hardware's processing power used for this research. This limits researcher’s ability to study algorithms like SVM, which requires excessive processing power.

Statement of the Methodology

The research methodology starts with synthesizing data and creating a simulated dataset from scratch to ensure private information is protected. A selected group of algorithms (e.g., decision tree, gradient boosting, neural network, etc.) are built using a proper ML tool (e.g., an open-source ML scripting language). The researcher uses a proper split strategy to train, test, and validate selected models. A cross-validation method is used to reduce model bias and overfitting. The researcher repeats all the steps above for both inpatient and outpatient claims. The result of the predictive models is then used to examine the hypothesis tests. The research methodology is explained in detail in chapter 3.

Definition of Terms

There is not a universally accepted terminology for machine learning, but the researchers tried to find the lowest common denominator amongst practical definitions and propose the following vocabularies:

- Data Mining is defined as “the application of specific algorithms for extracting patterns from data” (Fayyad et al., 1996). In practice, data mining has often referred to a preliminary study of data to find interesting but unknown patterns from which an initial hypothesis could be derived.
- Analytics is the discovery, interpretation, and communication of meaningful patterns in data (Ravishanker, 2018). This definition focuses on the function of Analytics and not on tools or methods. This definition includes all statistical techniques applied to both big and small datasets.
- Data Science (DS) is the interdisciplinary sciences and art of using computers, algorithms, mathematics, statistics, and subject matter expertise to extract insights from structured and unstructured data (Parks, 2017) (Manish, 2017). Extracting insight from unstructured data (e.g., images, text, and voice) has a significant implication here. Most conventional quality tools use structured data like nominal, categorical, and ratio data. In this dissertation, the terms Analytics and Data Science are used interchangeably.
- Big Data Analytics (BDA): ISO/IEC JT defines BDA as “analytical functions to support the integration of results derived in parallel across distributed pieces of one or more data sources” (ISO/IEC JTC 1 Information technology, 2015). This definition is technically accurate but too broad to specify the type of data subject to analytical functions. The researcher offers a slightly more focused description: BDA is the science of applying computer science and statistical learning to discover non-trivial patterns in massive datasets. This definition emphasizes two critical aspects of BDA: “non-trivial” patterns and “massive datasets.” “Non-trivial” is not a scientific notion but a practical Occam's razor heuristic. If a problem is not complex and could be solved using simple methods, it is not economical to

use complex BDA methods. Many studies use BDA and Data Mining interchangeably (*What Is Data Mining*, n.d.). However, in this dissertation, data mining is defined as the exploratory analysis of massive datasets to find interesting (and unknown) patterns, especially before an exact research problem is formulated.

- Machine Learning (ML) is a collective name for a group of algorithms based on statistical learning and artificial intelligence principles that enable computers to learn patterns or make predictions without being explicitly programmed (Simon, 2013) (Kohavi & Provost, 1998). Machine learning is not limited to big data; however, DBA shows its strengths when using machine learning to solve problems with massive datasets. ML a subset of BDA with a focus on the method (algorithm) by which data is mined.
- Structured data refers “to any data that resides in a fixed field within a record or file” (Beal, 2018). Examples of structured data are data that reside in a relational database or a spreadsheet.
- Unstructured data is defined as “information that either does not have a pre-defined data model or is not organized in a pre-defined manner” (Wikipedia, n.d.). Examples of structured data are emails, PDFs, images, videos, audios, social media posts, text files, websites, MP3 files, and most sensor data. About 80% of the data in the world is unstructured (Cano, 2014).
- Optimization: is defined as “searching for the extreme value of some objective function. The variables of the objective function represent the system parameters, and the extreme value corresponds to the optimized state of the system” (Knobloch et al., 2017).
- Cost function: when trying to optimize a situation by finding the minimum of the objective function, it is referred to it as a cost function (Knobloch et al., 2017).

- Protected Health Information (PHI): under the US Health Insurance Portability and Accountability Act of 1996 (HIPAA), any information that is created or received by a health care provider and relates to the past, present, or future physical or mental health or condition of any individual and identifies the individual or could be used to determine the individual, is called PHI. 45 CFR 164.514 Federal law (HIPAA Protected Health Information Identifiers) lists 18 specific PHIs items. This list includes name, all geographical identifiers, dates (other than year) directly related to an individual, phone, fax, email, social security number, medical record number, health insurance beneficiary numbers, account numbers, certificate/license numbers, vehicle identifiers and serial numbers, including license plate numbers, device identifiers, and serial numbers, URLs, IP addresses, biometric identifiers, full face images, any other unique identifying number, and record or payment history (Other Requirements Relating to Uses and Disclosures of Protected Health Information, 1996).

Chapter Summary

Medicaid provides medical services to 67 million Americans permanent residents and costs taxpayers $600 billion a year. Medicaid runs through two models, MCO (covering most Medicaid beneficiaries) and FFS. In the FFS model, the Medicaid claims are processed by an FI contractor and prone to processing errors. This costs taxpayers over $20 billion a year. FIs often use conventional inferential statistics to estimate population parameters using sample statistics. This method is incapable of detecting all quality issues in the claim adjudication process. ML algorithms are technically capable of computing the population parameters directly, resulting in millions of dollars saving for taxpayers. The goal is to find the most accurate supervised learning algorithm to detect FI claim adjudication issues. Seven research questions are formulated with the corresponding hypothesis and defined the research objective, need, limitation, assumption, and briefly explain our methodology. This chapter is concluded by defining some key terms.

CHAPTER 2

REVIEW OF LITERATURE

Chapter Overview

This chapter starts with some background information about the Medicaid program and its importance as a massive government-funded entitlement program for Americans and how Medicaid operates. Two standard Medicaid models are presented and explained why detecting quality issues in Medicaid claim adjudication is essential. Next, the overall impact of machine learning as a dominant “Disruptive Technology” and its increasing importance due to the exponential growth of analytical data in industry and society is discussed. The evolution of quality and machine learning as two analytical disciplines are explained. Then, it is shown how big data analytics fits into the quality paradigm. Different classes of machine learning algorithms are briefly discussed, and then the researcher reviews the variety of machine learning applications to detect quality issues in various industries. The researcher then summarizes multiple approaches to protect PHI and propose “synthesizing a simulated dataset” as the proper approach for this dissertation. A summary of the literature on the process of building, training, and validating supervised learning models (including selection criteria to choose an ML algorithm) is provided. Based on the literature review results, the researcher selects a few specific algorithms to be tested. Finally, the researcher reviews the literature's lessons and then focuses on the problem's scope and concludes the literature review with a few guidelines in this research's next chapters.

Medicaid Background

Medicaid is a government-funded healthcare program managed by each State to assist low-income beneficiaries and people with disability in paying for healthcare costs and long-term medical and custodial care costs (*What Is "Medicaid,"* n.d.). Medicaid is authorized by the Title XIX of the Social Security Act and runs since 1965. The federal government and each State mutually fund Medicaid programs. Centers for Medicare & Medicaid Services (CMS) is the administrator of the Medicaid and Medicare programs on a federal level. A department under the Health and Human Services Agency (HHS) administers the Medicaid program on a state level in each state. The collective name for state organizations administrating the Medicaid programs on the state level is State Medicaid Agency (SMA).

As of March 2018, over 67 million individuals were enrolled in Medicaid in 50 states and the District of Colombia (Medicaid.gov, 2018). The number of Medicaid enrollee has increased to over 71 million in October 2019 (The Centers for Medicare and Medicaid Services, 2020). In the Fiscal year 2016, the total Medicaid expenditure (on both federal and state levels), including administrative costs and accounting adjustments, was $574.2 billion (The Henry J. Kaiser Family Foundation, 2018). This number increased to $616.1 billion in 2018 (The Centers for Medicare and Medicaid Services, 2020). To put things in perspective, in FY 2016, the U.S. government collected $2.99 trillion in tax revenues and had a $587 billion budget deficit, so Medicaid spending was equaled to 19% of tax revenue and almost equal to the U.S. government budget deficit (USAID, 2016) and (Congressional Budget Office, 2017). CMS has predicted the total

Medicaid expenditures to increase from 3.1 percent of GDP in 2017 to 3.3 percent of GDP in 2027 (The Centers for Medicare and Medicaid Services, 2020).

Medicaid programs are usually divided into two subcategories:

1. Fee-For-Services (FFS): in this model, a Medicaid provider (e.g., a physician, pharmacy, a long term facility, hospital, etc.) provides medical services to a Medicaid beneficiary and sends the bill (medical claim) to an SMA. SMAs either process claims directly, or more commonly, through a claim adjudication contractor known as Fiscal Intermediary (FI). FIs leverage a state-owned computer system known as the Medicaid Management Information System (MMIS) to adjudicate and process claims.
2. Managed Care Organization (MCO): in this model, the State hires Health Plans (HP) and assign Medicaid beneficiaries to them through a complex enrollment process. This model (also known as the Capitated model) pays each HP a fixed monthly amount per covered beneficiary through a Per Member Per Month (PMPM) method. SMAs or FIs do not process capitated claims; however, SMAs request HPs to submit an informative document calls encounter form for each service provided by MCO contractors. Encounters give the detail of medical services provided to the Medicaid beneficiary and the amount MCO paid to the provider (if any). SMAs use the encounter data to adjust the PMPM rates and communicate the Medicaid program's effectiveness to the state legislators and regulators (Henderson, 2017).

Medicaid Improper and Erroneous Payments

It is vital to ensure Medicaid programs are administrating the taxpayer funds effectively and detect, correct, and prevent improper or erroneous payments both at administrative and policy levels. Total Medicaid improper payments are estimated to be over $29.1 billion in FY 2015 with an estimated improper payment rate of %9.8 (*Testimony of Ann Maxwell Before the United States House of Representatives*, 2017). The term improper has a broad range of applications and subject to various interpretations. This research offers the following definitions of these terms (these are not standard definitions but will allow the research to clarify the scope of the study):

1. Intentional improper payments: including fraud, waste, and abuse (FWA). These types of issues are usually outside of the control of an FI and mainly initiated by providers.
2. Unintentional improper payments: unintentional improper payments are operational quality issues in the claim adjudication processes and systems. FIs, health plans, and SMAs have more control over such problems compared to FWA cases. In this research, these terms are interchangeable with unintentional improper payments: "erroneous payments," "payment errors," and "quality issues in the claim adjudication process" (which include erroneous denials). Any of the classic five Ms could cause an erroneous payment:
   - Men power: claim examiners processing errors, Miscalculation of manually priced claims, data entry issues, etc.
   - Machines: scanners, optical character recognition (OCR), and MMIS systems (logic issues in MMIS applications).
   - Methods: claim adjudication procedures, MMIS software defects, system update, and documentation issues (e.g., lack of timely updates in MMIS tables)
   - Materials: inputs to the process of quality claims submitted by providers like illegible or duplicative claims.

- Measurement: issues in SMA and CMS policies and policy materials and execution, including problems with error codes, operational instruction letters (OILs), and standard operating procedures (SOPs) resulting in a claim under or overpayment

This research focuses on erroneous payments, specifically those caused by quality issues in the process of adjudicating claims.

Both FFS and MCO models usually require a Medicaid claim to be proceeds either by a health plan or an FI. As a result, it is crucial to ensure all Medicaid claims are processed correctly and free from payment errors. There are three types of payment errors in the processing of Medicaid claims (The Centers for Medicare and Medicaid Services, 2014):

1. Medical review errors include claims deemed medically unnecessary or not following the State's written policies.
2. Inaccurate eligibility determination: including paying a Medicaid claim for an ineligible person.
3. Claim processing and adjudication error: including errors cause by the FI and MMIS operations. The two major types of these errors are:
    a. Claim pricing issues (caused by the MMIS software defects or medical examiners' miscalculations)
    b. Data entry issues (caused by the claim adjudicators and data entry staff)

The focus of this study is the third type of erroneous payments. This research's proposed approach could be applied to other kinds of erroneous payments, subject to labeled data availability.

In the past few decades, FI contractors and MCO health plans have invented and implemented several methods to detect and prevent improper payments, fraud, waste, and abuse in Medicaid programs. They have also implemented quality systems augmented by conventional quality control reviews and process audits. Such programs are usually based on traditional inferential statistics and sampling processes. A sample of processed claims is reviewed to detect claim processing errors and extrapolate the sample statistics to estimate population parameters. If used correctly and effectively, these methods can provide a point or interval estimate of population parameters. For example, a sample of a few hundred manually reviewed random claims could give us an interval estimate of the quality of millions of claims processed during a week with an acceptable confidence level. However, if the client wants to detect and identify all quality issues in the entire population (e.g., all claims processed erroneously), a sampling method cannot answer the question. Quality reviews based on sampling and inferential statistic methods, by nature, are not able to "directly" compute population parameters.

Improper payments in the context of two Medicaid Models

Both Medicaid models (FFS and MCO) are dealing with improper payment issues. Study shows as of 2015, 9.8% of all Medicaid payments (over $50 billion) were contributed to improper payments (The Centers for Medicare and Medicaid Services, 2015). However, there is a significant difference between the rate of improper payment for FFS and MCO. While the average improper payment for MCO is less than 0.1%, a similar FFS measure is estimated to be closer to 10.6%. This means just the federal share of FFS improper payments was over $20 billion (The Centers for Medicare and Medicaid Services, 2015). As a result, while the FFS portion of the Medicaid population is on average 20% or less, they contribute to 70% of improper payments (The Centers for Medicare and Medicaid Services, 2015).

This dissertation focuses on “erroneous payments” in the FFS claim adjudication process caused by FI and MMIS operations. Not only do these issues have a significant contribution to overall waste Medicaid waste ($20 billion a year), but also there are much easier to be fixed compared to intentional improper payments. There two reasons for this: Firstly, fraud and abuse are complex and ever-changing patterns that are adjusted deliberately by those who want to take advantage of the system and try to make them undetectable. Secondly, fixing a quality issue, if it is found before the payment, is easy and requires a simple reprocessing of the claim. In contrast, even after the fraudulent activity is detected in many cases, it needs to go through a complicated legal process to recover any overpayment.

Figure 1 illustrates the two Medicaid models (explained earlier in this chapter), the role of parties in each model, and how Medicaid fraud analytics is different from this dissertation's scope. This research aims to create a training data set using the known patterns of erroneous payment in a given Medicaid FFS. Then, a few well-known supervised learning algorithms will be trained and tested using synthesized data.

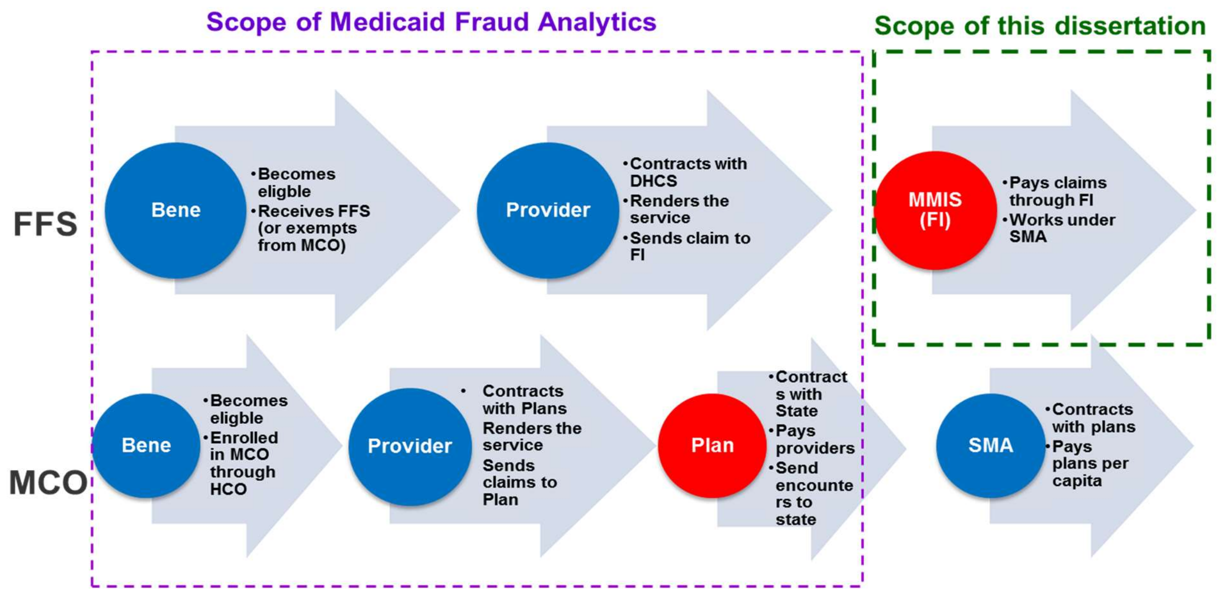


Figure 1. Improper payments in the context of two Medicaid Models

Disruptive Technologies, Big Data Analytics, and Machine Learning

Global Technology Innovation Insights is the title of a series of annual studies conducted by the big consulting firm KPMG to offer the clients a view of the upcoming challenges and issues in the world of technology. "The Changing Landscape of Disruptive Technologies" is one of these reports published in 2015. The study was based on the data gathered and analyzed in 2014 based on 768 surveys conducted with tech-leaders from 14 countries, including start-up, Mid-market and large corporations (each represented about 30% of the interviewees) as well as angel investors and venture capitalists (Matuszak et al., 2015). The central premise of this article is to analyze the impact of disruptive technologies on business models. The challenges and opportunities will become a result of the emergence of such technologies. Cloud and mobile computing, Internet of Things (IoT), Big Data Analytics (BDA), Machin-2-Machin (M2M) interactions, Biotech, and 3D printing, Autotech (including self-driving and parking cars) are the leading technologies listed in this report as disruptive. Cloud computing and mobile technology are named as the two significant foundations for technology that will continue to be the basis for cutting-edge advances, while they are losing their high positions (compare to last years) as a disruptor. Cloud computing and mobile technology continue to drive more efficiency and cost-saving by transforming business models and making decision making faster and more efficient by streamlining the process of gathering and analyzing data. But the underlying technology for all disruptive technologies is BDA. As an ever-increasing technology trend, BDA plays a more critical role in driving incremental business value.

The exponential growth in the amount of data generated through customers' and devices' interaction caused a rapid expansion of the need for better analytics and data science practices. The rapid growth of IoT, artificial intelligence, and robotics directly improves the power of machine learning and big data analytics. The evolution of analytics and data science improved productivity on a global scale. It has also shortened the innovation cycle and improved customer loyalty. BDA has disrupted many industries in the past two decades and introduced new challenges and opportunities in many disciplines. Quality, as a scientific and practical body of knowledge, is no exception. The exponential growth of data in industry and society in the past few decades calls for new analytical paradigms. New methodologies, algorithms, and solutions powered by machine learning are opening the door to exciting breakthroughs. Supervised and unsupervised learning methods have solved many operational and quality problems.

How much data is “Big Data”?

The volume of operational data grows daily, so the need for extracting meaningful quality insight from massive datasets becomes increasingly essential. Research shows the global dataset size to analyze will increase by a factor of 50 to 5.2 ZB in 2025 and by a factor of 100 to 1.4 ZB in 2025. (Reinsel et al., 2017). Figure 2 illustrates the global data growth between 2004 and 2017. The scale is not exact, but the picture fully demonstrates the exponential growth of global data. These are some statistics to help estimate the typical size of a dataset (*How Much Data Does Each DVD Format Hold?*, 2018), (Tozzi, 2017), and (J. Lee, 2014):

- 700 MB: is the size of a CD-ROM
- 1.8 GB: is the data an average U.S. customer uses on her or his cell phone per month
- 16 GB: is the size of a double-sided double-layer DVD
- 1 TB: equals approximately 1,500 CD-ROMs
- 1.5 PT: is the size of 10 billion Facebook photos
- 20 PT: is the data processed by Google every day

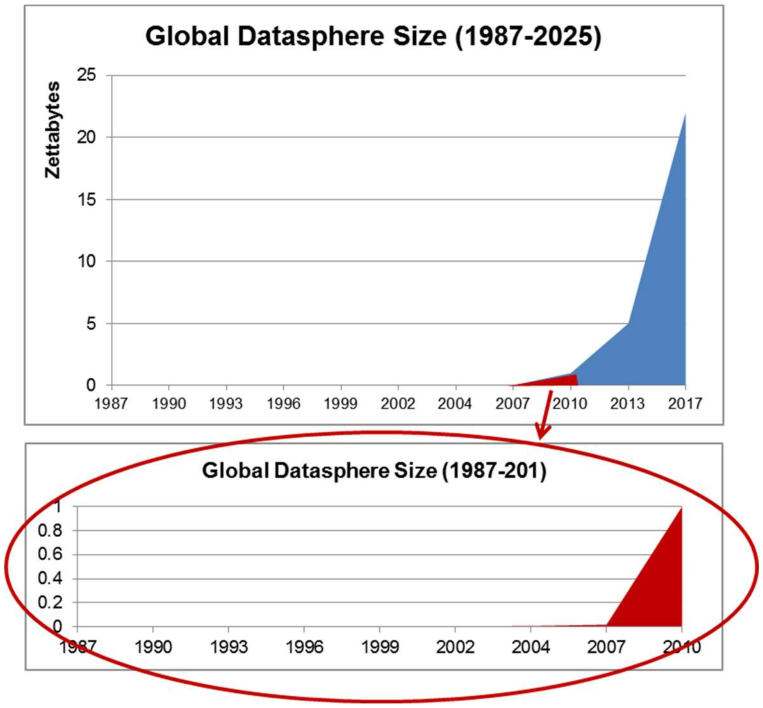


Figure 2. Global Growth of Data 2004-2017.

The **V**olume of the data is essential and directly impacts the complexity and the computational power needed to solve a problem. However, the volume is not the only important factor, and there are three other classic **V**'s that impact the complexity:

- **V**elocity: velocity is defined as how fast data should be processed or a decision should be made. Google processes over 40,000 search queries per second (Frican, 2017), so any technology that uses such a velocity of input data has to be very efficient to produce timely results. Data streaming platforms and broad application of IoT rapidly increasing the data velocity and amplifying the need for real-time decision-making technologies (e.g., in-memory analytics).
- **V**ariety: quality professionals are familiar with structured data. However, extracting insights from a massive unstructured dataset is a relatively new challenge for them. Over 80% of data in the world is unstructured (Cano, 2014). Many unstructured data sources (e.g., videos, images, social media comments, doctor notes, medical images, etc.) contain Critical to Quality (CtQ) information and need to be adequately analyzed.
- **V**eracity: refers to the "biases, noise, and abnormality in data" (Normandeau, 2013). Lack of data quality is the most significant barrier to the proper implementation of DBA. Quality professionals spend substantial time to prepare data. As the size and complexity of data increase, data veracity becomes more critical. Low data quality costs the US economy over $3 B annually, and one in three business leaders doesn't trust the quality of data they use for decision making (IBM Big Data & Analytics Hub, n.d.).

As a practical rule-of-thumb, a messy set of structured and unstructured data more massive than 1 TB is a sensible minimum for a massive dataset. You can apply BDA technologies to small, structured, and static datasets, but a conventional quality approach would be more cost-effective.

Evolution of Quality and BDA over time

Quality discipline is evolving, but the emergence of revolutionary quality management concepts and methodologies seems to have reached a plateau. When was the last time you encounter a groundbreaking quality management idea, something like TQM, SPC, Lean, or Six Sigma? It appears in the past two decades, many of the advancements in quality management fall into one of these two categories:

1. New quality standards, including Capability Maturity Model Integration (CMMI), Information Technology Infrastructure Library (ITIL), and standards developed by various standard organizations including International Organization for Standardization (ISO) and Institute of Electrical and Electronics Engineers (IEEE). ISO 9000 family of standards and IEE 730 are examples of quality standards developed and evolved in the past couple of decades.
2. New applications of the old concepts (e.g., leveraging social media to measure customer satisfaction).

In contrast, as depicted in Figure 3, new and innovative BDA concepts, tools, and techniques are introduced every year. Quality and BDA are data-driven disciplines and have overlaps so quality practitioners can review the DBA advancements and leverage BDA to tackle quality problems using new tools and techniques.

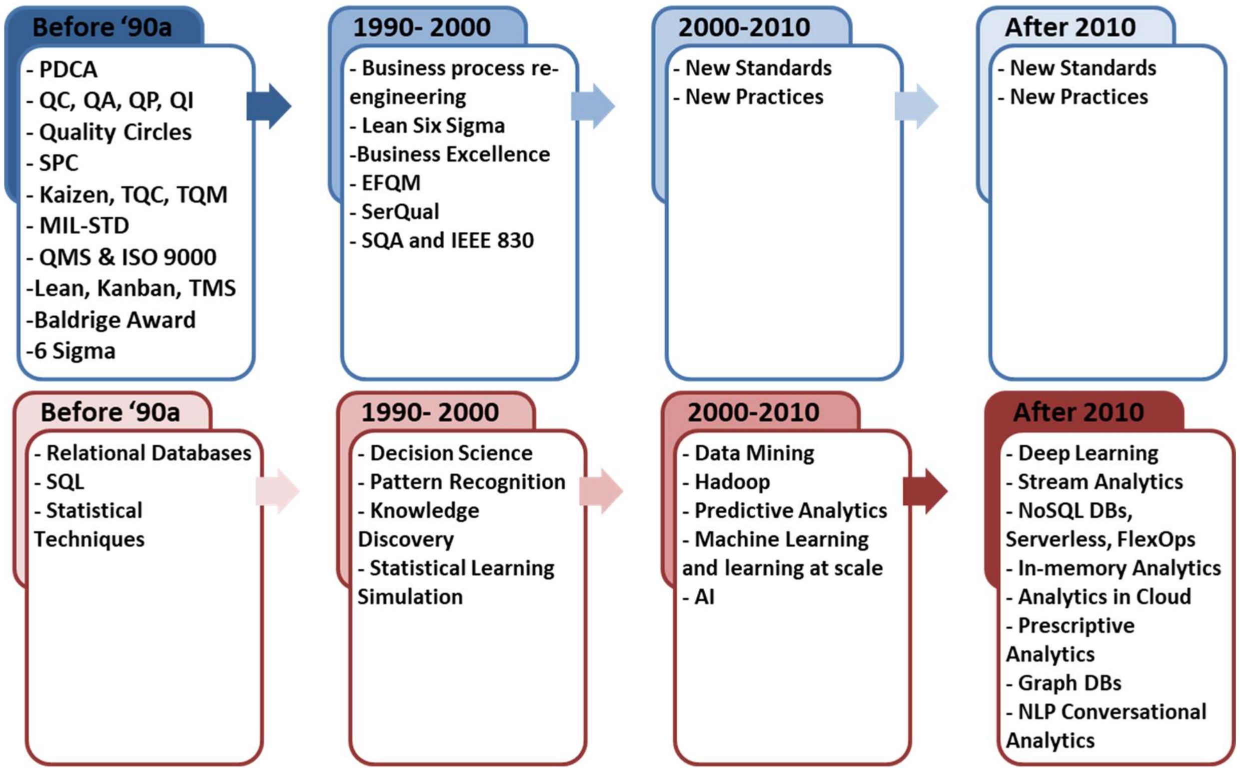


Figure 3. Evolution of Quality and BDA over time.

How does BDA fit into the Quality paradigm?

There are three data analysis areas in statistics: descriptive statistics, inferential statistics (or inferential analysis), and the Design of Experiments (DoE) (Bartz-Beielstein et al., 2010). As illustrated in figure 4, inferential statistics and descriptive statistics are observational, and DoE is experimental by nature. Descriptive statistics could be easily applied on large datasets and the first candidate to be augmented by BDA. Rapid progress in data management technologies is changing the landscape. Now, it is possible to run analytics on the entire operational data and directly measure population parameters. BDA is also used to search patterns and find associations among variables on a massive scale (Han & Kamber, 2001) and apply the detected patterns to new subsets of data and validate the findings (Hastie et al., 2009).

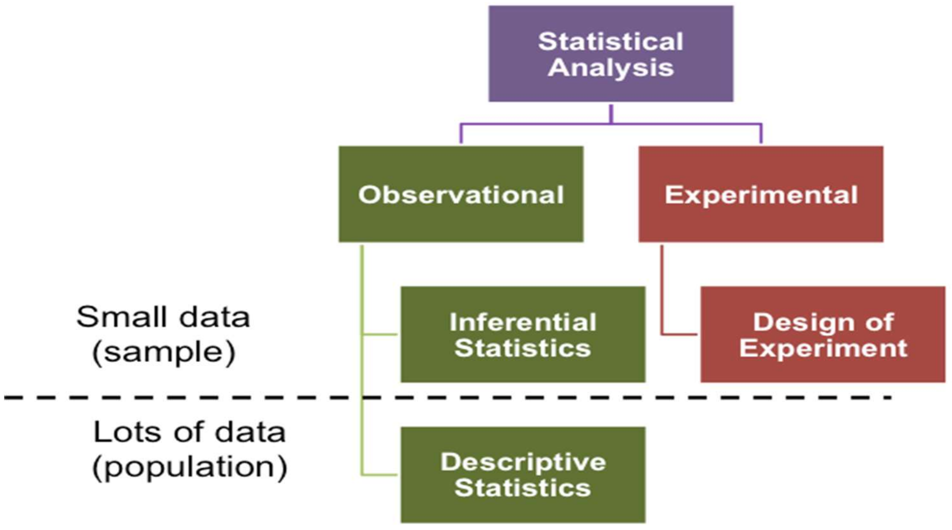


Figure 4. statistical analysis and data size.

Inferential statistics is a popular data analysis approach among quality practitioners, especially in the service industry. The idea is to review a relatively small sample and extrapolate the sample statistics to the entire population. In this sense, inferential statistics is the art of extracting meaningful information from small data. The core of many quality systems in the service industry is to design a set of checks and balances, find a proper sampling plan, and execute a series of measurements to ensure processes can produce the intended results. How could BDA support inferential statistics? By automating the measurement. Detecting quality problems is essentially a "classification" problem and could be solved through supervised learning. Researchers only need to have examples of conforming and non-conforming samples to use them as labeled data. DoE also deals with small datasets. It seems quality professionals know the art of small data, but big data is a new paradigm.

Machine Learning (ML) algorithms at the heart of BDA

ML is the main engine of DBA. ML methods have been around for decades, but recent increases in the computers' processing power, made them available to a broader user community. Advancements in data processing, like matrix, parallel, and distributed computation, also made it possible to leverage massive datasets and implemented scalable deep-learning solutions. As a result, the performance of the algorithms improves as data grows. All these made ML a popular and economical method to solve many operational and quality problems.

ML algorithms use probabilistic models. A probabilistic model is a way to prove the existence of a structure by creating a probability space (choosing random elements) and "prove any random element from the space has both a positive probability and the properties sought after" (*Probabilistic: Definition, Models and Theory Explained*, n.d.). ML algorithms are commonly clustered in four primary categories (Fumo, 2017) and (Ayodele, 2010):

1. *Supervised learning*: the principal function of supervised learning is to infer a function from labeled data. In this group of algorithms, the researcher uses historical data as a baseline (training data) to produce an inferred function that predicts a pattern in the future data (test data) (Mohri et al., 2014). A regression model is an example of supervised learning in which the future behavior of a dependent variable is predicted based on the independent variables. The most used supervised learning algorithms are Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), and Neural Networks (NN). The biggest challenge with supervised learning is the labeling process, usually a manual, time consuming, and expensive process.

2. *Unsupervised learning*: These algorithms analyze a set of “unlabeled” data to find and model hidden structures in the dataset. Cluster analysis is an example of unsupervised methods in which an algorithm analyzes and divides a dataset into different clusters. K-means clustering and Association Rules (AR) algorithms are examples of unsupervised learning methods.
3. *Semi-supervised (or Hybrid) learning*: semi-supervised algorithms use a combination of labeled and unlabeled data for training. In these algorithms, a small set of labeled data is used to identify specific groups of data elements present in the labeled and could be used to label unlabeled data. The algorithm is then trained on the unlabeled data to define the boundaries of those data elements and even found new characteristics for labeling (Castle, 2018).
4. *Reinforcement learning:* These types of algorithms use an iterative approach to learning. In these methods, the algorithm (a.k.a. agent) gathers observations from the interaction with the environment and tries to optimize the outcome (minimize risk or maximize the profit) using a reward feedback (a.k.a. reinforcement signal) and continues with the iterations until it explores the full range of possible states (Fumo, 2017). Q-learning, Temporal Difference (TD), and Deep Adversarial Networks (DAN) are examples of reinforcement learning methods.

Application of ML to solve quality problems

Machine learning techniques have been used to solve production control problems since the late 90s. As an example, Bowden and Bullington successfully applied an unsupervised learning method called the Genetic algorithm rule discovery system (GARDS) to solve a flexible cellular manufacturing system (eMS) problem (Bowden & Bullington, 1996). However, leveraging machine learning to solve quality problems is a much newer trend. The literature has reviewed extensively to find examples of the application of machine learning to solve quality problems. In most cases, machine learning methods are used to detect quality issues in a product, process, or service. Our literature review's focus was to summarize cases that are more relevant to this dissertation's scope. This includes studies about leveraging ML to augment a quality control or quality assurance process or detect quality issues in service. The researcher has also included studies in which different machine learning algorithms' performance in solving a quality problem was compared.

Examples of ML application to detect quality issues in the service industry

Yussupova and colleagues used several machine learning techniques, including Text Mining, Aspect Sentiment Analysis, and Decision Trees, to emulate the service quality and customer satisfaction for hotel customers (Yussupova et al., 2016). Peddamuthu & Srivastava used statistical and rule-based natural language processing (NLP) techniques to categorize sentiment and emotions, detect quality issues (e.g., anger and delays) to augment the QA process for calls center conversations (Peddamuthu & Srivastava, 2014). Ucar and colleagues leveraged Extreme Learning Machine (ELM) to detect smart-grid Power Quality Events (PQE) in a dataset. They combined a histogram-based method with a Discrete Wavelet Transform (DWT) to improve their algorithm's performance and then tested the result on a real-world like PQE database and validated the accuracy of the classification method (Ucar et al., 2018). To augment the QA process and detect inaccurate coding in financial contribution, Blomquist & Möller applied supervised classification methods (e.g., SVM) and tested ten classification models. They concluded the Adaboost procedure performed better and more steadily on most models, primarily when an ensemble classifier supports it (Blomquist & Möller, 2015). Honda leveraged

Text Analytics and Machine Learning to extract and classify useful information from the unstructured feedback it receives from a pull of our 20 million customers resulting in an 80% reduction in the time required by Quality Assurance staff (Kumar, 2018). To estimate the mobile phone provider service quality using social media (Twitter), Calvin & Setiawan developed a supervised learning model and Naïve Bayes classifier. This study has several limitations, including limited use of training data, trying just one supervised learning algorithm, and a lack of proper validation method of test results (Calvin & Setiawan, 2014). Paprzycki et al. compared the performance of Six ML algorithms to predict the quality of call center services. They compared Multi-layer perceptron (MLP), Linear neural networks (LNN), Probabilistic neural networks (PNN), Classification and regression trees (CART), SVM (with a third degree polynomial kernel), and a hybrid decision tree-ANN. They concluded the CART algorithm prfomes the best in overal prediction accuracy and also resulting in lowest false positive and false negative prediction (over 80% customer service satisfaction prediction accuracy and close to 90% in predicting business need satisfaction) (Paprzycki et al., 2004).

Examples of ML application to detect quality issues in a product or manufacturing process

As an alternative to expensive chemical QC test for olive oils, Ordukaya & Karlik used classification ML algorithms. First, they used Principal Component Analysis (PCA) to reduce the number of predictors from 32 to 8. Then they compared the performance of classifiers algorithms, including SVM, ANN, Naïve Bayesian, -Nearest Neighbors (-NN), Linear Discriminate Analysis (LDA), and Decision Tree. Then performances of these classifiers were compared according to their accuracies finding Naïve Bayes performs the best as a classifier algorithm in their case (Ordukaya & Karlik, 2017). Escobar and Morales-Menendez applied supervised learning classification methods, including the least absolute shrinkage and selection operator (LASSO) and Logistic Regression, to detect rare quality defects in a high-conformance manufacturing environment with a near-perfect result (Escobar & Morales-Menendez, 2018). Ribeiro proposed a classification model based on SVMs (*C*-SVM and *v*-SVM algorithms) to monitor plastic injection molding quality issues. He used process data (cycle time, metering time, injection time, barrel temp, cushion, and injection velocity) as predictors (independent variables). The accuracy results ranged from 70.83% up to 99.16%, which shows an acceptable performance. She also validated the model using a radial basis function (RBF) NNs as an alternative classification method and concluded SVMs have a performance advantage over NNs (Ribeiro, 2005). ML and Predictive modeling in junction with IoT is used for predictive maintenance and predicting the equipment failure, which resulted in a significant increase in the overall equipment efficiency (OEE) and reliability (Tracy, 2018). Researchers used a massive dataset containing three years of process and sensor data from 350 lines across 33 manufacturing plants and 15 countries. They built a machine learning model using N-dimensional Euclidean distance-based scoring algorithms normalized that predicts potential quality failures of a process and then applied the model to 1,400 manufacturing lines and over 20 million sensor events resulting in a potential $300 million annual saving (RTInsights Team, 2017). Lieber and colleagues proposed a hybrid ML model consisting of unsupervised (k-Means) and supervised algorithms (nearest neighbor and SVM) to predict intermediate products' quality in interlinked manufacturing processes. With the k-NN algorithm (k=11), they achieved a 97% accuracy level in prediction, while SVM produced about 90% prediction accuracy (Lieber et al., 2013). Irgens and colleagues proposed a hybrid model consisting of unsupervised (Cluster Analysis) and Supervised ML algorithms (SVM) to improve the product and process quality in the manufacturing process (Irgens et al., 2012). In a detailed study, researchers used the manufacturing quality control data. They proved deep learning algorithms (specifically deep restricted Boltzmann machine and the stacked autoencoder) have a performance advantage over shallow learning methods (in this case, a feed-forward neural network with one hidden layer and the least squares support vector machine with no hidden layers). Also, they found the performance of deep learning algorithms is proportional to the sample size (Bai et al., 2017). Chou, Ho, & Hoang collected ten years' worth of data from 20 reservoirs in Taiwan and applied four ML algorithms (ANN, SVM, classification and regression trees, and linear regression) and concluded the ANN model performs best in predicting the water quality (Chou et al., 2018). To correlate arc sound with the weld quality, Sumesh et al. used ML classification algorithms (namely J48 and Random Forest) and concluded J48 is outperforming the random forest in detecting weld quality issues (88.69% accuracy for J48 vs. 70.78% for random forest) (Sumesh et al., 2015).

Examples of application of ML to detect quality issues in the food industry

To identify quality issues in Salmon using a computer vision, Sture applied ML and classification methods using Support Vector Machines (SVM) with a nonlinear kernel function (RBF), with success and then validated the geometric classification results using nearest-neighbor classification, with slightly less accuracy (Sture, 2015). Lotfi and the team successfully applied vision processing techniques augmented by an ML method based on neural networks to detect French fries' quality issues (Lotfi et al., 2008). To expand machine vision and measure the quality of mixed raisins, Karimi and the team built an ML model with 146 predictors and then applied a Principal Components Analysis (PCA) method to reduce them predictors to an optimal level. They used SVM and ANN to classify the raisin mixtures and found SVM surpasses ANN in this classification problem and produces a high-level accuracy in prediction (Karimi et al., 2017). Gonzalez Viejo et al. leveraged ML and used physical measurements of color and foam (as independent values) to predict beer quality (defined by the intensity levels of sensory descriptors like perceived foam–related parameters and beer color). Their research used principal component analysis to identify the most relevant predictors and an artificial neural network (ANN) regression algorithm, which resulted in an R = 0.91correlation to predict the beer's quality (Gonzalez Viejo et al., 2017). To check the quality of Pistachio automatically and through machine vision, Çitak & Genç formulated a classification problem and used support vector regression (SVR) and deep convolutional networks with over 98% accuracy (Çitak & Genç, 2017).

Examples of ML application to detect quality issues in Software and Information Technology

ML and predictive modeling have long been used to augment the quality assurance and testing practices in software engineering and information technology. Morales used a generic title for all such algorithms as "Predictive quality analytics" and defined it as: "the process of extracting useful insights from test data from various sources by applying statistical algorithms and machine learning to determine patterns and predict future outcomes and trends." The core statistical algorithms used in such methods include various classification, clustering, regression, time series, and association techniques (Morales, 2017). Parra and colleagues used a supervised learning model and text analytics approach to classify the software requirements (into two classes of good and bad quality requirements), emulating quality experts' assessments using two different methods. They compared the accuracy, stability, and efficiency for six different machine learning algorithms: PART, C4.5, bagging PART, bagging C4.5, boosting PART, and boosting C4.5. They concluded C4.5 has the highest efficiency and accuracy rate (87.72%) but has the highest standard deviation (7.12%). They found bagging-PART has the lowest standard deviation (2.44%) and is the most stable algorithm with the second-highest accuracy of 87.02% (Parra et al., 2015). Khoshgoftaar & Seliya used tree-based software quality classification models, software metrics, and SPRINT decision tree algorithm to build a predictive model to forecast if a software module is fault-prone or not. They also collected software metrics from extensive telecommunications systems and compared the SPRINT decision tree algorithm's result with a CART decision tree algorithm (a more generic form of SPRINT). They found advantages in how the SPRINT decision tree algorithm uses a unique tree pruning technique based on the Minimum Description Length (MDL) principle resulting in improved accuracy and stability of the model (Khoshgoftaar & Seliya, 2003). Many researchers have leveraged machine learning to predict which module in software has a higher chance of failure. Elish & Elish compared the performance of 8 different ML algorithms, including SVM, Logistic regression (LR), k-nearest neighbor (kNN), multi-layer perceptrons (MLP), Radial basis function (RBF), Bayesian belief network (BBN), Naïve Bayes (NB), Random forest (RF), and Decision tree (DT). They found SVM outperforms other algorithms in predicting defect-prone modules on the four NASA datasets (Elish & Elish, 2008). In a similar attempt, Gondra used the same NASA open-source dataset and 21 software product metrics as predictors and compared SVM and ANN's performance as supervised learning methods. He concluded SVM outperforms ANN in this binary classification problem (87.4%, to 72.61% accuracy rate). He also noted the advantage of SVM and ANN as nonlinear approaches to solving this problem compared to linear ones and even the benefit of sensitivity analysis to PCA in selecting software metrics that are more robust indicators of a defect in a module (Gondra, 2008). To overcome the challenge of skewness in defect-prediction datasets, Pelayo & Dick applied Synthetic Minority Over-sampling Technique (SMOT). This improved the geometric mean classification accuracy by over 20% (Pelayo & Dick, 2007).

Bouguil and the team applied a Bayesian model based on finite Dirichlet mixture models to predict software quality of fault-prone and non-fault-prone program modules. They used Gibbs sampler to implement their Bayesian algorithm (Bouguila et al., 2008). ML algorithms like SVM and Bayesian methods are widely applied to predict the quality of Web Services (Kai et al., 2016).

ML application examples of detecting quality issues in healthcare

A group of researchers used over 7,500 human-rated samples and applied specific supervised ML techniques (kernelized Support Vector Machine and Gradient Boosted Decision Trees classifiers) to detect meshes of failing quality. They improved the accuracy of quality assessment of MRI-derived data conducted by medical professionals resulting in a human workload reduction by 30-70% (Petrov et al., 2017). Gupta applied NN and SVM algorithms to determine the dependency of wine quality on different physicochemical characteristics. He first used linear regression to measure the dependency of the target variable (wine quality) on predictors and then leveraged the SVM and NN and concluded that SVM performs better than NN to predict the wine quality (Gupta, 2018).

Examples of application of ML to detect quality issues in healthcare fraud detection

ML in healthcare services is frequently used to detect Fraud, Waste, and Abuse (FWA). Anomaly detection, which mainly uses “unsupervised learning” methods, has been used often for this purpose (Anbarasi & Dhivya, 2017). Some researchers advanced anomaly detection methods through a combination of several algorithms. For example, to detect Korean outpatient clinics with abusive utilization patterns, Shin and the team proposed an algorithm consisting of a scoring model to measure the degree of abusiveness and segmentation method to cluster clinics with similar utilization patterns. Their algorithm leveraged decision tree for clustering and conditional probability distributions of the composite degree of anomaly (CDA) score to categorize clinics in intervention or non-intervention subgroups (Shin et al., 2012). Kose and colleagues proposed an interactive machine learning approach combined with the pairwise comparison method of analytic hierarchical processing (AHP) for weighting the actors and attributes and expectation maximization (EM) to detect electronic fraud and abuse in the healthcare system. To overcome the limitation of normality and outlier free assumptions of parametric methods, they invented a non-parametric unsupervised method (Kose et al., 2015). Gallardo used convolutional deep neural networks (CNN), and recurrent neural networks (RNN) have also been patented as an algorithm to create an analytics engine for detecting medical frauds (Gallardo, 2017). Van Capelleveen et al. used multivariate clustering and outlier detection along with an expert system to detect fraud cases in Medicaid dental claims. A group of experts manually reviewed the flagged cases and validated 71% of cases proved to be fraudulent, a considerable improvement over the conventional method, which is, on average, detect 10% of fraud cases (van Capelleveen et al., 2016).

Summary of examples ML application to detect quality issues in different industries

Table 1 summarizes the studies discussed in the literature review.

Table 1. Summary of literature review of the application of ML to detect quality issues.

Item	Description
Industry	Food
Quality Problem	Detect the quality issues in Salmon
Data Type	Unstructured
Inputs	Images
ML Algorithms Applied	• Support Vector Machines (SVM) with a nonlinear kernel function (RBF) [B][1] • Nearest-neighbor classification
Note and Reference	(Sture, 2015)
Industry	Food
Quality Problem	QC test for olive oils
Data Type	Structured and Unstructured
Inputs	Chemical and visual measures
ML Algorithms Applied	• Naïve Bayesian [B] • SVM • ANN • k-Nearest Neighbors (kNN) • Linear Discriminate Analysis (LDA) • Decision Tree
Note and Reference	Researchers first used Principal Component Analysis to reduce the number of predictors from 32 to 8 (Ordukaya & Karlik, 2017)
Industry	Food
Quality Problem	Detect quality issues in French fries
Data Type	Unstructured
Inputs	Images
ML Algorithms Applied	Vision processing techniques augmented by an ML method based on neural networks
Note and Reference	(Lotfi et al., 2008)
Industry	Food
Quality Problem	Measure the quality of mixed raisins
Data Type	Unstructured
Inputs	Images
ML Algorithms Applied	• SVM [B] • ANN
Note and Reference	First used Principal Components Analysis (PCA) method to reduce 146 predictors to an optimal level prediction (Karimi et al., 2017)
Industry	Food
Quality Problem	Predict wine quality
Data Type	Structured
Inputs	Physicochemical characteristics
ML Algorithms Applied	• SVM [B] • NN
Note and Reference	First used linear regression to measure the dependency of the target variable (wine quality) on predictors (Gupta, 2018)

[1] [B] best performing algorithm among tested ones

Item	Description
Industry	Food
Quality Problem	Predict the quality of the beer (defined by the intensity levels of sensory descriptors like perceived foam–related parameters and beer color)
Data Type	Structured and Unstructured
Inputs	Physicochemical characteristics
ML Algorithms Applied	• ANN • Regression
Note and Reference	Researchers first used principal component analysis to identify the most relevant predictors. The final resulted in an R = 0.91correlation to predict the quality of the beer (Gonzalez Viejo et al., 2017)
Industry	Manufacturing
Quality Problem	Detect rare quality defects in a high-conformance manufacturing environment
Data Type	Structured
Inputs	Sensor data
ML Algorithms Applied	• least absolute shrinkage and selection operator (LASSO) • Logistic Regression (LR)
Note and Reference	Supervised Learning (Escobar & Morales-Menendez, 2018)
Industry	Manufacturing
Quality Problem	Predict potential quality failures of a process
Data Type	Structured
Inputs	Process and sensor data
ML Algorithms Applied	N-dimensional Euclidean distance-based scoring algorithms
Note and Reference	Resulting in a potential $300 million annual saving (RTInsights Team, 2017)
Industry	Manufacturing
Quality Problem	Detect manufacturing quality issues
Data Type	Structured
Inputs	QC data
ML Algorithms Applied	• Deep Learning (deep restricted Boltzmann machine and the stacked autoencoder) [B] • Shallow learning (feed-forward neural network with one hidden layer and the least-squares SVM with no hidden layers)
Note and Reference	Researchers proved deep learning methods outperforming shallow learning methods. They also found the performance of deep learning algorithms is proportional to the sample size (Bai et al., 2017)
Industry	Manufacturing
Quality Problem	Predict the quality of intermediate products in interlinked manufacturing processes
Data Type	Structured
Inputs	Process and sensor data
ML Algorithms Applied	A hybrid model of unsupervised (k-Means) and supervised (k-nearest neighbor [B] and SVM) algorithms
Note and Reference	With the k-NN algorithm (k=11), researchers achieved a 97% accuracy level in prediction (Lieber et al., 2013)
Industry	Manufacturing
Quality Problem	Improve the product and process quality in the manufacturing process
Data Type	Structured
Inputs	Process data
ML Algorithms Applied	The hybrid model consists of unsupervised (Cluster Analysis) and Supervised ML algorithms (SVM)
Note and Reference	(Irgens et al., 2012)

Item	Description
Industry	Manufacturing
Quality Problem	Predict the quality of welding
Data Type	Unstructured
Inputs	Arc sound and images
ML Algorithms Applied	• J48 [B] • Random Forest
Note and Reference	88.69% accuracy rate for J48 vs. 70.78% for random forest (Sumesh et al., 2015)
Industry	Manufacturing
Quality Problem	Monitor quality issues in plastic injection molding
Data Type	Structured
Inputs	Process data (cycle time, metering time, injection time, barrel temp., cushion, and injection velocity)
ML Algorithms Applied	SVMs (C-SVM and v-SVM) radial basis function (RBF) NNs
Note and Reference	Accuracy results ranged from 70.83% up to 99.16%. SVMs have a performance advantage over NNs (Ribeiro, 2005).
Industry	Call Center
Quality Problem	Detect quality issues (e.g., anger or delay) and categorize sentiment and emotions
Data Type	Unstructured
Inputs	Calls, Text
ML Algorithms Applied	Rule-based natural language processing (NLP)
Note and Reference	(Peddamuthu & Srivastava, 2014)
Industry	Call Center
Quality Problem	Predict the quality of service in a call center
Data Type	Structured
Inputs	Call Performance Evaluation data
ML Algorithms Applied	• CART [B] • Multi-layer perceptron (MLP) • Linear NN • Probabilistic NN • SVM (with a third-degree polynomial kernel) • Hybrid decision tree-ANN
Note and Reference	Over 80% of customer service satisfaction prediction accuracy and close to 90% in predicting business need satisfaction (Paprzycki et al., 2004)
Industry	Hospitality
Quality Problem	Measure service quality and customer satisfaction
Data Type	Unstructured
Inputs	Text
ML Algorithms Applied	• Aspect Sentiment Analysis • Decision Trees
Note and Reference	(Yussupova et al., 2016)
Industry	Utilities
Quality Problem	Detect smart-grid Power Quality Events (PQE)
Data Type	Structured
Inputs	Network data
ML Algorithms Applied	Discrete Wavelet Transform (DWT)
Note and Reference	Ucar, Alcin, Dandil, & Ata, 2018)

Item	Description
Industry	Financial services
Quality Problem	Detect inaccurate coding in financial contribution
Data Type	Structured
Inputs	Financial data
ML Algorithms Applied	• Adaboost supported by ensemble classifier [B] • SVM
Note and Reference	Supervised Learning (Blomquist & Möller, 2015)
Industry	Medical
Quality Problem	Improve the accuracy of quality assessment of MRI-derived data conducted by medical professionals
Data Type	Unstructured
Inputs	Medical images
ML Algorithms Applied	• Kernelized SVM • Gradient Boosted Decision Trees classifiers
Note and Reference	Supervised learning using over 7,500 human-rated samples resulting in a human workload reduction by 30-70% (Petrov et al., 2017)
Industry	Natural Resource
Quality Problem	Measure water quality
Data Type	Structured
Inputs	Lab data
ML Algorithms Applied	• ANN [B] • SVM • Classification and regression trees (CART) • LR
Note and Reference	Very large sample size (10 years' worth of data from 20 reservoirs in Taiwan) (Chou et al., 2018)
Industry	Telecommunication
Quality Problem	Estimate the mobile phone provider service quality using social media
Data Type	Unstructured
Inputs	Twits from Twitter
ML Algorithms Applied	Naïve Bayes
Note and Reference	Lack of proper validation method of test results (Calvin & Setiawan, 2014).
Industry	Telecommunication
Quality Problem	Predict if a software module is fault-prone or not
Data Type	Structured
Inputs	Software metrics
ML Algorithms Applied	• The SPRINT decision tree [B] • CART decision tree
Note and Reference	They found the way SPRINT decision tree leverages a unique tree pruning technique based on the Minimum Description Length (MDL) principle improves the accuracy and stability of the model (Khoshgoftaar & Seliya, 2003)
Industry	Software
Quality Problem	Predict which module in software has a higher chance of failure.
Data Type	Structured
Inputs	Software metrics

Item	Description
ML Algorithms Applied	• SVM [B] • LR • KNN • Multi-layer perceptrons (MLP) • Radial basis function (RBF) • Bayesian belief network (BBN) • Naïve Bayes (NB) • Random forest (RF) • Decision tree (DT)
Note and Reference	(Elish & Elish, 2008)
Industry	Healthcare
Quality Problem	Detect fraud in outpatient clinics claims
Data Type	Unstructured
Inputs	Hospital visit information
ML Algorithms Applied	• Unsupervised clustering methods: • Decision tree
Note and Reference	Used conditional probability distributions of the composite degree of anomaly (CDA) to adjust the risk score (Shin et al., 2012)
Industry	Healthcare
Quality Problem	Detect fraud in electronic claim data
Data Type	Unstructured
Inputs	Hospital visit information
ML Algorithms Applied	Non-parametric unsupervised method
Note and Reference	AHP was used for weighting (Kose et al., 2015)
Industry	Healthcare
Quality Problem	Detect fraud in electronic claim data
Data Type	Unstructured
Inputs	Hospital visit information
ML Algorithms Applied	• convolutional deep neural networks (CNN) • recurrent neural networks (RNN)
Note and Reference	Patented proprietary algorithm (Gallardo, 2017)

Application of ML to fraud in Medicaid and its challenges

Medicaid fraud analytics is the most common application of analytics and ML in Medicaid.

Most ML methods used for fraud detection in Medicaid are based on unsupervised methods. In supervised Medicaid fraud detection techniques, subject matter experts use prior information on class membership to select a set of training data (Travaille et al., 2011). This training data set is used by the algorithm to label each reviewed value as a suspicious or legal item. For example, subject matter experts review a set of paid and denied Medicaid claims and label each one legitimate or fraudulent. There are many supervised techniques listed as potential algorithms for Medicaid Fraud detection, including (Copeland et al., 2012) (Li et al., 2008) (Phua, C., Lee, V., Smith-Miles, K. and Gayler, 2005) (Valdes & Skinner, 2000), (Getchius, 2014) (Li et al., 2008):

- Multi-layer perceptron network
- Artificial Neural Network including Backpropagation (backward propagation of errors)
- Support Vector Machine
- Decision Trees
- Fuzzy Logic
- Bayesian Network including Bayesian Belief Network (BBN)
- Probable Graph Model (PGM)
- Linear Gaussian Model

Researchers also presented some innovative methods to evaluate supervised methods' performance by calculating their return on investment and costs versus benefits. At least two metrics are introduced in research to measure the performance of anomaly detection methods tradeoff between detection probability and false alarm ratio and the tradeoff between false alarm ratio and detection delay (Siris & Papagalou, 2004). Phua and the team described how statistical methods should be classified, utilized, interpreted, and validated to detect healthcare fraud. They compared statistical methods applied to health care fraud detection by focusing on, reviewing, and analyzing the investigations conducted in this field (Phua, C., Lee, V., Smith-Miles, K. and Gayler, 2005). While supervised methods are more accurate in detecting previous fraud types, they also suffer from some shortcomings, primarily when used to detect Medicaid fraud. The shortcomings include:

- Bolton and hand (2001) indicated that supervised methods' accuracy relies on the accurate identification of fraudulent and non-fraudulent transactions in the historical dataset where information is missed or limited.
- Creating training data is a tedious task and requires expensive subject matter experts to develop training data sets (especially nonfraudulent or legal data sets). One way to overcome this burden is to create just fraudulent datasets, which are much more limited.
- The constant change in the healthcare landscape changes the claim patterns. This includes demographical changes (e.g., aging population, shrinkage of middle-class, baby boomer retirement) and policy changes (e.g., the Affordable Care Act). As a result, there is a constant need to update training data sets.
- Periodic changes in healthcare systems force the users of supervised methods to update their training data sets regularly. Examples of such changes include implementing the hospital's performance-based payment, HIPAA 5010, and ICD-10 code migrations.

Unsupervised Medicaid fraud detection techniques do not assume prior class labels of legitimate or fraudulent behavior (Travaille et al., 2011). These algorithms mine the entire dataset and try to find a Medicaid data pattern and identify outliers. For example, an unsupervised algorithm could analyze the utilization of a specific prescription drug among a group of Medicaid beneficiaries and identify the anomalies. All anomalies then need to be researched and validated. The important fact about all probabilistic fraud detection models is that they will find likely fraud cases, and each case requires future research and validation. These algorithms' strengths are their sensitivity (detecting most true positives) and specificity (reducing false positives). Examples of unsupervised methods that could be used for fraud detection (including Medicaid fraud) are Anomaly Detection, K-mean, Benford’s Law, Linear Regression, Modified Batch Library Method (MBLM), Adaptive threshold algorithm, and CUSUM (Cumulative SUM) algorithms (Heino & Toivonen, 2003), (Siris & Papagalou, 2004), (Issa & Vasarhelyi, 2011), (Tsung et al., 2007). Unsupervised Medicaid Fraud algorithms also have strengths and weaknesses. A few of these are listed below:

- The neural network is an excellent method to handle complex data sets and manage noisy data, but its black-box approach limits the researchers' understanding of how a system works. (Li et al., 2008)
- Anomaly Detection is not robust toward the number of metrics used to flag fraudulent cases. This means too many metrics negatively impact the anomaly detection method’s efficiency. When metrics features are increasing, more cases will be flagged as potential fraud cases, increasing the number of false-positive and making the process ineffective (Copeland et al., 2012).
- A decision tree is useful for handling missing data and generates rule from a tree (white-box approach) but is not suitable for complex datasets. (Li et al., 2008)
- Fuzzy Logic allows approximate reasoning but is difficult to tune and lack sufficient learning capability (Li et al., 2008)
- Genetic Algorithm could be used very well for systematic random search but is difficult to tune.

Application of ML to detect quality issues in FFS Medicaid claim adjudication process

The researcher found several instants in the literature on applying BDA and ML in detecting fraud in many industries, including Medicaid, and finding quality issues in several industries. However, the researcher did not find any reference in the literature (focusing on master and Ph.D. dissertations and peer-reviewed journals) about applying ML to detect quality issues in adjudicating Medicaid FFS claims. Conventional methods of quality control, including statistical sampling methods and acceptance sampling (e.g., military standards) is still the most common method to control and manage the quality of operation in healthcare, from the claim adjudication process to pharmaceuticals (Fu et al., 2004) and (Borget et al., 2006).

How to prevent the PHI from potential breaches

As explained in chapter 1, PHI's protection is a stringent requirement enforced by the HIPAA law and impacting all the research using health information, including studies on Medicaid claim and payment data. Researchers proposed many methods to protect PHI, but all follow one of these two approaches: “PHI de-identification” and “Synthesizing Data.” In the next few pages, each procedure is briefly introduced, and the justification for this research method is provided.

PHI de-identification approach: in this approach, the researcher starts with a dataset containing actual PHI and then use different tools and techniques to either remove the identifiable data elements or scramble or mask them in a way they cannot be used to identify any PHI data element. In his book, El Emam explained in detail the process by which a researcher should de-identify PHI. Per him, this process has two significant steps (El Emam, 2013):

1. Prove that the applied method has a negligible chance that allows any identification of protecting data by the data recipient in any stage of the data processing
2. Provide proper documentation including methodology and test results that prove such determination

Researchers have offered many advanced methods to de-identify PHI, including combining knowledge-driven (dictionaries and rules) and data-driven (machine learning) methods (Dehghan et al., 2015). Neamatullah and colleagues offered another way to use lexical look-up tables, regular expressions, and simple heuristics to locate PHI and replace them with imitated data (Neamatullah et al., 2008). The Office for Civil Rights classifies all PHI de-identification methods under two major categories (The Office for Civil Rights & Malin, 2012), as depicted in Figure 5.

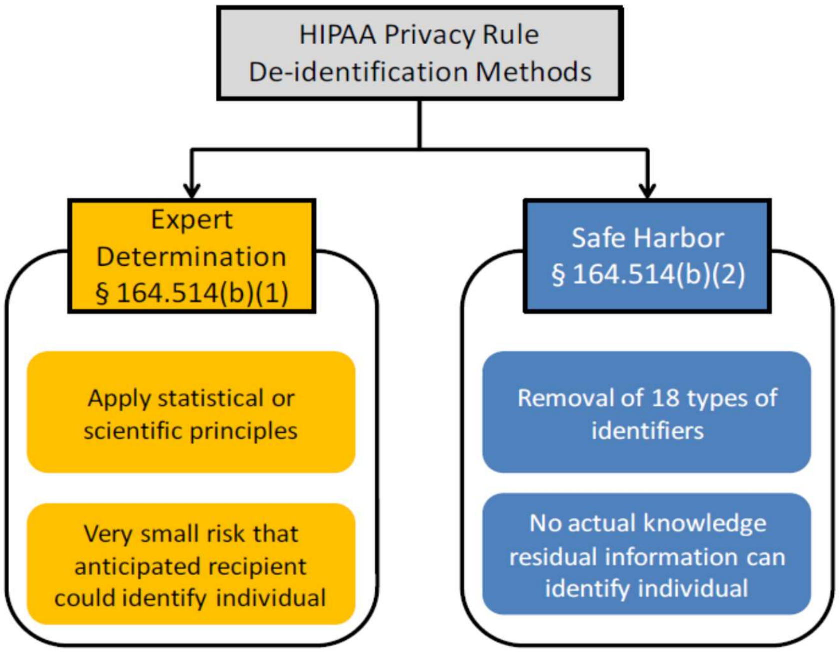

HIPAA Privacy Rule
De-identification Methods
Expert Determination § 164.514(b)(1)
Apply statistical or scientific principles
Very small risk that anticipated recipient could identify individual
Safe Harbor § 164.514(b)(2)
Removal of 18 types of identifiers
No actual knowledge residual information can identify individual

Figure 5. Two methods to achieve de-identification per the HIPAA Privacy Rule.

1. Expert Determination" method: in these methods, an expert with proper subject matter expertise and knowledge of statistical and scientific principles ensures that the method used would not allow any PHI disclosure or decrease the chance of PHI disclosure to a minimal amount and documents the technique.
2. Safe harbor: in this method, the researcher finds and removes any instance of 18 PHI data elements as described in chapter 1 or replace them with a dummy data element.

Meystre and the team conducted a comprehensive literature review and compared 18 methods in de-identifying PHI information. They concluded all methods have some strengths, but neither of them obliterates the possibility of data disclosure, and de-identified data may have less value for researchers due to the removal of information-rich contents (Meystre et al., 2010).

Synthesizing Data: in this approach researcher starts with understating the behavior of the actual data (e.g., population proportion, underlying distribution, descriptive statistics, etc.). The next step is to generate a dummy dataset from scratch without using any PHI input (e.g., using a computer algorithm). The advantage of this approach is that it eliminates the chance of any PHI disclosure. The biggest drawback of using a synthesized dataset is the limitation of multivariate modeling and model validation. As explained in the CMS's DE-SynPUF codebook, when the dynamic relationships between variables are altered (in this case, demographic, clinical, financial, and provider data), analyses from multivariate modeling should be interpreted with caution because the generated dataset may or may not inherit all the critical characteristic of the original dataset (The Centers for Medicare and Medicaid Services, 2013a). The researcher uses this method for data preparation in this research to eliminate all concerns about the privacy and confidentiality of the PHI.

Criteria to consider before selecting an ML algorithm

There are quite a few supervised algorithms in the literature, and the list is evolving. Not all these methods are appropriate for this research. The researcher looked at the literature to find some guidelines to help us select an initial list for the study, and then the results of the preliminary research guide the final list.

- *Accuracy*: perhaps the most popular metric to select an appropriate machine learning algorithm is the model accuracy, which is defined as total true positives and true negatives of all results produced by an algorithm. Numerous articles compared the accuracy of different supervised classifiers in solving different types of problems. As an example, Singh and Kaur compared the accuracy of J48 and REP Tree to predict the performance of computer science students and found J48 outperforms REP Tree in model accuracy 67.37% to 56.78% (Singh & Kaur, 2016).
- *Stability*: an algorithm is defined as β-stable (or stable, in general) when its losses incurred by the corresponding hypotheses on two similar but different datasets are equal or less than β (Mohri et al., 2014). There are specific limitations to how big β could grow and still a training algorithm to be convergence. Convergence to a particular measure is required for the learning algorithm to have an endpoint. Convergence to the optimal point is necessary to ensure the algorithm effectively solves the problem (it stops when it finds the global optimum or a reasonable local optimum).
- *Interpretability vs. prediction accuracy:* there is an inherited trade-off between the statistical models' interpretability and prediction accuracy (James et al., 2015). Many flexible and accurate ML algorithms (e.g., SVM and NN) lack interpretability. Such algorithms are black-

box approaches and are not good choices if there is a need to look inside the box. When there is a lack of trust in the ML model, or stakeholders demand the model logic to be clearly explained, the choice is to compromise the flexibility and choose a more interpretable model (e.g., LASSO).

- *Bias-variance trade-off:* In the context of ML, bias is the overall fitness of the model to the training data, and variance is the overall fitness of the model to the test and unseen data. As a rule of thumb, as model flexibility and complexity increase, the bias will decrease, and variance will increase, so the complex models (e.g., NN and SVM) have an inherited tendency to over-fit and should be used with caution (James et al., 2015). Brian and Webb found the size of the dataset may impact variance and bias in an ML model. Specifically, the hypothesized variance can be expected to decrease as training set size increases, but no apparent effect of training set size on bias was observed (Brain & Webb, 1999). These results have profound implications for data mining from large data sets, indicating that developing effective learning algorithms for large data sets is not merely a matter of finding computationally efficient variants of existing learning algorithms.
- *The impact of data reduction*: one way to increase the efficiency of a machine learning model is to appropriately reduce the version of data or a lower number of attributes. El-hasnony and colleagues compared the accuracy of classification nine algorithms using data reduction techniques like Correlation Feature Selection (CFS), Rough Set Attribute Reduction (RSAR), Fuzzy Rough Feature Selection (FRFS), PCA, and gain ratio. These are the list of classification algorithms they tested in their research: C4.5, fuzzy rough nearest neighbor, Multi-layer perceptron (MLP), Nearest-neighbor-like algorithm using non-nested generalized

exemplars (NNGE), Fuzzy nearest neighbor, sequential minimum optimization (SMO), classification via clustering, NB-tree and Naïve Bayes. They concluded that fuzzy rough feature selection outperforms rough set attribute selection, gain ratio, correlation feature selection, and principal components analysis (El-hasnony et al., 2015). The importance of this study is not only in the selection of the ten popular classification techniques (that could be used as a data input to select the most commonly used classifier algorithms for this study) but also in understanding the performance of different data reduction techniques in data validation and testing phases.

- *Model fitness*: Akaike's Information Criterion Akaike's information criterion (AIC) is a statistical measure of the goodness of fit for a particular model. It maximizes the expression $-2(LL+k)$ where k is the number of features, and *LL* is the maximized value of the log-likelihood function for the given model. The smaller the AIC, the better the model fits the data. Because of the k term, the smaller number of model parameters is favored. (Dean, 2014)

Summary of literature review in applying ML to detect the quality issue and Medicaid fraud and select supervised algorithms for this research

In previous sections, detail of several use cases was provided. These researches were related to ML's application to solve quality problems and detect Medicaid fraud cases. Table 2 summarizes the appearance of various ML algorithms in these researches as a tested algorithm and, in some cases, as the best performing one.

Table 2. ML algorithms in used literature to solve quality problems or detect Medicaid fraud

Algorithm	Frequency of application	Times listed as the best performer
Decision Tree	9	
Support Vector Machines	9	2
Logistic Regression	7	
Neural Network	6	1
Naïve Bayes	4	1
k-Nearest Neighbor	3	1
Random Forest	2	
Discriminant Analysis	1	
Gradient Boosting	1	
Others	6	1

These are a few lessons learned from the researches reviewed and summarized on the application of ML in detecting quality problems (in general) and Medicaid fraud:

- The most common application of machine learning in the examined cases is to "detect quality issues through supervised learning." This problem could be formulated as a classification problem. The second most common application was to "predict quality" using supervised or semi-supervised learning. Depends on the nature of the quality measure (variable vs. attribute), the problem could be formulated as a regression or a classification problem.
- Supervised learning has more applications in detecting quality issues. Unsupervised learning methods are mainly based on unsupervised methods. This is due to the changing nature of the fraud patterns and the cost of data labeling.
- There is no magic bullet for the best algorithm, and each algorithm may perform better in solving a specific problem. The most popular choices among algorithms are Decision Tree

(various configurations for both supervised and unsupervised learning), SVM (for supervised learning for binary classification problems), Logistic Regression (supervised), NN (for unsupervised and semi-supervised learnings), and Naïve Bayes (supervised).

- Complex (aka black box) algorithms like SVM and NN may perform better in many cases, but they are hard to explain. Simpler models like decision trees also have disadvantages, including low stability. Computation time for complex models tends to be higher. SVM requires massive computational power for mixed and massive datasets.
- The biggest challenge in leveraging supervised learning is the labeling process, usually a tedious and expensive process.
- Principal Components Analysis (PCA) is frequently used to reduce the number of features without compromising the model performance.

Based on all these studies, the researcher proposes to use the following algorithms to build, test, and validate our supervised model:

1. Decision tree with two configurations Entropy (DTE) and Gini coefficient (DTG)
2. Random forests with two configurations Entropy (RFE) and Gini configuration (RFG)
3. Naïve Bayes (NN)
4. K Nearest Neighbor (kNN)
5. Logistic Regression (LR)
6. Neural Network (NN)
7. Discriminant Analysis (DA)
8. Gradient Boosting (GB)

Statistics for Model Selection

Our research problem uses both ratio and categorical data (as independent value) and predicts a categorical data type target. As a result, different algorithms' performances should be compared using proper statistics that work with categorical data. The most common statistics to compare supervised learning models' performance with categorical data are accuracy, recall, and precision extracted from the confusion matrix using the formula explained in Table 3 (Vihinen, 2012).

Table 3. Confusion Matrix and its related metrics

		Actual		Measure
		+	-	
Predicted	+	True Positive (TP)	False Positive (FP)	Positive Predictive Value (PPV) = TP/(TP+FP)
	-	False Negative (FN)	True Negative (TN)	Negative Predictive Value (NPV) = TN/(FN+TN)
Measure		Recall or Sensitivity = TP/(TP+FN)	Precision or Specificity = TN/(TP+TN)	Accuracy= (TP+TN)/(TP+FP+FN+TN) F-measure (F1 Score) = 2.(Precision.Recal)/(Percision+Recal)

There are many metrics to select the optimal model to solve an ML problem with the massive dataset, including:

- Computation and tuning time
- Algorithm performance metric (e.g., accuracy, specificity, recall, balanced or weighted accuracy, F1 score, prediction power, etc.)
- Explainability of the model and the associated algorithm, stability, and scalability (Vijaya Beeravalli, 2018), (Rácz et al., 2019). T

here are also many other metrics used to precisely evaluate the performance of supervised learning models. Each has its advantages and disadvantages based on the problem statement and the nature of the data. Researchers have offered various metrics for model validation and selecting the most appropriate model for each particular problem (Shah et al., 2016), (Marc-Oliver Arsenault, 2017).

- Average squared error (ASE): The sum of squared errors (SSE) divided by the number of observations.
- Area under the curve (C-statistic): A measure of goodness of fit for a binary outcome. It is the concordance rate, and it is calculated as the area under the curve.
- Area under the ROC curve: The area under the ROC curve presents accuracy.
- Captured response: The number of response events in each bin divided by the events' total number.
- Cumulative captured response: Cumulative of the captured response.
- Kolmogorov- Smirnov statistic (KS): A goodness-of-fit statistic representing the maximum separation between the model ROC curve and the baseline ROC curve.
- F1 score: The weighted average of precision (positive predicted value) and recall (sensitivity). It is also known as the F-score or F-measure.
- False discovery rate: he expected proportion of type error I - incorrectly reject the null hypothesis (false positive rate).
- Gini: A measure of the quality of the model. It has values between -1 and 1. Closer to 1 is better. It is also known as Somer's D.
- KS (Youden): A goodness-of-fit index representing the maximum separation between the model ROC curve and the baseline ROC curve.

- Lift: A measure of the advantage (or lift) uses a predictive model, compares the target to when the model is not used and improves it. It is a measure of the predictive model's effectiveness calculated as the ratio between the results obtained with and without the predictive model.
- Misclassification (Event): Considers only the classification of the event level versus all other levels. Thus, a non-event level classified as another non-event level does not count in the misclassification. For binary targets, these two measures are the same. It is computed in the context of the ROC report. That is, at each cutoff value, this measure is calculated.
- Misclassification (MCE): A measure of how many observations are incorrectly classified for each response variable's value.
- Multiclass log loss: The loss function applied to a multinomial target. It is the negative log-likelihood of the true labels given a probabilistic classifier's prediction.
- ROC separation: ROC separation allows a ROC-based cutoff to compare the model's performance under different accuracy ranges.
- The root-mean-square deviation (RMSD) or root-mean-square error (RMSE): is the square root of the average of squared errors and measures the differences between the predicted and observed value.

Accuracy is the most popular measure to compare supervised learning models' performance for categorical targets. It is easy to calculate and understand. However, when dealing with rare events as the target group (dependent variable), accuracy could be misleading. This is because of many true negatives in the denominator; accuracy would be a near-perfect measure even for low performing models. Recall, precision, and F-measure (a balanced measure of precision and recall) focus on the target category (a rare event) and are more relevant to the problem. For our specific problem statement, recall is the most critical measure. This is because quality problems in a massive claim adjudication data are rare events, and it is paramount not to lose the chance to detected all of them. Precision has less importance; however, having many false positives will result in extra work by the quality team to manually research each false positive. To ensure our model considers the number of false positives as performance metrics, the researcher selected the F-measure (F1 score) as the second most essential metrics to compare model performance without losing sight of the recall. ROC separation is another popular metric; however, research shows F1-score's superiority to ROC separation in selecting the best performing model when dealing with imbalanced data (Yahya, 2018). The datasets used for this research are imbalanced datasets; thus, F1-score is selected as the second performance metric.

Chapter Summary

This chapter provided some background information about the Medicaid program and its critical impact on the lives of millions of people in the U.S. and its overall financial implications on expenditure budget on a national and state level. Medicaid has two models: MCO and FFS. The researcher also reviewed two models' characteristics and shows how FFS, the smaller subset of Medicaid, contributes to the highest amount of potential fraud, waste, abuse, and quality issues mainly because, in this model, the risk is not transferred to the providers. The researcher reviewed the impact of machine learning and its importance on a global scale. The evolution of quality and machine learning, and then it was reviewed how ML could detect quality issues in the Medicaid FFS claim adjudication process. Different classes of machine learning methods were then discussed, and an extensive literature review was presented on applying machine learning algorithms to solve various quality problems. The conclusion was that detecting quality issues through supervised learning is a general approach in different industries and often formulated as a classification problem. It was also concluded no ML algorithm generally performs better than others. Still, SVM and Naïve Bayes (supervised learning) and NN (unsupervised and semi-supervised learnings) are very popular among researchers. Labeling is the most challenging part of a supervised learning process and immensely impacts the solution's accuracy. Constant changes in the healthcare landscape changes and claim patterns negatively impact fraud analytics solutions, which heavily rely on unsupervised methods. A few lessons about the performance of different algorithms were learned. For example, NN is useful in managing complex and noisy datasets, and the decision tree helps to handle missing data. It is still not suitable for complex datasets, and the Fuzzy Logic and Genetic Algorithm are challenging to tune.

The researcher reviewed two approaches to protect PHI (de-identification and synthesizing) and explained why the latter is more favorable for this dissertation. The literature was reviewed on building, training, and validating supervised learning models and proposed a unique research process and the selection criteria to choose an ML algorithm, including model Accuracy, model Stability, Interpretability-prediction trade-off, Bias-variance trade-off, and its impact data reduction. While unsupervised learning methods (including anomaly detection and clustering) have been frequently used to detect fraud in Medicaid claims, a supervised method is a more proper approach for our problem. This is because the starting point is a selection of labeled data (defective and non-defective Medicaid claims). Then models were built to find a similar pattern in the test dataset. As in fraud analytics, the pattern of fraud is usually unknown and frequently changing, so staring with a set of label data is not practical. The researcher concluded this chapter by selecting ten algorithms to compare and two primary performance metrics.

CHAPTER 3

METHODOLOGY

Chapter Overview

This chapter starts by reviewing the statement of the problem, research questions, and hypotheses. Then five steps of the research procedure are evaluated in detail, including the type of statistical tests used to validate that the synthesized data have the same error pattern as the actual one. This chapter concludes with some preliminary findings and a chapter summary.

Statement of the Problem

Conventional quality methods used to detect quality issues in the Medicaid FFS claim adjudication process only provide a population proportion estimate of the claims processed erroneously. This study aims to apply supervised learning as a method to detect erroneously adjudicated claims in the entire population and find out the most important feature and most effective supervised learning algorithm in detecting if an inpatient or outpatient Medicaid FFS claim has been erroneously adjudicated.

Research Questions and Hypotheses

The following research questions will be answered, and hypotheses will be tested to complete this study.

Research Question 1 (RQ1): What supervised machine learning algorithms can be used to determine Medicaid claim payment issues?

Research Question 2 (RQ2): What are the most critical measures to compare the performance of different machine learning algorithms (resulted from RQ1) for our problem?

Research Question 3 (RQ3): What are the claim attributes that could predict if a given FFS claim has been adjudicated correctly or erroneously (also known as predictors or independent variables)?

Research Question 4 (RQ4): Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid outpatient claims?

- Hypothesis 4.1: there is no significant difference among the feature importance score (Gini importance) of the predictors in identifying erroneously processed Medicaid outpatient claims:
  - $H_{0_{4.1}}$: $\mu_{1.1} = \mu_{1.2} = \cdots = \mu_{1.n}$ where μ_{1i} is the feature importance score (Gini importance) of the i-th outpatient predictor.
  - $H_{1_{4.1}}$: There is at least one predictor (j) for which μ_{1j} is not equal to the rest.

Research Question 5 (RQ5): Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid inpatient claims?

- Hypothesis 5.1: there is no significant difference among the feature importance score (Gini importance) of the predictors in identifying erroneously processed Medicaid inpatient claim:

- $H_{0_{5.1}}$: $\mu_{2.1} = \mu_{2.2} = \cdots = \mu_{2.n}$ where μ_{2i} is the feature importance score (Gini importance) of the i-th inpatient predictor.
- $H_{1_{5.1}}$: There is at least one algorithm (j) for which μ_{2j} is not equal to the rest.

Research Question 6 (RQ6): Is there any statistically significant difference among the selected supervised learning algorithms (the result of RQ1) in identifying erroneously adjudicated Medicaid outpatient claims (as measured by the result of the RQ2)?

- Hypothesis 6.1: there is no significant difference among the average recall of selected supervised learning algorithms in identifying erroneously processed Medicaid outpatient claims:
    - $H_{0_{6.1}}$: $\mu_{3.1} = \mu_{3.2} = \cdots = \mu_{3.10}$ ($\mu_{3.i}$: the mean recall of algorithm *i*).
    - $H_{1_{6.1}}$: There is at least one algorithm (j) for which $\mu_{3.j}$ is not equal to the rest.
- Hypothesis 6.2: there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid outpatient claims:
    - $H_{0_{6.2}}$: $\mu_{41} = \mu_{42} = \cdots = \mu_{4.10}$ ($\mu_{4.i}$: the mean F1-score of algorithm *i*).
    - $H_{1_{6.2}}$: There is at least one algorithm (j) for which $\mu_{4.j}$ is not equal to the rest.

Research Question 7 (RQ7): Is there any statistically significant difference among the selected supervised learning algorithms (the result of RQ1) in identifying erroneously adjudicated Medicaid inpatient claims (as measured by the result of the RQ2)?

- Hypothesis 7.1: there is no significant difference among the average recall of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims:
  - $H_{0_{7.1}}$: $\mu_{5.1} = \mu_{5.2} = \cdots = \mu_{5.10}$ (μ_{5i}: the mean recall of algorithm i).
  - $H_{1_{7.1}}$: There is at least one algorithm (j) for which μ_{5j} is not equal to the rest.
- Hypothesis 7.2: there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims:
  - $H_{0_{7.2}}$: $\mu_{6.1} = \mu_{6.2} = \cdots = \mu_{6.10}$ (μ_{4i}: the mean F1-score of algorithm i)
  - $H_{1_{7.2}}$: There is at least one algorithm (j) for which μ_{4j} is not equal to the rest.

Research Question 8 (RQ8): Are the most powerful predictors different between outpatient and inpatient claims? Are the most accurate algorithms in detecting erroneously paid claims different between the two types of Medicaid claims studied?

Research Design and Procedures

There are many popular books on ML, and each proposes a slightly different process to build, train, validate, and implement an ML model. Some of them focus on the statistical learning process, proof ability, and validity of an approach like “Machine Learning: A Probabilistic Perspective” by Kevin Murphy (Kevin P. Murphy, 2014). Some focus more on the technical aspects of implementing a scalable and effective model using proper technologies or tools like “Machine Learning: Hands-On for Developers and Technical Professionals” by Jason Bell (Bell, 2014). Machine learning researchers offer a verity of models and process to build an effective machine learning process and propose different steps, sometimes as little as three steps and sometimes up to 20 steps (Bernardi et al., 2019), (Levinger, 2019), (Niwratti, 2020), (Mayo, 2020), (Chang, 2017). In his book Finlay offers an iterative 10-step model construction process, which is an excellent example of how a predictive model could be built (Finlay, 2014):

1. Explore data landscape
2. Sampling and shaping
3. Data preparation (data cleansing)
4. Create derived data
5. Visualization and understanding
6. Preliminary variable selection
7. Pre-processing
8. Model construction
9. An iterative process to select the best model
10. Model completion

Bell summarizes the ML model building process in five steps: Planning, Developing, Testing, Reporting, Refining, and Production (Bell, 2014). Similarly, Witten and colleagues explained the ML model building process in five steps: data understanding, data preparation, modeling, evaluation, development (Witten et al., 2016). Building and implementing a machine learning model follow the same steps as any improvement project and Plan, Do, Study, Act (PDSA) cycle:

- Define the problem and parameters based on the plan (Plan)
- Collect, prepare, and label data, select split strategy, train, and test the model(s) (Do)

- Review the performance of the model(s), fine-tune the model (s), and check the performance of the modified model(s) (S)
- Execute the fine-tuned model(s) and start solving the business problem (A).

Using the lessons learned from the literature, the researcher proposes a three-step process to build, train, validate, and implement our model, as depicted in Figure 6:

1. Manage data, including:
    a. Selecting proper machine learning toolset
    b. Collecting and creating research dataset
    c. Feature Selection
2. Build, train, and test models, including:
    a. Dealing with rare events and sampling strategy
    b. Select the split strategy and cross-validation
    c. Train and test Decision Tree, Random Fore, and Gradient Boosting
    d. Review feature importance
    e. Train and test the rest of the selected models and review the results
3. Evaluate models' performances, including:
    a. Calculate and compare the performance of the models
    b. Evaluate the statistical difference among models' performances
    c. Review the results and discussions

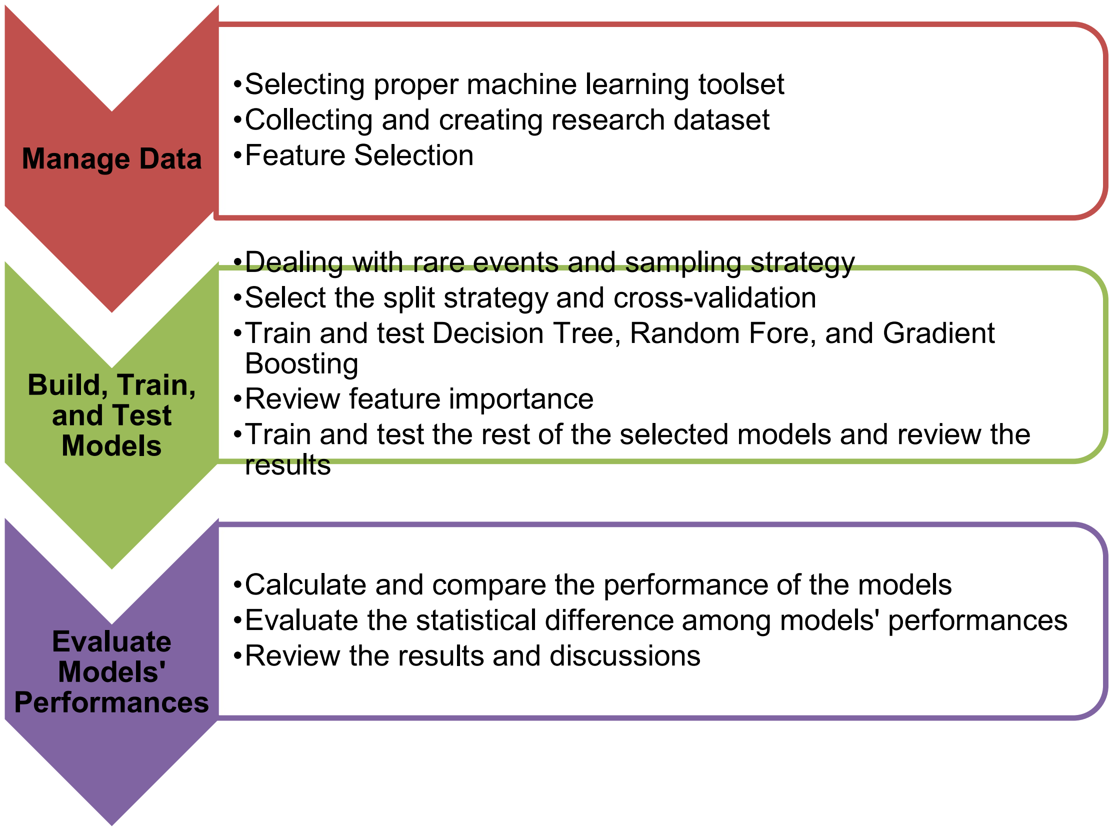

Manage Data
•Selecting proper machine learning toolset
•Collecting and creating research dataset
•Feature Selection
Build, Train, and Test Models
•Dealing with rare events and sampling strategy
•Select the split strategy and cross-validation
•Train and test Decision Tree, Random Fore, and Gradient Boosting
•Review feature importance
•Train and test the rest of the selected models and review the results
Evaluate Models' Performances
•Calculate and compare the performance of the models
•Evaluate the statistical difference among models' performances
•Review the results and discussions

Figure 6. Research procedure steps.

Table 4 describes research procedures and statistical methods used to test the research hypotheses, answer research questions, and relate to each research methodology step.

Table 4. Research procedures and statistical methods.

Research Question	Goal	Method	Research Step
RQ1	Selecting ML algorithms for this research	Literature review and select appropriate algorithms	Literature Review and Build, Train, and Test Models
RQ2	Selecting performance metrics to compare ML algorithms performance for the problem statement	Literature review and select appropriate metrics	Literature Review and Build, Train, and Test Models

Research Question	Goal	Method	Research Step
RQ3	Finding the most critical claim attributes (features) to predict the outcome (classify the claim correctly)	Review the data and PCA	Manage Data
RQ4 (H4.1)	Examine if there exist differences among predictors' predictability power in identifying erroneously processed Medicaid outpatient claims	Mean Decrease in Impurity (MDI) [Gini importance] and one-way ANOVA	Build, Train, and Test Models
RQ5 (H5.1)	Examine if there exist differences among predictors' predictability power in identifying erroneously processed Medicaid inpatient claims	Mean Decrease in Impurity (MDI) [Gini importance] and one-way ANOVA	Build, Train, and Test Models
RQ6 (H6.1, H6.2)	Examine if there are differences among the performance of selected algorithms to detect quality issues in outpatient claims	Classification Algorithms (Recall and F1-score), Gini Measure, and one-way ANOVA	Evaluate Models' Performances
RQ7 (H7.1, H7.2)	Examine if there are differences among the performance of selected algorithms to detect quality issues in inpatient claims	Classification Algorithms (Recall and F1-score), Gini Measure, and one-way ANOVA	Evaluate Models' Performances
RQ8	Review the differences among the feature importance and algorithms performance between two selected claim types	Review the results	Evaluate Models' Performances

Selecting proper machine learning toolset

The researcher reviewed several machine learning software, including open source predictive modeling applications, with the following criteria in mind:

- Ease of use and model building: Python and R with readily available libraries (e.g., Scikit-learn library) are usually prime choices to build machine learning models. The challenge is that using Python and R requires scripting skills. Non-scripting tools (e.g., KNIME and RapidMiner) do not need coding and usually are easier for model building.

- Cost-effectiveness: some applications offer compelling features (e.g., SAS Viya and Microsoft Azure ML); however, they could be expensive and cost-prohibitive for research projects. KNIME, RapidMiner, and Microsoft ML offer free or low-cost options for education purposes, with some limitations in the free version features. R and Python libraries are mainly free of charge.
- User community: a large user community, offers comprehensive support to the users. Among all tools, Python and R have the largest user community, in which a researcher can find examples of codes, similar problems, and tested solutions. Some (like KNIME and RapidMiner) have smaller user communities.
- Capabilities: It is essential to ensure ML algorithms selected for this research are supported by the chosen tool. As an example, available ML supervised classifier in KNIME (KNIME, n.d.) include: decision tree, NN, naive Bayes, gradient boosted trees, logistic regression, SVM, and decision tree ensemble (including random forests). Among all tools, Python, R, and SAS Viya offer the most supported algorithms targeted by this research.

The researcher tried the free version of several tools (namely KNIME, Microsoft Azure ML, SAS Viya in an educational mode with the predefined dataset, and Python and its libraries like Scikit-learn). He also contacted several other researchers and data scientists to gather their inputs about ML tools to solve problems with similar datasets. The summary of the results is shown in Table 5.

Table 5. Tool selection criteria.

Selection Criteria	Python	R	KNIME	Microsoft Azure ML	RapidMiner	SAS Viya
Ease of use	Moderate	Moderate	Excellent	Excellent	Excellent	Excellent
Cost-effectiveness	Excellent	Excellent	Excellent	Good	Moderate	Cost prohibitive
User community	Excellent	Excellent	Moderate	Good	Good	Very Good
Capabilities	Excellent	Very Good	Excellent	Very good	Excellent	Excellent

Python is a high-level and general-purpose programming language and very popular among data scientists as the top selection (Piatetsky, 2018). As a result, it has an excellent user community. It is easy to find examples of scripts that solved problems with Python. Its modular architecture and simple syntax improve its readability. Python is easy to install and performs well with complex datasets. Online documentation of Python features and libraries also is considerable and makes it easier to get the result. The researcher selected Python based on the criteria listed in Table 5. Many models are fully configured in the Scikit-learn library, and when there are features to choose to fine-tune a model, the researcher provides the selected features and the logic for such selection.

Collecting and Creating research dataset

To create a dataset for this research, the researcher has taken several steps to collect data from publicly available sources, protect the sensitive data, test our hypotheses, and examine our models' performance in a safe environment. All steps for collecting and creating a research dataset are explained in detail in this section to ensure an independent researcher can reproduce similar results.

Step 1- Select a source for the claim data

This dissertation focuses on finding quality issues in the Medicaid FFS claim data, containing sensitive data. An alternative to using real-world data is to use publicly available PHI-free data. As discussed in chapter two, there are two principal methods to protect PHI data: de-identifying a real dataset and synthesizing a dataset that behaves like a real one. CMS has created five massive datasets called Synthetic Public Use File (DE-SynPUF) (The Centers for Medicare and Medicaid Services, 2013b). To make such a dataset, CMS has selected five percent random samples of Medicare beneficiaries in 2008 and their claims from 2008 to 2010 and used them to generate millions of Medicare claim data samples for various claim types, as shown in Table 6.

Table 6. Tool selection criteria.

File Name	Type of Data	Claim Type	Number of the records per year		
			2008	2009	2010
Beneficiary Summary DE-SynPUF	Beneficiary	N/A	2,326,856	2,291,320	2,255,098
Inpatient Claims DE-SynPUF	Claim	Inpatient	547,800	504,941	280,081
Outpatient Claims DE-SynPUF	Claim	Outpatient	5,673,808	6,519,340	3,633,839
Carrier Claims DE-SynPUF	Claim	Medical	34,276,324	37,304,993	23,282,135
Prescription Drug Events (PDE) DE-SynPUF	Claim	Pharmacy	39,927,827	43,379,293	27,778,849

A unique identification number is assigned to each synthetic beneficiary to link synthetic claims to a synthetic beneficiary. This beneficiary ID carries no information about the enrollee or any patient records and is provided solely for reference and data processing purposes. Table 7 shows the list of features (data elements) for outpatient claims with their data types (The Centers for Medicare and Medicaid Services, 2013a):

Table 7. List of features (data elements) for outpatient claims.

#	Feature names	Labels	Data Type
1	DESYNPUF_ID	DESYNPUF: Beneficiary Code	Categorical/Nominal

#	Feature names	Labels	Data Type
2	CLM_ID	DESYNPUF: Claim ID	Categorical/Nominal
3	SEGMENT	DESYNPUF: Claim Line Segment	Categorical/Nominal
4	CLM_FROM_DT	DESYNPUF: Claims start date	Ordinal (Date)
5	CLM_THRU_DT	DESYNPUF: Claims end date	Ordinal (Date)
6	PRVDR_NUM	DESYNPUF: Provider Institution	Categorical/Nominal
7	CLM_PMT_AMT	DESYNPUF: Claim Payment Amount	Ratio/Real number (Payment)
8	NCH_PRMRY_PYR_CLM_PD_AMT	DESYNPUF: NCH Primary Payer Claim Paid Amount	Ratio/Real number (Payment)
9	AT_PHYSN_NPI	DESYNPUF: Attending Physician – National Provider Identifier Number	Categorical/Nominal
10	OP_PHYSN_NPI	DESYNPUF: Operating Physician – National Provider Identifier Number	Categorical/Nominal
11	OT_PHYSN_NPI	DESYNPUF: Other Physician – National Provider Identifier Number	Categorical/Nominal
12	NCH_BENE_BLOOD_DDCTBL_LBLTY_AM	DESYNPUF: NCH Beneficiary Blood Deductible Liability Amount	Ratio/Real number (Payment)
13-22	ICD9_DGNS_CD_1 – ICD9_DGNS_CD_10	DESYNPUF: Claim Diagnosis Code 1 – Claim Diagnosis Code 10	Categorical/Nominal
23-28	ICD9_PRCDR_CD_1 – ICD9_PRCDR_CD_6	DESYNPUF: Claim Procedure Code 1 – Claim Procedure Code 6	Categorical/Nominal
29	NCH_BENE_PTB_DDCTBL_AMT	DESYNPUF: NCH Beneficiary Part B Deductible Amount	Ratio/Real number (Payment)
30	NCH_BENE_PTB_COINSRNC_AMT	DESYNPUF: NCH Beneficiary Part B Coinsurance Amount	Ratio/Real number (Payment)
31	ADMTNG_ICD9_DGNS_CD	DESYNPUF: Claim Admitting Diagnosis Code	Categorical/Nominal
32-76	HCPCS_CD_1 – HCPCS_CD_45	DESYNPUF: Revenue Center HCFA Common Procedure Coding System 1 –	Categorical/Nominal
77	Year	Year	Ordinal (Date/year)

Table 8 shows the list of features (data elements) for inpatient claims with their data types (The Centers for Medicare and Medicaid Services, 2013a).

Table 8. List of features (data elements) for inpatient claims.

#	Variable names	Labels	
10	*OP_PHYSN_NPI*	DESYNPUF: Operating Physician – National Provider Identifier Number	Categorical/Nominal
11	*OT_PHYSN_NPI*	DESYNPUF: Other Physician – National Provider Identifier Number	Categorical/Nominal

#	Variable names	Labels	
12	*CLM_ADMSN_DT*	DESYNPUF: Inpatient admission date	Ordinal (Date)
13	*ADMTNG_ICD9_DGNS_CD*	DESYNPUF: Claim Admitting Diagnosis Code	Categorical/Nominal
14	*CLM_PASS_THRU_PER_DIEM_AMT*	DESYNPUF: Claim Pass-Thru Per Diem Amount	Ratio/Real number (Payment)
15	*NCH_BENE_IP_DDCTBL_AMT*	DESYNPUF: NCH Beneficiary Inpatient Deductible Amount	Ratio/Real number (Payment)
16	*NCH_BENE_PTA_COINSRNC_LBLTY_AM*	DESYNPUF: NCH Beneficiary Part A Coinsurance Liability Amount	Ratio/Real number (Payment)
17	*NCH_BENE_BLOOD_DDCTBL_LBLTY_AM*	DESYNPUF: NCH Beneficiary Blood Deductible Liability Amount	Ratio/Real number (Payment)
18	*CLM_UTLZTN_DAY_CNT*	DESYNPUF: Claim Utilization Day Count	Categorical/Nominal
19	*NCH_BENE_DSCHRG_DT*	DESYNPUF: Inpatient discharged date	Ordinal (Date)
20	*CLM_DRG_CD*	DESYNPUF: Claim Diagnosis Related Group Code	Categorical/Nominal
21-30	*ICD9_DGNS_CD_1 – ICD9_DGNS_CD_10*	DESYNPUF: Claim Diagnosis Code 1 – Claim Diagnosis Code 10	Categorical/Nominal
31-36	*ICD9_PRCDR_CD_1 – ICD9_PRCDR_CD_6*	DESYNPUF: Claim Procedure Code 1 – Claim Procedure Code 6	Categorical/Nominal
37-81	*HCPCS_CD_1 – HCPCS_CD_45*	DESYNPUF: Revenue Center HCFA Common Procedure Coding System 1 – Revenue Center HCFA Common Procedure Coding System 45	Categorical/Nominal

Step 2- Ingest the claim data

CMS's DE-SynPUF database contains five massive datasets and over 120 separate flat files. Using the data for any data manipulation and machine learning purposes require creating a database, ingesting all the data, and providing reporting services to query easily, sample, and study the data. With a database developer's help, a data pipeline was created with a reporting layer to ingest all 120 flat files. The separate files were then merged using the data pipeline's capabilities, creating five massive tables: one for beneficiary data and four for each claim type (Inpatient, Outpatient, Medical, and Pharmacy claims). Our database is a platform-agnostic solution using a web server and a firebase database that consists of four components: webserver in Express.js, UI/UX design using React, Postgres for persistence database, and Firebase real-time database as the message broker. Researchers do not necessarily need to create a data pipeline to use CMS's DE-SynPUF datasets. The purpose of the data pipeline for this project was to ensure our simple random sampling of the synthesized claims is genuinely random. This requires the entire population of synthesized claims data to be available for random sampling, and each member of the population to have the same chance of being selected. APPENDIX A provides the architecture designed for the data pipeline.

A researcher can use cluster sampling, which does not require the entire CMS's DE-SynPUF datasets to be ingested and is less costly. To do so, first, a random sample of files should be drawn from the 120 available files (depends on the study's scope). Then, the required claim samples can be drawn from the selected files. This could be done quickly in Excel. The drawback is that cluster sampling makes the statistical analysis more complicated and increases the sampling error. Research shows if the heterogeneity among clusters is low and homogeneity of the sampled item in each cluster is high, the accuracy will be suffered (Kerry & Bland, 1998). The 120 files available in the CMS's DE-SynPUF dataset are cultured so that claims for a particular beneficiary are in samples with the same number are in the same file as much as possible (The Centers for Medicare and Medicaid Services, 2019b). As a result, records within each file are similar (homogenous); however, the variation across clusters (files) is high, which means cluster sampling will increase error in the estimations.

Step 3: Study the selected dataset to verify it works for the Medicaid data

"Medicaid Analytic eXtract Data (MAX)" was used as the primary source of available public data to understand Medicaid inpatient and outpatient claims' typical coverage (Williams & Baugh, 2016). MAX is defined as a "research-ready data source for Medicaid and CHIP calendar year person-level data on eligibility, service utilization and payment information in the 50 states and the District of Columbia (DC)" (Williams & Baugh, 2016). The procedure codes (CPT and ICD codes) on our claim dataset were compared against the Medicaid coverage of several states, including California, New Hampshire, Arizona, Nevada, and Hawaii, to verify the state level coverage of the Medicare-covered inpatient and outpatient FFS claims (Hawaii Department of Human Services & Med-QUEST, n.d.), (NM Human Services Department (HSD), n.d.), (Arizona Health Care Cost Containment System, n.d.), and (Nevada Department of Health and Human Services, n.d.). As appropriate, the researcher used the state level to add some adjustments and try to make the dataset as generic as possible (Hawaii Department of Human Services & Med-QUEST, n.d.). In general, Medicaid FFS programs tend to cover more procedures than the standard coverage for Medicare FFS. As a second payer, Medicaid FFS covers all inpatient and outpatient procedures covered through Medicaid part A and B; however, the scope of Medicaid coverage is usually beyond Medicare, especially on other claim types like long term care. As a result, one can consider the Medicare Synthesized dataset equivalent to a subset of a generic Medicaid claim payment dataset. The impact of this difference in our research is that our predictive models will be more generic. If the predictive models resulted inform this research are applied to Medicaid FFS payment data, the applicability of the model will not suffer because the same predictors (features) to exist in any Medicaid FFS payment dataset. The difference is that an actual Medicaid FFS data set has more features so the predictive model could improve by adding new features. The researcher will discuss how one could improve this research using a real or synthesized Medicaid FFS payment dataset in the recommendations section.

Step 4- Make Synthetic claim data available for sampling

The designed data pipeline was essential to ingest and merged split files into a coherent dataset. However, running queries in the pipeline is not fast, and it is costly. To overcome this challenge and draw random samples from various tables in the dataset quickly, the researcher, with the help of a database admin, moved the datasets into Amazon Web Services (AWS) cloud. Two AWS services were utilized for this purpose:

1. Amazon DynamoDB: a fully managed, multi-region, multi-master database with built-in security, backup and restore, and in-memory caching for applications requiring massive scale of handling data (Amazon, n.d.-b).
2. Amazon Athena: an interactive query service that makes it easy to analyze Amazon's data restored in Amazon DynamoDB tables using standard SQL (Amazon, n.d.-a).

Moving the cohesive datasets to the AWS cloud allowed the researcher to design and run complex queries and test various scenarios with proper speed and accuracy. The dissertation committee and peer reviewers can request access to this dataset for independent validation and verification of the data and the study results. An instruction to access AWS instant established for this research is provided in APPENDIX B.

Step 5- Select a random sample from the dataset

Our ingested dataset consists of over 1.3 million inpatient claims and close to 16 million outpatient claim payments. Two random samples were selected to build our supervised learning models, as illustrated in Table 9. The researcher used the SQL queries like the following example, which pulls 5,00 random sample of inpatient claims for the year 2010:

```
SELECT * FROM "de_synpuf"."inpatient_claims"
WHERE "de_synpuf"."inpatient_claims"."clm_from_dt" >= 20100101 and
"de_synpuf"."inpatient_claims"."clm_from_dt" <= 20101230
ORDER BY random() limit 5000;
```

Independent researchers can use SQL query or Excel function (RAND and RANDBETWEEN) and get the same results.

Table 9. Selected sample size.

Claim Type	Category	2008	2009	2010	Total
Inpatient Claims	Total Population in DB	547,800	504,941	280,081	1,332,822
	Initial Sample Size	5,478	5,049	2,801	13,328
Outpatient Claims	Total Population in DB	5,673,808	6,519,340	3,633,839	15,826,987
	Initial Sample Size	56,738	65,193	36,338	158,269

Step 6- Cleansing sampled dataset

data cleansing is an extensive task in real data and mainly consists of finding and dealing with missing data and anomalies. Our ingested dataset consists of over 1.3 million inpatient claims and close to 16 million outpatient claim payments. Two random samples were selected to build our supervised learning models, as illustrated in Table 9. Many such issues are not prevalent in our synthesized dataset. Two steps were taken in our datasets:

a. Eliminate negative payments: in the world of Medicaid FFS claims, a negative payment amount is usually an indication that the provider was overpaid in the past and had a negative balance. In those situations, the claim will result in a zero payment with a negative number on the claim explanation documents, also known as Remittance Advice Details (RAD) Codes. The researcher removed such incidents from the sample to avoid unnecessary complications.
b. The researcher also eliminated rows with missing data in key fields (e.g., payment, NPIs).

Step 7- *Understand the type and percentage of quality issues in Medicaid FFS claims adjudication using PERM audits*

The most comprehensive public source for Medicaid FFS quality issue information is the Payment Error Rate Measurement (PERM) program reports. PERM is an independent audit on all Medicaid programs. PERM's goal is "to measure and report a national improper payment rate for Medicaid and the Children's Health Insurance Program (CHIP) to comply with the requirements of the Improper Payments Elimination and Recovery Improvement Act" (The Centers for Medicare & Medicaid Services, 2015). PERM measures the quality issues on both Medicaid MCO and FFS programs. For Medicaid FFS, PERM measures a wide verity of quality issues, including improper payments caused by beneficiary eligibility problems, provide enrollment, provider diagnosis, policy-related matters, claim adjudication related issues, etc. Typical Medicaid claim adjudication processing errors more or less follow the same pattern as edited criteria, including the following categories: Eligibility, Pricing, Correct Coding, third part liability related issues, suspected duplicate, and Frequency or Utilization of Services (*Iowa Medicaid Enterprise Performance Report*, 2006). PERM groups all the problems found during the audit of Medicaid FFS claims into several categories, including the following groups (The Centers for Medicare and Medicaid Services, 2019a):

1. Pricing Error
2. Data Entry Error
3. Erroneously Paid Claims [should have been denied] (including Duplicate Claim Error, Non-covered Service/Beneficiary Error, FFS Payment for a Managed Care Service Error, etc.)
4. Erroneously Denied Claims [should have been paid]
5. Third-Party Liability Error
6. System Logic Edit Error
7. Claim Filed Untimely Error
8. Administrative/Other Error

The researcher selected the first four categories from the abovementioned list to label our target group based on these two types of quality errors. The reason for choosing these groups is that they contribute to the most erroneous payment found in Medicaid FFS PERM audits. There are two significant challenges with generalizing the findings of PERM audit reports to Medicaid FFS MMIS claim adjudication processes:

a) PERM reports aggregate results and presents statistics on the highest level and do not provide granule level information to allow researchers to see detailed level data and compare the results on claim level or even state level.
b) The definition and application of claim issues may differ from one MMIS program to another. For example, suppose a claim adjudication logic is programmed into one MMIS and is applied to claim manually in another state. In that case, the related pricing errors may be categorized as “System Logic Edit Error” and “Pricing Error” in another one.

To get an overall sense of the percentage of the pricing and data entry errors, the researcher used the PERM reports for 2019 and 2017, as shown in Table 10 (The Centers for Medicare and Medicaid Services, 2017) (The Centers for Medicare and Medicaid Services, 2019a).

Table 10. Selected error rates in PERM audit reports for 2017 and 2019.

Year[2]	Claim Type	Claims Sampled	Improper Payments	Improper Payment Rate	95% CI
2017	Inpatient and Outpatient Hospital Services- Medical Review Errors	3,672	38	1.3%	0.9%-1.8%
2017	Inpatient and Outpatient Hospital Services- Data Processing Errors	3,672	235	3.8%	2.9%-4.8%
2019	Inpatient Hospital Services- Medical Review Errors	2,651	9	0.69%	0-1.47%
2019	Outpatient Hospital Services- Medical Review Errors	1,670	29	1.24%	0.39%-2.10%
2019	Inpatient Hospital Services- Data Processing Errors	2,651	156	2.17 %	1.30% - 3.05%
2019	Outpatient Hospital Services- Data Processing Errors	1,670	78	3.27%	1.01% - 5.52%

This study focuses on a subset of the improper payments reported in PERM audits related to claim adjudication, including claim pricing issues and data entry issues. The results of PERM audits show a 1%-5% should be an overall acceptable interval estimate for data entry and claim pricing issues in a Medicaid FFS program.

Step 8- Estimate the overall percentage of claim adjudication errors (population proportion) in a given Medicaid FFS program

As mentioned before, the PERM audit report only provides a high-level understanding of the type of quality issues in a typical Medicaid FFS claim adjudication process. However, to label the target group in our synthesized dataset, detailed level information about the pattern of quality issues in each Medicaid program is needed. To achieve this goal, the researcher selected the California Medicaid Management Information System and submitted several formal "Public

[2] *2018 PERM is not available on the CMS website*

Information Request (PRA)" to the California Department of Health Care Services to receive weekly and monthly quality reports. CA-MMIS was selected for several reasons:

a. CA-MMIS is the most extensive Medicaid FFS program in the nation that processes over 200K claims per day. CA-MMIS is a data-rich program in which hundreds of reports are available.
b. CA-MMIS quality reports generate large sample sizes, increasing the chance to see samples of rare events (quality issues) in the quality reports.
c. The familiarity of the research team with CA-MMIS data streamlines the data gathering process.

Redacted copies of two quality reports requested from CA-MMIS:

1. Monthly Quality Management Performance Report (MQMPR)
2. Weekly Payment Data Review (WPDR) report for three years (2016-2019).

The researcher received the PHI-free version of these reports in which individually identifiable information (e.g., Providers' names and NPIs) were redacted. MQMPR contains large samples (over 1,000 samples from all adjudicated claims per month), and WPDR contains statistically significant (over 300 per week) samples of high-dollar claims. MQMPR did not provide detailed information on each quality issue per claim type, so the researcher used it to estimate the population proportion of quality issues and study the pattern of data entry issues in general. Per the MQMPR report of 2018, the estimated overall error rate on the CA-MMIS claim adjudication process is 2.1%, as shown in Table 11.

Table 11. Summary of claim adjudication errors per CA-MMIS MQMPR for 2018.

Month	Sample Size.	Overpayments	Underpayments	Overpayment%	Underpayment%	Erroneous Payment%
Jan	1,040	11	12	1.1%	1.2%	2.2%
Feb	927	9	6	1.0%	0.6%	1.6%
Mar	987	11	8	1.1%	0.8%	1.9%
Apr	954	8	12	0.8%	1.3%	2.1%
May	921	12	11	1.3%	1.2%	2.5%
Jun	938	8	9	0.9%	1.0%	1.8%
Jul	907	10	11	1.1%	1.2%	2.3%
Aug	911	13	9	1.4%	1.0%	2.4%
Sep	902	8	8	0.9%	0.9%	1.8%
Oct	945	7	3	0.7%	0.3%	1.1%
Nov	969	13	17	1.3%	1.8%	3.1%
Dec	1,008	19	8	1.9%	0.8%	2.7%
			Average			2.1%

Step 9- Understand the underlying pattern of quality issues in Medicaid FFS claims adjudication using detailed reports from a selected Medicaid program

CA-MMIS MQMPR reports do not provide detailed information on each quality issue per claim type. However, CA-MMIS WPDR reports include detail of the quality issues per claim type, so the researcher extracted the detail of the quality issues (erroneous payments) in the CA-MMIS WPDR 9/9/2016- 5/1/2020. Details of these reported findings are presented in APPENDIX C, and Table 12 summarizes the high-level statistics for the results.

Table 12. Claim adjudication errors reported in CA-MMIS WPDR 9/9/2016- 5/1/2020.

Claim Types	Count of Error
Outpatient	376
Inpatient	52
Medical	117
Long Term Care	6
Pharmacy	4
Total	555

The researcher used the detail of the WPDR results to study and simulate the pattern of erroneous payments. It is important to note all reported findings in the CA-MMIS WPDR reports were caught and fixed before the providers' final payment, so they are not essentially erroneous "payments." However, they carry the same characteristics of erroneous payments in a typical Medicaid FFS. CA-MMIS WPDR targets high dollar value claims, and it is not using a simple random sample of all adjudicated claims. Therefore, while the findings of CA-MMIS WPDR reports are appropriate for data labeling, they do not provide a statistically valid interval estimate of the population proportion (estimated rate of erroneous payments in a typical Medicaid FFS). Therefore, PERM audit reports in step 8 were used to provide an interval estimate of erroneous payments in a typical Medicaid FFS program.

As explained in step 6, the researcher simulated four types of erroneous payments to create labeled data for the target group: pricing errors, data entry error, erroneously paid, and erroneously denied claims. As depicted in Figure 7, these target groups are categorized into two major categories:

- P0- Negatives (not targeted group). As explained later in this chapter, the researcher did not change the payment amount for these groups and only applied the feature engineering for data cleansing.
- P1- Positives (target group). These are simulated labels and are categorized into three subgroups:
    - P1.1- Erroneously denied (should have been paid)
    - P1.2- Erroneously paid (should have been Denied)
    - P1.3- Pricing and data entry issues

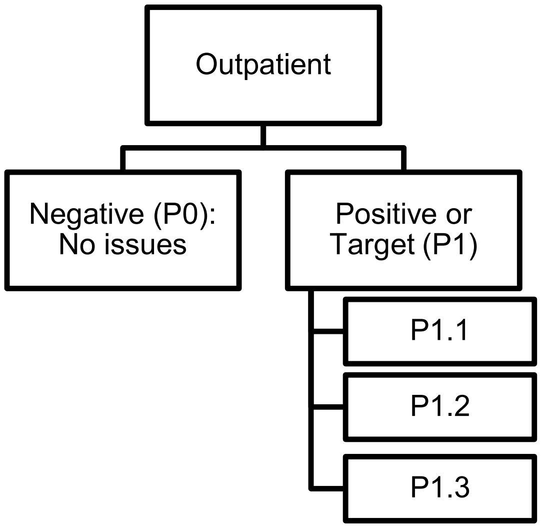


Figure 7. Hierarchy of target labels.

Table 13 presents the number of labels created for the target category for both inpatient and outpatient datasets.

Table 13. Number of labels created for the target category.

		Outpatient		Inpatient	
Labels		Count	Percentage	Count	Percentage
P0		112,335	98.65%	12,833	97.07%
P1		1,532	1.35%	388	2.93%
	P1.1	800	0.70%	200	0.18%
	P1.2	400	0.35%	100	0.09%
	P1.2	332	0.29%	88	0.08%
Total		113,867	100.00%	13,221	100.00%

A detailed explanation of this simulation process is provided in steps 9.1 to 9.3.

Step 9.1- simulating pricing errors and data entry error

This category contains the most complicated erroneous payments. CA-MMIS WPDR findings for inpatient and outpatient claims between 9/9/2016- 5/1/2020 were used to simulate these labels. First, results related to other erroneously paid and documentation issues were removed because they are not related to pricing issues and simulated in step 9.2. The claim adjudication errors' underlying probability distribution function (PDF) for the percentage difference for outpatient and inpatient claims was studied. Two separate Anderson-Darling (AD) tests were conducted. The AD test has been widely used to test the goodness of fit (GoF) of a distributional family. AD method compares the fit of the observed cumulative distribution function with that expected. AD test is derived as a modification of the Cramér–von Mises test, and the test statistics are given by (Upton & Cook, 2014):

$$A^2 = -\frac{1}{n}\sum_{j=1}^{n\infty}(2\mathrm{j}-1)\left[\ln\{F\left(x_{(j)}\right\} + \ln\{1 - F\left(x_{(j)}\right\}\right] - n$$

where F is the hypothesized cumulative distribution function, n is the sample size and $x_{(j)}$ is the jth ordered observation ($x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$). Researchers have been using the AD test for estimating the distribution function of operational risk, which is very similar to how it is used in this research (Feuerverger, 2016). The researcher uses the AD statistic to determine if our dataset (here, the distribution of the pricing errors in each Medicaid FFS program) follows a specified distribution (null hypothesis) or not (alternate hypothesis). The smaller the AD statistics, the better are the fit for the specific distribution. The researcher also looks at the p-value. If the p-value is less than alpha (0.05 or 0.10), the null hypothesis (that assumes the data will come from that distribution) is rejected. The Anderson-Darling statistic can be used to compare the fit of several distributions to determine which one is the best. However, to claim that one distribution is the best, its Anderson-Darling statistic must be substantially lower than the others. When the statistics are close together, one should use additional criteria, such as probability plots, to choose between them. Minitab was used for this test because it tests 16 different distribution functions for the AD test (Minitab, 2017). AD tests were run in Minitab for the absolute percentage difference (between the paid and correct amounts) for outpatient and inpatient samples. The complete results are presented in APPENDIX D. Table 13 summarizes the results of the GoF and AD tests for the outpatient claim issues.

Table 14. The goodness of Fit Test using AD for outpatient claim issues at CA-MMIS.

Distribution	AD	P	LRT P
Normal	115.373	<0.005	
Box-Cox Transformation	22.670	<0.005	
Lognormal	27.851	<0.005	
3-Parameter Lognormal	26.811	*	0.001
Exponential	270.653	<0.003	
2-Parameter Exponential	277.286	<0.010	0.112
Weibull	41.427	<0.010	
3-Parameter Weibull	39.024	<0.005	0.000
Smallest Extreme Value	122.999	<0.010	
Largest Extreme Value	93.428	<0.010	
Gamma	67.728	<0.005	
3-Parameter Gamma	64.935	*	0.000

Distribution	AD	P	LRT P
Logistic	84.572	<0.005	
Loglogistic	26.410	<0.005	
3-Parameter Loglogistic	25.357	*	0.000

There is no statistically significant proof (with P>0.05) that any tested distributions fit the data. Per AD test recommendations, the Loglogistic distribution offers the lowest AD (AD=26.410) and the best fit among the 2-parameter distributions (Minitab, 2019). Please note because the goal is to find the actual data's underlying distribution function. As a result, no transformation is selected, even if they have the lowest AD. Also, adding a third parameter significantly improves the fit of the Loglogistic distribution (LRT P = 0.000). Also, per guideline, a distribution with data points that roughly follow a straight line should be selected. This means Loglogistic and 3-parameter Loglogistic are acceptable rough estimates (Griffith, 2015). Table 14 is the summary of the results of the GoF and AD tests for the inpatient claims.

Table 15. The goodness of the Fit Test using AD for inpatient claim issues at CA-MMIS.

Distribution	AD	P	LRT P
Normal	10.531	<0.005	
Box-Cox Transformation	4.130	<0.005	
Lognormal	5.555	<0.005	
3-Parameter Lognormal	5.297	*	0.169
Exponential	8.078	<0.003	
2-Parameter Exponential	8.473	<0.010	0.051
Weibull	6.733	<0.010	
3-Parameter Weibull	6.287	<0.005	0.004
Smallest Extreme Value	11.243	<0.010	
Largest Extreme Value	9.380	<0.010	
Gamma	7.646	<0.005	
3-Parameter Gamma	7.191	*	0.011
Logistic	8.817	<0.005	
Loglogistic	4.641	<0.005	
3-Parameter Loglogistic	4.449	*	0.134
Johnson Transformation	0.190	0.894	

For the inpatient data, the researcher came to a similar conclusion as outpatient data. There is no statistically significant proof (with P>0.05) that any tested distributions fit the data, but the Loglogistic distribution offers the lowest AD among two-parameter distributions (AD=4.641). Here adding the third parameter does not improve the fit (LRT P = 0.134). While the underlying pattern of pricing issue percentage difference is relatively close to a Loglogistic distribution, there is no perfect among 16 tested PDFs to fit these data. As a result, during the labeling process, the exact pricing issue percentage differences are directly applied to the labeled data (instead of using a function to generate a dataset with the desired PDF) following these steps:

1. Select a random sample of 332 outpatient claims and 88 inpatient claims with a paid amount greater than zero. This could be done in SQL, as described in step 6 of using Excel RAND function.
2. Extract the absolute difference between paid and correct payment amounts from the CA-MMIS WPDR reports listed in APPENDIX C. Excel ABS function could be used for this purpose: ABS (Paid Amount-Correct Amount)
3. Select a random percentage from step 2 and multiple it by the selected sample from step 1. Sampling without replacement should be used. When the number of samples is greater than the number of percentage differences (Sept 2), repeat the process from setp1. All selected samples were labeled P1.

Step 9.2- simulating erroneously denied claims

This category contains the erroneously denied claims. Using numbers presented in Table 13, 800 random samples of outpatient claims and 200 samples of inpatient claims were selected.

This could be done in SQL, as described in step 6 of using Excel RAND function. The total payment amount of these selected samples was changed to zero, and they were labeled as P1.

Step 9.3- simulating erroneously paid claims

This category contains the erroneously paid claims. Using numbers presented in Table 13, 400 random samples of outpatient claims with payment amount equal zero, and 100 samples of inpatient claims with payment amount equal zero were selected. For each claim, the researcher found the first ICD code used in the claim and queried claims with similar first similar ICD codes. A random sample of those claims was selected, and their claim paid amounts were used to replace the zero paid amount. This could be done in SQL, as described in step 6 of using Excel RAND function. The total payment amount of these selected samples was changed to zero, and they were labeled as P1.

Chapter Summary

This chapter started by reviewing the statement of the problem and research questions and hypotheses. Then, three steps of the research procedure were explained in detail, including the type of statistical tests that will be used to validate that the synthesized data has the same error pattern as the actual one. Nine steps of collecting and creating a research dataset and labeling data were explained in detail. Anderson-Darling and GoF tests were applied to the data collected from California Medicaid FFS (aka CA-MMIS) to detect the underlying distribution function of potential erroneous payments and to simulate labels for the target group. PERM audit results were reviewed, and an overall interval estimate of population proportion for erroneous payments in a typical Medicaid FFS was extracted. Finally, two complete datasets were created and labeled to be used to train and test ML models.

CHAPTER 4

RESULTS

Chapter Overview

In this chapter, the results of this study are reviewed, hypotheses are tested, and the findings' details are discussed. As reviewed in the previous chapter, this chapter is organized into three major sections:

1. This chapter is tarted with completing the last part of manage data: "Feature Selection."
2. Then, ML models are built, and training to be conducted. The research approach and result of event-based sampling, oversampling, and stratification will be discussed. The selected split strategy will be addressed, two different cross-validation strategies will be compared, and one will be selected. Decision tree models, random forest models, and gradient boosting model will be trained and tested, and outcomes will be presented. The results from these three models will be used to calculate the feature importance for both datasets (inpatient and outpatient claims). After that, all other selected models will be trained and tested; the results will be presented.
3. This chapter will be concluded by evaluating selected models' performances. Models' performance metrics will be calculated and compared, the statistical difference among models' performance metrics will be assessed, and the results will be reviewed and discussed. The result of the hypothesis tests will be discussed in detail.

Feature Selection

This section discusses methods applied for feature selection to reduce the feature space dimension for our data. Feature selection aims to find and eliminate the independent variables (from all N features), which are less informative and make the computation difficult and even cause confusion. The idea here is to create an M-dimensional hyperspace, such M (where M < N) features that the data variates most (PCA For Categorical Features? - Stack Overflow, n.d.). Feature selection is a collective name for a group of methods that are a sub-class of data dimensionality reduction techniques that aim to reduce data complexity and improve machine learning and statistical learning results. As discussed by many data scientists, there is no universal best method for feature selection, and the complexity of the problem, nature of the dataset and business, and computational power availability and limitation may dictate choosing one method over another (Shamsaei & Gao, 2016), (Brownlee, 2019). While they are several methods that for feature selections, the most popular methods are Principal component analysis (PCA), which appropriate for continuous (ratio) data, and Multiple Correspondence Analysis (MCA), which works with categorical data.

Our problem consists of a mix of ratio and categorical features (independent variables) and categorical target (dependent variable). MCA function in Minitab only supports binary categories and does not perform well on the massive datasets with high dimensions in our experience. The researcher has tried methods recommended by researchers to conduct PCA/MCA on mixed data (ration and categorical) but found they require extreme and cost-prohibitive processing power (Stackexchange, n.d.). For example, to run a Python code to implement a mix-data PCA/MCA method recommended by Nancy Chelaru-Centea, a test was conducted (Chelaru-Centea, 2019). After several unsuccessful attempts on a very powerful server (in this case, centOS server with 16 intel CPUs with 64 GB RAM), it was found that more than 205 GB memory should be allocated the data to run the algorithm. This is logistically unfeasible. Even if the processing capability is available to conduct complex PCA/MCA on mixed data to identify all variables' statistical significance, one should not assume the features selected by such methods are the best predictors of the target group. As shows by Lo et al., relying solely on the statistical significance of the predictor in a similar situation may increase classification errors (Lo et al., 2015). For this research, the researcher used a heuristic and hybrid method for feature selection consist of three steps:

1. Eliminating data with no-low information value
2. Conduct PCA for continuous (ratio) features
3. Use business knowledge to select categorical features

Eliminating data with no-low information value

Independent values with one (unique), irrelevant, identical, or near-identical values do not improve a machine learning model's prediction power. This is even more important when dealing with high dimensional data (datasets with over 15-20 features) and massive (hundreds and thousands of rows). Our original datasets each had over 70 elements and more than ten thousand rows. The researcher reviewed datasets and found the following features to have unique or near-unique values, so they were removed from the model:

- Segment
- QAREVIEW_RESULT
- hcpcs_cd_45

- nch_bene_blood_ddctbl_lblty_am (for outpatient dataset had single value)

The researcher also removed feature with no-information values, including identifiers, as they have no prediction power by nature, so the following features were removed:

- at_physn_npi
- op_physn_npi
- ot_physn_npi
- prvdr_num
- clm_from_dt
- clm_thru_dt

Conduct PCA for continuous (ratio) features

As discussed in the literature review, PCA has been frequently used as a successful method to reduce the dimensionality of the data, improving the performance of the algorithms, and enhance the model interpretability without losing the performance. The PCA function of the Minitab (Stat > Multivariate > Principal Components) was used on the reduced dataset resulted from the previous steps and selected five (5) features (independent variables) for the outpatient dataset PCA, including:

- clm_pmt_amt
- nch_prmry_pyr_clm_pd_amt
- nch_bene_ptb_ddctbl_amt
- nch_bene_ptb_coinsrnc_amt
- clm_utlztn_day_cnt

Following the same process, six (6) features (independent variables) selected for the inpatient dataset PCA, including

- clm_pmt_amt
- nch_prmry_pyr_clm_pd_amt
- clm_pass_thru_per_diem_amt
- nch_bene_pta_coinsrnc_lblty_am
- nch_bene_blood_ddctbl_lblty_am
- nch_bene_ip_ddctbl_amt

The summary result of PCA analysis from Minitab is illustrated in Tables 16 and 17, and details are presented in APPENDIX E.

Table 16. PCA Analysis for five ratio features for the outpatient dataset.

Eigen analysis of the Correlation Matrix					
Eigenvalue	1.2556	1.1104	0.9846	0.8969	0.7525
Proportion	0.251	0.222	0.197	0.179	0.151
Cumulative	0.251	0.473	0.67	0.849	1
12794 cases used; 727 cases contain missing values					
Eigenvectors					
Variable	PC1	PC2	PC3	PC4	PC5
clm_pmt_amt	0.698	-0.017	0.09	-0.071	0.707
nch_prmry_pyr_clm_pd_amt	0.211	0.214	-0.944	0.116	-0.071
clm_pass_thru_per_diem_amt	0.1	-0.683	-0.204	-0.677	-0.158
nch_bene_ip_ddctbl_amt	0.047	-0.691	-0.059	0.718	0.017
clm_utlztn_day_cnt	0.676	0.1	0.236	0.088	-0.685

The first principal component for outpatient claims accounts for 22.2% of the total variance. To keep the most information in the selected features, a 90% cumulative cut-ff threshold was set for the Correlation Matrix's Eigen analysis. This means all ratio features must be chosen to explain the least 90% of the variance in our dataset. As shown in Table 16, even if one ratio feature is removed, the remaining ratio features only account for up to 82.2% of the variation. As a result, all ratio features are kept in the outpatient model.

Table 17. PCA Analysis for five ratio features for the inpatient dataset.

Eigen analysis of the Correlation Matrix					
Eigenvalue	1.1655	1.1056	1.0034	0.9832	0.8896
Proportion	0.194	0.184	0.167	0.164	0.148
Cumulative	0.194	0.379	0.546	0.71	0.858
12504 cases used; 717 cases contain missing values					

Eigenvectors					
Variable	PC1	PC2	PC3	PC4	PC5
clm_pmt_amt	0.658	-0.209	-0.057	-0.007	-0.075
nch_prmry_pyr_clm_pd_amt	0.298	-0.351	0.338	-0.738	-0.019
clm_pass_thru_per_diem_amt	0.38	0.567	-0.071	-0.11	0.71
nch_bene_pta_coinsrnc_lblty_am	0.518	-0.233	0.059	0.612	-0.079
nch_bene_blood_ddctbl_lblty_am	0.071	-0.174	-0.935	-0.219	-0.059

The first principal component for inpatient claims accounts for 19.4% of the total variance. To keep the most information in the selected features, a 90% cumulative cut-ff threshold was set for the Correlation Matrix's Eigen analysis. As shown in Table 17, even if just one ration feature is removed, the remaining ration features only account for up to 85.8% of the variation. As a result, all ratio features are kept in the inpatient model.

Use business knowledge to select categorical features

Medicaid FFS claims data allows a researcher to use business knowledge and make a judgment call to eliminate categorical features with low predictability power. Two right candidates for such heuristic feature selection are diagnosing and treating medical codes (CPT and ICD codes). A Medicaid FFS claim may contain several diagnoses and treatments, so there are up to 45 columns (features) to capture all codes related to one claim in the dataset. However, most claims contain less than a handful of CPT and ICD codes. A simple exploratory analysis of the data shows the only the first three CPT, and ICD codes for each claim have considerable values. The rest are usually with no value. Table 18 listed all features selected at the end of the feature selection process to build our supervised learning models.

Table 18. PCA Analysis for five ratio features for the inpatient and outpatient dataset.

Inpatient	Outpatient
clm_pmt_amt	clm_pmt_amt
nch_prmry_pyr_clm_pd_amt	nch_prmry_pyr_clm_pd_amt
at_physn_npi	at_physn_npi
op_physn_npi	op_physn_npi
ot_physn_npi	ot_physn_npi
admtng_icd9_dgns_cd	icd9_dgns_cd_1
clm_pass_thru_per_diem_amt	icd9_dgns_cd_2
nch_bene_ip_ddctbl_amt	icd9_dgns_cd_3
nch_bene_pta_coinsrnc_lblty_am	icd9_prcdr_cd_1
nch_bene_blood_ddctbl_lblty_am	nch_bene_ptb_ddctbl_amt
clm_utlztn_day_cnt	nch_bene_ptb_coinsrnc_amt
clm_drg_cd	admtng_icd9_dgns_cd
icd9_dgns_cd_1	hcpcs_cd_1
icd9_dgns_cd_2	hcpcs_cd_2
icd9_dgns_cd_3	hcpcs_cd_3
hcpcs_cd_1	YEAR
hcpcs_cd_2	
hcpcs_cd_3	
YEAR	

Build, Train, and Test Models

In this section, ML models are built, and algorithms will be trained. First, the challenge of dealing with rare events and proposing event-based sampling to improve model performance will be discussed. Approach to oversampling, stratification strategy and the cross-validation method will be presented. Decision Tree, Random Forest, and Gradient Boosting models are the first models to be trained. The outputs of these models are required to calculate the feature importance score for both datasets' features. Finally, the ML model for the rest of the chosen algorithms will be built and trained, and test results will be presented.

Dealing with rare events and event-based sampling

The biggest challenge in applying supervised learning to detect quality issues is dealing with rare events. Quality issues tend to be a rare event. When a process operates around four sigma level (a typical Sigma Level for large operations), the percentage of quality issues (if the quality data is attribute data) is less than one percent. A supervised algorithm needs to see lots of similar data to be trained on a specific pattern. In applying machine learning to quality data, a researcher has a considerably more extensive set of non-target events in the dataset (non-quality issues) than target events (quality issues). Event-based sampling is a method to overcome this challenge through oversampling of the rare-events. This method also comes with guidelines to apply various techniques (like adjusting posterior probability) to offset the impact of oversampling and avoid overfitting the algorithm (James et al., 2015). The researcher will be using event-based sampling within our cross-validation process.

Select the split strategy and cross-validation

The classic method to create train and test data for supervised learning is based on a simple split strategy in which a specific percentage of the dataset (e.g., 80%) to train the model, and the trained model is tested on the rest of the data. More modern supervised learning models usually use cross-validation. The researcher selected an n-fold cross-validation strategy for our problem. In this method, the entire data is split into n equal folds (partitions) randomly. Then, the model is trained on n-1 folds in each iteration and tested on the n-th fold.

Several researchers have discussed various methods to deal with unbalanced data through oversampling, before, during, and after cross-validation. In summary, they emphasize the separation of oversampled training data and test or validation data and suggest the following guidelines (Altini, 2015) (Mencar, 2016):

- Oversampling and stratification to be conducted during the cross-validation.
- Ensure the oversampled train data is not used for other purposes (e.g., features selection, test, or validation).
- Draw an oversample from the rare events (target subgroup, which is P1 for our case) after excluding the samples taken before.
- Create test data by combining excluded samples (from training phase), oversampled the rare events, and majority class (P0 in our case).
- Repeat the process n-times, where n is the number of cross-validation folds.

Figure 8 illustrates the oversampling implementation within a cross-validation strategy. This method prevents the model from leaking the training data to the test that causes over-optimism.

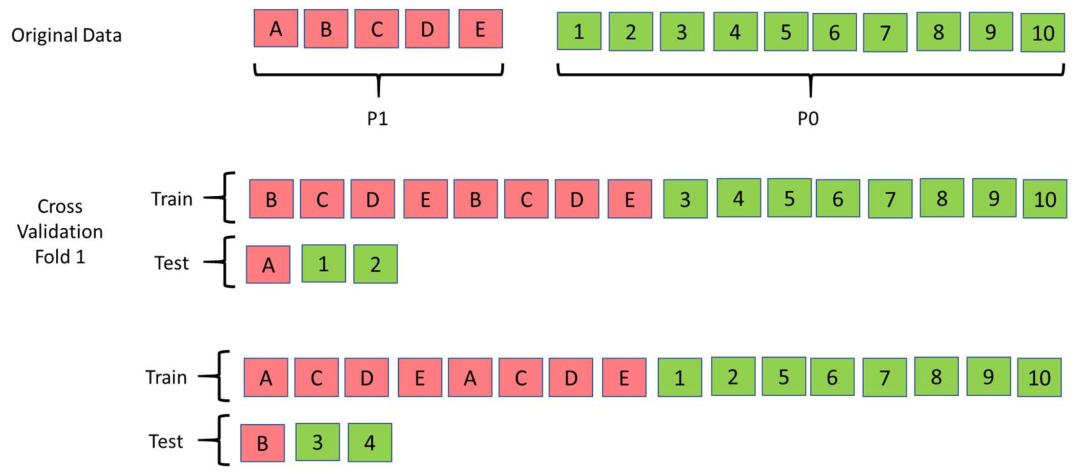


Figure 8. Oversampling within cross-validation.

Popular choices for n are n=5 and n=10, and while there almost no limit on how large n can be, many experts recommend keeping n less than 10 (James et al., 2015). N=5 reduces the complexity of our calculations. However, to validate the there is no significant difference between n=5 and n=10, a test was designed. First, a balanced dataset (by applying oversampling on both train and test data) was created. Please note, this method is not the same as the proper oversampling method described before, and it is only designed to compare the performance on n=10 and n=5 cross-validation. The detailed result of this test is in APPENDIX F. Table 19 shows a summary of the result showing no significant difference in the performance of 5 and 10 fold cross-validation on a balanced dataset (the outpatient dataset, with multiclass classification and Decision Tree and Random Forest with entropy configuration, were used).

Table 19. The average difference in calculating various performance measures between 5-fold and 10-fold validation on a balanced dataset.

	Precision	Recall	Accuracy	F Score	ROC
DTE	0.83%	0.80%	0.33%	0.89%	0.51%
RFE	-0.39%	-0.51%	-0.11%	-0.57%	-0.18%

Adjusting posterior probability after the oversampling as recommended by some researchers in our case skewed the outcomes by assigning the majority (or all) of rare events (in our case, P1) to the none-targeted category (P0) (Bhalla, 2016), (SAS Institute Inc., 2020). The researcher used the Synthetic Minority Oversampling Technique (SMOTE) within the oversampling procedure to create synthetic samples. SMOTE uses nearest neighbors and Naive Bayes classifier to generate new and synthetic data examples (Chawla et al., 2002). Python Scikit-learn *SMOTE()* function was used to implement this method.

Train and test Decision Tree, Random Fore, and Gradient Boosting

Decision Tree, Random Fore, and Gradient Boosting are the first three models to be trained and tested. The researcher needed their output to calculate the feature importance score (Gini importance) for independent variables and test the first two hypotheses. One crucial notion here is the difference between an "algorithm" and a predictive (or supervised leaning) "model." An algorithm is a generic method that could apply to various datasets and could usually be tuned through some configuration features. For example, a Decision Tree is the name of a group of algorithms with similar approaches and have several configuration features like a maximum number of leaves. A model (in our definition) is a mathematical equation that resulted from running an algorithm with specific configuration features on a dataset. It is impractical to compare "algorithms performance" in the abstract due to the massive permutation among configuration features. The focus of this research is on the models derived from running selected algorithms. As a result, an algorithm with and one without oversampling creates two different predictive models. If common themes are found across various models, it shows that the algorithm performs similarly across different configurations and datasets.

To build and train the Decision Tree, the researcher used the *DecisionTreeClassifier* function in Python Scikit-learn library with the following features:

- Decision Tree with Gini classifier (DTG): *default*, *random_state=0*, *max_depth=10*, *max_leaf_nodes=115*
- Decision Tree with Entropy classifier (DTE): *critrion=entropy, random_state=0, max_depth=10, max_leaf_nodes=115*

DTE and DTG models were trained on both inpatient and outpatient datasets with and without oversampling. A partial view of the inpatient DTE is depicted in Figure 9 to provide an example that shows the claim payment amount as the highest predictor (splitter) for inpatient claims.

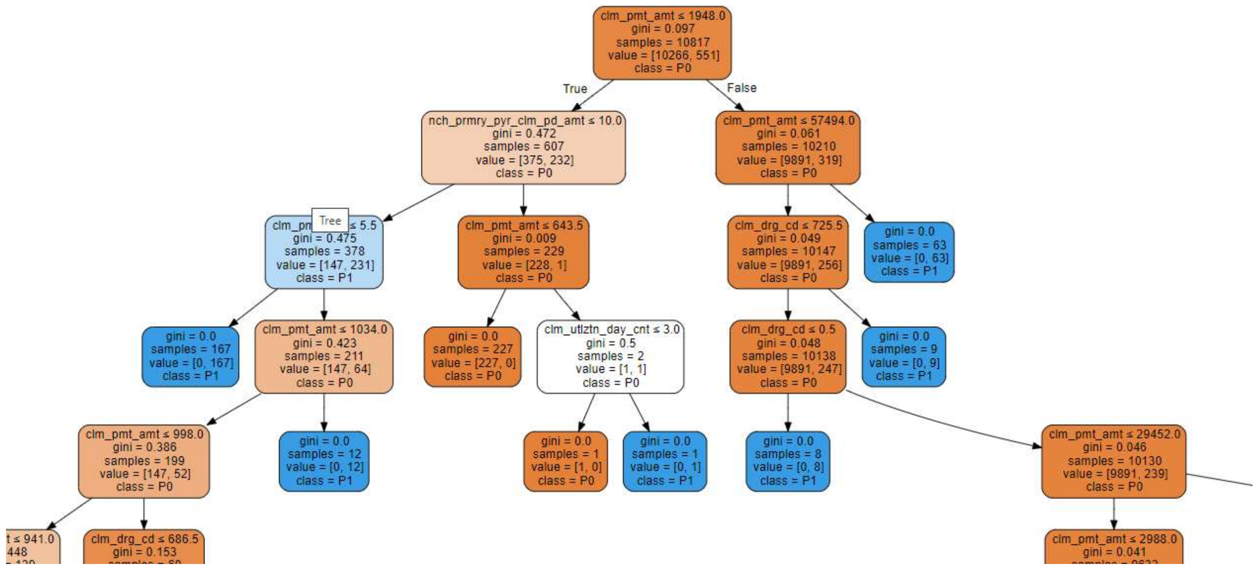


Figure 9. A partial view of the inpatient DTE.

RFG, RFE, and GB algorithms on inpatient and outpatient datasets were trained with and without applying the oversampling methods. *RandomForestClassifier* and *GradientBoostingClassifier* from the Python Scikit-learn library were utilized, and results were tested. Test results for all algorithms are presented together in APPENDIX H.

Review feature importance

Mean Decrease in Impurity (MDI) is a method in the Python Scikit-learn library to calculate the feature importance score (Gini importance). This method and associated codes were applied for the independent variables the researcher selected for inpatient and outpatient models. The result of these calculations is presented in APPENDIX G and discussed in detail in the "Results of Testing Research Hypotheses" section.

Train and test the rest of the selected models and review the results

Table 20 listed the Python Scikit-learn library functions and configuration selected to build, train, and test the rest of the selected algorithms.

Table 20. Scikit-learn configuration selected to build the ML models.

Al	Configurations	Main SL Function
NB	default + fit_prior=False	BernoulliNB
kNN	default + weights='distance	KNeighborsClassifier
LR	default + solver='liblinear', random_state=0, dual=False	LogisticRegression
NN	default + solver='adam'	MLPClassifier
DA	default + solver='lsqr', shrinkage='auto'	LinearDiscriminantAnalysis

Detailed results for all tested algorithms and models on both datasets are presented in APPENDIX H. Table 21 summarizes the result of the performance metrics (recall and F1-score) for all tested algorithms for both outpatient and inpatient claims.

Table 21. Performance metrics for tested algorithms.

Algorithm	Outpatient Data		Inpatient Data	
	Recall	F1-Score	Recall	F1- Score
DTG	6.27%	10.92%	58.47%	71.38%
DTG (OS)	42.24%	6.78%	60.55%	53.05%
DTE	5.68%	10.18%	58.50%	72.70%
DTE (OS)	47.00%	7.19%	58.77%	54.88%
RFG	2.74%	5.32%	57.19%	72.52%
RFG(OS)	11.03%	8.80%	49.73%	50.52%
RFE	2.61%	5.06%	55.41%	71.22%

Algorithm	Outpatient Data		Inpatient Data	
	Recall	F1-Score	Recall	F1- Score
RFE (OS)	11.42%	9.07%	52.32%	52.35%
NB	52.42%	19.74%	54.39%	38.25%
NB (OS)	52.29%	4.23%	55.94%	20.15%
kNN	0.46%	0.70%	1.55%	2.12%
kNN (OS)	16.38%	2.57%	13.65%	3.71%
LR	76.39%	2.71%	1.29%	1.96%
LR (OS)	78.25%	2.67%	39.08%	4.70%
NN	0.78%	0.45%	2.05%	0.77%
NN (OS)	36.53%	2.19%	48.40%	5.53%
DA	0.20%	0.39%	0.00%	0.00%
DA (OS)	48.56%	2.76%	60.02%	7.10%
GB	11.29%	14.00%	59.76%	73.46%
GB (OS)	25.00%	19.51%	63.89%	66.33%

Evaluate Models' Performances

This section will use the previous section's algorithm results to test the main hypotheses for this research. Appropriate statistical tests will be used, hypotheses will be tested, results will be reviewed, and the outcomes will be discussed.

Results of Testing Research Hypotheses

The results of testing the four main hypotheses of our research will be discussed in this section. Here assumptions, results of the primary hypothesis test, and post-hoc analysis performed are discussed.

Research Hypothesis 1 (H4.1)

The first hypothesis (hypothesis 4.1) stated there is no significant difference among the predictors' feature importance in identifying erroneously processed Medicaid outpatient claims. This means all predictors (selected independent variables) are the same from the feature importance or Mean Decrease in Impurity (MDI) perspective. Detailed results of these calculations are presented in APPENDIX G. A one-way ANOVA was run on the means of feature importance score (Gini importance) for selected outpatient claims independent variables. Detailed results of the one-way ANOVA test from Minitab are included in Appendix I. Table 22 indicates that there was enough evidence at the α = 0.05 significance level to reject the null hypothesis (p = 0.001). This means at least one of the feature importance scores (Gini importance) for outpatient claims independent variables performed significantly different from the rest of the features.

Table 22. One-Way ANOVA results for comparing feature importance scores (Gini importance) for outpatient claims independent variables.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	15	0.6359	0.042394	8.02	0.000
Error	64	0.3383	0.005286		
Total	79	0.9742			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 23 shows clm_pmt_amt has the highest Mean F-value, significantly higher than the rest of the features.

Table 23. Tukey pairwise comparisons for feature importance scores (Gini importance) for outpatient claims independent variables.

Factor	N	Mean	Grouping	
clm_pmt_amt	5	0.389	A	
icd9_dgns_cd_1	5	0.1100		B
at_physn_npi	5	0.0983		B
hcpcs_cd_1	5	0.0539		B
icd9_dgns_cd_2	5	0.0521		B
hcpcs_cd_2	5	0.0484		B
nch_bene_ptb_coinsrnc_amt	5	0.04511		B
ot_physn_npi	5	0.04454		B
icd9_dgns_cd_3	5	0.04191		B

Factor	N	Mean	Grouping
hcpcs_cd_3	5	0.03497	B
admtng_icd9_dgns_cd	5	0.03077	B
op_physn_npi	5	0.02340	B
YEAR	5	0.01583	B
nch_bene_ptb_ddctbl_amt	5	0.00787	B
nch_prmry_pyr_clm_pd_amt	5	0.004190	B
icd9_prcdr_cd_1	5	0.000005	B

Means that do not share a letter are significantly different.

The Fisher pairwise comparisons using the LSD method and a 95% confidence level in Table 24 shows clm_pmt_amt, which has the highest Mean F-value and significantly higher than the rest of the features. Fisher LSD test also presents a slight difference among other predictors' importance. In both tests, icd9_dgns_cd_1 and at_physn_npi have the second and third highest F-value for the feature importance significantly lower than clm_pmt_amt.

Table 24. Fisher LSD pairwise comparisons for feature importance scores (Gini importance) for outpatient claims independent variables.

Factor	N	Mean	Grouping			
clm_pmt_amt	5	0.389	A			
icd9_dgns_cd_1	5	0.1100		B		
at_physn_npi	5	0.0983		B	C	
hcpcs_cd_1	5	0.0539		B	C	D
icd9_dgns_cd_2	5	0.0521		B	C	D
hcpcs_cd_2	5	0.0484		B	C	D
nch_bene_ptb_coinsrnc_amt	5	0.04511		B	C	D
ot_physn_npi	5	0.04454		B	C	D
icd9_dgns_cd_3	5	0.04191		B	C	D
hcpcs_cd_3	5	0.03497		B	C	D
admtng_icd9_dgns_cd	5	0.03077		B	C	D
op_physn_npi	5	0.02340		B	C	D
YEAR	5	0.01583			C	D
nch_bene_ptb_ddctbl_amt	5	0.00787			C	D
nch_prmry_pyr_clm_pd_amt	5	0.004190				D
icd9_prcdr_cd_1	5	0.000005				D

Means that do not share a letter are significantly different.

Research Hypothesis 2 (H5.1)

The second hypothesis (hypothesis 5.1) stated there is no significant difference among the predictors' feature importance in identifying erroneously processed Medicaid inpatient claims. This means all predictors (selected independent variables) are the same from the feature importance or Mean Decrease in Impurity (MDI) perspective. Detailed results of these calculations are presented in APPENDIX G. A one-way ANOVA was run on the means of feature importance score (Gini importance) for selected inpatient claims independent variables. Detailed results of the one-way ANOVA test from Minitab are included in Appendix I. Table 25 indicates that there was enough evidence at the α = 0.05 significance level to reject the null hypothesis (p = 0.001). This means at least one of the feature importance scores (Gini importance) for outpatient claims independent variables performed significantly different from the rest of the features.

Table 25. One-Way ANOVA results for comparing feature importance scores (Gini importance) for inpatient claims independent variables.

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	18	0.8838	0.049103	23.60	0.000
Error	76	0.1581	0.002080		
Total	94	1.0419			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 26 shows clm_pmt_amt and nch_prmry_pyr_clm_pd_amt have the highest Mean F-value, whereas clm_pmt_amt is significantly higher than the rest of the features.

Table 26. Tukey pairwise comparisons for feature importance scores (Gini importance) for inpatient claims independent variables.

Factor	N	Mean	Grouping	
clm_pmt_amt	5	0.3601	A	
nch_prmry_pyr_clm_pd_amt	5	0.2898	A	
clm_drg_cd	5	0.08992		B
at_physn_npi	5	0.0441		B
icd9_dgns_cd_1	5	0.0352		B
icd9_dgns_cd_2	5	0.0334		B
icd9_dgns_cd_3	5	0.0332		B
admtng_icd9_dgns_cd	5	0.0302		B
op_physn_npi	5	0.02552		B
clm_utlztn_day_cnt	5	0.01947		B
clm_pass_thru_per_diem_amt	5	0.01288		B
YEAR	5	0.00761		B
nch_bene_ip_ddctbl_amt	5	0.00721		B
ot_physn_npi	5	0.00694		B
nch_bene_pta_coinsrnc_lblty_am	5	0.002416		B
nch_bene_blood_ddctbl_lblty_am	5	0.002045		B
hcpcs_cd_3	5	0.000000		B
hcpcs_cd_2	5	0.000000		B
hcpcs_cd_1	5	0.000000		B

Means that do not share a letter are significantly different.

The Fisher pairwise comparisons using the Fisher LSD method and a 95% confidence level are also shown in Table 27, showing clm_pmt_amt has the highest F-measure and significantly higher than the rest of the predictors. nch_prmry_pyr_clm_pd_amt has the second-highest F-measure and significantly higher than the rest of the predictors. Fisher LSD test results are slightly different from the Tukey test results in the grouping, but the predictors' ranking is similar.

Table 27. Fisher LSD pairwise comparisons for feature importance scores (Gini importance) for inpatient claims independent variables

Factor	N	Mean	Grouping			
clm_pmt_amt	5	0.3601	A			
nch_prmry_pyr_clm_pd_amt	5	0.2898		B		
clm_drg_cd	5	0.08992			C	
at_physn_npi	5	0.0441			C	D
icd9_dgns_cd_1	5	0.0352			C	D
icd9_dgns_cd_2	5	0.0334			C	D
icd9_dgns_cd_3	5	0.0332			C	D
admtng_icd9_dgns_cd	5	0.0302				D
op_physn_npi	5	0.02552				D
clm_utlztn_day_cnt	5	0.01947				D
clm_pass_thru_per_diem_amt	5	0.01288				D
YEAR	5	0.00761				D
nch_bene_ip_ddctbl_amt	5	0.00721				D
ot_physn_npi	5	0.00694				D
nch_bene_pta_coinsrnc_lblty_am	5	0.002416				D
nch_bene_blood_ddctbl_lblty_am	5	0.002045				D
hcpcs_cd_3	5	0.000000				D
hcpcs_cd_2	5	0.000000				D
hcpcs_cd_1	5	0.000000				D

Means that do not share a letter are significantly different.

Research Hypothesis 3 (H6.1)

The third hypothesis (hypothesis 6.1) stated there is no significant difference among the average recall of algorithms in identifying Medicaid outpatient claim issues. The researcher runs a one-way ANOVA based on 5-fold cross-validation recall results across all selected algorithms to investigate this hypothesis. Detailed results of the ANOVA test from Minitab are included in Appendix J. Table 28 shows that there was enough evidence at the $\alpha = 0.05$ significance level to reject the null hypothesis ($p = 0.001$). This means the recall for at least one algorithm is significantly higher than the others in detecting quality issues in outpatient claims.

Table 28. One-Way ANOVA results for comparing algorithms' recall for outpatient claim issues

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	19	6.2851	0.33080	29.21	0.000
Error	80	0.9060	0.01132		
Total	99	7.1911			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 29 shows the Logistic Regression model (with oversampling) has the highest F-Value (not be confused with the F1-score used as a performance metrics) and significantly outperforms other algorithms in detecting erroneous payments in the research outpatient data. This will make the Logistic Regression (regardless of sampling approach) the favorite algorithm in finding true positives in outpatient claim adjudication. Logistic Regression (with oversampling) and Naïve Bayes (with and without oversampling) are the next best performers and grouped together (group B). Tuckey test also shows the recall of some algorithms, including Discriminant Analysis, Neural Network, and Decision Trees, improves with oversampling. This means these algorithms may not perform well on unbalanced data. Oversampling improves the performance of algorithms like Logistic Regression, Naïve Bays, and Gradient Boosting. However, there is no statistical significance to this difference (Naïve Bays slightly performed better without oversampling, which could be due to random effects in the cross-validation process). Neural Networks recall for inpatient claims is significantly improved by oversampling.

Tukey grouped algorithms' recalls into several overlapping groups, whereas the groups that do not share a letter are significantly different. For example, Naïve Bayes (with and without oversampling) is in groups B and C. Gradient Boosting (with oversampling) listed in groups D, E, and F. This means there is a significant difference between their recalls on the outpatient dataset because they do not share same letters in the grouping. This will make the Logistic Regression (regardless of sampling approach) the favorite algorithm in finding true positives in outpatient claim adjudication. The Tukey test does not recognize a statistically significant difference in the recall rate for the following algorithms compare the Logistic Regression: Discriminant Analysis (with oversampling), Naïve Bayes (with and without oversampling), Decision Tree (both with Gini and Entropy feature and oversampling), Neural Network (with oversampling), and Gradient Boosting (with and without oversampling) with various Mean F-value.

Table 29. Tukey pairwise comparisons for algorithms' recall for outpatient claim issues

Factor	N	Mean	Grouping						
LR (OS)	5	0.7825	A						
LR	5	0.7639	A	B					
NB	5	0.5242		B	C				
NB (OS)	5	0.5229		B	C				
DA (OS)	5	0.48565			C	D			
DTE (OS)	5	0.4700			C	D			
DTG (OS)	5	0.4224			C	D			
NN (OS)	5	0.365			C	D	E		
GB (OS)	5	0.25000				D	E	F	
kNN (OS)	5	0.16383					E	F	G
RFE (OS)	5	0.11424						F	G
GB	5	0.1129						F	G
RFG (OS)	5	0.1103						F	G
DTG	5	0.06266						F	G
DTE	5	0.05680						F	G
RFG	5	0.02741						F	G
RFE	5	0.02611						F	G
NN	5	0.00782						F	G
kNN	5	0.00457						F	G
DA	5	0.00196							G

Means that do not share a letter are significantly different.

The Fisher LSD pairwise comparisons with a 95% confidence level in Table 30 presents the same ranking of the algorithms' recall means with slightly different pairwise grouping.

Table 30. Fisher LSD pairwise comparisons for algorithms' recall for outpatient claim issues

Factor	N	Mean	Grouping						
LR (OS)	5	0.7825	A						
LR	5	0.7639	A						
NB	5	0.5242		B					
NB (OS)	5	0.5229		B					
DA (OS)	5	0.48565		B	C				
DTE (OS)	5	0.4700		B	C				
DTG (OS)	5	0.4224		B	C				
NN (OS)	5	0.365			C	D			
GB (OS)	5	0.25000				D	E		
kNN (OS)	5	0.16383					E	F	
RFE (OS)	5	0.11424						F	G
GB	5	0.1129						F	G
RFG (OS)	5	0.1103						F	G
DTG	5	0.06266						F	G
DTE	5	0.05680						F	G
RFG	5	0.02741							G
RFE	5	0.02611							G
NN	5	0.00782							G
kNN	5	0.00457							G
DA	5	0.00196							G

Means that do not share a letter are significantly different.

Research Hypothesis 4 (H6.2)

The fourth hypothesis (hypothesis 6.2) stated there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid outpatient claims. This means all selected algorithms have similar recall and F1-score. The researcher runs a one-way ANOVA based on 5-fold cross-validation F1-score results across all selected algorithms to investigate this hypothesis. Detailed results of the ANOVA test from

Minitab are included in Appendix J. Table 31. they are indicating that there was enough evidence at the α = 0.05 significance level to reject the null hypothesis (p = 0.001). This means the F1-score for at least one algorithm is significantly higher than the others in detecting quality issues in outpatient claims.

Table 31. One-Way ANOVA results for comparing algorithms' F1-score for outpatient claim issues

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	19	0.31920	0.016800	126.77	0.000
Error	80	0.01060	0.000133		
Total	99	0.32980			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 32 shows the Naïve Bays (without oversampling), and Gradient Boosting (with oversampling) outperformed the rest of the algorithms from the F1 score perspective. Gradient Boosting (without oversampling) is the third performing algorithm on recall for outpatient claims. Oversampling significantly improves the F1-score of the Gradient Boosting algorithm on the outpatient data. Various configuration of Decision Tree and Random Forest (with entropy and Gini, with and without oversampling) are the next group in the rank. There is evidence to support a statistically significant difference between their performance and the rest of the algorithms. Decision Tree (with Gini feature) without oversampling has its own group, followed by Gradient Boosting with oversampling. Decision Tree and Gradient Boosting algorithms (regardless of their feature configuration and sampling approach) are the favorite algorithms from the F1 score point of view (a balance between recall and precision).

Table 32. Tukey pairwise comparisons for algorithms' F1 score for outpatient claim issues

Factor	N	Mean	Grouping									
NB	5	0.19742	A									
GB (OS)	5	0.19513	A									
GB	5	0.1400		B								
DTG	5	0.10924			C							
DTE	5	0.10178			C							
RFE (OS)	5	0.09074			C	D						
RFG (OS)	5	0.08799			C	D						
DTE (OS)	5	0.07190				D	E					
DTG (OS)	5	0.06783				D	E	F				
RFG	5	0.05316					E	F	G			
RFE	5	0.05063					E	F	G	H		
NB (OS)	5	0.042270						F	G	H	I	
DA (OS)	5	0.027570							G	H	I	J
LR	5	0.027106							G	H	I	J
LR (OS)	5	0.026677							G	H	I	J
kNN (OS)	5	0.02572								H	I	J
NN (OS)	5	0.02195									I	J
kNN	5	0.00702										J
NN	5	0.00449										J
DA	5	0.00390										J

Means that do not share a letter are significantly different.

The Fisher LSD pairwise comparisons with a 95% confidence level in Table 33 presents the same ranking of the algorithms' F1 score means with slight differences in the grouping of the low performing models:

Table 33. Fisher LSD pairwise comparisons for algorithms' F1 score for outpatient claim issues

Factor	N	Mean	Grouping				
NB	5	0.19742	A				
GB (OS)	5	0.19513	A				
GB	5	0.1400		B			
DTG	5	0.10924			C		
DTE	5	0.10178			C	D	
RFE (OS)	5	0.09074				D	
RFG (OS)	5	0.08799				D	
DTE (OS)	5	0.07190					E
DTG (OS)	5	0.06783					E

Factor	N	Mean	Grouping		
RFG	5	0.05316	F		
RFE	5	0.05063	F		
NB (OS)	5	0.042270	F		
DA (OS)	5	0.027570		G	
LR	5	0.027106		G	
LR (OS)	5	0.026677		G	
kNN (OS)	5	0.02572		G	
NN (OS)	5	0.02195		G	
kNN	5	0.00702			H
NN	5	0.00449			H
DA	5	0.00390			H

Means that do not share a letter are significantly different.

Research Hypothesis 5 (H7.1)

The fifth hypothesis (hypothesis 7.1) stated there is no significant difference among the average recall of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims. The researcher runs a one-way ANOVA based on 5-fold cross-validation recall results across all selected algorithms to investigate this hypothesis. Detailed results of the ANOVA test from Minitab are included in Appendix J. Table 34 indicates that there was enough evidence at the α = 0.05 significance level to reject the null hypothesis (p = 0.001). This means the recall for at least one algorithm is significantly higher than the others in detecting quality issues in inpatient claims.

Table 34. One-Way ANOVA results for comparing algorithms' recall for inpatient claim issues

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	19	6.0195	0.316815	33.38	0.000
Error	80	0.7594	0.009492		
Total	99	6.7789			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 35 shows the Gradient Boosting (with oversampling) outperformed the rest of the models in recalling quality issues of the inpatient claim data. However, the Tukey test does not suggest a statistical difference among the recall of the top 14 algorithms and models. This test shows k-Nearest Neighbor, Neural Network, Discriminant Analysis, and Logistic Regressions (without sampling) have significantly lower recall than the rest of the algorithms. Oversampling has significantly improved the recall of the Logistic Regression model, which is a sign of the impact of imbalanced data on this algorithm's performance.

Table 35. Tukey pairwise comparisons for algorithms' recall for inpatient claim issues

Factor	N	Mean	Grouping		
GB (OS)	5	0.6389	A		
DTG (OS)	5	0.6055	A	B	
GB	5	0.5976	A	B	
DTE (OS)	5	0.5877	A	B	
DTE	5	0.5850	A	B	
DTG	5	0.5847	A	B	
RFG	5	0.5719	A	B	
NB (OS)	5	0.5594	A	B	
RFE	5	0.5541	A	B	
NB	5	0.5439	A	B	
RFE (OS)	5	0.5232	A	B	
RFG (OS)	5	0.4973	A	B	
NN (OS)	5	0.484	A	B	
LR (OS)	5	0.391		B	
kNN (OS)	5	0.1365			C
NN	5	0.0205			C
kNN	5	0.01548			C
LR	5	0.01289			C
DA (OS)	5	0.000000			C
DA	5	0.000000			C

Means that do not share a letter are significantly different.

The Fisher LSD pairwise comparisons with a 95% confidence level in Table 36 presents the same ranking of the algorithms' recall means with slightly different pairwise grouping:

Table 36. Fisher LSD pairwise comparisons for algorithms' recall for inpatient claim issues

Factor	N	Mean	Grouping				
GB (OS)	5	0.6389	A				
DTG (OS)	5	0.6055	A	B			
GB	5	0.5976	A	B			
DTE (OS)	5	0.5877	A	B			
DTE	5	0.5850	A	B			
DTG	5	0.5847	A	B			
RFG	5	0.5719	A	B			
NB (OS)	5	0.5594	A	B			
RFE	5	0.5541	A	B			
NB	5	0.5439	A	B			
RFE (OS)	5	0.5232	A	B			
RFG (OS)	5	0.4973		B	C		
NN (OS)	5	0.484		B	C		
LR (OS)	5	0.391			C		
kNN (OS)	5	0.1365				D	
NN	5	0.0205				D	E
kNN	5	0.01548				D	E
LR	5	0.01289					E
DA (OS)	5	0.000000					E
DA	5	0.000000					E

Means that do not share a letter are significantly different.

Research Hypothesis 6 (H7.2)

The sixth hypothesis (hypothesis 7.2) stated there is no significant difference among the average F1-score of selected supervised learning algorithms in identifying erroneously processed Medicaid inpatient claims. The researcher runs a one-way ANOVA based on 5-fold cross-validation F1-score results across all selected algorithms to investigate this hypothesis. Detailed results of the ANOVA test from Minitab are included in Appendix J. Table 37 indicates that there was enough evidence at the $\alpha = 0.05$ significance level to reject the null hypothesis ($p = 0.001$). This means the F1-score for at least one algorithm is significantly higher than the others in detecting quality issues in inpatient claims.

Table 37. One-Way ANOVA results for comparing algorithms' F1-score for inpatient claim issues

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	19	8.2962	0.436642	237.27	0.000
Error	80	0.1472	0.001840		
Total	99	8.4434			

The Tukey pairwise comparisons using the Tukey method and a 95% confidence level in Table 38 shows Gradient Boosting (both without oversampling), Decision Tree (using Entropy and Gini), Random Forest (with entropy and without oversampling) outperformed the rest of the algorithms from the F1 score perspective on inpatient data. There is evidence to support a statistically significant difference between their performance and the rest of the algorithms. Tukey's ranking suggests that oversampling has reduced the F1-score (by decreasing the precision) for inpatient claims. This may result from oversampling and applying the SMOTE method on a very complex dataset that resulted in simulated samples not carrying the characteristics that could be easily modeled as a feature.

Table 38. Tukey pairwise comparisons for algorithms' recall for inpatient claim issues

Factor	N	Mean	Grouping					
GB	5	0.7346	A					
DTE	5	0.7270	A					
RFG	5	0.7252	A					
DTG	5	0.7138	A					
RFE	5	0.7122	A					
GB (OS)	5	0.6633	A	B				
DA	5	0.6002		B	C			
DTE (OS)	5	0.5488			C			
DTG (OS)	5	0.5305			C			
RFE (OS)	5	0.5235			C			
RFG (OS)	5	0.5052			C			
NB	5	0.3825				D		
NB (OS)	5	0.2015					E	
DA (OS)	5	0.07102						F

Factor	N	Mean	Grouping
NN (OS)	5	0.05532	F
LR (OS)	5	0.04701	F
kNN (OS)	5	0.03712	F
kNN	5	0.02125	F
LR	5	0.01959	F
NN	5	0.00766	F

Means that do not share a letter are significantly different.

The Fisher LSD pairwise comparisons with a 95% confidence level in Table 39 presents the same ranking of the algorithms' F1 score, with some differences in the algorithms' grouping.

Table 39. Tukey pairwise comparisons for algorithms' recall for inpatient claim issues

Factor	N	Mean	Grouping							
GB	5	0.7346	A							
DTE	5	0.7270	A							
RFG	5	0.7252	A							
DTG	5	0.7138	A	B						
RFE	5	0.7122	A	B						
GB (OS)	5	0.6633		B						
DA	5	0.6002			C					
DTE (OS)	5	0.5488			C	D				
DTG (OS)	5	0.5305				D				
RFE (OS)	5	0.5235				D				
RFG (OS)	5	0.5052				D				
NB	5	0.3825					E			
NB (OS)	5	0.2015						F		
DA (OS)	5	0.07102							G	
NN (OS)	5	0.05532							G	H
LR (OS)	5	0.04701							G	H
kNN (OS)	5	0.03712							G	H
kNN	5	0.02125							G	H
LR	5	0.01959							G	H
NN	5	0.00766								H

Means that do not share a letter are significantly different.

Chapter Summary

In this chapter, all research questions were reviewed, and the results of hypotheses tests and their implications were discussed. Table 40 summarizes research questions, hypothesis, results, and findings.

Table 40. Summary of Research Results and Key Findings

Item	Description
Research Question 1	What supervised machine learning algorithms can be used to determine Medicaid claim payment issues?
Results	There is no set of universally accepted "best" supervised learning algorithms, but SVM, Naïve Bays, and Neural Network are among the algorithms most used in solving quality problems. SVM requires more computational power, especially for massive datasets. The researcher selected ten algorithms for our research: Decision tree with Entropy configuration (DTE), Decision tree with Gini coefficient configuration (DTG), Random forests with Entropy configuration (RFE), Random forests with Gini configuration (RFG), Naïve Bayes (NN), K Nearest Neighbor (kNN), Logistic Regression (LR), Neural Network (NN), Discriminant Analysis (DA), and Gradient Boosting (GB).
Findings	1. Principal Components Analysis (PCA) is the most used dimensionality reduction technique. 2. There are several trade-offs in selecting classification algorithms, including interpretability vs. prediction accuracy and bias-variance trade-offs.
Research Question 2	What are the most critical measures to compare the performance of different machine learning algorithms (resulted from RQ1) for our problem?
Results	Based on the nature of our problem at hand and the literature review, the researcher selected Recall and F1-score as two metrics to compare our supervised learning models' performance
Findings	1. Most popular performance metrics are derived from the confusion metrics, including recall, precision, accuracy, and F1-score. 2. For the problem dealing with imbalanced data, accuracy will be inflated due to the high number of true negatives in the denominator. Recall and F1-score can measure the sensitivity and specificity of the model more realistically. 3. F1-score outperformers the ROC charts measure the performance of an algorithm on imbalanced data.
Research Question 3	What are the claim attributes that could predict if a given FFS claim has been adjudicated correctly or erroneously (also known as predictors or independent variables)?

Item	Description
Results	The researcher used PCA on the ratio features and concluded all independent ratio variables should be kept in the model to account for over 90% of the variation. The researcher also used business knowledge and subject matter expertise to eliminate the low-information categorical features and selected 16 features for outpatient and 19 features for the inpatient dataset.
Findings	1. Dealing with mixed predictors (categorical and ratio) adds to the complexity of feature reduction. 2. PCA works with ratio and continuous data, while Multiple Correspondence Analysis (MCA) works with categorical data. Methods that could handle both are complex and requires massive computational powers on large datasets.
Research Question 4	Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid outpatient claims?
Hypotheses	$H_{0_{4.1}}: \mu_{11} = \mu_{12} = \cdots = \mu_{116}$, μ_{1i} = feature importance score (Gini importance) of the i-th outpatient predictor.
Tests	Gini importance was calculated, and the hypothesis was tested with one-way ANOVA.
Results	The hypothesis was rejected at $\alpha = 0.05$ significance level
Findings	clm_pmt_amt has the highest Mean F-value, significantly higher than the rest of the features. icd9_dgns_cd_1 and at_physn_npi have the second and third highest F-value for the feature importance significantly lower than clm_pmt_amt. The "claim payment amount" is the most powerful predictor of detecting quality issues in our outpatient claim dataset.
Research Question 5	Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid inpatient claims?
Hypotheses	$H_{0_{5.1}}: \mu_{21} = \mu_{22} = \cdots = \mu_{2.19}$, μ_{1i} = feature importance score (Gini importance) of i-th inpatient predictor.
Tests	Gini importance was calculated, and the hypothesis was tested with one-way ANOVA.
Results	The hypothesis was rejected at $\alpha = 0.05$ significance level.
Findings	clm_pmt_amt and nch_prmry_pyr_clm_pd_amt have the highest Mean F-value, whereas clm_pmt_amt is significantly higher than the rest of the features. The "claim payment amount" is the most powerful predictor of detecting quality issues in our inpatient claim dataset.
Research Question 6	Is there any statistically significant difference among the selected supervised learning algorithms in identifying erroneously adjudicated Medicaid outpatient claims?
Hypotheses	$H_{0_{6.1}}: \mu_{3.1} = \mu_{3.2} = \cdots = \mu_{3.10}$, $\mu_{3.i}$ = the mean recall of algorithm i $H_{0_{6.2}}: \mu_{41} = \mu_{42} = \cdots = \mu_{4.10}$ $\mu_{4.i}$: the mean F1-score of algorithm i

Item	Description
Tests	Two hypotheses were tested with one-way ANOVA.
Results	Two hypotheses were rejected at $\alpha = 0.05$ significance level.
Findings	1. LR (with oversampling) significantly outperforms other algorithms in detecting erroneous payments in the research outpatient data. 2. LR (with oversampling), NB (with and without oversampling) are the next best performers and group together (group B). 3. Oversampling improves LR, NB, GB recall, but there is no statistical significance to this difference. 4. NB slightly performed better without oversampling. This could be due to random effects in the cross-validation process. 5. NN recall for inpatient claims is significantly improved by oversampling. 6. There is a statistically significant difference in the recall rate for the following algorithms compare the LR: DA (with oversampling), NB (with and without oversampling), DT (both with Gini and Entropy feature and oversampling), NN (with oversampling), and GB (with and without oversampling). 7. NB (without oversampling) and GB (with oversampling) outperformed the rest of the algorithms from the F1 score perspective. 8. GB (without oversampling) is the third performing algorithm on recall for outpatient claims. 9. Oversampling significantly improves the F1-score of the GB in outpatient data. 10. DT and GB (regardless of their feature configuration and sampling approach) have the highest F1 score for inpatient data.
Research Question 7	Is there any statistically significant difference among the selected supervised learning algorithms) in identifying erroneously adjudicated Medicaid inpatient claims?
Hypotheses	$H_{0_{7.1}}: \mu_{5.1} = \mu_{5.2} = \cdots = \mu_{5.10}$ (μ_{5i}: the mean recall of algorithm i) $H_{0_{7.2}}: \mu_{6.1} = \mu_{6.2} = \cdots = \mu_{6.10}$ (μ_{4i}: the mean F1-score of algorithm i)
Tests	Two hypotheses were tested with one-way ANOVA.
Results	Two hypotheses were rejected at $\alpha = 0.05$ significance level.
Findings	1. GB (with oversampling) outperformed the rest of the models in recalling quality issues of the inpatient claim data. 2. kNN, NN, DA, and LR (without sampling) have significantly lower recall than the rest of the algorithms. 3. Oversampling has significantly improved LR recall. A sign of the potential impact of imbalanced data on the performance of this algorithm. 4. GB (both without oversampling), DT (using Entropy and Gini), RF (with entropy and without oversampling) have significantly higher F1 on inpatient data.

Item	Description
Research Question 8	Are the most powerful predictors different between outpatient and inpatient claims? Are the most accurate algorithms in detecting erroneously paid claims different between the two types of Medicaid claims studied?
Results	1. *clm_pmt_amt* is the most powerful predictor for both datasets. 2. For outpatient claims, *icd9_dgns_cd_1* and *at_physn_npi* have the next highest importance scores. 3. For inpatient claims *nch_prmry_pyr_clm_pd_amt,* and *clm_drg_cdn* the have the next highest importance scores. 4. GB and DT (with various configurations and sampling strategies) outperform the rest of the algorithms in recall and F1-measure on both datasets.
Findings	1. NB and LR performed considerably better in outpatient data than inpatient data. This shows this algorithm requires more labeled data to train and performed well. 2. DA, kNN, and NN algorithms did not perform as well as the rest of the algorithm. 3. In general, oversampling has improved the performance of the algorithms.

CHAPTER 5

DISCUSSIONS, CONCLUSIONS, AND RECOMMENDATIONS

Chapter Overview

This chapter discusses the research result in more detail for each research question, provides conclusions, and offers further studies. This chapter is concluded by giving a summary table that presents research questions, hypotheses, results and findings, and discussions of significant findings concisely and tabularly.

Discussion of the Results

The purpose of this study was to develop a supervised learning model to detect the Medicaid FFS claims with the high chance of being adjudicated erroneously. Eight research questions were articulated, seven hypotheses were formulated, hypothesis testing and other statistical tests were performed, and the results were reviewed. Here crucial findings will be reviewed and discussed.

Research Question 1 (RQ1)- What supervised machine learning algorithms can be used to determine Medicaid claim payment issues?

The researcher started this research by reviewing the literature to understand what supervised machine learning algorithms can determine Medicaid claim payment issues. The researcher examined the literature extensively and found out among major classes of machine learning algorithms, supervised learning, and the most used method to detect quality issues, which is essentially a classification problem. While there is no set of universally accepted "best" supervised learning algorithms, SVM, Naïve Bays, and Neural Network are among the algorithms most used in solving quality problems. The researcher also found Principal Components Analysis (PCA) is the most used dimensionality reduction technique allowing researchers to limit the number of features without compromising overall performance. Many factors should be considered in selecting proper algorithms, including their accuracy, detection power (recall), stability, and computation time. There are several trade-offs in selecting classification algorithms, including interpretability vs. prediction accuracy and bias-variance trade-offs. The researcher concluded this review by selecting ten different supervised learning algorithms for our problem.

Research Question 2 (RQ2)- What are the most critical measures to compare the performance of different machine learning algorithms (resulted from RQ1) for our problem?

The second research question was around finding the most critical measures to compare different machine learning algorithms' performance. While there are many metrics that could be used to compare machine learning algorithms' performance, the problem's nature dictates the superior metrics. Significant findings related to this research question could be summarized in three points:

1. The most famous performance metrics are around false and true positives and all derived from the confusion metrics, which is easy to calculate and communicate.
2. For the problem dealing with imbalanced data, accuracy will be inflated due to the high number of true negatives in the denominator. Recall and F1-score can measure the sensitivity and specificity of the model more realistically.

3. F1-score outperformers the ROC charts measure the performance of an algorithm on imbalanced data.

Based on the nature of our problem at hand and the literature review, Recall and F1-score were selected as two metrics to compare the supervised learning models' performance.

Research Question 3 (RQ3)- What are the claim attributes that could predict if a given FFS claim has been adjudicated correctly or erroneously (also known as predictors or independent variables)?

The third research question was about finding the predictors to detect quality issues FFS inpatient and outpatient claims? There were two massive datasets with over 100 features. In theory, all features could be used to build a supervised learning model. The researcher found thorough the literature review and some tests that this will increase the models' complexity without improving their performances. Dealing with mixed predictors (categorical and ratio) adds to the complexity of feature reduction. PCA on the ratio features showed all ratio features should be kept in our model to account for over 90% of the variation. Business knowledge and subject matter expertise were applied to eliminate the low-information categorical features. The researcher finally reduced the number of features by over 80% and selected 16 features for outpatient models and 19 features for inpatient models.

Research Question 4 (RQ4) and Research Question 5 (RQ5)- Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid outpatient claims? Is there any statistically significant difference among predictors' predictability power in identifying erroneously processed Medicaid inpatient claims?

Research questions 4 and 5 were related to finding the most powerful predictors in detecting quality issues in our claim datasets for outpatient claims (RQ4) and inpatient claims (RQ5). Gini importance was calculated, and the hypothesis was tested with one-way ANOVA. For both outpatient and inpatient claims, the Tukey pairwise comparisons using the Tukey method and a 95% confidence level shown the features (*clm_pmt_amt*, *nch_prmry_pyr_clm_pd_amt*, and *clm_drg_cd*) have the highest Mean F-value, whereas *clm_pmt_amt* is significantly higher than the rest of the features. The Fisher LSD method also presented a significant difference in *clm_pmt_amt*'s feature importance scores compared to other features. However, the Fisher test results are slightly different from the Tukey test results for inpatient claims. This test grouped *nch_prmry_pyr_clm_pd_amt* and *clm_drg_cdn* together with a significant difference between these two features and the rest of the features. This is because the Tukey test is more conservative (Andy, 2015). Also, Fisher's LSD does not control the family-wise error rate (FWER) (Scheffe, 1953). Some researchers suggested using a more liberal method like Fisher's LSD to avoid being unnecessarily conservative (with weak statistical power) and detect a real difference when the goal is to prevent type II errors (S. Lee & Lee, 2018). In our case, it is preferred not to lose the opportunity to detect a real power of a feature in predicting the result, so both Tukey and Fisher LSD tests were conducted, and results were presented. From a practical point of view, proving that "Claim Payment Amount" (*clm_pmt_amt*) is our most crucial predictor was somehow expected. Differences in claim payment, especially for large under or overpayments, make a payment an outlier in our dataset. The researcher applied the pattern of adjudication errors on the claim payments using the claim errors distribution function, without providing and adjusting relevant data elements, including:

- Who has processed the claim?
- What were applicable policies impacting the claim pricing methodology:
- What was the degree of difficulty in interpreting specific claim pricing policies?

The second most powerful predictor is a payment-related feature, too: "NCH Primary Payer Claim Paid Amount" (nch_prmry_pyr_clm_pd_amt). Its prediction power is probability contributed to the fact that this amount provides a comparison point across various payments. The third predictor with a significantly higher feature important score is: “Claim Diagnosis Related Group Code” (clm_drg_cd). Its prediction power could be explained by the fact that for inpatient claims, pay-per-performance methods require providers like hospitals to bundle the related claims, thus providing another point of comparison for the algorithm to create a baseline and detect anomalies.

Research Question 6 (RQ6)- Is there any statistically significant difference among the selected supervised learning algorithms in identifying erroneously adjudicated Medicaid outpatient claims?

To examine if there is a significant difference among the selected supervised learning algorithms in identifying quality issues in outpatient claims, the researcher conducted a one-way ANOVA test. The result showed the Logistic Regression model (with oversampling) has the best detection power (recall). Logistic Regression model (with oversampling) and Naïve Bayes (with and without oversampling) are the next best performers and group together (group B). The oversampling strategy applied in this research (as explained in chapter 3) has improved the recall and F1-score of models most of the time. In a few cases, the improvement done by oversampling is statistically significant (e.g., Neural Network recall improvement for inpatient claims, F1-score of the Gradient Boosting in outpatient data, etc.). Naïve Bays (without oversampling) and Gradient Boosting (with oversampling) outperformed the rest of the algorithms from the F1 score perspective in outpatient data. Oversampling significantly improves the F1-score of the Gradient Boosting algorithm in outpatient data. As described in chapter 3, the SMOTE algorithm was used for oversampling of the minority class and improved the Gradient Boosting performance. This is congruent with the findings of other researchers working on imbalanced data (Cahyana et al., 2019).

Research Question 7 (RQ7)- Is there any statistically significant difference among the selected supervised learning algorithms) in identifying erroneously adjudicated Medicaid inpatient claims?

To examine if there is a significant difference among the selected supervised learning algorithms in identifying quality issues in inpatient claims, the researcher conducted a one-way ANOVA test. Gradient Boosting (with oversampling) outperformed the rest of the models in recalling quality issues of the inpatient claim data. k-Nearest Neighbor, Neural Network, Discriminant Analysis, and Logistic Regression (without sampling) have significantly lower recall than the rest of the algorithms on the inpatient dataset. Gradient Boosting (both without oversampling), Decision Tree (using Entropy and Gini), and Random Forest (with entropy and without oversampling) have significantly higher F1 on inpatient data. Oversampling has significantly improved LR recall. A sign of the potential impact of imbalance data on the performance of this algorithm

Research Question 8 (RQ8)- Are the most powerful predictors different between outpatient and inpatient claims? Are the most accurate algorithms in detecting erroneously paid claims different between the two types of Medicaid claims studied?

Research question 8 has two parts:

1. Are the most powerful predictors different between outpatient and inpatient claims? The result of our ANOVA tests shows *clm_pmt_amt* is the most powerful predictor for both datasets, which was expected. For outpatient claims, *icd9_dgns_cd_1*, and *at_physn_npi* have

the second-highest importance scores. For inpatient claims *nch_prmry_pyr_clm_pd_amt,* and *clm_drg_cdn* the have the second-highest importance scores.

2. Are the most accurate algorithms in detecting erroneously paid claims different between the two types of Medicaid claims studied?

    This question is complex and requires comparing several measures. Table 41 illustrates the difference among various tested algorithms on both datasets. Tested algorithms are sorted based on their recall and F1-score from 1 to 20, whereas 1 shows the algorithm with the highest score for recall or F-score. Then an average rank is calculated, and algorithms are sorted based on their average rank.

Table 41. Summary of Research Results and Key Findings

Ranking of Algorithms on Outpatient Data				Ranking of Algorithms on Inpatient Data			
Algorithm	Recall Rank	F1-Score Rank	Average Rank	Algorithm	Recall Rank	F1-Score Rank	Average Rank
NB	3	1	2.0	GB	4	1	2.5
GB (OS)	9	2	5.5	GB (OS)	1	6	3.5
DTE (OS)	6	8	7.0	DTE	6	2	4.0
GB	12	3	7.5	DTG (OS)	2	8	5.0
DTG (OS)	7	9	8.0	DTG	7	4	5.5
LR	2	14	8.0	RFG	8	3	5.5
LR (OS)	1	15	8.0	DTE (OS)	5	7	6.0
NB (OS)	4	12	8.0	RFE	10	5	7.5
RFE (OS)	11	6	8.5	DA (OS)	3	13	8.0
DA (OS)	5	13	9.0	NB (OS)	9	12	10.5
DTG	14	4	9.0	RFE (OS)	12	9	10.5
DTE	15	5	10.0	NB	11	11	11.0
RFG (OS)	13	7	10.0	RFG (OS)	13	10	11.5
NN (OS)	8	17	12.5	NN (OS)	14	14	14.0
kNN (OS)	10	16	13.0	LR (OS)	15	15	15.0
RFG	16	10	13.0	kNN (OS)	16	16	16.0
RFE	17	11	14.0	kNN	18	17	17.5
kNN	19	18	18.5	NN	17	19	18.0
NN	18	19	18.5	LR	19	18	18.5
DA	20	20	20.0	DA	20	20	20.0

Table 41 shows there are commonalities between the best and least effective algorithms across both datasets. Gradient Boosting (with or without sampling), Decision Tree (both with entropy and Gini classifiers), and Random Forests have the best overall performance from both recall and F1-scores. There are some critical differences in the performance of some algorithms on two different datasets. Naïve Bayes is the best performing algorithm on outpatient data. This algorithm's precision, and consequently its F1-score, decreased with oversampling. This could have caused because of overfitting of the data. Logistics Regression has an excellent recall on an outpatient dataset but not so much for inpatient claims. Random Forest models performed better on the inpatient dataset.

Importance of the findings and practical implications

As mentioned in previous chapters, sampling and inferential statistics are the dominant quality tools utilized by Medicaid FFS programs. Such methods simply offer an interval estimate of the size of the problem in payments (e.g., estimated erroneous payments). They are not designed to detect potential erroneous payment at scale. This study aims to apply supervised learning as a method to detect such possible erroneous payments at scale. The average accuracy and recall rate for the test algorithms are not as high as one would like to see. However, if this method is implemented successfully using operational data, it would be a game-changer. Let us provide a rough estimate of the potential saving that could be realized by implementing this method in a typical Medicaid FFS. Assume the following assumptions:

- The number of adjudicated claims a year 1,000,000 (many Medicaid FFS programs process a considerably higher volume).
- Percentage of erroneous payments 5% (a conservative estimate per PERM reports).

- Estimate the number of erroneously processed claims: 50,000.
- The average cost per erroneously adjudicated claim: $300. The average cost of a claim varies based on the claim type, but $300 is a conservative estimate (Substance Abuse and Mental Health Services Administration, 2015).
- Total potential saving: $15 million a year.
- Average manually sampled and review claims by the QM team at a typical Medicaid FFS: 50 a day, 10,000 reviewed claim a year.
- Potential saving through detecting potential erroneous payment (if they are detected pre-payment): 10,000*%5*$300= $150K.
- Cost of each review: $5, so the total cost of review 10,000 samples is $50 K.
- Actual saving for the Medicaid FFS program: $150k-$50k= $100K.

So, in this scenario, conventional sampling will save about $100K. If the same program can design and implement a model with a recall rate of 50% and precision of 10% (which equals an F1-score of 16.7%), the researcher calculates the potential saving using the assumptions listed in the previous scenario:

- Recall 50% means the model could potentially detect half of 50K erroneous payments or 25K claims.
- 10% precision means to detect the 25K erroneous claims, 250K claims should be reviewed.
- Potential saving through detecting possible erroneous payment (if they are caught pre-payment): 25,000*$300=$7.5M
- The total cost of reviewing 55K claims (25K true positives and 250K false positives): 255K*$10=$2.5M

- Actual saving for the Medicaid FFS program: $7.5-$2.5M= $5M

When real-world data is used, and the model is fine-tuned, then the accuracy rate will increase, and the cost-saving would be even more substantial. One recommendation is to use several algorithms in tandem and create a scoring model to reduce the number of to improve overall accuracy. A Medicaid FFS program can set up a multi-phase approach in which algorithms with higher recall used to collect as many true positives as possible and then use algorithms with higher accuracy to detect true positives.

Recommendations for Further Research

The most prominent recommendation for further research based on this dissertation's findings is to apply the same method to detect quality issues on other Medicaid FFS claim types. This research methodology can be used for other Medicaid FFS claim typos, including medical (e.g., DME, vision, etc.), dental, and crossover claims. Medicare does not pay for most of the long-term care (LTS) claims. So, one should start with finding a set of Medicaid FFS LTC claims before applying the method recommended by this dissertation. As shown in the result gathered from the CA-MMIS project, quality issues in the Medicaid FFS claim adjudication process are rare events. As a result, collecting enough data to label target training datasets for specific claim types may require access to years’ worth of data and be very time-consuming. One way to overcome this challenge is to study the possibility of using text analytics and unsupervised learning methods to automate the data labeling process and then train the supervised algorithms built in this project.

Another aspect of this research that could be expanded is to apply the same process and include other algorithms. For example, SVM was not selected for testing because it requires massive computation power on the mix and massive datasets like datasets used for this research. However, it is recommended that other researchers with more computational power at their disposal to include SVM in their list of algorithms. As shown in chapter two- SVM has been successfully applied to solve many classification quality issues and frequently came on the top of this list from the accuracy and recall point of view.

As discussed in the manage data section, a synthesized dataset created by CMS was used for this research. The method explained in CMS's DE-SynPUF codebook user manual (The Centers for Medicare and Medicaid Services, 2013b) has described the data synthesizing process in detail. Then, to create labels, the general pattern of quality issues on a Medicaid FFS program was studied (e.g., population proportion, underlying distribution function, descriptive statistics, etc.). This approach eliminates the chance of any PHI disclosure. The biggest drawback of using a synthesized dataset is the limitation of multivariate modeling and model validation. As explained in the CMS's DE-SynPUF codebook, when the dynamic relationships between variables are altered (in this case, demographic, clinical, financial, and provider data), analyses from multivariate modeling should be interpreted with caution because the generated dataset may or may not inherit all the critical characteristic of the original dataset (The Centers for Medicare and Medicaid Services, 2013a). It may be appropriate for future research to use the claim and quality issues data from the same Medicaid program and start with the deidentification of sensitive data. As explained in chapter two, the de-identification of PHI data has its disadvantages compared to the method used in this study (data de0identification). However, using a real or de-identified dataset will result in a more useful predictive model. This may require the researcher(s) to include other related datasets to the research data, including beneficiary data, claim examiners' information and information related to the pricing policies (e.g., error codes). These following data elements are examples of data elements that were not used in this research. A researcher can include them to their dataset, follow the steps explained in chapter 4, and examine if they can find better predictors for quality issues in the Medicaid FFS claim adjudication data:

- Media Type Electronic vs. paper claims)
- Policy information (e.g., Error Code)
- Procedure Code/NDC/Revenue Code
- MMIS internal processing codes including Roll Numbers and Remittance Advice Details (RAD)
- Programs (e.g., FQHC, GHPP, etc.)
- Provider Type
- Information about claim examiners

During the review of the claim adjudication quality issues (in our sample gathered from the CA-MMIS project), the researcher has also observed anecdotal evidence that answering yes or no to the following question could be used as a predictor for quality issues:

- Is the Error Code an audit type or not?
- Is claim a crossover or not?
- Is it paid as billed? (Billed Amount = Paid Amount)
- Is payment based on the Provider Master File?
- Is the Provider's Frequency over the limits?

- Is the Zip Code, including DRG/ACA, negotiated ZIP Codes?
- Is the procedure code Hemophilia?
- Is payment based on modifiers?

Chapter Summary

In this chapter, the researcher reviewed all research questions, discuss the results of hypotheses tests and their implications, and concluded the section with recommendations for further research.

REFERENCES

Altini, M. (2015). *Dealing with imbalanced data: undersampling, oversampling, and proper cross-validation*. https://www.marcoaltini.com/blog/dealing-with-imbalanced-data-undersampling-oversampling-and-proper-cross-validation

Amazon. (n.d.-a). *Amazon Athena FAQs – Serverless Interactive Query Service – Amazon Web Services*. Retrieved October 24, 2020, from https://aws.amazon.com/athena/faqs/

Amazon. (n.d.-b). *Amazon DynamoDB | NoSQL Key-Value Database | Amazon Web Services*. Retrieved October 24, 2020, from https://aws.amazon.com/dynamodb/

Anbarasi, M. S., & Dhivya, S. (2017). Fraud detection using outlier predictor in health insurance data. *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, 1–6. https://doi.org/10.1109/ICICES.2017.8070750

Andy. (2015, August 31). *Tukey's Test or LSD Fisher - Cross Validated*. Stats.Stackexchange. https://stats.stackexchange.com/questions/169412/tukeys-test-or-lsd-fisher

Arizona Health Care Cost Containment System. (n.d.). *AHCCCS Fee-For-Service Fee Schedules Proposed Fee Schedules*. Retrieved June 17, 2018, from https://www.azahcccs.gov/PlansProviders/RatesAndBilling/FFS/

Ayodele, T. O. (2010). Types of Machine Learning Algorithms. In Y. Zhang (Ed.), *New Advances in Machine Learning*. InTech. https://doi.org/http://dx.doi.org/10.5772/46845

Bai, Y., Sun, Z., Deng, J., Li, L., Long, J., & Li, C. (2017). Manufacturing Quality Prediction Using Intelligent Learning Approaches: A Comparative Study. *Sustainability*, *10*, 85.

Bartz-Beielstein, T., Chiarandini, M., Paquete, L., & Preuss, M. (2010). Experimental methods for the analysis of optimization algorithms. In *Experimental Methods for the Analysis of Optimization Algorithms*. https://doi.org/10.1007/978-3-642-02538-9

Beal, V. (2018). *What is Structured Data?* Webopedia Definition. https://www.webopedia.com/TERM/S/structured_data.html

Bell, J. (2014). Machine Learning : Hands-On for Developers and Technical Professionals. In *Machine Learning: Hands-On for Developers and Technical Professionals* (1st ed.). Wiley. https://doi.org/10.1002/9781119183464

Bernardi, L., Mavridis, T., & Estevez, P. (2019). 150 Successful Machine Learning Models: 6 Lessons Learned at Booking.Com. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery &amp; Data Mining*, 1743–1751. https://doi.org/10.1145/3292500.3330744

Bhalla, D. (2016). *Oversampling for rare event*. https://www.listendata.com/2015/04/oversampling-for-rare-event.html

Blomquist, H., & Möller, J. (2015). *Anomaly detection with Machine learning – Quality assurance of statistical data in the Aid community*. http://www.utn.uu.se/sts/cms/node/908

Borget, I., Laville, I., Paci, A., Michiels, S., Mercier, L., Desmaris, R.-P., & Bourget, P. (2006). Application of an acceptance sampling plan for post-production quality control of chemotherapeutic batches in an hospital pharmacy. *European Journal of Pharmaceutics and Biopharmaceutics*, *64*(1), 92–98. https://doi.org/https://doi.org/10.1016/j.ejpb.2006.04.002

Bouguila, N., Wang, J. H., & Hamza, A. Ben. (2008). A Bayesian approach for software quality prediction. *2008 4th International IEEE Conference Intelligent Systems*, *2*, 11–54. https://doi.org/10.1109/IS.2008.4670508

Bowden, R., & Bullington, S. F. (1996). Development of manufacturing control strategies using unsupervised machine learning. *IIE Transactions*, *28*(4), 319–331.

Brain, D., & Webb, G. I. (1999). On The Effect of Data Set Size on Bias And Variance in Classification Learning. *Proceedings of the Fourth Australian Knowledge Acquisition Workshop (AKAW '99)*, 117–128.

Brownlee, J. (2019). *How to Choose a Feature Selection Method For Machine Learning*. https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/

Cahyana, N., Khomsah, S., & Aribowo, A. S. (2019). Improving Imbalanced Dataset Classification Using Oversampling and Gradient Boosting. *2019 5th International Conference on Science in Information Technology (ICSITech)*, 217–222. https://doi.org/10.1109/ICSITech46713.2019.8987499

Calvin, & Setiawan, J. (2014). Using Text Mining to Analyze Mobile Phone Provider Service Quality (Case Study: Social Media Twitter). *International Journal of Machine Learning and Computing*, *4*(1), 106–109. https://doi.org/10.7763/IJMLC.2014.V4.395

Cano, J. (2014). *The V's of Big Data: Velocity, Volume, Value, Variety, and Veracity*. https://www.xsnet.com/blog/bid/205405/the-v-s-of-big-data-velocity-volume-value-variety-and-veracity

Castle, N. (2018). *What is Semi-Supervised Learning?* https://www.datascience.com/blog/what-is-semi-supervised-learning

Chang, M. (2017, September 30). *4 Stages of the Machine Learning (ML) Modeling Cycle | LinkedIn*. https://www.linkedin.com/pulse/4-stages-machine-learning-ml-modeling-cycle-maurice-chang/

Chawla, N. V, Bowyer, K., Hall, L., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority

Over-sampling Technique. *J. Artif. Intell. Res.*, *16*, 321–357.

Chelaru-Centea, N. (2019). *Calculate principal components of mixed-type data – Nextjournal*. https://nextjournal.com/pc-methods/calculate-pc-mixed-data

Chou, J.-S., Ho, C.-C., & Hoang, H.-S. (2018). Determining quality of water in reservoir using machine learning. *Ecological Informatics*, *44*, 57–75. https://doi.org/10.1016/J.ECOINF.2018.01.005

Çitak, E., & Genç, Y. (2017). Machine Learning For Product Quality Inspection. *Signal Processing and Communications Applications Conference (SIU)*, 1–4.

Congressional Budget Office. (2017). *The Federal Budget in 2016: An Infographic | Congressional Budget Office*. https://www.cbo.gov/publication/52408

Copeland, L., Edberg, D., Panorska, A. K., & Wendel, J. (2012). Applying Business Intelligence Concepts to Medicaid Claim Fraud Detection. *Journal of Information Systems Applied Research (JISAR)*, *5*(1), 51–61. www.aitp-edsig.org

Dean, J. (2014). Big data, data mining, and machine learning: value creation for business leaders and practitioners. In *Wiley & SAS business series*.

Dehghan, A., Kovacevic, A., Karystianis, G., Keane, J. A., & Nenadic, G. (2015). Combining knowledge- and data-driven methods for de-identification of clinical narratives. *Journal of Biomedical Informatics*, *58*, S53–S59. https://doi.org/10.1016/J.JBI.2015.06.029

El-hasnony, I. M., Bakry, H. M. El, & Saleh, A. A. (2015). Comparative Study among Data Reduction Techniques over Classification Accuracy. *International Journal of Computer Applications*, *122*(2), 8–15. https://doi.org/10.5120/21671-4752

El Emam, K. (2013). *Guide to the De-Identification of Personal Health Information*. CRC Press.

Elish, K. O., & Elish, M. O. (2008). Predicting defect-prone software modules using support vector machines. *Journal of Systems and Software*. https://doi.org/10.1016/j.jss.2007.07.040

Escobar, C. A., & Morales-Menendez, R. (2018). Machine learning techniques for quality control in high conformance manufacturing environment. *Advances in Mechanical Engineering*, *10*(2), 1–16. https://doi.org/10.1177/1687814018755519

Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, *17*(3), 37. https://doi.org/10.1609/aimag.v17i3.1230

Feuerverger, A. (2016). On Goodness of Fit for Operational Risk. *International Statistical Review*, *84*(3), 434–455. https://doi.org/10.1111/insr.12112

Finlay, S. (2014). How to Build a Predictive Model. In *Predictive Analytics, Data Mining and Big Data. Myths, Misconceptions and Methods* (pp. 157–178). Palgrave Macmillan. https://doi.org/https://doi.org/10.1057

Firican, G. (2017). *The 10 Vs of Big Data*. Transforming Data with Intelligence. https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx

Fu, H.-H., Tsai, H.-T., Lin, C.-W., & Wei, D. (2004). Application of a single sampling plan for auditing medical-claim payments made by Taiwan National Health Insurance. *Health Policy*, *70*(2), 185–195. https://doi.org/10.1016/J.HEALTHPOL.2004.02.006

Fumo, D. (2017). *Types of Machine Learning Algorithms You Should Know*. https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861

Gallardo, K. (2017). *Analytics Engine for Detecting Medical Fraud, Waste, and Abuse* (Patent No. US20170270435A1). https://patents.google.com/patent/US20170270435A1/en

Getchius, J. M. (2014). *Healthcare fraud detection with machine learning* (Patent No. US20140149128 A1).

Gondra, I. (2008). Applying machine learning to software fault-proneness prediction. *Journal of Systems and Software*, *81*(2), 186–195. https://doi.org/10.1016/j.jss.2007.05.035

Gonzalez Viejo, C., Fuentes, S., Torrico, D., Howell, K., & Dunshea, F. R. (2017). Assessment of beer quality based on foamability and chemical composition using computer vision algorithms, near infrared spectroscopy and machine learning algorithms. *Journal of the Science of Food and Agriculture*. https://doi.org/10.1002/jsfa.8506

Griffith, M. (2015). *Identifying the Distribution of Your Data*. Https://Blog.Minitab.Com/. https://blog.minitab.com/blog/meredith-griffith/identifying-the-distribution-of-your-data

Gupta, Y. (2018). Selection of important features and predicting wine quality using machine learning techniques. *Procedia Computer Science*, *125*, 305–312. https://doi.org/10.1016/J.PROCS.2017.12.041

Han, J., & Kamber, M. (2001). Data Mining: Concepts and Techniques. In *Systems Research*. https://doi.org/10.1017/CBO9781107415324.004

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning. In *Data mining, inference, and prediction*. https://doi.org/10.1007/978-0-387-84858-7

Hawaii Department of Human Services, & Med-QUEST. (n.d.). *Medicaid Fee-For-Service Eligibility*. Retrieved June 17, 2018, from http://humanservices.hawaii.gov/mqd/medicaid-fee-for-service/#Eligibility-0

Heino, J., & Toivonen, H. (2003). Automated Detection of Epidemics from the Usage Logs of a Physicians' Reference Database. *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 180–191. https://doi.org/10.1007/978-3-540-39804-2_18

Henderson, S. (2017). *What Health Plan Need to Know about Medicaid Encounter Data*.

http://www.burchfieldgroup.com/health-plan-insurer-blog/what-health-plans-need-to-know-about-medicaid-encounter-data

How Much Data Does Each DVD Format Hold? (2018). Lifewire. https://www.lifewire.com/dvd-formats-data-limits-1130690

IBM Big Data & Analytics Hub. (n.d.). *Infographic: The Four V's of Big Data*. Retrieved May 29, 2018, from http://www.ibmbigdatahub.com/infographic/four-vs-big-data

Iowa Medicaid Enterprise Performance Report. (2006). https://www.legis.iowa.gov/docs/publications/IH/4906.pdf

Irgens, C., Wuest, T., & Thoben, K.-D. (2012). Product state based view and machine learning: A suitable approach to increase quality? *IFAC Proceedings Volumes*, *45*(6), 1733–1738. https://doi.org/10.3182/20120523-3-RO-2023.00083

ISO/IEC JTC 1 Information technology. (2015). Big data Preliminary Report 2014. In *Iso / Iec* (pp. 1–36). ISO. https://doi.org/10.1016/B978-0-7020-2920-2.50020-0

Issa, H., & Vasarhelyi, M. A. (2011). Application of Anomaly Detection Techniques to Identify Fraudulent Refunds. In *Available at SSRN 1910468* (pp. 1–19). https://doi.org/10.2139/ssrn.1910468

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). An Introduction to Statistical Learning. In *Springer Texts in Statistics*. https://doi.org/10.1016/j.peva.2007.06.006

Kai, J., Miao, H., & Gao, H. (2016). *A Survey of Quality Prediction Methods of Service-oriented Systems*. *9*(4), 183–198.

Karimi, N., Ranjbarzadeh Kondrood, R., & Alizadeh, T. (2017). An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms. *Measurement*, *107*, 68–76.

https://doi.org/10.1016/J.MEASUREMENT.2017.05.009

Kerry, S. M., & Bland, J. M. (1998). The intracluster correlation coefficient in cluster randomisation. *BMJ*, *316*(7142), 1455–1460. https://doi.org/10.1136/bmj.316.7142.1455

Kevin P. Murphy. (2014). *Machine Learning : A Probabilistic Perspective*. MIT Press.

Khoshgoftaar, T. M., & Seliya, N. (2003). Software Quality Classification Modeling Using the SPRINT Decision Tree Algorithm. *International Journal on Artificial Intelligence Tools*, *12*(03), 207–225. https://doi.org/10.1142/S0218213003001204

KNIME. (n.d.). *Classification and Predictive Modelling*. Retrieved June 16, 2018, from https://www.knime.com/nodeguide/analytics/classification-and-predictive-modelling

Knobloch, R., Mlýnek, J., & Srb, R. (2017). The classic differential evolution algorithm and its convergence properties. *Applications of Mathematics*, *62*(2), 197–208. https://doi.org/10.21136/AM.2017.0274-16

Kohavi, R., & Provost, F. (1998). *Glossary of Terms*. Machine Learning. http://ai.stanford.edu/~ronnyk/glossary.html

Kose, I., Gokturk, M., & Kilic, K. (2015). An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Applied Soft Computing Journal*. https://doi.org/10.1016/j.asoc.2015.07.018

Kumar, A. (2018). *How Honda is using cognitive search to drive real changes*. IBM Big Data & Analytics Hub. http://www.ibmbigdatahub.com/blog/how-honda-using-cognitive-search-drive-real-changes-quality-assurance

Lee, J. (2014). *Memory Sizes Explained - Gigabytes, Terabytes &amp; Petabytes in Layman's Terms*. https://www.makeuseof.com/tag/memory-sizes-gigabytes-terabytes-petabytes/

Lee, S., & Lee, D. K. (2018). What is the proper way to apply the multiple comparison test?

Korean Journal of Anesthesiology, *71*(5), 353–360. https://doi.org/10.4097/kja.d.18.00242

Levinger, D. (2019). *Six Steps to Master Machine Learning with Data Preparation*. Kdnuggets.Com. https://www.kdnuggets.com/2018/12/six-steps-master-machine-learning-data-preparation.html

Li, J., Huang, K.-Y., Jin, J., & Shi, J. (2008). A survey on statistical methods for health care fraud detection. *Health Care Management Science*, *11*(3), 275–287. https://doi.org/10.1007/s10729-007-9045-4

Lieber, D., Stolpe, M., Konrad, B., Deuse, J., & Morik, K. (2013). Quality prediction in interlinked manufacturing processes based on supervised & unsupervised machine learning. *Procedia CIRP*. https://doi.org/10.1016/j.procir.2013.05.033

Lo, A., Chernoff, H., Zheng, T., & Lo, S.-H. (2015). Why significant variables aren't automatically good predictors. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(45), 13892–13897. https://doi.org/10.1073/pnas.1518285112

Lotfi, E., Yaghoobi, M., & Pourreza, H. R. (2008). A new approach for automatic quality control of fried potatoes using machine learning. *2008 7th IEEE International Conference on Cybernetic Intelligent Systems*, 1–4. https://doi.org/10.1109/UKRICIS.2008.4798934

Manish. (2017). *How to build a career in Data Science*. Admission Table. https://www.admissiontable.com/ms-data-science/

Marc-Oliver Arsenault. (2017). *KOLMOGOROV–SMIRNOV TEST. A needed tool in your data science… | by Marc-Olivier Arsenault | Towards Data Science*. https://towardsdatascience.com/kolmogorov-smirnov-test-84c92fb4158d

Matuszak, G., Hanley, R., & Rios, P. (2015). *The changing landscape of disruptive technologies - Global Technology Innovation Survey*.

Mayo, M. (2020). *Frameworks for Approaching the Machine Learning Process*. KDnuggets. https://www.kdnuggets.com/2018/05/general-approaches-machine-learning-process.html

Medicaid.gov. (2018). *March 2018 Medicaid &amp; CHIP Enrollment Data Highlights*. Medicaid.Gov. https://www.medicaid.gov/medicaid/program-information/medicaid-and-chip-enrollment-data/report-highlights/index.html

Mencar, C. (2016). *Should oversampling be done before or within cross-validation?* Researchgate.Net. https://www.researchgate.net/post/should_oversampling_be_done_before_or_within_cross-validation

Meystre, S., Friedlin, F., South, B., Shen, S., & Samore, M. (2010). Automatic de-identification of textual documents in the electronic health record: a review of recent research. *BMC Medical Research Methodology*, *10*(1), 70. http://scholar.google.co.uk/scholar.bib?q=info:lqS6CSi29RoJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&as_ylo=2007&ct=citation&cd=2

Minitab. (2017). *The Anderson-Darling statistic*. https://support.minitab.com/en-us/minitab/18/help-and-how-to/statistics/basic-statistics/supporting-topics/normality/the-anderson-darling-statistic/

Minitab. (2019). *Example of Individual Distribution Identification*.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2014). Foundations of Machine Learning. In *Adaptive Computation and Machine Learning Serries*. MIT Press.

Morales, K. (2017). *Predictive Quality Analytics: delivering better quality, faster*. Atlassian Blog. https://www.atlassian.com/blog/add-ons/predictive-quality-analytics-delivering-better-quality-faster

Neamatullah, I., Douglass, M. M., Lehman, L. W. H., Reisner, A., Villarroel, M., Long, W. J., Szolovits, P., Moody, G. B., Mark, R. G., & Clifford, G. D. (2008). Automated de-identification of free-text medical records. *BMC Medical Informatics and Decision Making*, *8*(32), 1–17. https://doi.org/10.1186/1472-6947-8-32

Nevada Department of Health and Human Services, D. of H. C. F. and P. (n.d.). *NHHS Fee Schedules*. Retrieved June 17, 2018, from http://dhcfp.nv.gov/Resources/Rates/FeeSchedules/

Niwratti. (2020, January 4). *6 stages to get success in Machine Learning project | by Niwratti | Analytics Vidhya | Medium*. Medium.Com. https://medium.com/analytics-vidhya/6-stages-to-get-success-in-machine-learning-project-2900555327bc

NM Human Services Department (HSD). (n.d.). *NM Fee for Service Fee Schedules*. Retrieved June 17, 2018, from http://www.hsd.state.nm.us/providers/fee-for-service.aspx

Normandeau, K. (2013). *Big data volume, variety, velocity and veracity*. https://insidebigdata.com/2013/09/12/beyond-volume-variety-velocity-issue-big-data-veracity/

Ordukaya, E., & Karlik, B. (2017). Quality Control of Olive Oils Using Machine Learning and Electronic Nose. *Journal of Food Quality*. https://doi.org/10.1155/2017/9272404

Other requirements relating to uses and disclosures of protected health information, Pub. L. No. §164.514 (1996). https://www.ecfr.gov/cgi-bin/text-idx?SID=e58a563f56b8cf8e6511be534d364a64&node=se45.1.164_1514&rgn=div8

Paprzycki, M., Abraham, A., Guo, R., & Abraham, A. (2004). Analyzing Call Center Performance: A Data Mining Approach. *2.Softcomputing.Net*, *Section 3*. http://2.softcomputing.net/icfai-km.pdf%5Cnhttp://link.springer.com/chapter/10.1007/978-

3-540-24677-0_112

Parks, D. M. D. (2017). *Defining Data Science and Data Scientist* [University of South Florida]. https://search.proquest.com/openview/1072d7452723a8138cfed03666f54ee7/1.pdf?pq-origsite=gscholar&cbl=18750&diss=y

Parra, E., Dimou, C., Llorens, J., Moreno, V., & Fraga, A. (2015). A methodology for the classification of quality of requirements using machine learning techniques. *Information and Software Technology*, *67*, 180–195. https://doi.org/10.1016/J.INFSOF.2015.07.006

PCA For categorical features? - Stack Overflow. (n.d.). 2020. Retrieved September 29, 2020, from https://stackoverflow.com/questions/40795141/pca-for-categorical-features

Peddamuthu, B., & Srivastava, S. (2014). *QualityAssurance in Real-time (QART)*. Xerox Research Centre India. http://www.xrci.xerox.com/quality-assurance-in-real-time-qart

Pelayo, L., & Dick, S. (2007). Applying Novel Resampling Strategies To Software Defect Prediction. *NAFIPS 2007 - 2007 Annual Meeting of the North American Fuzzy Information Processing Society*, 69–72. https://doi.org/10.1109/NAFIPS.2007.383813

Petrov, D., Gutman, B. A., Shih-Hua, Yu, van Erp, T. G. M., Turner, J. A., Schmaal, L., Veltman, D., Wang, L., Alpert, K., Isaev, D., Zavaliangos-Petropulu, A., Ching, C. R. K., Calhoun, V., Glahn, D., Satterthwaite, T. D., Andreasen, O. A., Borgwardt, S., Howells, F., … Thompson, P. M. (2017). *Machine Learning for Large-Scale Quality Control of 3D Shape Models in Neuroimaging*. http://arxiv.org/abs/1707.06353

Phua, C., Lee, V., Smith-Miles, K. and Gayler, R. (2005). A Comprehensive Survey of Data Mining-based Fraud Detection Research. *School of Business Systems, Faculty of Information Technology, Monash University*.

Piatetsky, G. (2018). *Python eats away at R: Top Software for Analytics, Data Science, Machine*

Learning in 2018: Trends and Analysis. KDnuggets. https://www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html/2

Probabilistic: Definition, Models and Theory Explained. (n.d.). Search Statistics How To. Retrieved May 25, 2018, from http://www.statisticshowto.com/probabilistic/

Rácz, A., Bajusz, D., & Héberger, K. (2019). Multi-Level Comparison of Machine Learning Classifiers and Their Performance Metrics. *Molecules (Basel, Switzerland)*, *24*(15), 2811. https://doi.org/10.3390/molecules24152811

Ravishanker, R. (2018). *From Reports to Analysis: Big Data and Data Analytics in Small Institutions*. The EvoLLLution. https://evolllution.com/managing-institution/operations_efficiency/from-reports-to-analysis-big-data-and-data-analytics-in-small-institutions/

Reinsel, D., Gantz, J., & Rydning, J. (2017). *Total WW Data to Reach 163ZB by 2025*. StorageNewsletter. https://www.storagenewsletter.com/2017/04/05/total-ww-data-to-reach-163-zettabytes-by-2025-idc/

Ribeiro, B. (2005). Support Vector Machines for Quality Monitoring in a Plastic Injection Molding Process. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, *35*(3), 401–410. https://doi.org/10.1109/TSMCC.2004.843228

RTInsights Team. (2017). *Predicting Quality Assurance Outcomes for Process Manufacturing*. https://www.rtinsights.com/predicting-quality-assurance-outcomes-for-process-manufacturing/

SAS Institute Inc. (2020). *Adjusting for oversampling the event level in a binary logistic model*. https://support.sas.com/kb/22/601.html

Scheffe, H. (1953). A Method for Judging Contracts in the Analysis of Variance. *Biometrika*, *40*(1–2), 87–110. https://doi.org/10.1093/biomet/40.1-2.87

Shah, N., Chatterjee, A., Bhargava, A., & Joshi, N. (2016). *Model Validation, KS Test and Lorenz Curve*. Kharagpur Data Analytics Group-KDAG. https://kgpdag.wordpress.com/2016/03/14/model-validation-ks-test-and-lorenz-curve/

Shamsaei, B., & Gao, C. (2016). Comparison of some machine learning and statistical algorithms for classification and prediction of human cancer type. *3rd IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2016*, 296–299. https://doi.org/10.1109/BHI.2016.7455893

Shin, H., Park, H., Lee, J., & Jhee, W. C. (2012). A scoring model to detect abusive billing patterns in health insurance claims. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2012.01.105

Simon, P. (2013). *Too Big to Ignore: The Business Case for Big Data*. Wiley.

Singh, W., & Kaur, P. (2016). Comparative Analysis of Classification Techniques for Predicting Computer Engineering Students ' Academic Performance. *International Journal of Advanced Research in Computer Science RESEARCH*, *7*(6).

Siris, V. a., & Papagalou, F. (2004). Application of anomaly detection algorithms for detecting SYN flooding attacks. *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, *4*, 2050–2054. https://doi.org/10.1109/GLOCOM.2004.1378372

Stackexchange. (n.d.). *Can principal component analysis be applied to datasets containing a mix of continuous and categorical variables? - Cross Validated*. Retrieved September 29, 2020, from https://stats.stackexchange.com/questions/5774/can-principal-component-analysis-be-applied-to-datasets-containing-a-mix-of-cont

Sture, Ø. (2015). Automatic Quality Control of Salmon - Using Machine Learning Algorithms based on Input from a 3D Machine Vision System. *138*. https://brage.bibsys.no/xmlui/handle/11250/2352578#.WwSR2hm3SjU.mendeley

Substance Abuse and Mental Health Services Administration. (2015). *Analyses of MAX Claims: SAMHSA Fee-for-Service Spending Estimates, Medicare-Medicaid Enrollee Analysis, and Managed Care Summary*.

Sumesh, A., Rameshkumar, K., Mohandas, K., & Babu, R. S. (2015). Use of machine learning algorithms for weld quality monitoring using acoustic signature. *Procedia Computer Science*. https://doi.org/10.1016/j.procs.2015.04.042

Testimony of Ann Maxwell Before the United States House of Representatives. (2017). http://docs.house.gov/meetings/IF/IF02/20170131/105493/HHRG-115-IF02-Wstate-ArchambaultJ-20170131.pdf

The Centers for Medicare & Medicaid Services. (2015). *Payment Error Rate Measurement Manual* (Issue August). https://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicaid-and-CHIP-Compliance/PERM/Downloads/CMSPERMMAnual1.pdf

The Centers for Medicare and Medicaid Services. (2013a). *CMS Linkable 2008 – 2010 Medicare Data Entrepreneurs ' Synthetic Public Use File (DE-SynPUF): Codebook* (p. 125). https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/index.html

The Centers for Medicare and Medicaid Services. (2013b). *CMS Linkable 2008 – 2010 Medicare Data Entrepreneurs ' Synthetic Public Use File (DE-SynPUF): User Namual* (p. 34). https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-

Files/SynPUFs/index.html

The Centers for Medicare and Medicaid Services. (2014). *Payment Error Rate Measurement (PERM) Overview Background & Regulations* (Issue January).

The Centers for Medicare and Medicaid Services. (2015). *Medicaid and CHIP 2015 Improper Payments Report*. https://www.cms.gov/Research-Statistics-Data-and-Systems/Monitoring-Programs/Medicaid-and-CHIP-Compliance/Downloads/2015MedicaidandCHIPImproperPaymentsReport.pdf

The Centers for Medicare and Medicaid Services. (2017). *2017 Medicaid & CHIP Supplemental Improper Payment* (Issue November).

The Centers for Medicare and Medicaid Services. (2019a). *2019 Medicaid & CHIP Supplemental Improper Payment Data* (Issue November).

The Centers for Medicare and Medicaid Services. (2019b). *CMS 2008-2010 Data Entrepreneurs' Synthetic Public Use File (DE-SynPUF) | CMS*. https://www.cms.gov/Research-Statistics-Data-and-Systems/Downloadable-Public-Use-Files/SynPUFs/DE_Syn_PUF

The Centers for Medicare and Medicaid Services. (2020). *Medicaid Facts and Figures | CMS*. https://www.cms.gov/newsroom/fact-sheets/medicaid-facts-and-figures

The Henry J. Kaiser Family Foundation. (2018). *Federal and State Share of Medicaid Spending*. https://www.kff.org/medicaid/state-indicator/federalstate-share-of-spending/?dataView=1&currentTimeframe=0&sortModel=%7B%22colId%22:%22Location%22,%22sort%22:%22asc%22%7D

The Office for Civil Rights, & Malin, B. (2012). Guidance Regarding Methods for de-identification of protected health information in accordance with the Health Insurance

Portability and Accountability Act (HIPAA) Privacy Rule. *Health Information Privacy*, 1–32.

Tozzi, C. (2017). *How Big is Big Data? A Look at the Definition of Big Data &amp; Examples of It in Action*. http://blog.syncsort.com/2017/07/big-data/how-big-is-big-data-definition-examples/

Tracy, R. B. (2018). *How Machine Learning Can Transform Quality Management*. EtQ Blog. https://blog.etq.com/how-machine-learning-can-transform-quality-management

Travaille, P., Thornton, D., Müller, R. M., & Hillegersberg, J. van. (2011). Electronic Fraud Detection in the U . S . Medicaid Healthcare Program : Lessons Learned from other Industries. *Seventeenth Americas Conference on Information Systems*, 1–10. http://doc.utwente.nl/78000/1/Travaille11electronic.pdf

Tsung, F., Zhou, Z., & Jiang, W. (2007). Applying manufacturing batch techniques to fraud detection with incomplete customer information. *IIE Transactions*, *39*(6), 671–680. https://doi.org/10.1080/07408170600897510

Ucar, F., Alcin, O., Dandil, B., & Ata, F. (2018). Power Quality Event Detection Using a Fast Extreme Learning Machine. *Energies*. https://doi.org/10.3390/en11010145

Upton, G., & Cook, I. (2014). Anderson–Darling test. In *A Dictionary of Statistics* (3rd ed.). Oxford University Press. https://doi.org/10.1093/acref/9780199679188.001.0001

USAID. (2016). *Fiscal Year 2016 Commodity Calculator*.

Valdes, A., & Skinner, K. (2000). Adaptive, model-based monitoring for cyber attack detection. *Recent Advances in Intrusion Detection*, 80–92. https://doi.org/10.1007/3-540-39945-3_6

van Capelleveen, G., Poel, M., Mueller, R. M., Thornton, D., & van Hillegersberg, J. (2016). Outlier detection in healthcare fraud: A case study in the Medicaid dental domain.

International Journal of Accounting Information Systems. https://doi.org/10.1016/j.accinf.2016.04.001

Vihinen, M. (2012). How to evaluate performance of prediction methods? Measures and their interpretation in variation effect analysis. *BMC Genomics*, *13*(Suppl 4), S2. https://doi.org/10.1186/1471-2164-13-S4-S2

Vijaya Beeravalli. (2018, September 30). *Comparison of Machine Learning Classification Models for Credit Card Default Data | by Vijaya Beeravalli | Medium*. Medium.Com. https://medium.com/@vijaya.beeravalli/comparison-of-machine-learning-classification-models-for-credit-card-default-data-c3cf805c9a5a

What is Data Mining. (n.d.). IGI Global. Retrieved May 29, 2018, from https://www.igi-global.com/dictionary/data-mining/6763

What is "Medicaid." (n.d.). Retrieved January 25, 2018, from https://www.investopedia.com/terms/m/medicaid.asp

Wikipedia. (n.d.). *Unstructured data*. Retrieved June 11, 2018, from https://en.wikipedia.org/wiki/Unstructured_data

Williams, S., & Baugh, D. (2016). *Medicaid Analytic eXtract Data (MAX) Providing Data to Researchers and Policy Analysts*. https://www.cms.gov/Research-Statistics-Data-and-Systems/Computer-Data-and-Systems/MedicaidDataSourcesGenInfo/Downloads/MAXintrotoData.zip

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical Machine Learning Tools and Techniques* (4th ed.). Morgan Kaufmann. https://books.google.com/books?id=1SylCgAAQBAJ

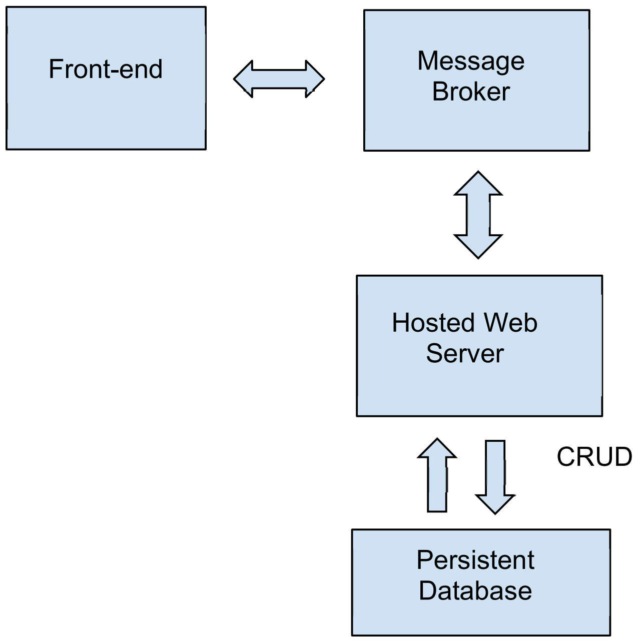
Yahya. (2018). *F1 Score vs ROC AUC*. Stack Overflow.

https://stackoverflow.com/questions/44172162/f1-score-vs-roc-auc

Yussupova, N., Kovács, G., Boyko, M., & Bogdanova, D. (2016). Models and methods for quality management based on artificial intelligence applications. *Acta Polytechnica Hungarica*.

APPENDIX A: DATA PIPELINE ARCHITECTURE

The dataset used for this research is extracted from CMS's DE-SynPUF database. This database contains five massive datasets and over 120 flat files in separate files. The researcher needed an integrated dataset for this research, so a data pipeline was developed (with support by a data engineer) to ingest and merged 120 segregated flat files. This pipeline is a platform-agnostic solution using a web server and a firebase database that consists of four components: webserver in Express.js, UI/UX design using React, Postgres for persistence database, and Firebase real-time database as the message broker. After ingesting and merging all the data, the was moved database to the cloud. The system is built using a web server with a firebase database client listening to change on the firebase real-time database. When a message is written on the firebase database, an action gets triggered on the server and operates. The system comprises four components - web server built using Express.js framework, front-end built using react, Postgres for persistence database, and Firebase real-time database as the message broker.



The system is designed to have a single administrator with a global scope of control over the message broker. CRUD operations on the database can only be performed by the administrator in which sockets on the webserver listens to the UID of the registered admin account on Firebase.

Message Broker

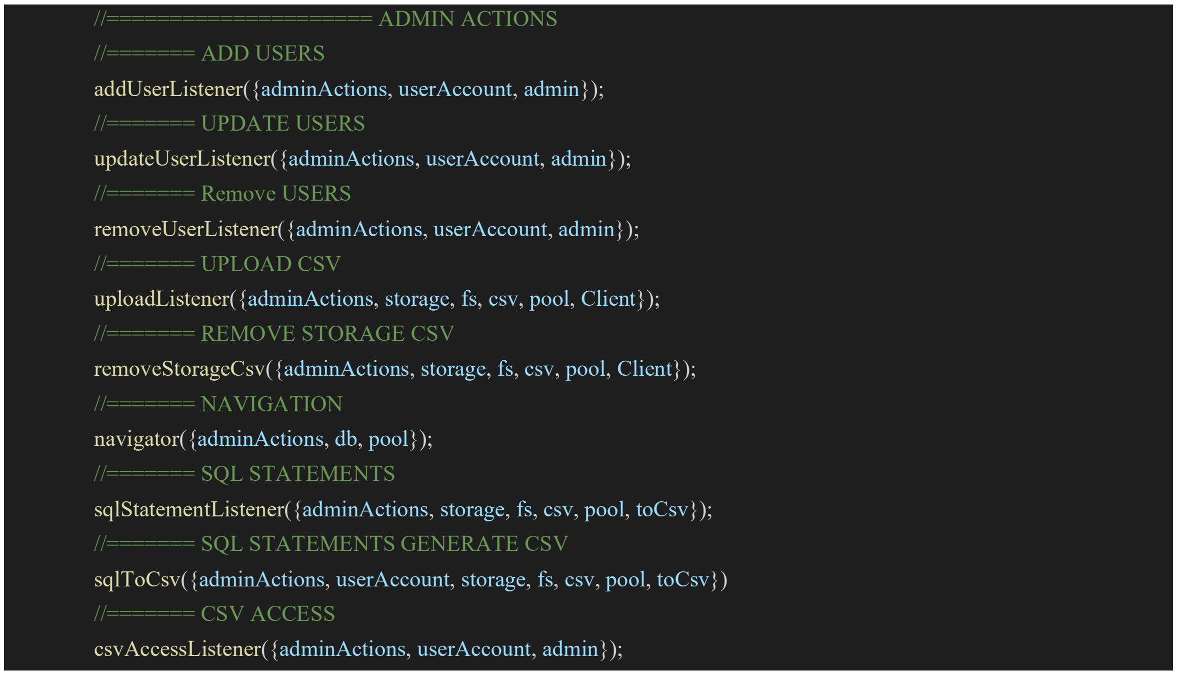
The system mainly listens to the message written by the admin on its UID. Actions under the admin UID are as follows:

- **columnsAvailable:** Contains available columns of all the tables in the database.
- **createCsv:** Stores all the available CSV files created by the admin with download links.
- **csvToTable:** Stores a record of CSV files being converted to database elements.
- **navigation:** Contains the record of 100 buffered data to be displayed as well as page control.
- **savedSqlStatements:** Contains the saved SQL statements created by the admin account.
- **userList:** Stores the record of users with their corresponding UID and action nodes created by the admin account.

Each of the following action nodes is being listened to on the webserver with corresponding processes upon detecting change on the action nodes.

Web Server

The web server is built using Express.js framework utilizing firebase database client to listen for changes. This setup serves as the real-time message detection to perform the action required by the admin. The server is decoupled from each of the components allowing flexibility on demand. If the server is not running, the process will be frozen as is, and the system will retain the last 100 rows of data retrieved previously. The process will resume running the server, and all the changes that have been made in the absence of the server will proceed. The server contains the following action listeners waiting for a message from the broker:

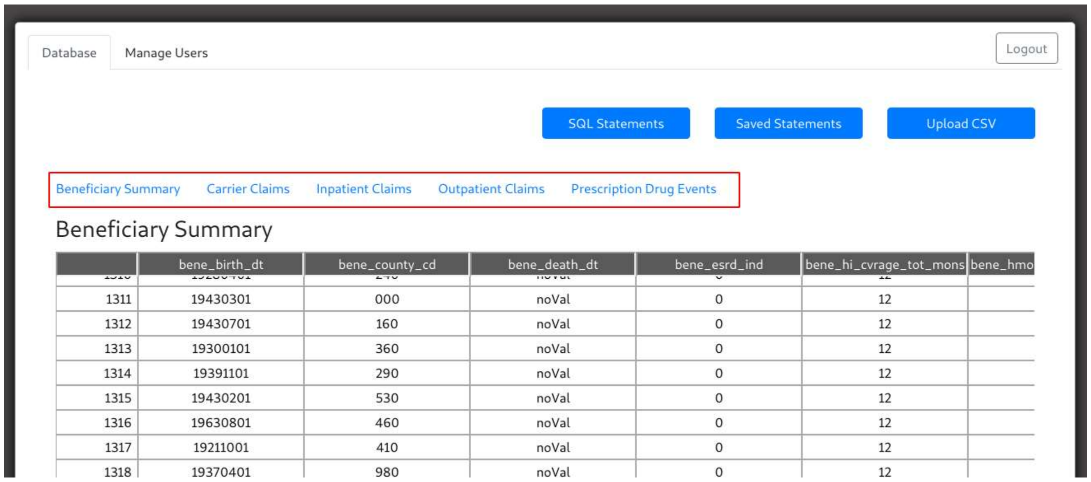


```
//===================== ADMIN ACTIONS
//======= ADD USERS
addUserListener({adminActions, userAccount, admin});
//======= UPDATE USERS
updateUserListener({adminActions, userAccount, admin});
//======= Remove USERS
removeUserListener({adminActions, userAccount, admin});
//======= UPLOAD CSV
uploadListener({adminActions, storage, fs, csv, pool, Client});
//======= REMOVE STORAGE CSV
removeStorageCsv({adminActions, storage, fs, csv, pool, Client});
//======= NAVIGATION
navigator({adminActions, db, pool});
//======= SQL STATEMENTS
sqlStatementListener({adminActions, storage, fs, csv, pool, toCsv});
//======= SQL STATEMENTS GENERATE CSV
sqlToCsv({adminActions, userAccount, storage, fs, csv, pool, toCsv})
//======= CSV ACCESS
csvAccessListener({adminActions, userAccount, admin});
```

Features and Usages

Navigation:

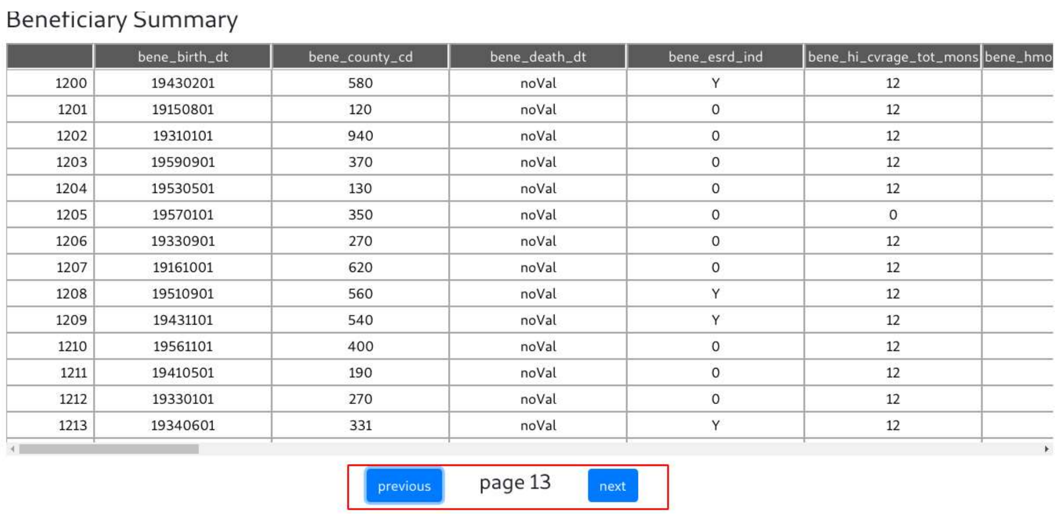
All tables created within the system will appear on the “Database” tab, allowing users to switch tables by selecting the table's name.



Beneficiary Summary

	bene_birth_dt	bene_county_cd	bene_death_dt	bene_esrd_ind	bene_hi_cvrage_tot_mons	bene_hmo
1311	19430301	000	noVal	0	12	
1312	19430701	160	noVal	0	12	
1313	19300101	360	noVal	0	12	
1314	19391101	290	noVal	0	12	
1315	19430201	530	noVal	0	12	
1316	19630801	460	noVal	0	12	
1317	19211001	410	noVal	0	12	
1318	19370401	980	noVal	0	12	

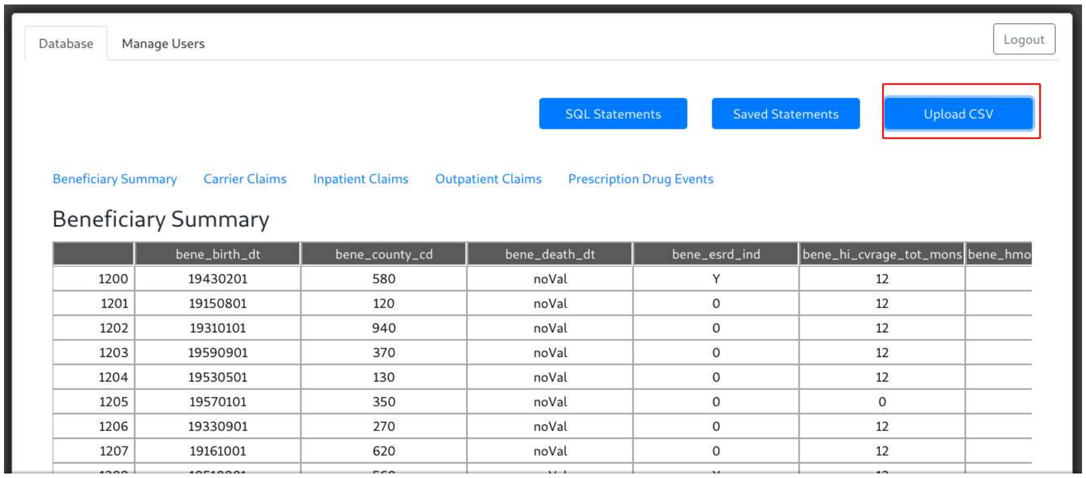
Page navigation is located at the bottom of each table and allows users to browse through the database. Each page is paginated to 100 rows of data.



Beneficiary Summary

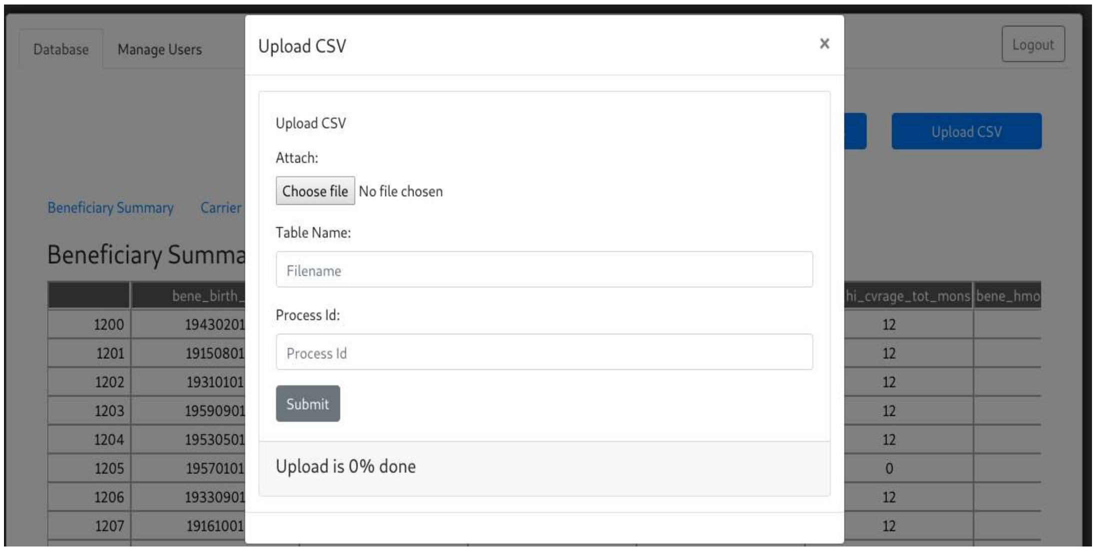
	bene_birth_dt	bene_county_cd	bene_death_dt	bene_esrd_ind	bene_hi_cvrage_tot_mons	bene_hmo
1200	19430201	580	noVal	Y	12	
1201	19150801	120	noVal	0	12	
1202	19310101	940	noVal	0	12	
1203	19590901	370	noVal	0	12	
1204	19530501	130	noVal	0	12	
1205	19570101	350	noVal	0	0	
1206	19330901	270	noVal	0	12	
1207	19161001	620	noVal	0	12	
1208	19510901	560	noVal	Y	12	
1209	19431101	540	noVal	Y	12	
1210	19561101	400	noVal	0	12	
1211	19410501	190	noVal	0	12	
1212	19330101	270	noVal	0	12	
1213	19340601	331	noVal	Y	12	

Uploading CSV:



Beneficiary Summary

	bene_birth_dt	bene_county_cd	bene_death_dt	bene_esrd_ind	bene_hi_cvrage_tot_mons	bene_hmo
1200	19430201	580	noVal	Y	12	
1201	19150801	120	noVal	0	12	
1202	19310101	940	noVal	0	12	
1203	19590901	370	noVal	0	12	
1204	19530501	130	noVal	0	12	
1205	19570101	350	noVal	0	0	
1206	19330901	270	noVal	0	12	
1207	19161001	620	noVal	0	12	



Upload CSV

Upload CSV

Attach:

No file chosen

Table Name:

Filename

Process Id:

Process Id

Upload is 0% done

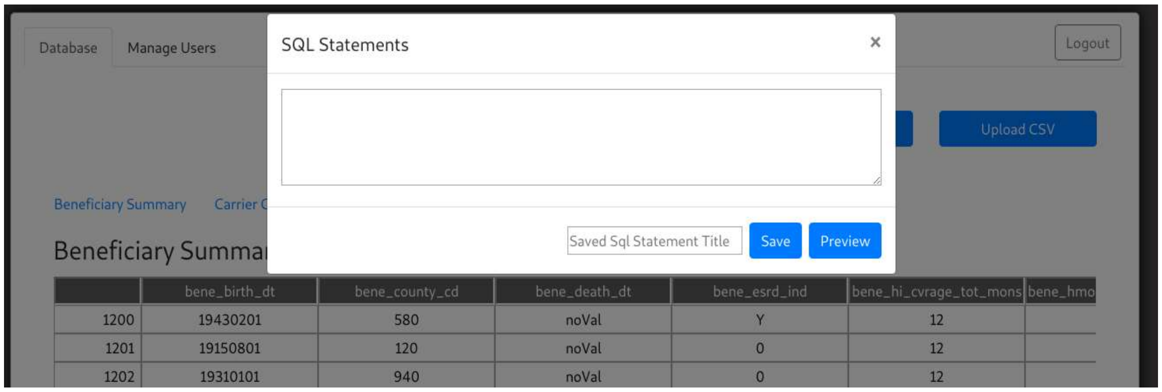
Beneficiary Summa

	bene_birth_	hi_cvrage_tot_mons	bene_hmo
1200	19430201	12	
1201	19150801	12	
1202	19310101	12	
1203	19590901	12	
1204	19530501	12	
1205	19570101	0	
1206	19330901	12	
1207	19161001	12	

Uploading CSV to create a table in the database requires two fields: the “Table Name” and the “ProcessId.”

NOTE: It is imperative to add a **unique** “ProcessId” for each upload to avoid file collision when creating a new table to insert existing ones.

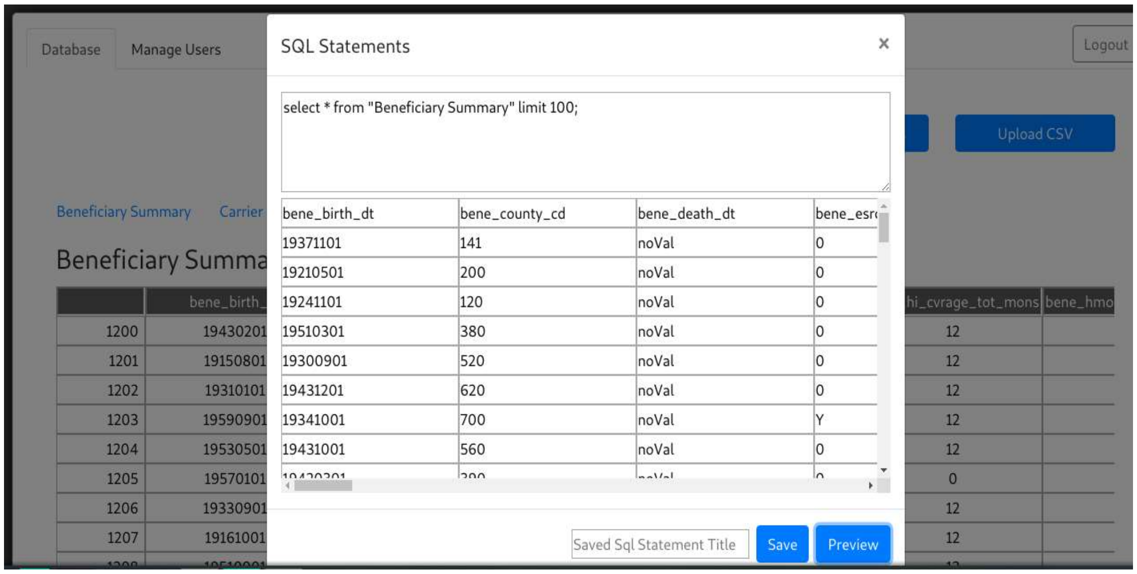
SQL Statements:



Beneficiary Summar

	bene_birth_dt	bene_county_cd	bene_death_dt	bene_esrd_ind	bene_hi_cvrage_tot_mons	bene_hmo
1200	19430201	580	noVal	Y	12	
1201	19150801	120	noVal	0	12	
1202	19310101	940	noVal	0	12	

This feature allows the user to test and qualify their SQL statements. Users can write their statements and test to see the results by clicking “preview.”



```
select * from "Beneficiary Summary" limit 100;
```

bene_birth_dt	bene_county_cd	bene_death_dt	bene_esr
19371101	141	noVal	0
19210501	200	noVal	0
19241101	120	noVal	0
19510301	380	noVal	0
19300901	520	noVal	0
19431201	620	noVal	0
19341001	700	noVal	Y
19431001	560	noVal	0

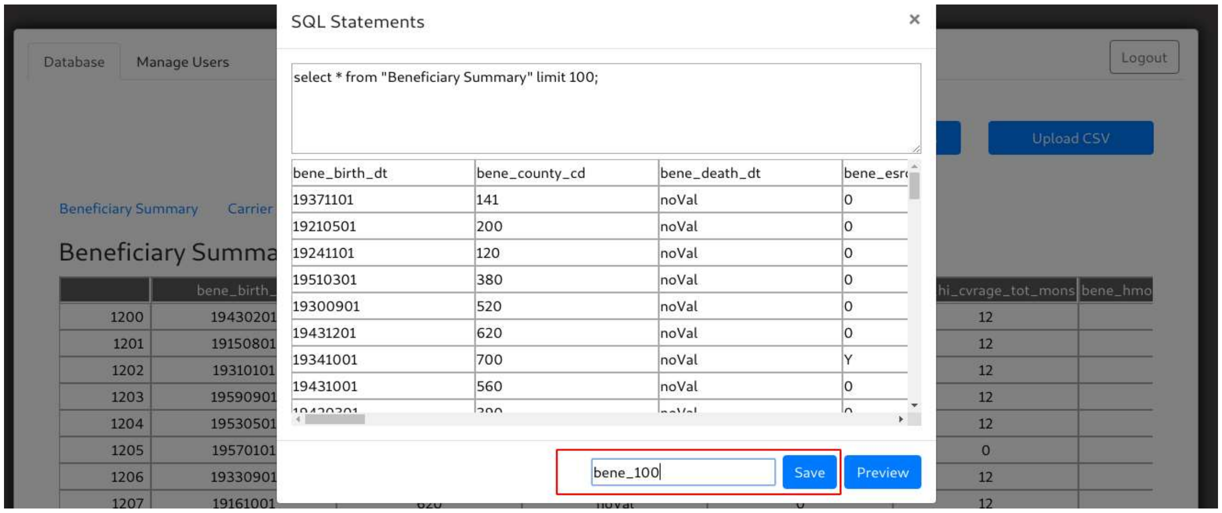
Beneficiary Summa

	bene_birth_	hi_cvrage_tot_mons	bene_hmo
1200	19430201	12	
1201	19150801	12	
1202	19310101	12	
1203	19590901	12	
1204	19530501	12	
1205	19570101	0	
1206	19330901	12	
1207	19161001	12	

NOTE: This feature is not intended to be used for huge queries, so it is essential to limit each request. Statements for huge processing size do not need to undergo the “Preview” function and may simply be saved for CSV generation.

CSV Generation:

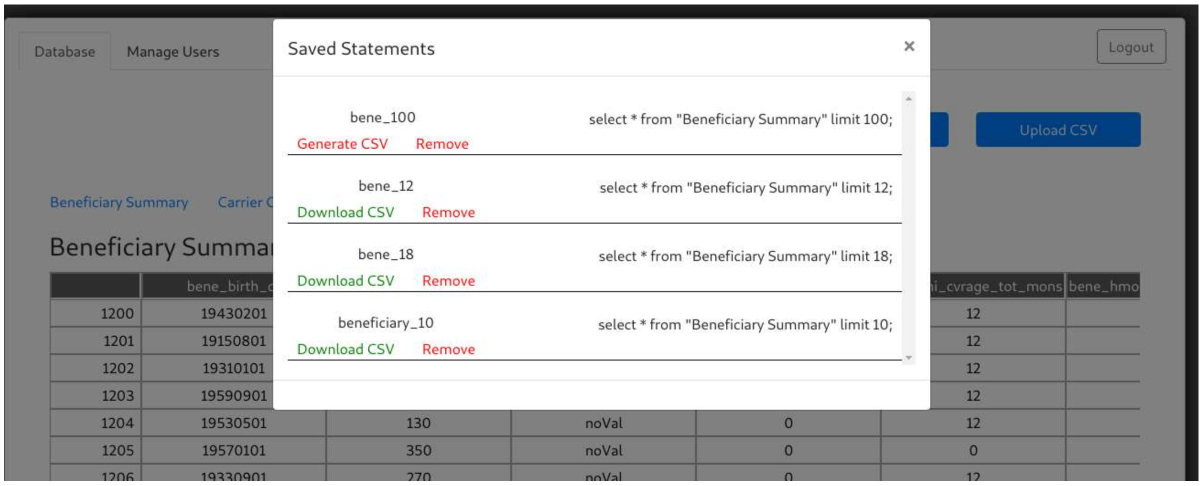
After saving the statement, a record should be saved on the "Saved SQL Statements."



```
select * from "Beneficiary Summary" limit 100;
```

bene_birth_dt	bene_county_cd	bene_death_dt	bene_esr
19371101	141	noVal	0
19210501	200	noVal	0
19241101	120	noVal	0
19510301	380	noVal	0
19300901	520	noVal	0
19431201	620	noVal	0
19341001	700	noVal	Y
19431001	560	noVal	0

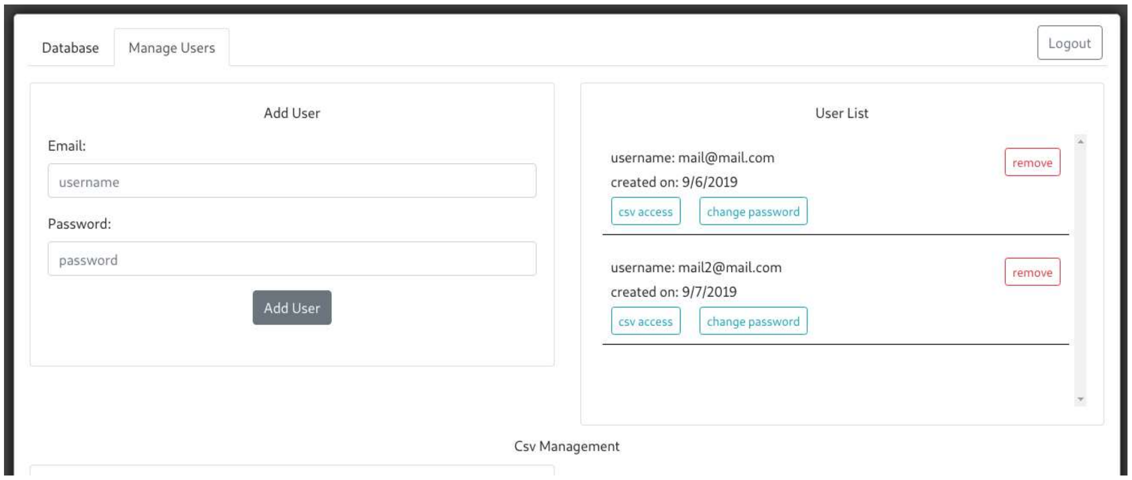
bene_100



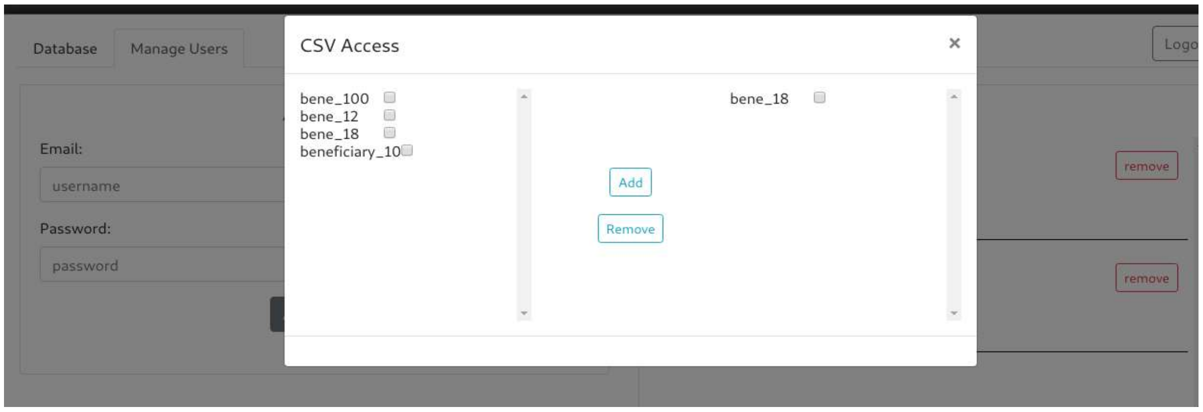
bene_100 (Generate CSV, Remove)	select * from "Beneficiary Summary" limit 100;
bene_12 (Download CSV, Remove)	select * from "Beneficiary Summary" limit 12;
bene_18 (Download CSV, Remove)	select * from "Beneficiary Summary" limit 18;
beneficiary_10 (Download CSV, Remove)	select * from "Beneficiary Summary" limit 10;

Clicking "Generate CSV" will process creating a CSV file that will be available for download.

Create Account and Adding CSV Access:

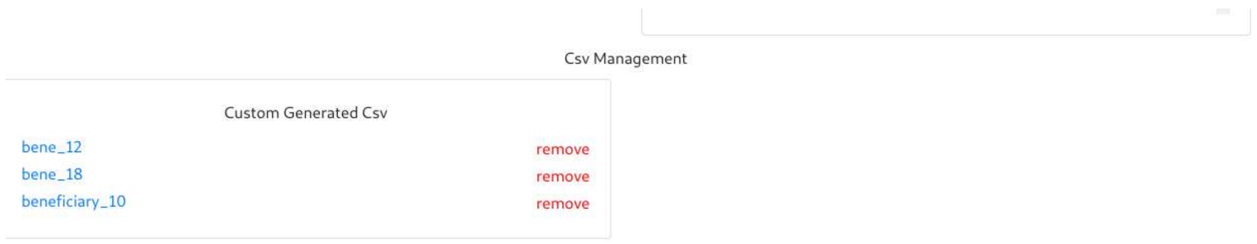


Accounts can be created by merely providing a username and password. Adding CSV access for a user can be done by clicking "CSV Access" and choosing the checkbox on the left to be added.



In revoking access, select the CSV on the left and click remove. The process is near real-time, so each action performed on the admin side can expect almost immediate change to the user accounts.

Clearing the CSV off the system:

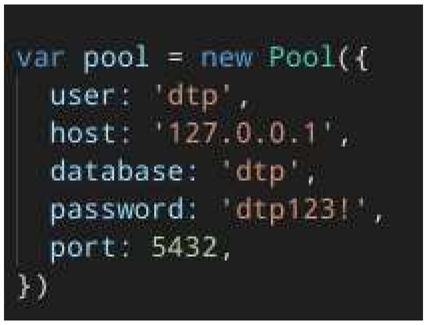


Removing the CSV file will remove the saved SQL statement, while removing the statement will still retain the created CSV file.

System Build and Operations

The system is built to run under Node Version 10 or higher and assumes to have a Postgres SQL installed on the host machine. To start the system, a few requirements will have to be met. The system will require the SQL credentials to make a connection, the Firebase service account to access a real-time database, and the admin user UID to start to create a message map and start listening for a request.

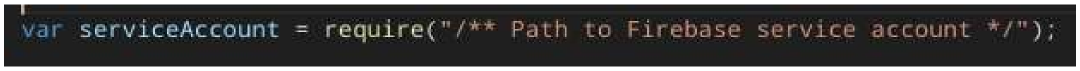
Provide the credentials to the database:



```
var pool = new Pool({
  user: 'dtp',
  host: '127.0.0.1',
  database: 'dtp',
  password: 'dtp123!',
  port: 5432,
})
```

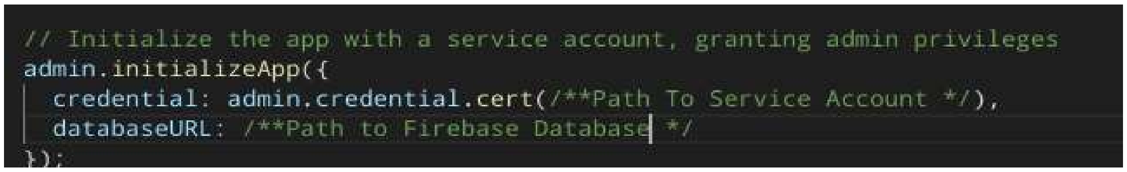
Provide the Firebase service account:

Import firebase service account



```
var serviceAccount = require("/** Path to Firebase service account */");
```

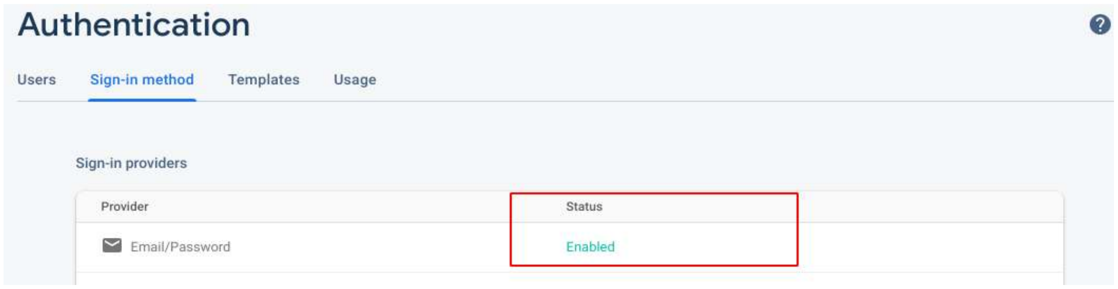
Add the service account



```
// Initialize the app with a service account, granting admin privileges
admin.initializeApp({
  credential: admin.credential.cert(/**Path To Service Account */),
  databaseURL: /**Path to Firebase Database */
});
```

Provide the firebase UID of the admin:

Enable the email sign-in method



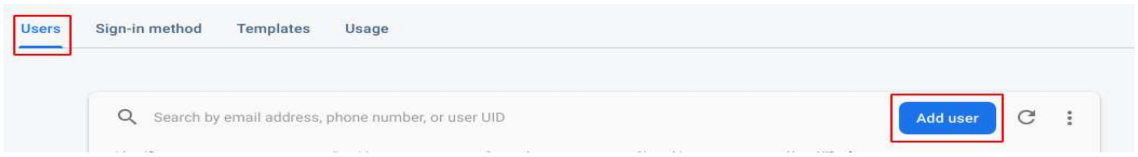
Authentication

Users | Sign-in method | Templates | Usage

Sign-in providers

Provider	Status
Email/Password	Enabled

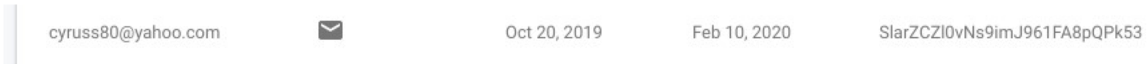
Register an admin account



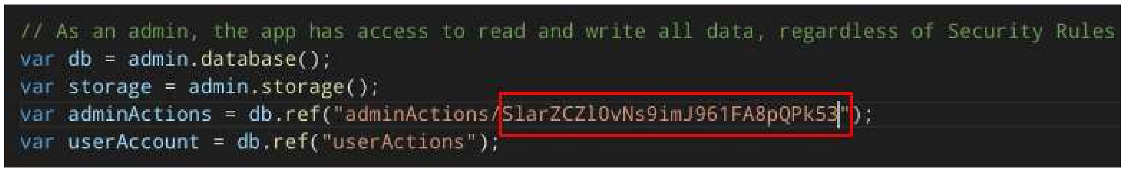
Users | Sign-in method | Templates | Usage

Search by email address, phone number, or user UID

Add user

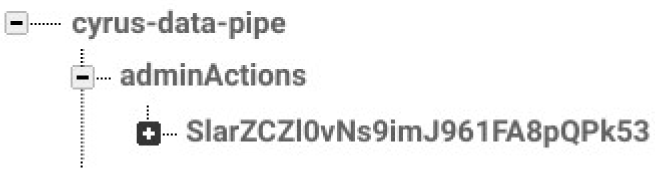


cyruss80@yahoo.com | Oct 20, 2019 | Feb 10, 2020 | SlarZCZl0vNs9imJ961FA8pQPk53

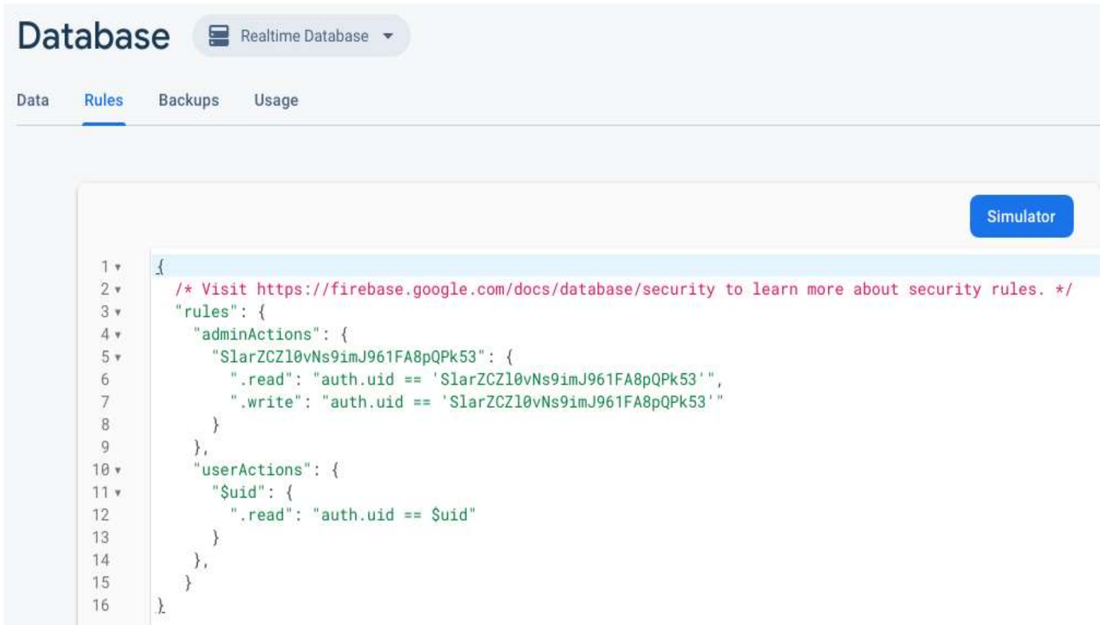


```
// As an admin, the app has access to read and write all data, regardless of Security Rules
var db = admin.database();
var storage = admin.storage();
var adminActions = db.ref("adminActions/SlarZCZl0vNs9imJ961FA8pQPk53");
var userAccount = db.ref("userActions");
```

Seed the initial database with the following pattern

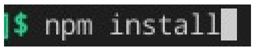


Set the database rules



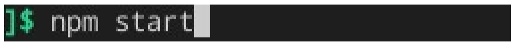
```
{
  /* Visit https://firebase.google.com/docs/database/security to learn more about security rules. */
  "rules": {
    "adminActions": {
      "SlarZCZl0vNs9imJ961FA8pQPk53": {
        ".read": "auth.uid == 'SlarZCZl0vNs9imJ961FA8pQPk53'",
        ".write": "auth.uid == 'SlarZCZl0vNs9imJ961FA8pQPk53'"
      }
    },
    "userActions": {
      "$uid": {
        ".read": "auth.uid == $uid"
      }
    },
  }
}
```

Install the dependencies:



```
]$ npm install
```

Execute the server:

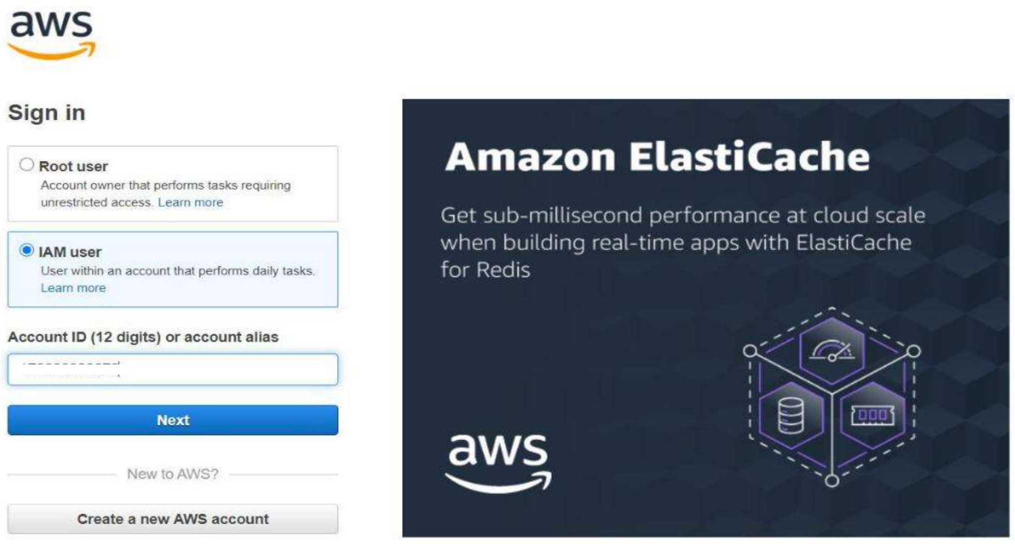


```
]$ npm start
```

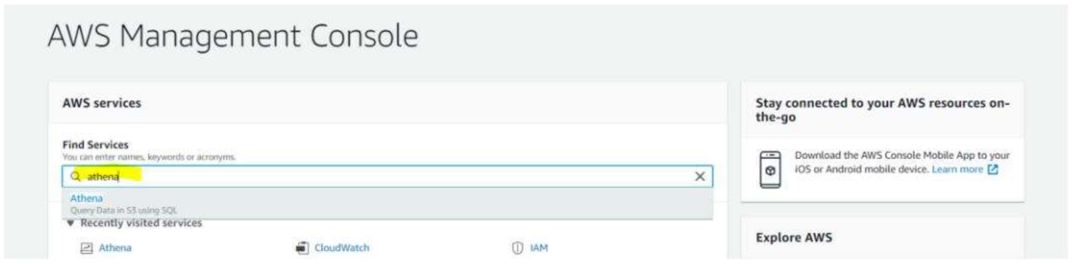
APPENDIX B: HOW TO CONNECT TO THE AWS ATHENA AND ACCESS THE DATASETS USED FOR THIS RESEARCH

Step 1: use the link provided below to get connected to the AWS Athena services. You need the 12-digit Service Account number (provided below) and the username and password you received via email to connect to this service and the datasets used in this dissertation in the cloud.

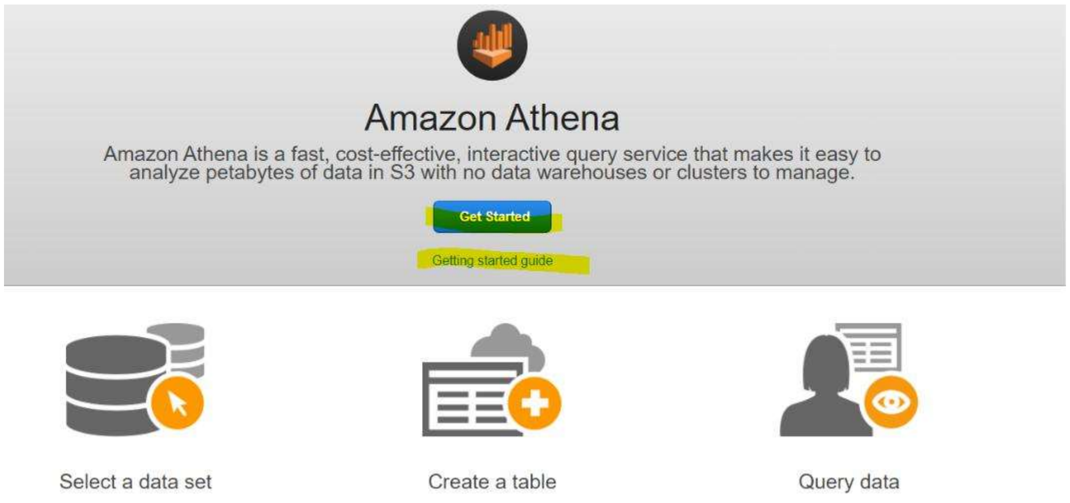
URL for Amazon Athena: https://signin.aws.amazon.com
Login Using IAM account Service Account: Use the account number provided by the researcher


aws
Sign in
Root user
Account owner that performs tasks requiring unrestricted access. Learn more
IAM user
User within an account that performs daily tasks. Learn more
Account ID (12 digits) or account alias
Next
New to AWS?
Create a new AWS account
Amazon ElastiCache
Get sub-millisecond performance at cloud scale when building real-time apps with ElastiCache for Redis
aws

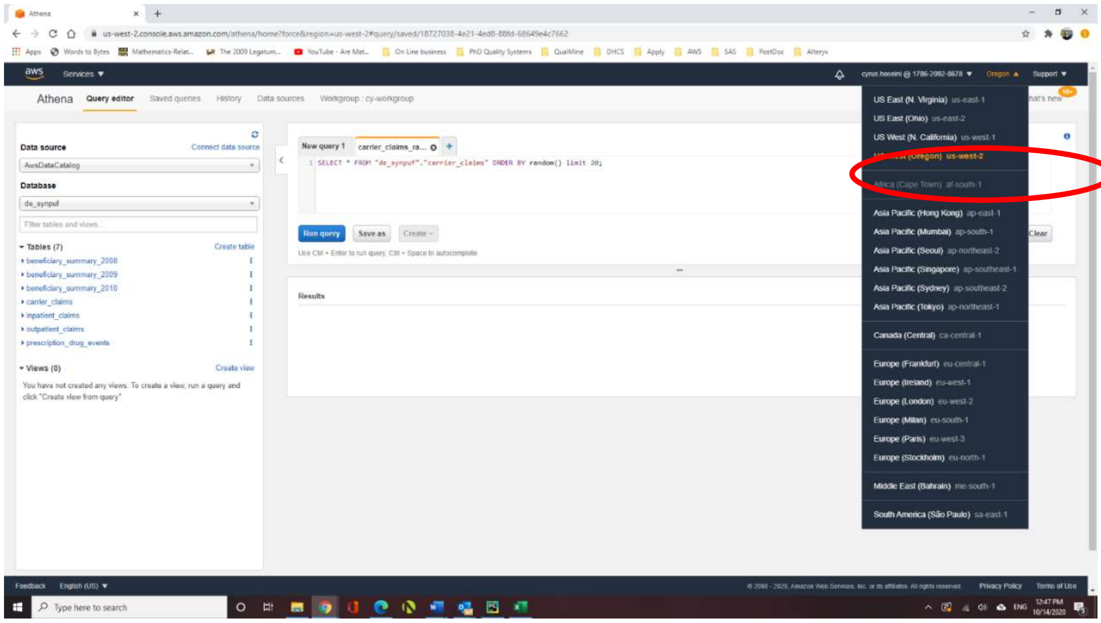
Step 2: search for Athena service, as shown below, and click on "Get Started" with Amazon Athena.


AWS Management Console
AWS services
Find Services
You can enter names, keywords or acronyms.
athena
Athena
Query Data in S3 using SQL
Recently visited services
Athena
CloudWatch
IAM
Stay connected to your AWS resources on-the-go
Download the AWS Console Mobile App to your iOS or Android mobile device. Learn more
Explore AWS

** Below the **Get Started** button is the link to the documentation of the service


Amazon Athena
Amazon Athena is a fast, cost-effective, interactive query service that makes it easy to analyze petabytes of data in S3 with no data warehouses or clusters to manage.
Get Started
Getting started guide
Select a data set
Create a table
Query data

Step 3: On the top right side of the page, select "US West (Oregon)us-west-2" as the server hub as shown in the below picture:

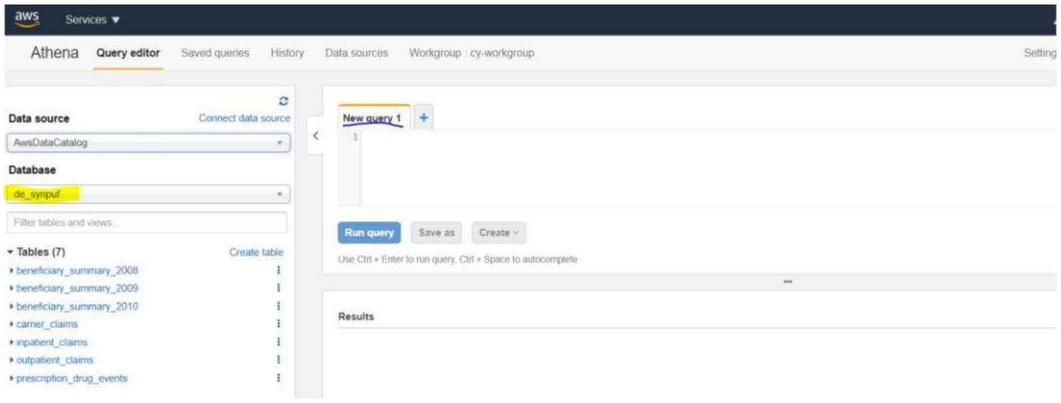

Athena
Query editor
Saved queries
History
Data sources
Workgroup : cy-workgroup
Data source
Connect data source
AwsDataCatalog
Database
de_synpuf
Tables (7)
Create table
beneficiary_summary_2008
beneficiary_summary_2009
beneficiary_summary_2010
carrier_claims
inpatient_claims
outpatient_claims
prescription_drug_events
Views (0)
Create view
New query 1
carrier_claims_ra...
Run query
Save as
Create
Results
Oregon
Support
US East (N. Virginia) us-east-1
US East (Ohio) us-east-2
US West (N. California) us-west-1
US West (Oregon) us-west-2
Africa (Cape Town) af-south-1
Asia Pacific (Hong Kong) ap-east-1
Asia Pacific (Mumbai) ap-south-1
Asia Pacific (Seoul) ap-northeast-2
Asia Pacific (Singapore) ap-southeast-1
Asia Pacific (Sydney) ap-southeast-2
Asia Pacific (Tokyo) ap-northeast-1
Canada (Central) ca-central-1
Europe (Frankfurt) eu-central-1
Europe (Ireland) eu-west-1
Europe (London) eu-west-2
Europe (Milan) eu-south-1
Europe (Paris) eu-west-3
Europe (Stockholm) eu-north-1
Middle East (Bahrain) me-south-1
South America (São Paulo) sa-east-1

```
SELECT * FROM "de_synpuf"."carrier_claims" ORDER BY random() limit 20;
```

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

You have not created any views. To create a view, run a query and click "Create view from query"

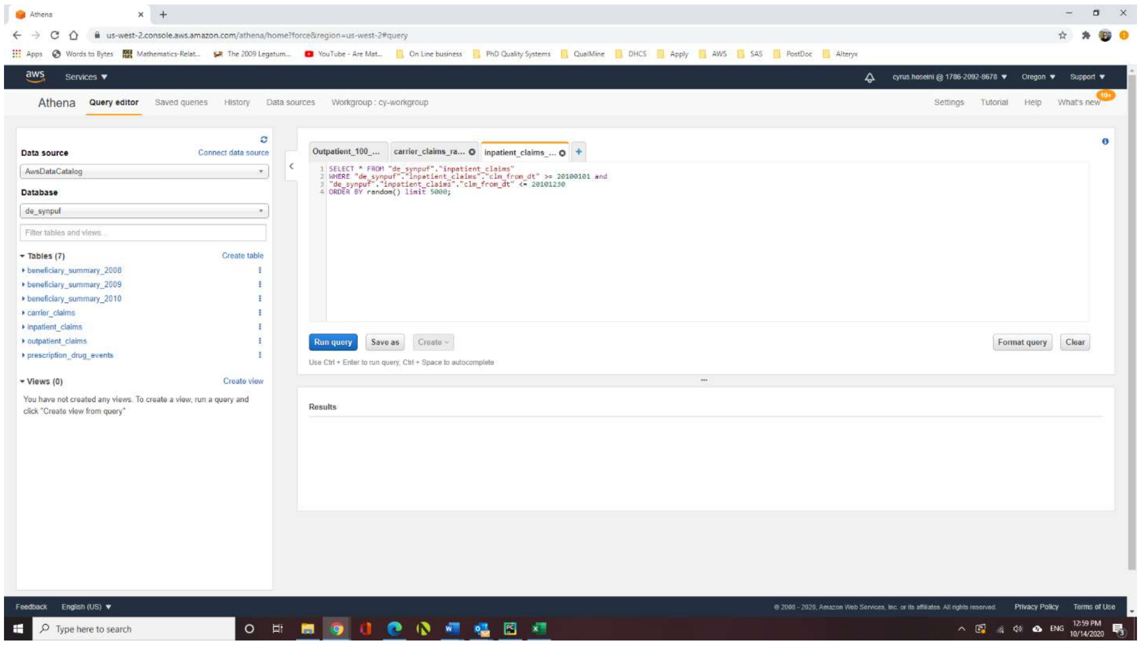
Step 4: If the database is not selected, switch it to de_synpuf (highlighted in yellow)



Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

Step 5: design and run your SQL queries in the query box. Please note Amazon Athena uses Presto with full standard SQL support and works with various standard data formats, including CSV, JSON, etc. However, some of the functions may be slightly different than standard SQL. This is a sample query to select 5,000 random inpatient claims from the 2010 table:

```
SELECT * FROM "de_synpuf"."inpatient_claims"
WHERE "de_synpuf"."inpatient_claims"."clm_from_dt" >= 20100101 and
"de_synpuf"."inpatient_claims"."clm_from_dt" <= 20101230
ORDER BY random() limit 5000;
```



```
SELECT * FROM "de_synpuf"."inpatient_claims"
WHERE "de_synpuf"."inpatient_claims"."clm_from_dt" >= 20100101 and
"de_synpuf"."inpatient_claims"."clm_from_dt" <= 20101230
ORDER BY random() limit 5000;
```

You have not created any views. To create a view, run a query and click "Create view from query"

Use Ctrl + Enter to run query, Ctrl + Space to autocomplete

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

You can also retrieve saved queries by going through the "Saved Queries" tab, as shown below:

APPENDIX C: DETAIL OF CA-MMIS WEEKLY PAYMENT DATA REVIEW FINDINGS

This appendix provides the detail of post-adjudication and pre-payment findings reported in CA-MMIS weekly payment data reports between 9/9/2016 and 5/1/2020. This dataset is used to extract the pattern of erroneous payments. It is important to note all these reported findings were caught and fixed before the providers' final payment, so they are not technically erroneous payments. However, they carry the same characteristics of typical erroneous payments in a Medicaid FFS. Reviewed samples in these reports are targeted samples because the process targets high dollar value claims. Therefore, these results are not used to provide an interval estimate for the overall error rate in a typical Medicaid FFS. As explained in detail in chapter 3, PERM audit reports were used for this purpose.

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
9/9/16	253	3	$59,483.32	$59,483.32		$0.00	Erroneously Paid
9/9/16	253	4	$7,947.72	$6,914.16		$1,033.56	768.97%
9/9/16	253	4	$7,947.72	$6,914.16		$1,033.56	768.97%
9/9/16	253	4	$7,947.72	$6,914.16		$1,033.56	768.97%
9/9/16	253	4	$3,853.44	$3,352.32		$501.12	768.97%
9/9/16	253	4	$6,829.92	$1,727.57		$5,102.35	133.86%
9/16/16	?	3	$177,427.97	$163,376.16		$14,051.81	1262.67%
9/16/16	?	3	$26,649.86	$5,204.56		$21,445.30	124.27%
9/16/16	?	5	$5,180.40	$5,180.40		$0.00	Erroneously Paid
9/16/16	?	5	$2,609.28	$194.75		$2,414.53	108.07%
9/23/18	325	4	$2,234.23		$39.31	$2,273.54	98.27%
9/23/18	325	4	$2,043.60	$25.95		$2,017.65	101.29%
9/23/18	325	4	$43,211.79	$35,863.33		$7,348.46	588.04%
9/30/16	287	4	$46,978.56	$27,404.56		$19,574.00	240.00%
9/30/16	287	4	$43,485.75	$38,549.44		$4,936.31	880.94%
9/30/16	287	4	$75,243.00		$1,024.80	$76,267.80	98.66%
9/30/16	287	5	$3,822.58	$42.62		$3,779.96	101.13%
10/7/18	215	4	$2,256.47	$2,256.47		$0.00	Erroneously Paid
10/7/18	215	4	$8,984.97	$6,532.51		$2,452.46	366.37%
10/14/16	287	3	$37,944.12	$29,555.08		$8,389.04	452.31%
10/21/16	306	3	$32,184.23		$466.10	$32,650.33	98.57%
10/21/16	306	4	$24,355.10		$989.00	$25,344.10	96.10%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
10/21/16	306	4	$25,882.00	$20.00		$25,862.00	100.08%
10/21/16	306	4	$21,482.30	$20.00		$21,462.30	100.09%
10/21/16	306	4	$25,279.75		$0.01	$25,279.76	100.00%
10/21/16	306	4	$30,507.12	$28,328.04		$2,179.08	1400.00%
10/21/16	306	5	$3,677.40	$3,542.40		$135.00	2724.00%
10/28/16	250	3	$31,434.78	$6,676.60		$24,758.18	126.97%
10/28/16	250	3	$35,660.36		$30.00	$35,690.36	99.92%
10/28/16	250	5	$3,455.28		$0.65	$3,455.93	99.98%
10/28/16	250	4	$57,081.40	$20.00		$57,061.40	100.04%
10/28/16	250	4	$45,780.68	$43,601.60		$2,179.08	2100.92%
11/4/16	477	4	$13,702.08	$160.44		$13,541.64	101.18%
11/4/16	477	4	$7,829.76	$91.68		$7,738.08	101.18%
11/4/16	477	4	$5,872.32	$68.76		$5,803.56	101.18%
11/4/16	477	4	$4,514.79	$1.99		$4,512.80	100.04%
11/11/16	396	4	$88,084.80		$29,361.60	$117,446.40	75.00%
11/11/16	396	4	$88,084.80		$29,361.60	$117,446.40	75.00%
11/11/16	396	4	$88,084.80		$29,361.60	$117,446.40	75.00%
11/18/16	426	1	$2,385.63	$0.75		$2,384.88	100.03%
11/18/16	426	3	$43,638.00	$1,530.00		$42,108.00	103.63%
11/18/16	426	4	$44,520.00	$34,966.20		$9,553.80	465.99%
11/18/16	426	4	$11,361.60		$2,210.30	$13,571.90	83.71%
11/18/16	426	4	$2,442.66	$488.53		$1,954.13	125.00%
11/18/16	426	4	$4,538.78		$47.41	$4,586.19	98.97%
11/23/16	424	4	$30,952.32		$366.72	$31,319.04	98.83%
11/23/16	424	4	$30,952.32		$366.72	$31,319.04	98.83%
11/23/16	424	5	$2,876.40	$2,876.40		$0.00	Erroneously Paid
11/23/16	424	4	$5,372.00		$268.60	$5,640.60	95.24%
12/2/16	322	5	$2,755.44	$194.75	$269.60	$2,830.29	97.36%
12/9/16	424	3	$45,757.59	$8.00	$270.60	$46,020.19	99.43%
12/9/16	424	4	$176,169.60	$29,361.60	$271.60	$147,079.60	119.78%
12/9/16	424	4	$176,169.60	$88,084.80	$272.60	$88,357.40	199.38%
12/9/16	424	4	$146,808.00	$58,723.20	$273.60	$88,358.40	166.15%
12/16/16	406	5	$5,180.40		$72.00	$5,252.40	98.63%
12/16/16	406	5	$4,630.35	$321.27		$4,309.08	107.46%
12/16/16	406	5	$5,328.18		$37.26	$5,365.44	99.31%
12/23/16	416	4	$15,476.16		$183.36	$15,659.52	98.83%
12/23/16	416	4	$10,018.68	$392.88		$9,625.80	104.08%
12/23/16	416	4	$6,261.68	$245.56		$6,016.12	104.08%
12/23/16	416	4	$2,504.67	$98.22		$2,406.45	104.08%
12/23/16	416	4	$6,261.68	$2,652.01		$3,609.67	173.47%
12/23/16	416	4	$6,261.68	$2,652.01		$3,609.67	173.47%
12/23/16	416	4	$17,565.12	$5,745.60		$11,819.52	148.61%
12/30/16	405	4	$2,137.41		$530.86	$2,668.27	80.10%
1/6/17	241	5	$6,385.50	$4,802.46		$1,583.04	403.37%
1/6/17	241	4	$61,083.83	$2,668.27		$58,415.56	104.57%
1/6/17	241	4	$2,475.50	$0.20		$2,475.30	100.01%
1/6/17	241	6	$16,199.96	$16,199.96		$0.00	Erroneously Paid
1/13/17	331	4	$5,768.18			$5,768.18	Documentation Issue
1/27/17	385	4	$9,622.80	$42.60		$9,580.20	100.44%
1/27/17	385	4	$2,679.09	$16.59		$2,662.50	100.62%
1/27/17	385	4	$2,679.09	$16.59		$2,662.50	100.62%
2/3/17	423	3	$36,750.00		$10,500.00	$47,250.00	77.78%

Report Date	Weekly Sample Size	Claim type	Amount Paid	Overpayment	Underpayment	Amount that should have been paid	Percentage difference to apply
2/3/17	423	4	$2,010.96	$131.95		$1,879.01	107.02%
2/10/17	365	2	$4,450.00		$0.90	$4,450.90	99.98%
2/10/17	365	4	$58,254.90	$1,082.67		$57,172.23	101.89%
2/10/17	365	4	$2,340.48	$0.02		$2,340.46	100.00%
2/17/17	387	3	$76,133.00		$62.00	$76,195.00	99.92%
2/17/17	387	3	$56,574.00		$28.00	$56,602.00	99.95%
2/17/17	387	4	$2,026.08		$57.89	$2,083.97	97.22%
2/17/17	387	5	$4,316.80		$236.09	$4,552.89	94.81%
2/24/17	411	3	$35,911.45		$6,605.19	$42,516.64	84.46%
2/24/17	411	4	$3,658.62	$3,628.94		$29.68	12326.89%
2/24/17	411	4	$5,745.64		$35.80	$5,781.44	99.38%
2/24/17	411	4	$8,810.29		$54.90	$8,865.19	99.38%
2/24/17	411	4	$2,950.00	$937.00		$2,013.00	146.55%
2/24/17	411	5	$2,560.70	$140.91		$2,419.79	105.82%
3/3/17	370	4	$24,180.84	$0.20		$24,180.64	100.00%
3/3/17	370	5	$3,659.06		$75.15	$3,734.21	97.99%
3/3/17	370	4	$5,386.56	$6.40		$5,380.16	100.12%
3/10/17	305	4	$5,325.00	$2,662.50		$2,662.50	200.00%
3/10/17	305	5	$10,139.40	$8,683.20		$1,456.20	696.29%
3/17/17	356	2	$4,732.89		$54.41	$4,787.30	98.86%
3/17/17	356	3	$72,318.75	$18,503.62		$53,815.13	134.38%
3/17/17	356	3	$26,185.60	$268.54		$25,917.06	101.04%
3/17/17	356	5	$3,226.74		$404.81	$3,631.55	88.85%
3/24/17	323	4	$2,279.20		$0.30	$2,279.50	99.99%
3/24/17	323	4	$5,049.58		$0.30	$5,049.88	99.99%
3/24/17	323	4	$28,059.17	$28,059.16		$0.01	280591700.04%
3/31/17	354	3	$60,037.97	$60,037.97		$0.00	Erroneously Paid
3/31/17	354	4	$19,985.00	$15,185.00		$4,800.00	416.35%
3/31/17	354	4	$4,800.00		$15,185.00	$19,985.00	24.02%
3/31/17	354	4	$8,344.71	$7,290.36		$1,054.35	791.46%
4/7/17	392	4	$10,610.22		$4,428.49	$15,038.71	70.55%
4/7/17	392	5	$3,281.65	$1,581.52		$1,700.13	193.02%
4/7/17	392	4	$83,575.70	$79,427.74		$4,147.96	2014.86%
4/7/17	392	5	$5,069.41	$260.15		$4,809.26	105.41%
4/7/17	392	4	$25,104.00	$50.00		$25,054.00	100.20%
4/14/17	407	3	$46,334.72		$1,882.87	$48,217.59	96.10%
4/14/17	407	4	$3,570.56		$124.67	$3,695.23	96.63%
4/14/17	407	4	$3,570.56		$124.67	$3,695.23	96.63%
4/21/17	438	4	$11,866.40	$8,833.82		$3,032.58	391.30%
4/21/17	438	4	$11,866.40	$8,833.82		$3,032.58	391.30%
4/21/17	438	4	$12,052.95	$8,972.69		$3,080.26	391.30%
4/21/17	438	4	$11,930.10	$8,889.24		$3,040.86	392.33%
4/21/17	438	5	$3,006.75		$16.24	$3,022.99	99.46%
4/28/17	419	3	$26,690.99	$4,500.00		$22,190.99	120.28%
4/28/17	419	4	$9,542.54	$475.72		$9,066.82	105.25%
4/28/17	419	4	$8,344.71	$7,290.36		$1,054.35	791.46%
5/5/17	367	3	$52,860.10	$8,428.26		$44,431.84	118.97%
5/5/17	367	3	$45,002.36	$2,336.02		$42,666.34	105.48%
5/5/17	367	5	$3,868.56		$348.17	$4,216.73	91.74%
5/5/17	367	5	$5,201.66	$8.64		$5,193.02	100.17%
5/12/17	338	3	$32,586.47	$18,949.78		$13,636.69	238.96%
5/12/17	338	4	$15,115.41	$13,115.92		$1,999.49	755.96%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
5/19/17	469	4	$4,633.96	$3,447.90		$1,186.06	390.70%
5/19/17	469	5	$4,531.85	$9.05		$4,522.80	100.20%
5/26/17	422	4	$31,500.00	$7,458.00		$24,042.00	131.02%
6/2/17	412	4	$9,589.80	$36.00		$9,553.80	100.38%
6/2/17	412	4	$4,087.78		$29.96	$4,117.74	99.27%
6/9/17	311	4	$71,512.00	$20.00		$71,492.00	100.03%
6/16/17	431	4	$19,866.00		$52.80	$19,918.80	99.73%
6/16/17	431	4	$5,318.28		$265.91	$5,584.19	95.24%
7/10/17	572	5	$4,599.94		$21.21	$4,621.15	99.54%
7/10/17	572	3	$44,532.34	$1,415.66		$43,116.68	103.28%
7/10/17	572	3	$39,563.13	$39,563.13		$0.00	Erroneously Paid
7/10/17	572	3	$36,208.39	$13,707.90		$22,500.49	160.92%
7/10/17	572	3	$34,166.95	$34,166.95		$0.00	Erroneously Paid
7/10/17	572	5	$3,313.34		$122.45	$3,435.79	96.44%
7/10/17	572	4	$2,368.00		$2.00	$2,370.00	99.92%
7/10/17	572	5	$4,019.57		$120.27	$4,139.84	97.09%
7/11/17	377	5	$5,039.58		$0.13	$5,039.71	100.00%
7/11/17	377	4	$6,986.40		$8.64	$6,995.04	99.88%
7/21/17	411	5	$4,469.38		$7.76	$4,477.14	99.83%
7/21/17	411	5	$2,509.26	$62.02		$2,447.24	102.53%
7/21/17	411	5	$3,770.46		$270.00	$4,040.46	93.32%
7/21/17	411	3	$28,184.88	$1,657.13		$26,527.75	106.25%
7/21/17	411	4	$7,139.78		$1,004.46	$8,144.24	87.67%
7/21/17	411	4	$6,317.19		$4.46	$6,321.65	99.93%
7/28/17	446	4	$23,839.20	$11,919.60		$11,919.60	200.00%
7/28/17	446	5	$9,895.50	$989.55		$8,905.95	111.11%
8/11/17	404	4	$11,678.20	$10,029.34		$1,648.86	708.26%
8/11/17	404	4	$23,356.40	$20,063.14		$3,293.26	709.22%
8/11/17	404	4	$20,188.13	$20,141.81		$46.32	43584.05%
8/18/17	485	4	$2,134.71	$23.04		$2,111.67	101.09%
8/18/17	485	4	$2,134.71	$23.04		$2,111.67	101.09%
8/18/17	485	4	$2,134.71	$23.04		$2,111.67	101.09%
8/18/17	485	4	$2,134.71	$23.04		$2,111.67	101.09%
8/18/17	485	5	$3,060.00	$1,727.71		$1,332.29	229.68%
8/18/17	485	4	$2,994.17		$153.88	$3,148.05	95.11%
8/18/17	485	3	$31,574.83		$3.00	$31,577.83	99.99%
8/25/17	445	4	$4,504.11		$0.84	$4,504.95	99.98%
8/25/17	445	4	$5,584.48		$0.32	$5,584.80	99.99%
8/25/17	445	5	$7,590.52		$0.46	$7,590.98	99.99%
8/25/17	445	5	$10,265.49	$144.18		$10,121.31	101.42%
9/1/17	401	4	$3,568.67	$2,695.08		$873.59	408.51%
9/1/17	401	4	$2,976.00		$172.05	$3,148.05	94.53%
9/8/17	400	4	$7,479.68			$7,479.68	Documentation Issue
9/8/17	400	4	$7,479.68			$7,479.68	Documentation Issue
9/8/17	400	4	$5,928.00			$5,928.00	Documentation Issue
9/8/17	400	4	$5,457.86			$5,457.86	Documentation Issue
9/8/17	400	4	$2,620.54			$2,620.54	Documentation Issue
9/8/17	400	4	$6,125.60			$6,125.60	Documentation Issue
9/8/17	400	4	$6,032.40			$6,032.40	Documentation Issue
9/8/17	400	4	$2,418.96			$2,418.96	Documentation Issue
9/8/17	400	4	$5,584.00			$5,584.00	Documentation Issue
9/15/17	388	4	$2,182.18		$14.34	$2,196.52	99.35%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
9/15/17	388	3	$64,583.54	$280.34		$64,303.20	100.44%
9/22/17	492	3	$167,862.46	$72,233.04		$95,629.42	175.53%
9/22/17	492	4	$4,451.00	$4,249.77		$201.23	2211.90%
9/22/17	492	5	$7,180.26	$10.37		$7,169.89	100.14%
9/29/17	368	4	$3,146.39		$1.66	$3,148.05	99.95%
9/29/17	368	4	$27,502.62	$27,502.62		$0.00	Erroneously Paid
9/29/17	368	5	$2,585.46	$0.16		$2,585.30	100.01%
10/6/17	380	5	$2,939.71	$2,939.71		$0.00	Erroneously Paid
10/20/17	420	4	$2,134.71	$23.04		$2,111.67	101.09%
10/20/17	420	4	$3,663.48	$698.08		$2,965.40	123.54%
10/20/17	420	4	$2,134.71	$23.04		$2,111.67	101.09%
10/20/17	420	4	$5,318.28		$265.91	$5,584.19	95.24%
10/27/17	428	4	$5,124.27	$51.55		$5,072.72	101.02%
10/27/17	428	4	$5,801.90	$160.00		$5,641.90	102.84%
10/27/17	428	4	$32,721.00	$32,721.00		$0.00	Erroneously Paid
10/27/17	428	4	$5,203.61	$2.20		$5,201.41	100.04%
11/17/17	448	3	$77,846.67		$359,168.25	$437,014.92	17.81%
11/17/17	448	4	$13,366.98	$11,507.76		$1,859.22	718.96%
11/17/17	448	4	$13,366.98	$11,507.78		$1,859.20	718.96%
11/17/17	448	4	$13,366.98	$11,507.76		$1,859.22	718.96%
11/17/17	448	4	$25,671.00	$20,469.59		$5,201.41	493.54%
11/24/17	413	4	$4,407.72	$2,189.15		$2,218.57	198.67%
12/1/17	394	4	$7,947.75	$7,265.61		$682.14	1165.12%
12/1/17	394	4	$8,344.71	$7,263.13		$1,081.58	771.53%
12/1/17	394	4	$8,344.71	$7,267.59		$1,077.12	774.72%
12/1/17	394	4	$8,344.71	$7,265.61		$1,079.10	773.30%
12/1/17	394	4	$8,344.71	$7,265.61		$1,079.10	773.30%
12/1/17	394	4	$8,344.71	$7,263.13		$1,081.58	771.53%
12/1/17	394	4	$4,045.92	$3,522.72		$523.20	773.30%
12/8/17	376	3	$236,894.61	$814.62		$236,079.99	100.35%
12/8/17	376	4	$10,520.00	$8,448.80		$2,071.20	507.92%
12/8/17	376	4	$6,803.75		$257.87	$7,061.62	96.35%
12/8/17	376	4	$7,945.00		$965.00	$8,910.00	89.17%
12/22/17	398	4	$21,208.76	$14,545.20		$6,663.56	318.28%
12/22/17	398	4	$80,117.10	$67,626.40		$12,490.70	641.41%
12/22/17	398	2	$5,482.70		$10.00	$5,492.70	99.82%
1/5/18	280	5	$17,595.88		$9.00	$17,604.88	99.95%
1/5/18	280	4	$2,010.10	$0.05		$2,010.05	100.00%
1/5/18	280	4	$2,010.10	$0.05		$2,010.05	100.00%
1/5/18	280	4	$2,010.10	$0.05		$2,010.05	100.00%
1/5/18	280	4	$2,010.10	$0.05		$2,010.05	100.00%
1/5/18	280	4	$2,010.10	$0.05		$2,010.05	100.00%
1/12/18	308	4	$18,691.00		$0.20	$18,691.20	100.00%
1/12/18	308	5	$4,567.81	$633.97		$3,933.84	116.12%
1/12/18	308	5	$3,527.16		$272.46	$3,799.62	92.83%
1/26/18	426	4	$10,493.05		$7,082.19	$17,575.24	59.70%
2/2/18	393	2	$4,591.06		$1,000.00	$5,591.06	82.11%
2/2/18	393	3	$26,891.21	$15,120.00		$11,771.21	228.45%
2/2/18	393	4	$65,993.40	$44,926.69		$21,066.71	313.26%
2/2/18	393	4	$2,402.02		$1,637.90	$4,039.92	59.46%
2/2/18	393	4	$4,955.66		$187.83	$5,143.49	96.35%
2/2/18	393	4	$16,318.80		$3,600.00	$19,918.80	81.93%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
2/16/18	426	5	$10,638.54		$95,746.86	$106,385.40	10.00%
2/16/18	426	4	$27,472.00		$6.00	$27,478.00	99.98%
2/16/18	426	4	$11,000.00	$9,900.00		$1,100.00	1000.00%
2/16/18	426	4	$5,826.88	$5,826.88		$0.00	Erroneously Paid
2/16/18	426	5	$7,328.20	$3.00		$7,325.20	100.04%
2/23/18	371	4	$2,071.62	$0.42		$2,071.20	100.02%
2/23/18	371	4	$2,644.37		$2,644.37	$5,288.74	50.00%
2/23/18	371	5	$11,203.02	$24.30		$11,178.72	100.22%
2/23/18	371	2	$5,503.90		$63.00	$5,566.90	98.87%
2/23/18	371	4	$13,139.20	$1,000.00		$12,139.20	108.24%
2/23/18	371	4	$4,855.68		$43,701.12	$48,556.80	10.00%
2/23/18	371	4	$4,855.68		$43,701.12	$48,556.80	10.00%
2/23/18	371	4	$2,123.17		$424.63	$2,547.80	83.33%
3/2/18	392	4	$2,086.81		$54.76	$2,141.57	97.44%
3/2/18	392	4	$2,072.14		$12.02	$2,084.16	99.42%
3/2/18	392	4	$2,071.20	$31.20		$2,040.00	101.53%
3/2/18	392	3	$61,593.08	$100.00		$61,493.08	100.16%
3/9/18	350	4	$4,095.12		$122.36	$4,217.48	97.10%
3/9/18	350	4	$11,250.00	$8,250.00		$3,000.00	375.00%
3/16/18	411	4	$12,136.50	$10,963.96		$1,172.54	1035.06%
3/16/18	411	3	$46,829.63	$71.26		$46,758.37	100.15%
3/16/18	411	3	$27,868.40	$692.27		$27,176.13	102.55%
3/16/18	411	4	$20,436.85	$17,436.74		$3,000.11	681.20%
3/23/18	432	4	$5,669.40	$467.99		$5,201.41	109.00%
3/30/18	388	4	$2,600.00	$2,600.00		$0.00	Erroneously Paid
3/30/18	388	4	$3,143.40		$2.99	$3,146.39	99.90%
3/30/18	388	5	$42,473.50	$29,174.39		$13,299.11	319.37%
3/30/18	388	5	$10,935.80	$8,359.14		$2,576.66	424.42%
4/6/18	441	4	$3,240.00	$1,167.86		$2,072.14	156.36%
4/6/18	441	5	$4,077.60		$7.76	$4,085.36	99.81%
4/6/18	441	5	$11,205.60	$6,534.94		$4,670.66	239.91%
4/6/18	441	4	$5,429.34	$4,295.13		$1,134.21	478.69%
4/6/18	441	5	$6,171.00	$5,928.94		$242.06	2549.37%
4/6/18	441	4	$30,663.60	$20,877.10		$9,786.50	313.33%
4/6/18	441	5	$5,201.41		$117.70	$5,319.11	97.79%
4/6/18	441	4	$6,663.56		$14,545.20	$21,208.76	31.42%
4/6/18	441	4	$4,653.99	$4,653.99		$0.00	Erroneously Paid
4/13/18	481	4	$6,381.12	$0.01		$6,381.11	100.00%
4/13/18	481	4	$3,667.93	$65.05		$3,602.88	101.81%
4/13/18	481	4	$3,547.80	$1,000.00		$2,547.80	139.25%
4/13/18	481	4	$2,435.87		$125.60	$2,561.47	95.10%
4/13/18	481	4	$2,248.20		$22,482.00	$24,730.20	9.09%
4/13/18	481	4	$2,263.50	$2,263.50		$0.00	Erroneously Paid
4/13/18	481	5	$6,665.87		$6,665.88	$13,331.75	50.00%
4/13/18	481	5	$5,426.22	$29.82		$5,396.40	100.55%
4/20/18	483	4	$2,980.00	$20.00		$2,960.00	100.68%
4/20/18	483	4	$6,293.49		$9,241.15	$15,534.64	40.51%
4/20/18	483	4	$9,487.30	$9,050.21		$437.09	2170.56%
4/27/18	447	4	$2,259.47	$2,259.47		$0.00	Erroneously Paid
4/27/18	447	4	$2,500.80		$107.53	$2,608.33	95.88%
4/27/18	447	4	$2,259.47	$17.34		$2,242.13	100.77%
4/27/18	447	5	$4,373.59		$159.48	$4,533.07	96.48%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
5/4/18	381	4	$2,276.15		$24.69	$2,300.84	98.93%
5/4/18	381	4	$2,276.15		$24.69	$2,300.84	98.93%
5/4/18	381	5	$4,104.00	$4,104.00		$0.00	Erroneously Paid
5/11/18	395	4	$12,564.00		$7,635.60	$20,199.60	62.20%
5/11/18	395	4	$32,708.00	$2.00		$32,706.00	100.01%
5/11/18	395	5	$4,419.32		$2.26	$4,421.58	99.95%
5/11/18	395	5	$6,664.96		$0.91	$6,665.87	99.99%
5/18/18	484	4	$6,320.00		$128.00	$6,448.00	98.01%
5/25/18	461	4	$4,367.11	$0.04		$4,367.07	100.00%
5/25/18	461	4	$5,868.00	$484.54		$5,383.46	109.00%
6/1/18	408	4	$21,964.80	$17,322.24		$4,642.56	473.12%
6/1/18	408	4	$81,232.20		$3.00	$81,235.20	100.00%
6/1/18	408	4	$4,945.14		$0.05	$4,945.19	100.00%
6/1/18	408	5	$10,143.90		$1,182.10	$11,326.00	89.56%
7/6/18	540	4	$15,836.34	$12,952.18		$2,884.16	549.08%
7/6/18	540	5	$22,696.80		$47.88	$22,744.68	99.79%
7/6/18	540	4	$17,828.56	$200.00		$17,628.56	101.13%
7/6/18	540	4	$3,124.80		$18.60	$3,143.40	99.41%
7/13/18	284	4	$2,136.00	$102.00		$2,034.00	105.01%
7/13/18	284	4	$5,026.32		$1,003.82	$6,030.14	83.35%
7/13/18	284	4	$2,760.00	$155.18		$2,604.82	105.96%
7/20/18	534	4	$45,511.20	$729.00		$44,782.20	101.63%
7/20/18	534	4	$40,454.40	$648.40		$39,806.00	101.63%
7/20/18	534	4	$45,511.20	$729.00		$44,782.20	101.63%
7/20/18	534	4	$50,568.00	$810.00		$49,758.00	101.63%
7/20/18	534	4	$15,170.40	$243.00		$14,927.40	101.63%
7/20/18	534	4	$40,454.40	$648.00		$39,806.40	101.63%
7/20/18	534	5	$4,015.32	$15.77		$3,999.55	100.39%
7/20/18	534	5	$4,093.71		$1,289.75	$5,383.46	76.04%
7/20/18	534	3	$43,352.83	$37,347.09		$6,005.74	721.86%
7/27/18	465	4	$2,632.09		$2,632.09	$5,264.18	50.00%
7/27/18	465	4	$2,614.66		$2,674.08	$5,288.74	49.44%
7/27/18	465	4	$2,518.09		$2,518.09	$5,036.18	50.00%
7/27/18	465	4	$4,532.83		$3.00	$4,535.83	99.93%
7/27/18	465	4	$2,614.66	$66.86		$2,547.80	102.62%
7/27/18	465	4	$65,798.88	$50.00		$65,748.88	100.08%
7/27/18	465	4	$65,498.88	$510.72		$64,988.16	100.79%
7/27/18	465	4	$9,213.79	$8,617.27		$596.52	1544.59%
7/27/18	465	5	$5,383.46		$156.75	$5,540.21	97.17%
7/27/18	465	5	$4,093.71		$1,446.50	$5,540.21	73.89%
8/10/18	400	5	$55,694.72		$10,000.00	$65,694.72	84.78%
8/10/18	400	3	$98,005.35	$83,081.65		$14,923.70	656.71%
8/10/18	400	4	$8,582.27	$5,547.42		$3,034.85	282.79%
8/10/18	400	4	$31,436.40	$21,649.90		$9,786.50	321.22%
8/10/18	400	4	$3,175.43	$91.99		$3,083.44	102.98%
8/10/18	400	4	$3,714.50		$252.70	$3,967.20	93.63%
8/17/18	449	3	$172,328.48	$165,668.35		$6,660.13	2587.46%
8/24/18	435	3	$1,526,574.33	$274,925.70		$1,251,648.63	121.97%
8/24/18	435	5	$5,535.75		$4.46	$5,540.21	99.92%
8/31/18	398	4	$6,430.06		$2,732.66	$9,162.72	70.18%
9/7/18	401	4	$6,004.42		$75.58	$6,080.00	98.76%
9/7/18	401	5	$3,171.40		$40.33	$3,211.73	98.74%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
9/7/18	401	5	$3,330.96		$303.16	$3,634.12	91.66%
9/14/18	340	4	$2,688.96		$1,000.00	$3,688.96	72.89%
9/14/18	340	5	$19,497.69	$4.72		$19,492.97	100.02%
9/21/18	500	4	$2,364.72	$292.58		$2,072.14	114.12%
9/21/18	500	5	$3,189.60		$254.37	$3,443.97	92.61%
9/21/18	500	5	$2,530.15		$133.17	$2,663.32	95.00%
9/28/18	407	4	$32,210.40	$17,712.00		$14,498.40	222.17%
9/28/18	407	5	$2,661.56		$249.04	$2,910.60	91.44%
10/5/18	318	2	$6,155.30	$333.25		$5,822.05	105.72%
10/12/18	367	4	$2,347.85		$770.65	$3,118.50	75.29%
10/12/18	367	4	$2,403.65	$1,413.99		$989.66	242.88%
10/12/18	367	4	$5,029.90	$1,112.07		$3,917.83	128.38%
10/12/18	367	4	$3,518.40	$2,909.55		$608.85	577.88%
10/19/18	393	4	$5,773.36	$3,708.00		$2,065.36	279.53%
10/19/18	393	4	$10,836.00	$264.00		$10,572.00	102.50%
10/19/18	393	4	$5,179.71	$0.10		$5,179.61	100.00%
10/26/18	441	5	$2,989.44	$319.20		$2,670.24	111.95%
10/26/18	441	4	$2,098.08		$27.12	$2,125.20	98.72%
10/26/18	441	4	$25,629.12		$377.28	$26,006.40	98.55%
10/26/18	441	4	$2,621.99	$87.19		$2,534.80	103.44%
10/26/18	441	3	$143,688.42	$6,717.22		$136,971.20	104.90%
10/26/18	441	5	$4,739.43	$17.87		$4,721.56	100.38%
10/26/18	441	4	$4,170.00	$2,002.80		$2,167.20	192.41%
11/2/18	283	4	$10,705.56		$3.00	$10,708.56	99.97%
11/2/18	283	4	$3,025.93		$4.46	$3,030.39	99.85%
11/16/18	408	4	$2,134.08		$273.78	$2,407.86	88.63%
11/16/18	408	4	$3,026.92		$4.46	$3,031.38	99.85%
11/16/18	408	5	$3,611.26		$478.42	$4,089.68	88.30%
11/16/18	408	5	$4,356.68		$4.66	$4,361.34	99.89%
11/23/18	403	4	$31,068.00		$648.00	$31,716.00	97.96%
11/23/18	403	4	$62,136.00		$1,296.00	$63,432.00	97.96%
11/23/18	403	4	$62,136.00		$1,296.00	$63,432.00	97.96%
11/23/18	403	4	$31,068.00		$648.00	$31,716.00	97.96%
11/23/18	403	4	$62,136.00		$1,296.00	$63,432.00	97.96%
11/23/18	403	4	$23,405.76	$339.55		$23,066.21	101.47%
11/23/18	403	5	$6,750.42		$0.04	$6,750.46	100.00%
11/23/18	403	4	$3,019.23	$0.02		$3,019.21	100.00%
11/23/18	403	4	$24,639.12	$24,639.12		$0.00	Erroneously Paid
11/30/18	373	4	$3,450.33	$133.96		$3,316.37	104.04%
11/30/18	373	5	$4,791.34	$310.47		$4,480.87	106.93%
11/30/18	373	5	$26,816.72		$0.03	$26,816.75	100.00%
12/7/18	361	4	$8,465.76	$278.40		$8,187.36	103.40%
12/7/18	361	5	$4,180.52		$160.31	$4,340.83	96.31%
12/14/18	370	1	$1,124.55	$530.74		$593.81	189.38%
12/14/18	370	4	$2,518.09		$2,518.09	$5,036.18	50.00%
12/14/18	370	3	$876,192.81	$835,784.23		$40,408.58	2168.33%
12/14/18	370	4	$10,125.00	$9,952.20		$172.80	5859.38%
12/14/18	370	5	$2,912.90		$0.02	$2,912.92	100.00%
12/14/18	370	5	$2,839.39		$379.87	$3,219.26	88.20%
12/21/18	471	4	$3,254.22		$2,710.62	$5,964.84	54.56%
12/21/18	471	4	$8,126.73		$1,073.61	$9,200.34	88.33%
1/11/19	252	4	$67,555.56	$53,190.01		$14,365.55	470.26%

Report Date	Weekly Sample Size	Claim type	Amount Paid	Overpayment	Underpayment	Amount that should have been paid	Percentage difference to apply
1/11/19	252	5	$8,414.24	$1,308.70		$7,105.54	118.42%
1/11/19	252	4	$6,962.04	$211.92		$6,750.12	103.14%
1/18/19	491	4	$12,016.83	$0.03		$12,016.80	100.00%
1/18/19	491	5	$4,477.63		$3.24	$4,480.87	99.93%
1/18/19	491	5	$2,894.08		$16.38	$2,910.46	99.44%
1/18/19	491	5	$4,216.46		$1.05	$4,217.51	99.98%
1/25/19	436	1	$27,526.22		$0.08	$27,526.30	100.00%
1/25/19	436	4	$2,031.94		$33.42	$2,065.36	98.38%
1/25/19	436	4	$2,116.44	$85.56		$2,030.88	104.21%
1/25/19	436	4	$2,325.00	$0.02		$2,324.98	100.00%
1/25/19	436	4	$23,424.40	$9,996.40		$13,428.00	174.44%
1/25/19	436	5	$10,131.66		$30.87	$10,162.53	99.70%
2/1/19	345	4	$20,731.20	$531.60		$20,199.60	102.63%
2/1/19	345	4	$51,828.00	$1,329.00		$50,499.00	102.63%
2/1/19	345	4	$51,828.00	$1,329.00		$50,499.00	102.63%
2/1/19	345	4	$62,193.60	$1,594.80		$60,598.80	102.63%
2/1/19	345	4	$82,924.80	$2,126.40		$80,798.40	102.63%
2/1/19	345	4	$103,656.00	$2,658.00		$100,998.00	102.63%
2/1/19	345	4	$62,193.60	$1,594.80		$60,598.80	102.63%
2/1/19	345	4	$41,642.40	$1,243.20		$40,399.20	103.08%
2/1/19	345	4	$20,731.20	$531.60		$20,199.60	102.63%
2/1/19	345	4	$2,677.14	$34.67		$2,642.47	101.31%
2/1/19	345	4	$2,241.19		$0.73	$2,241.92	99.97%
2/1/19	345	4	$2,241.19		$0.72	$2,241.91	99.97%
2/8/19	360	4	$4,594.09	$1,710.28		$2,883.81	159.31%
2/8/19	360	4	$16,380.45	$15,907.72		$472.73	3465.08%
2/8/19	360	4	$3,255.00	$575.00		$2,680.00	121.46%
2/8/19	360	4	$4,594.09	$3,099.55		$1,494.54	307.39%
2/15/19	398	4	$9,370.00	$7,451.16		$1,918.84	488.32%
2/15/19	398	5	$6,690.60	$5,982.48		$708.12	944.84%
2/15/19	398	5	$3,765.45	$0.04		$3,765.41	100.00%
3/8/19	318	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/8/19	318	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/8/19	318	4	$21,840.60	$21,367.87		$472.73	4620.10%
3/8/19	318	4	$30,677.00	$50.00		$30,627.00	100.16%
3/8/19	318	5	$5,750.12		$1,000.00	$6,750.12	85.19%
3/15/19	346	4	$10,336.59	$10,329.05		$7.54	137090.05%
3/15/19	346	4	$4,696.16	$4,696.16		$0.00	Erroneously Paid
3/15/19	346	4	$7,089.20	$5,914.07		$1,175.13	603.27%
3/15/19	346	4	$10,336.59	$10,329.05		$7.54	137090.05%
3/15/19	346	4	$6,120.18	$5,974.65		$145.53	4205.44%
3/15/19	346	4	$6,120.18	$5,970.19		$149.99	4080.39%
3/15/19	346	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/15/19	346	4	$6,418.67	$5,362.24		$1,056.43	607.58%
3/15/19	346	4	$5,697.47	$4,760.14		$937.33	607.84%
3/22/19	589	4	$4,474.95	$4,285.61		$189.34	2363.45%
3/22/19	589	4	$8,763.87	$8,671.76		$92.11	9514.57%
3/22/19	589	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/22/19	589	4	$6,025.14	$4,850.00		$1,175.14	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,026.13	$4,850.00		$1,176.13	512.37%
3/22/19	589	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,025.14	$4,850.01		$1,175.13	512.72%
3/22/19	589	4	$6,026.13	$4,851.00		$1,175.13	512.81%
3/29/19	532	4	$6,349.32		$57,143.88	$63,493.20	10.00%
4/19/19	509	4	$43,200.00	$34,734.24		$8,465.76	510.29%
4/19/19	509	5	$5,051.97	$4,131.63		$920.34	548.92%
4/26/19	383	4	$7,600.00	$10.00		$7,590.00	100.13%
5/3/19	367	5	$5,040.53		$41.19	$5,081.72	99.19%
5/10/19	392	5	$4,453.60	$325.12		$4,128.48	107.88%
5/17/19	434	4	$8,029.46	$2,140.00		$5,889.46	136.34%
5/17/19	434	4	$3,463.29		$42.86	$3,506.15	98.78%
5/17/19	434	4	$20,245.68	$1,557.36		$18,688.32	108.33%
5/17/19	434	4	$13,497.12	$1,038.24		$12,458.88	108.33%
5/24/19	432	4	$6,424.46	$535.00		$5,889.46	109.08%
6/7/19	345	4	$115,416.00	$113,236.92		$2,179.08	5296.55%
6/7/19	345	4	$6,359.40		$30.96	$6,390.36	99.52%
6/7/19	345	4	$6,359.40		$30.96	$6,390.36	99.52%
6/7/19	345	5	$30,902.51		$1,163.67	$32,066.18	96.37%
6/14/19	361	5	$5,787.72	$9.85		$5,777.87	100.17%
7/5/19	541	1	$149,994.00	$30,941.10		$119,052.90	125.99%
7/5/19	541	3	$30,316.10	$3,146.87		$27,169.23	111.58%
7/5/19	541	5	$32,066.14		$0.04	$32,066.18	100.00%
7/12/19	291	5	$9,790.12	$2,711.37		$7,078.75	138.30%
7/12/19	291	5	$19,236.96	$406.02		$18,830.94	102.16%
7/19/19	480	3	$123,104.44		$36,678.00	$159,782.44	77.05%
7/19/19	480	4	$3,791.04	$3,205.44		$585.60	647.38%
7/19/19	480	3	$228,611.85	$228,611.85		$0.00	Erroneously Paid
7/19/19	480	4	$12,458.34		$27.00	$12,485.34	99.78%
7/19/19	480	4	$4,066.89		$4,066.88	$8,133.77	50.00%
8/2/19	397	4	$15,480.63	$15,480.63		$0.00	Erroneously Paid
8/9/19	486	3	$29,356.83	$16,560.00		$12,796.83	229.41%
8/16/19	405	4	$10,890.99			$10,890.99	Documentation Issue
8/16/19	405	4	$10,890.99			$10,890.99	Documentation Issue
8/23/19	521	4	$100,003.46		$273,001.00	$373,004.46	26.81%
8/23/19	521	4	$100,003.46		$273,001.00	$373,004.46	26.81%
8/30/19	391	4	$2,355.84		$3.92	$2,359.76	99.83%
8/30/19	391	4	$2,355.84		$3.92	$2,359.76	99.83%
9/6/19	389	3	$96,626.70	$5,692.63		$90,934.07	106.26%
9/6/19	389	4	$2,152.43		$9.01	$2,161.44	99.58%
9/6/19	389	4	$6,551.87		$115.71	$6,667.58	98.26%

Report Date	Weekly Sample Size	Claim type	Amount Paid	Overpayment	Underpayment	Amount that should have been paid	Percentage difference to apply
9/6/19	389	5	$4,015.44	$260.19		$3,755.25	106.93%
10/11/19	358	5	$5,756.40	$72.00		$5,684.40	101.27%
10/18/19	482	5	$9,391.95	$9,391.95		$0.00	Erroneously Paid
10/18/19	482	5	$17,232.00		$129.14	$17,361.14	99.26%
10/25/19	436	5	$3,905.28	$600.48		$3,304.80	118.17%
10/25/19	436	4	$6,324.00	$4.00		$6,320.00	100.06%
10/25/19	436	4	$4,830.66		$1.48	$4,832.14	99.97%
11/1/19	418	5	$4,465.71	$8.91		$4,456.80	100.20%
11/1/19	418	4	$7,064.16		$1,328.16	$8,392.32	84.17%
11/1/19	418	4	$111,333.00	$111,333.00		$0.00	Erroneously Paid
11/1/19	418	4	$111,333.00	$111,333.00		$0.00	Erroneously Paid
11/1/19	418	4	$111,333.00	$111,333.00		$0.00	Erroneously Paid
11/1/19	418	4	$111,333.00	$111,333.00		$0.00	Erroneously Paid
11/1/19	418	4	$37,114.46	$37,114.46		$0.00	Erroneously Paid
11/1/19	418	4	$37,114.46	$37,114.46		$0.00	Erroneously Paid
11/1/19	418	4	$37,114.46	$37,114.46		$0.00	Erroneously Paid
11/1/19	418	4	$37,114.46	$37,114.46		$0.00	Erroneously Paid
11/1/19	418	4	$37,114.46	$37,114.46		$0.00	Erroneously Paid
11/1/19	418	4	$111,334.46	$111,334.46		$0.00	Erroneously Paid
11/1/19	418	4	$39,938.89	$39,938.89		$0.00	Erroneously Paid
11/1/19	418	4	$57,051.25	$57,051.25		$0.00	Erroneously Paid
11/1/19	418	4	$57,051.25	$57,051.25		$0.00	Erroneously Paid
11/1/19	418	4	$171,133.65	$171,133.65		$0.00	Erroneously Paid
11/1/19	418	4	$171,133.65	$171,133.65		$0.00	Erroneously Paid
11/1/19	418	4	$171,133.65	$171,133.65		$0.00	Erroneously Paid
11/8/19	405	5	$5,474.03		$219.29	$5,693.32	96.15%
11/8/19	405	4	$2,309.26		$4,609.60	$6,918.86	33.38%
11/8/19	405	5	$6,015.49		$31.54	$6,047.03	99.48%
11/15/19	361	5	$30,902.46		$1,163.72	$32,066.18	96.37%
11/15/19	361	5	$33,620.95	$1,554.77		$32,066.18	104.85%
11/15/19	361	5	$2,669.33	$5.33		$2,664.00	100.20%
11/22/19	455	4	$7,097.62	$7,097.62		$0.00	Erroneously Paid
12/13/19	394	4	$62,489.20	$40.00		$62,449.20	100.06%
12/13/19	394	4	$11,000.00	$9,900.00		$1,100.00	1000.00%
12/20/19	472	4	$14,665.00	$1,982.00		$12,683.00	115.63%
1/3/20	369	4	$8,263.14		$739.23	$9,002.37	91.79%
1/3/20	369	4	$8,787.14		$786.11	$9,573.25	91.79%
1/10/20	318	3	$94,681.94	$94,681.94		$0.00	Erroneously Paid
1/17/20	550	5	$24,128.34		$9,492.61	$33,620.95	71.77%
1/17/20	550	3	$30,757.30		$2,010.00	$32,767.30	93.87%
1/24/20	418	4	$2,366.54	$1,883.40		$483.14	489.82%
2/7/20	406	4	$8,817.18		$106.33	$8,923.51	98.81%
2/7/20	406	4	$4,147.28		$50.02	$4,197.30	98.81%
2/21/20	437	3	$50,538.00	$50,538.00		$0.00	Erroneously Paid
3/6/20	369	4	$2,545.60		$100.00	$2,645.60	96.22%
3/13/20	438	5	$10,775.70	$594.00		$10,181.70	105.83%
3/13/20	438	5	$5,245.16	$4.68		$5,240.48	100.09%
3/13/20	438	4	$16,143.93		$4.46	$16,148.39	99.97%
3/20/20	484	3	$581,730.13	$12,908.21		$568,821.92	102.27%
3/20/20	484	4	$4,489.61		$400.00	$4,889.61	91.82%
3/20/20	484	5	$3,056.82		$1,678.80	$4,735.62	64.55%
3/20/20	484	5	$5,868.56		$2,661.04	$8,529.60	68.80%

Report Date	*Weekly Sample Size*	*Claim type*	*Amount Paid*	*Overpayment*	*Underpayment*	*Amount that should have been paid*	*Percentage difference to apply*
3/20/20	484	5	$5,868.56		$533.10	$6,401.66	91.67%
3/20/20	484	5	$12,755.91	$10,933.64		$1,822.27	700.00%
3/27/20	456	3	$143,048.13	$3,292.80		$139,755.33	102.36%
3/27/20	456	4	$32,000.00		$4.46	$32,004.46	99.99%
3/27/20	456	4	$4,274.82	$83.82		$4,191.00	102.00%
3/27/20	456	5	$3,721.81		$483.59	$4,205.40	88.50%
3/27/20	456	5	$4,638.33		$261.19	$4,899.52	94.67%
4/3/20	364	5	$14,419.84		$26.96	$14,446.80	99.81%
4/3/20	364	5	$5,868.56		$533.10	$6,401.66	91.67%
4/18/20	457	4	$4,920.74		$232.13	$5,152.87	95.50%
4/18/20	457	5	$4,695.62		$95.95	$4,791.57	98.00%
4/24/20	453	3	$47,439.01	$47,439.01		$0.00	Erroneously Paid
4/24/20	453	4	$7,824.22		$8,756.90	$16,581.12	47.19%
5/1/20	349	4	$16,458.79	$14,812.92		$1,645.87	1000.01%

APPENDIX D: DETAIL RESULTS OF ANDERSON-DARLING AND HYPOTHESES TEST TO IDENTIFY ERRNOUSE PAYMENTS' PROBABILITY DISTRIBUTION FUNCTION

This appendix provides the detailed result of two separate Anderson-Darling (AD) tests on the CA-MMIS WPDR findings on inpatient and outpatient claims as detailed in APPENDIX C. Minitab was used as the tool to conduct 16 goodness of fit (GoF) of a distributional family for each distribution and AD results are presented.

Distribution Identification for Outpatient (CA-MMIS WPDR Data)

```
* NOTE * Fail to select a Johnson transformation function with P-Value > 0.05. No
transformation is made.
```

2-Parameter Exponential

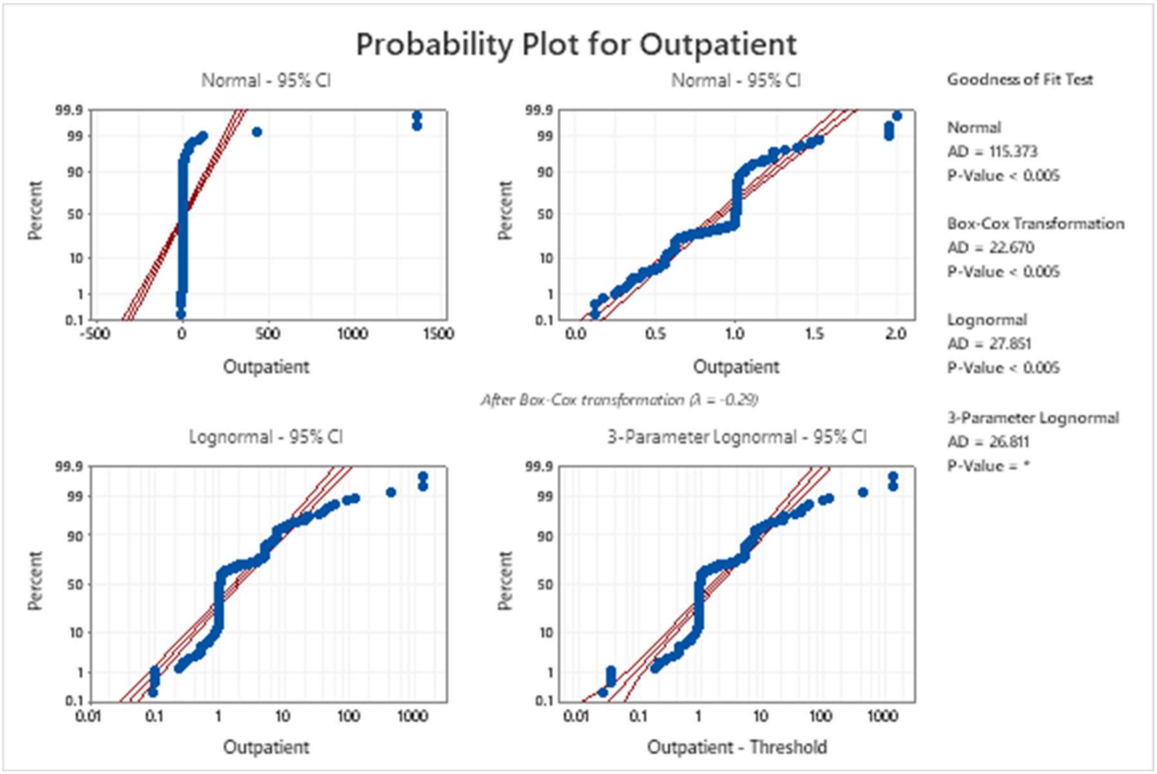
```
* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold
parameter is assumed fixed when calculating confidence intervals.
```

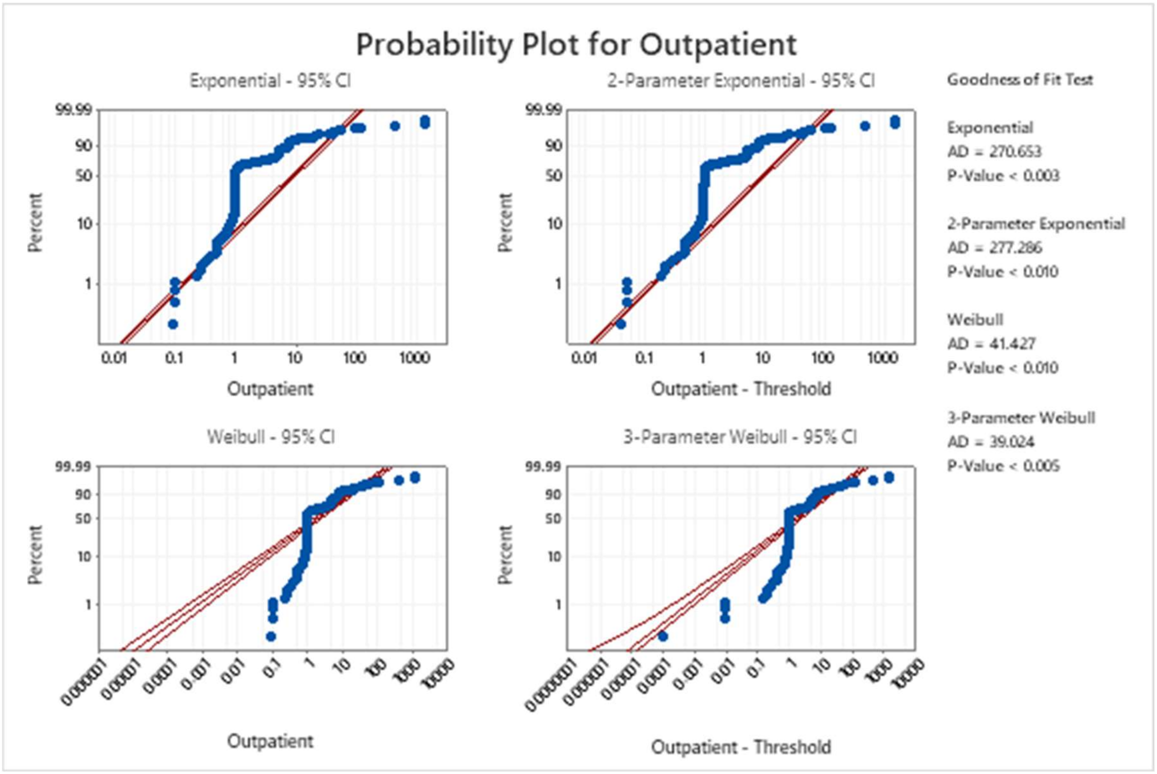
3-Parameter Weibull

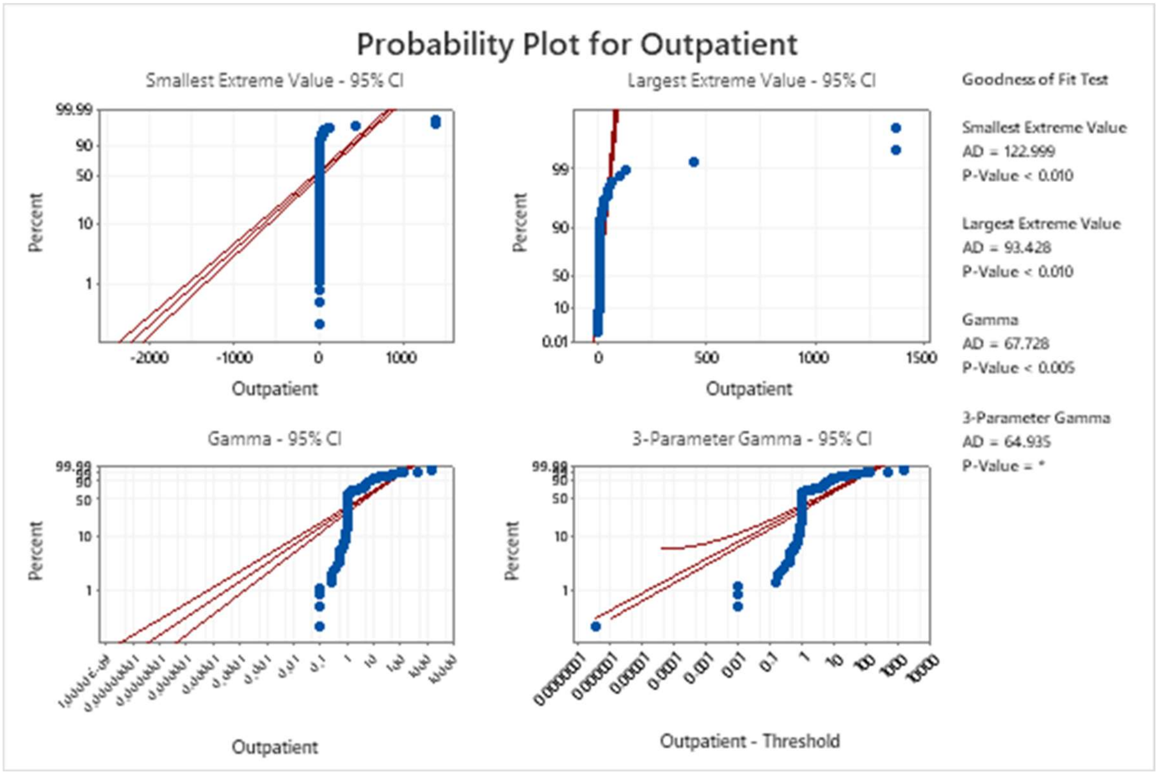
```
* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold
parameter is assumed fixed when calculating confidence intervals.
```

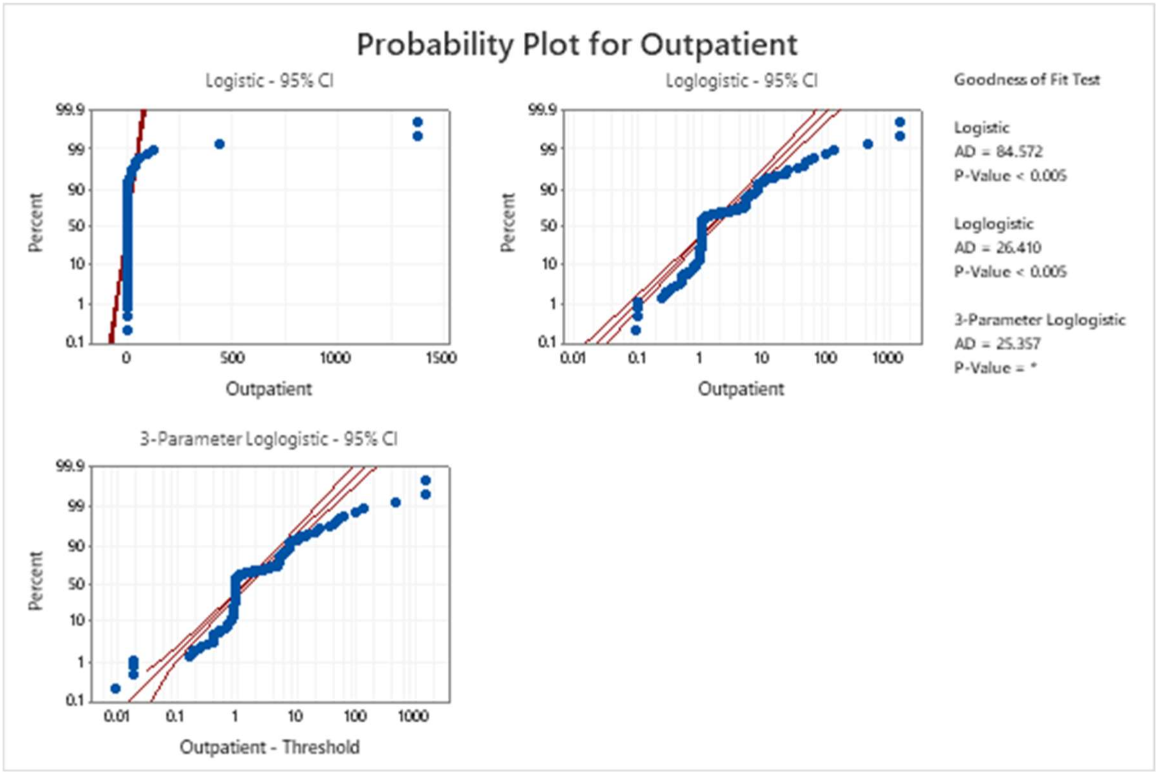
3-Parameter Gamma

```
* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold
parameter is assumed fixed when calculating confidence intervals.
```


Probability Plot for Outpatient
Normal - 95% CI
Normal - 95% CI
After Box-Cox transformation (λ = -0.29)
Lognormal - 95% CI
3-Parameter Lognormal - 95% CI
Percent
Outpatient
Outpatient - Threshold
Goodness of Fit Test
Normal
AD = 115.373
P-Value < 0.005
Box-Cox Transformation
AD = 22.670
P-Value < 0.005
Lognormal
AD = 27.851
P-Value < 0.005
3-Parameter Lognormal
AD = 26.811
P-Value = *


Probability Plot for Outpatient
Exponential - 95% CI
2-Parameter Exponential - 95% CI
Weibull - 95% CI
3-Parameter Weibull - 95% CI
Percent
Outpatient
Outpatient - Threshold
Goodness of Fit Test
Exponential
AD = 270.653
P-Value < 0.003
2-Parameter Exponential
AD = 277.286
P-Value < 0.010
Weibull
AD = 41.427
P-Value < 0.010
3-Parameter Weibull
AD = 39.024
P-Value < 0.005





Descriptive Statistics

N	N*	Mean	StDev	Median	Minimum	Maximum	Skewness	Kurtosis

```
  335      0  13.4829  108.489     1.01529     0.0909091        1370.90        11.9812       147.094
Box-Cox transformation: λ = -0.287532
```

Goodness of Fit Test

Distribution	AD	P	LRT P
Normal	115.373	<0.005	
Box-Cox Transformation	22.670	<0.005	
Lognormal	27.851	<0.005	
3-Parameter Lognormal	26.811	*	0.001
Exponential	270.653	<0.003	
2-Parameter Exponential	277.286	<0.010	0.112
Weibull	41.427	<0.010	
3-Parameter Weibull	39.024	<0.005	0.000
Smallest Extreme Value	122.999	<0.010	
Largest Extreme Value	93.428	<0.010	
Gamma	67.728	<0.005	
3-Parameter Gamma	64.935	*	0.000
Logistic	84.572	<0.005	
Loglogistic	26.410	<0.005	
3-Parameter Loglogistic	25.357	*	0.000

ML Estimates of Distribution Parameters

Distribution	Location	Shape	Scale	Threshold
Normal*	13.48288		108.48887	
Box-Cox Transformation*	0.88942		0.25790	
Lognormal*	0.58927		1.23092	
3-Parameter Lognormal	0.52174		1.29323	0.06535
Exponential			13.48288	
2-Parameter Exponential			13.43207	0.05081
Weibull		0.53945	3.63364	
3-Parameter Weibull		0.52622	3.38616	0.09081
Smallest Extreme Value	103.15020		338.72883	
Largest Extreme Value	2.59818		11.53885	
Gamma		0.33636	40.08457	
3-Parameter Gamma		0.31890	41.99428	0.09091
Logistic	3.29619		11.29919	
Loglogistic	0.41604		0.61167	
3-Parameter Loglogistic	0.35048		0.65303	0.08141

** Scale: Adjusted ML estimate*

Distribution Identification for Inpatient (CA-MMIS WPDR Data)

2-Parameter Exponential

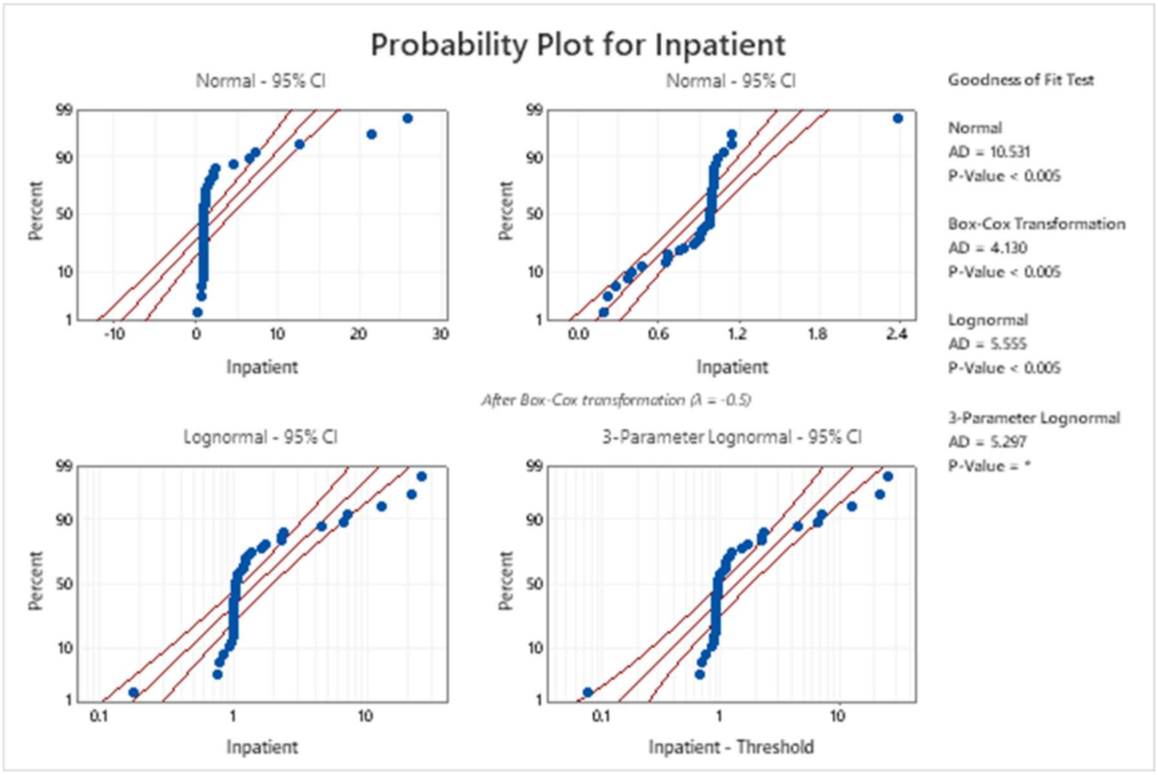
* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold parameter is assumed fixed when calculating confidence intervals.

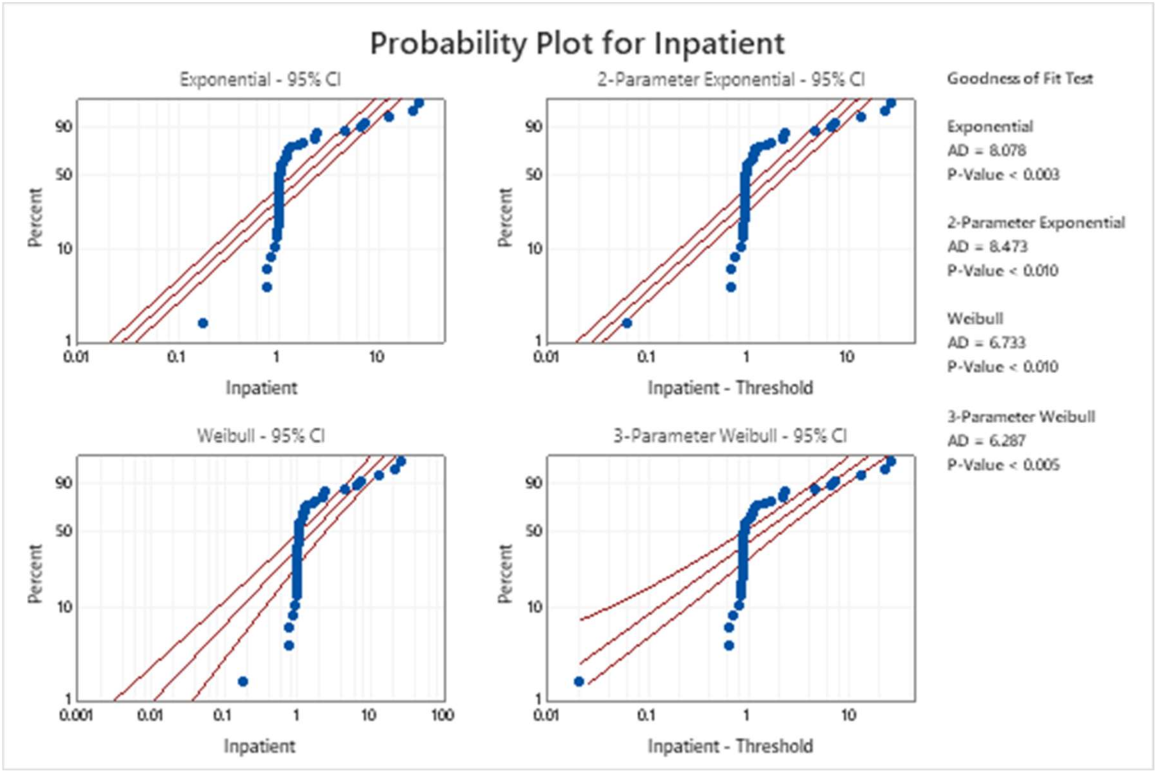
3-Parameter Weibull

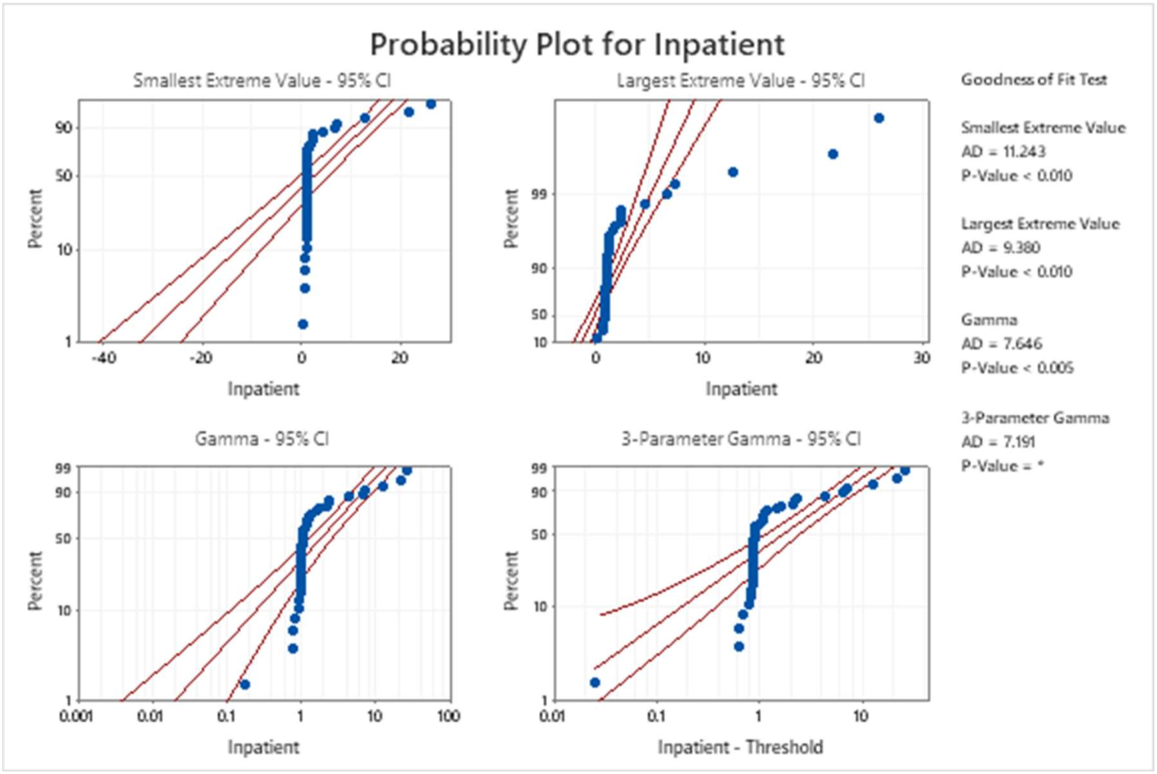
* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold parameter is assumed fixed when calculating confidence intervals.

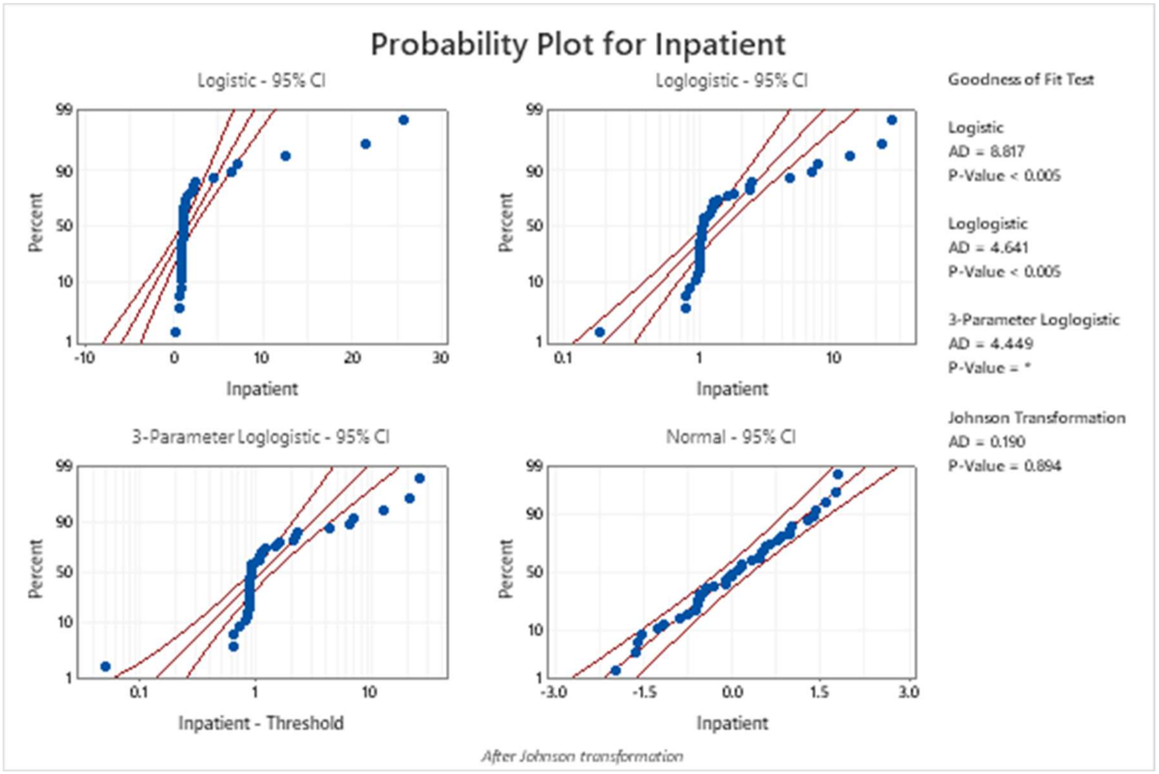
3-Parameter Gamma

* WARNING * Variance/Covariance matrix of estimated parameters does not exist. The threshold parameter is assumed fixed when calculating confidence intervals.




Probability Plot for Inpatient
Exponential - 95% CI
2-Parameter Exponential - 95% CI
Weibull - 95% CI
3-Parameter Weibull - 95% CI
Percent
Inpatient
Inpatient - Threshold
Goodness of Fit Test
Exponential
AD = 8.078
P-Value < 0.003
2-Parameter Exponential
AD = 8.473
P-Value < 0.010
Weibull
AD = 6.733
P-Value < 0.010
3-Parameter Weibull
AD = 6.287
P-Value < 0.005


Probability Plot for Inpatient
Smallest Extreme Value - 95% CI
Largest Extreme Value - 95% CI
Gamma - 95% CI
3-Parameter Gamma - 95% CI
Percent
Inpatient
Inpatient - Threshold
Goodness of Fit Test
Smallest Extreme Value
AD = 11.243
P-Value < 0.010
Largest Extreme Value
AD = 9.380
P-Value < 0.010
Gamma
AD = 7.646
P-Value < 0.005
3-Parameter Gamma
AD = 7.191
P-Value = *



Descriptive Statistics

N	N*	Mean	StDev	Median	Minimum	Maximum	Skewness	Kurtosis
44	0	2.77840	5.13097	1.04269	0.178133	25.8746	3.60498	13.0846

Box-Cox transformation: λ = -0.5

Johnson transformation function:

-0.558044 + 0.273480 × Asinh((X - 1.00096) / 0.00924874)

Goodness of Fit Test

Distribution	AD	P	LRT P
Normal	10.531	<0.005	
Box-Cox Transformation	4.130	<0.005	
Lognormal	5.555	<0.005	
3-Parameter Lognormal	5.297	*	0.169
Exponential	8.078	<0.003	
2-Parameter Exponential	8.473	<0.010	0.051
Weibull	6.733	<0.010	
3-Parameter Weibull	6.287	<0.005	0.004
Smallest Extreme Value	11.243	<0.010	
Largest Extreme Value	9.380	<0.010	
Gamma	7.646	<0.005	
3-Parameter Gamma	7.191	*	0.011
Logistic	8.817	<0.005	
Loglogistic	4.641	<0.005	
3-Parameter Loglogistic	4.449	*	0.134
Johnson Transformation	0.190	0.894	

ML Estimates of Distribution Parameters

Distribution	Location	Shape	Scale	Threshold
Normal*	2.77840		5.13097	
Box-Cox Transformation*	0.89477		0.33382	
Lognormal*	0.38792		0.91148	
3-Parameter Lognormal	0.28350		0.97898	0.10306
Exponential			2.77840	
2-Parameter Exponential			2.66074	0.11766
Weibull		0.84741	2.45644	
3-Parameter Weibull		0.79093	2.17195	0.15739
Smallest Extreme Value	5.95838		8.39066	
Largest Extreme Value	1.34502		1.69089	
Gamma		0.91970	3.02099	
3-Parameter Gamma		0.79068	3.31980	0.15350
Logistic	1.63515		1.62851	
Loglogistic	0.22854		0.40720	
3-Parameter Loglogistic	0.11605		0.45333	0.12744
Johnson Transformation*	0.04042		0.94763	

** Scale: Adjusted ML estimate*

APPENDIX E: DETAIL RESULTS OF PRINCIPLE COMPONENT ANALYSIS (PCA) FOR RATIO FEATURES FOR INPATIENT AND OUTPATIENT DATASETS

This appendix provides the detail result of PCA analysis from Minitab for ratio features in outpatient and inpatient datasets.

OUTPATIENT_LABELLEDSAMPLE_ALL

Principal Component Analysis for ratio features- outpatient

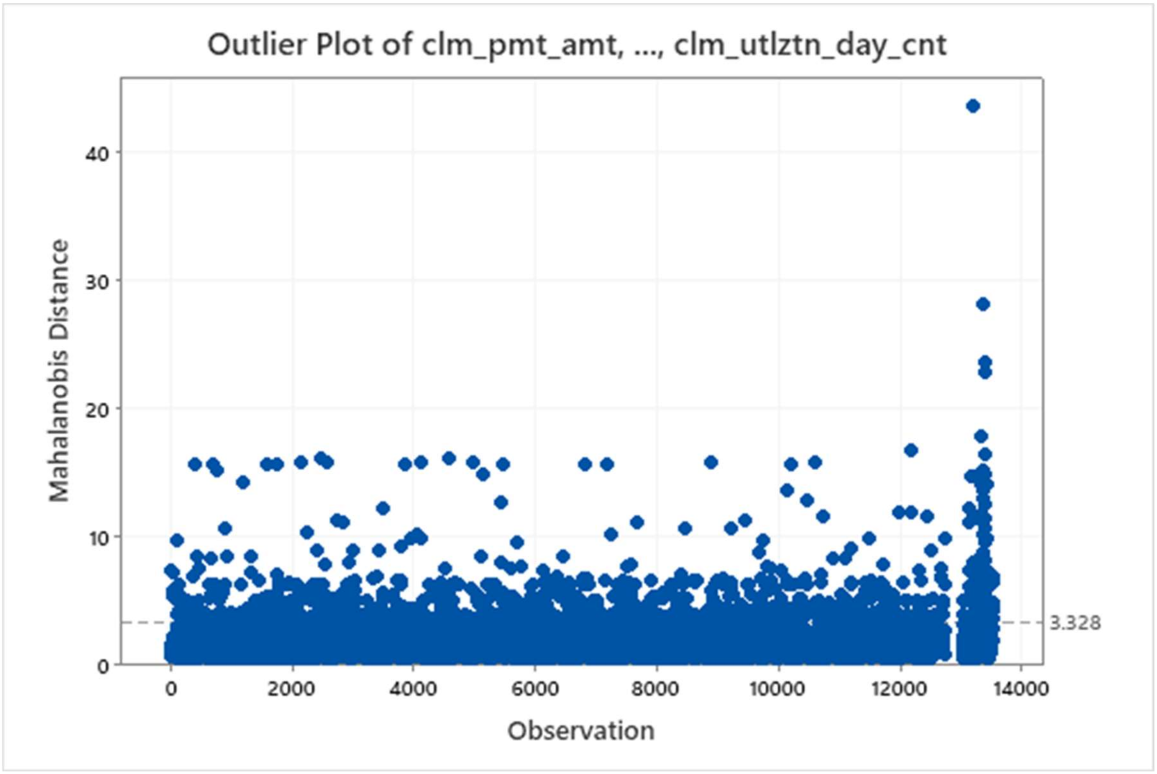
Eigenanalysis of the Correlation Matrix

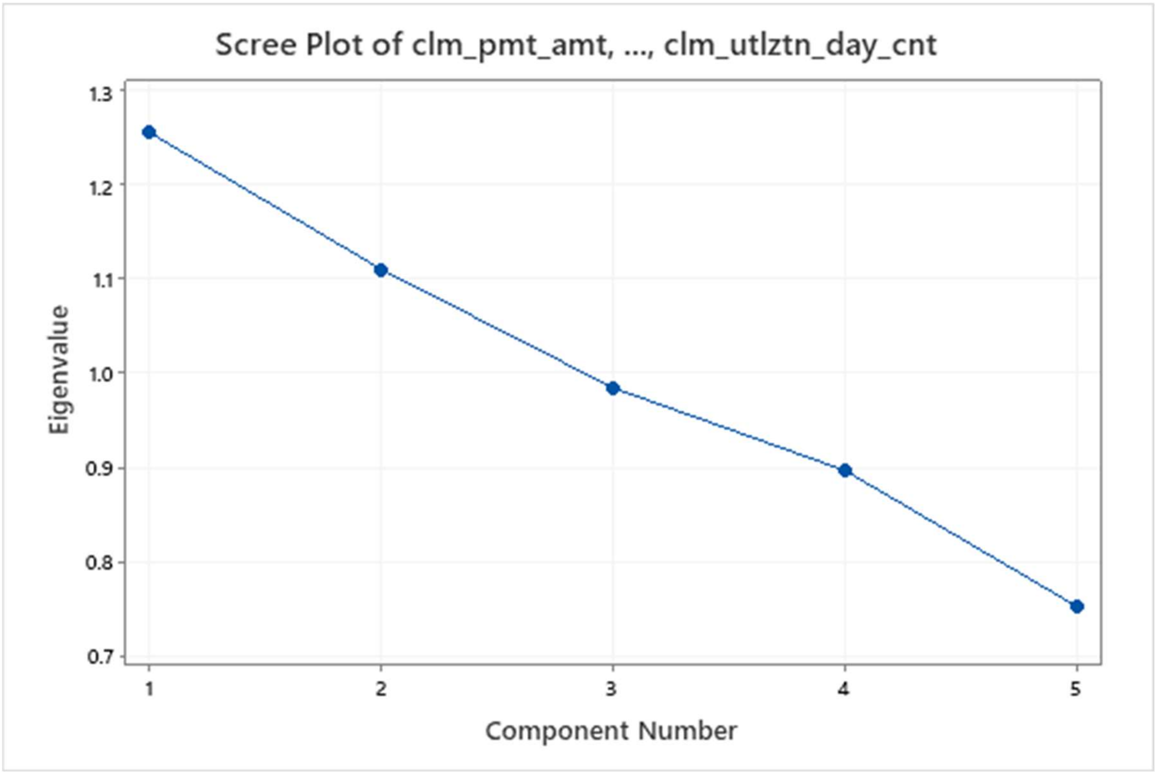
Eigenvalue	1.2556	1.1104	0.9846	0.8969	0.7525
Proportion	0.251	0.222	0.197	0.179	0.151
Cumulative	0.251	0.473	0.670	0.849	1.000

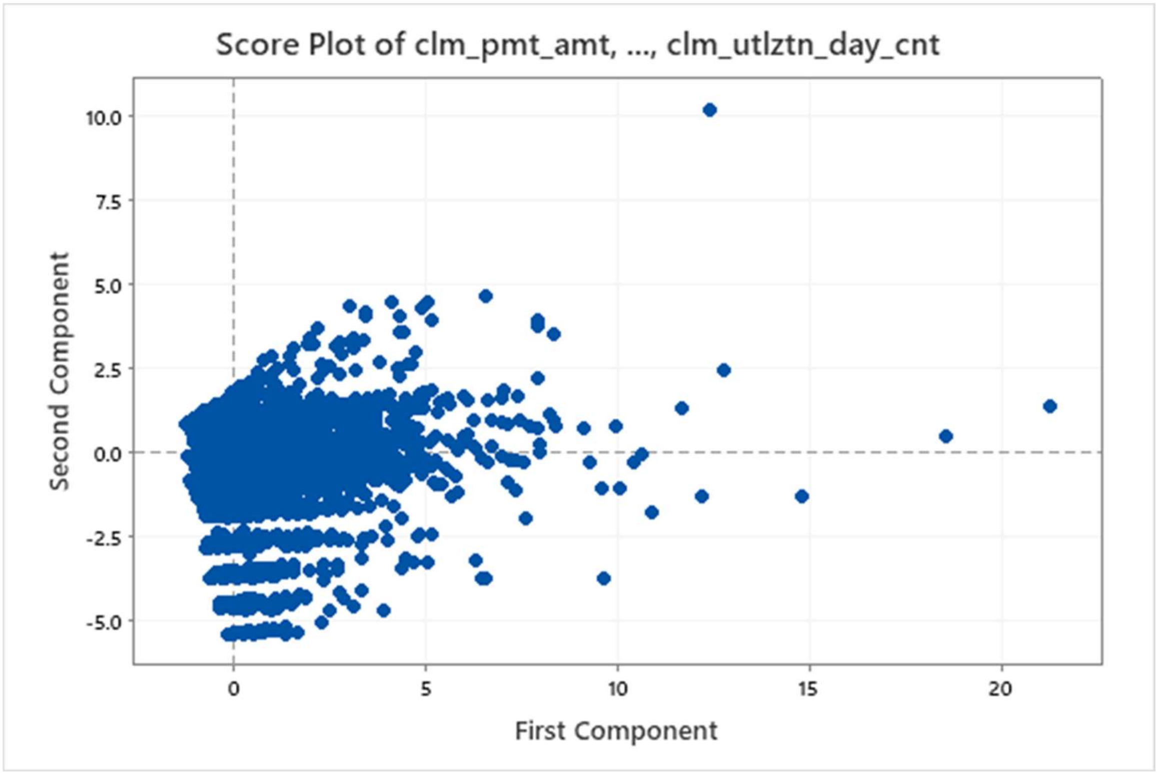
12794 cases used, 727 cases contain missing values

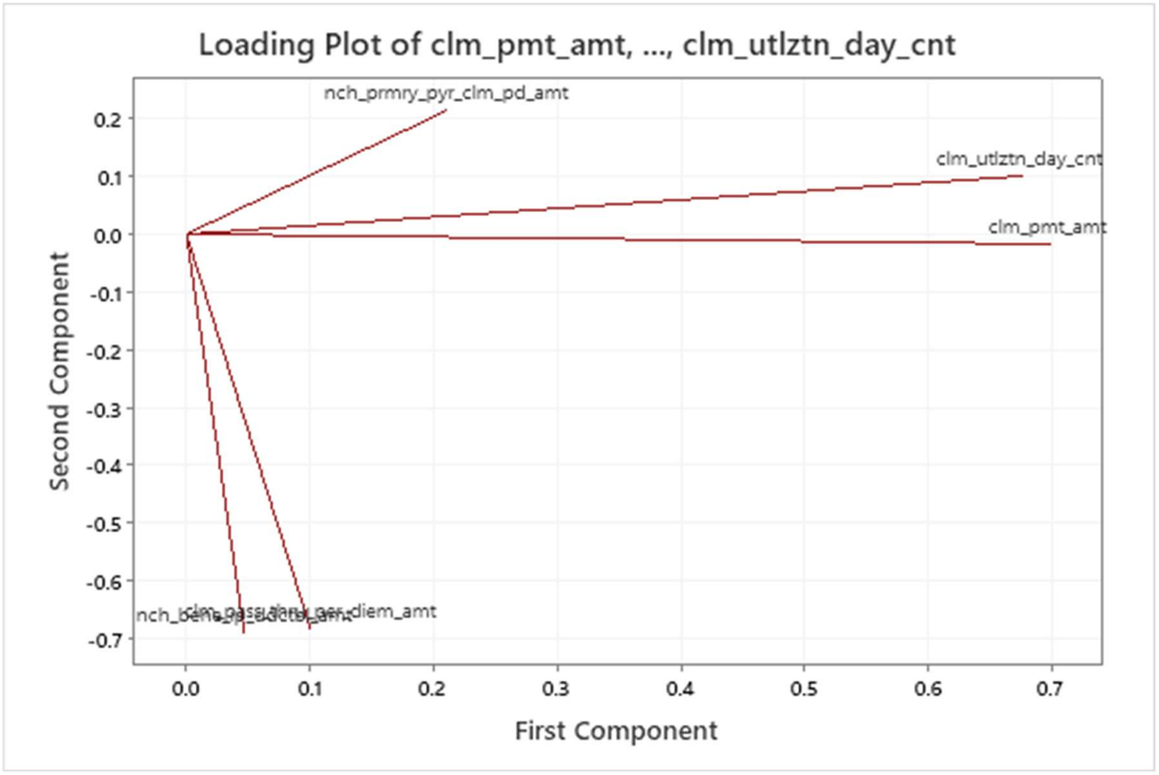
Eigenvectors

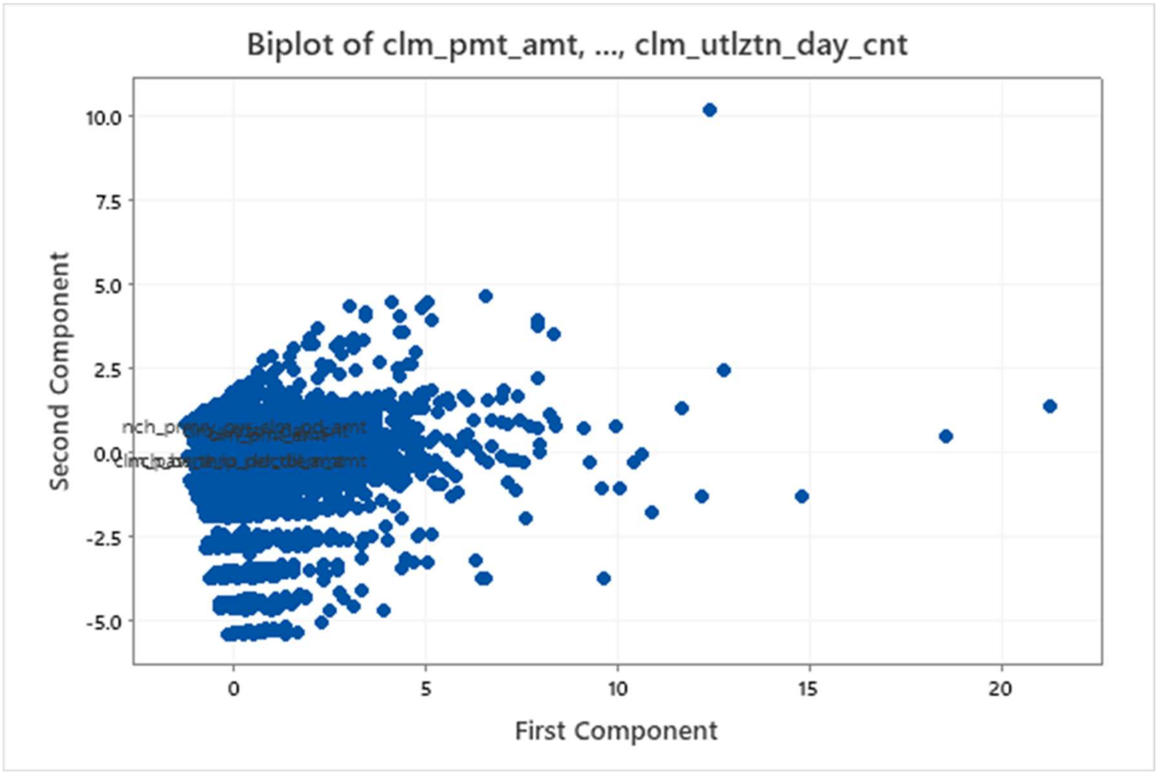
Variable	PC1	PC2	PC3	PC4	PC5
clm_pmt_amt	0.698	-0.017	0.090	-0.071	0.707
nch_prmry_pyr_clm_pd_amt	0.211	0.214	-0.944	0.116	-0.071
clm_pass_thru_per_diem_amt	0.100	-0.683	-0.204	-0.677	-0.158
nch_bene_ip_ddctbl_amt	0.047	-0.691	-0.059	0.718	0.017
clm_utlztn_day_cnt	0.676	0.100	0.236	0.088	-0.685


Outlier Plot of clm_pmt_amt, ..., clm_utlztn_day_cnt
Mahalanobis Distance
Observation
3.328
0
10
20
30
40
0
2000
4000
6000
8000
10000
12000
14000


Scree Plot of clm_pmt_amt, ..., clm_utlztn_day_cnt
Eigenvalue
Component Number
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1
2
3
4
5


Score Plot of clm_pmt_amt, ..., clm_utlztn_day_cnt
Second Component
First Component
10.0
7.5
5.0
2.5
0.0
-2.5
-5.0
0
5
10
15
20


Loading Plot of clm_pmt_amt, ..., clm_utlztn_day_cnt
nch_prmry_pyr_clm_pd_amt
clm_utlztn_day_cnt
clm_pmt_amt
clm_pass_thru_per_diem_amt
nch_bene_ip_ddctbl_amt
Second Component
First Component
0.2
0.1
0.0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7



INPATIENT_LABELLEDSAMPLE_ALL

Principal Component Analysis for ratio features- Inpatient

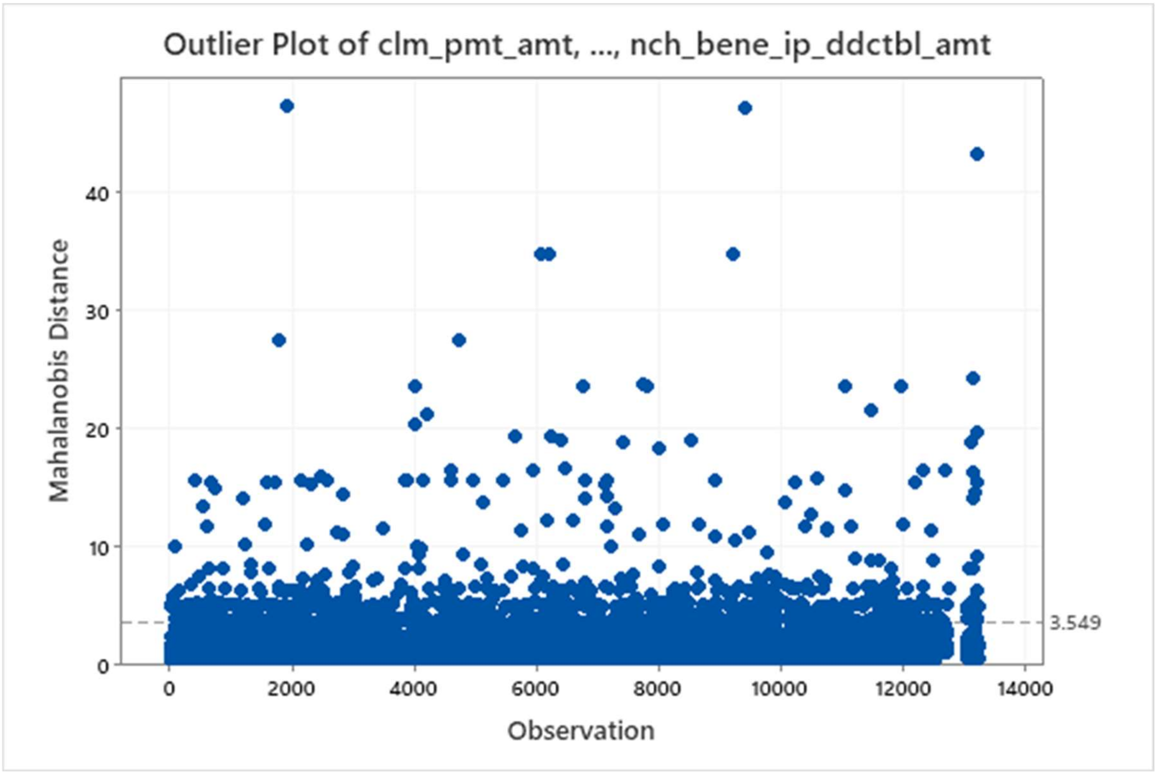
Eigenanalysis of the Correlation Matrix

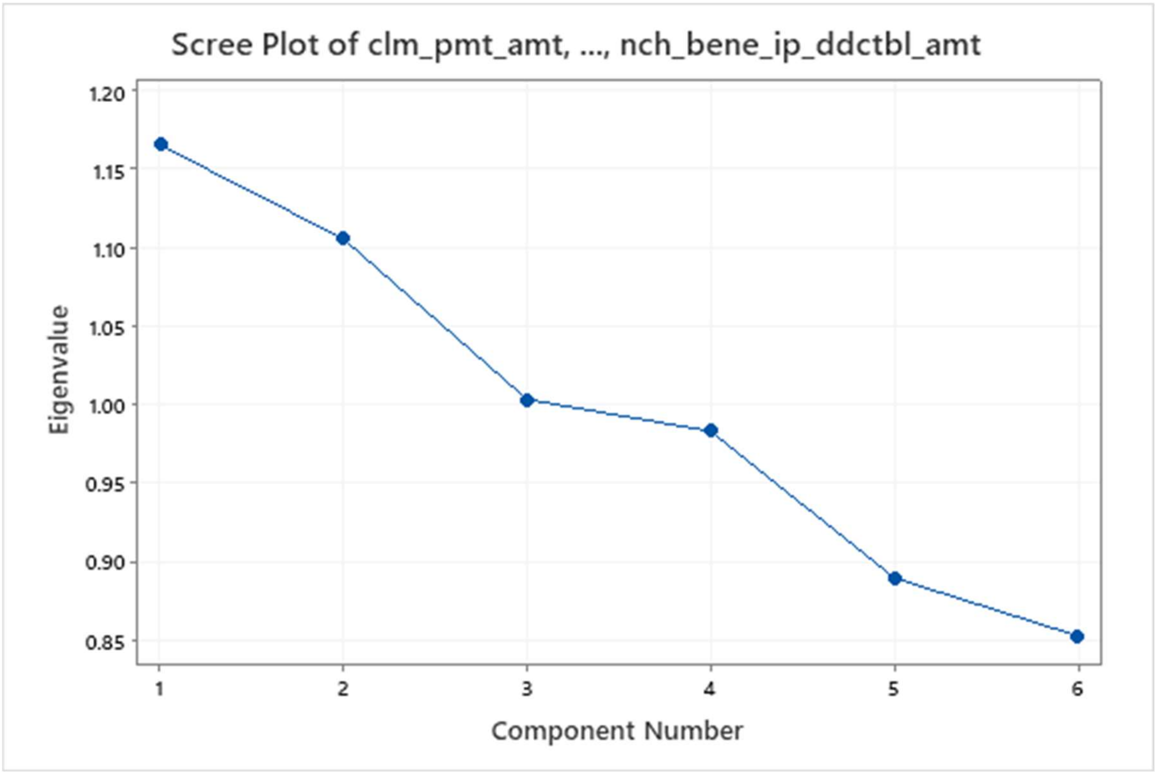
Eigenvalue	1.1655	1.1056	1.0034	0.9832	0.8896	0.8527
Proportion	0.194	0.184	0.167	0.164	0.148	0.142
Cumulative	0.194	0.379	0.546	0.710	0.858	1.000

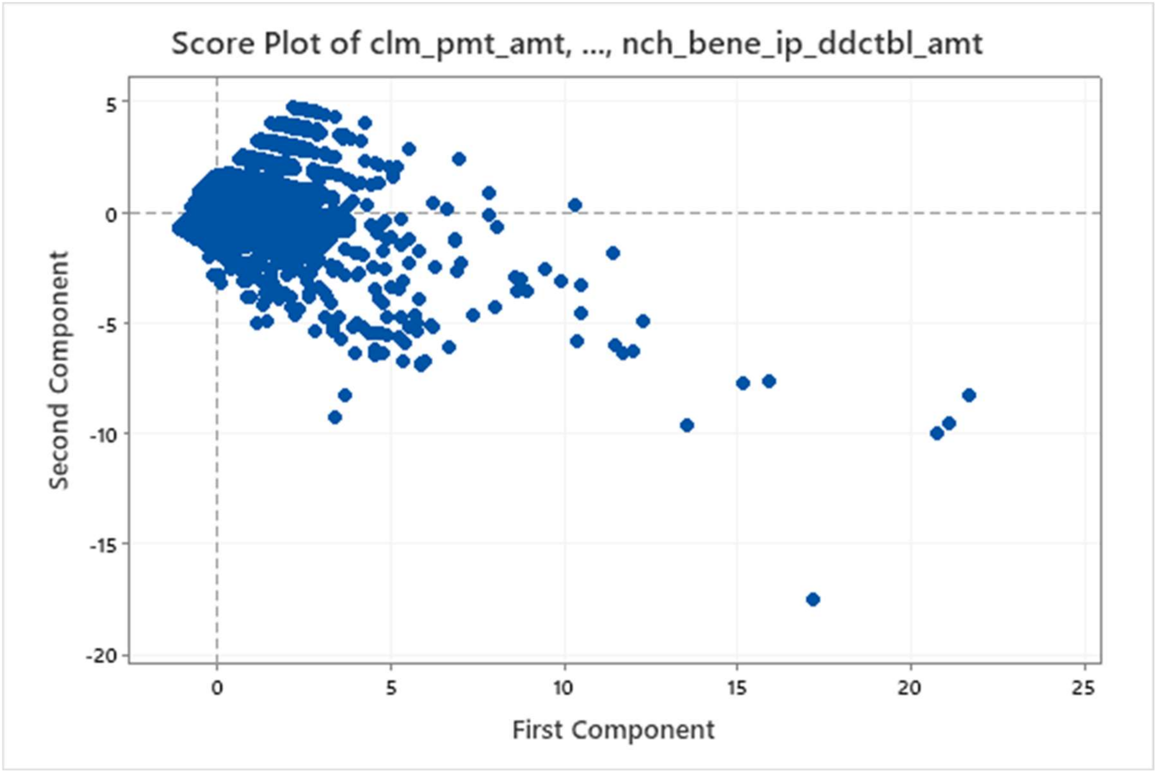
12504 cases used, 717 cases contain missing values

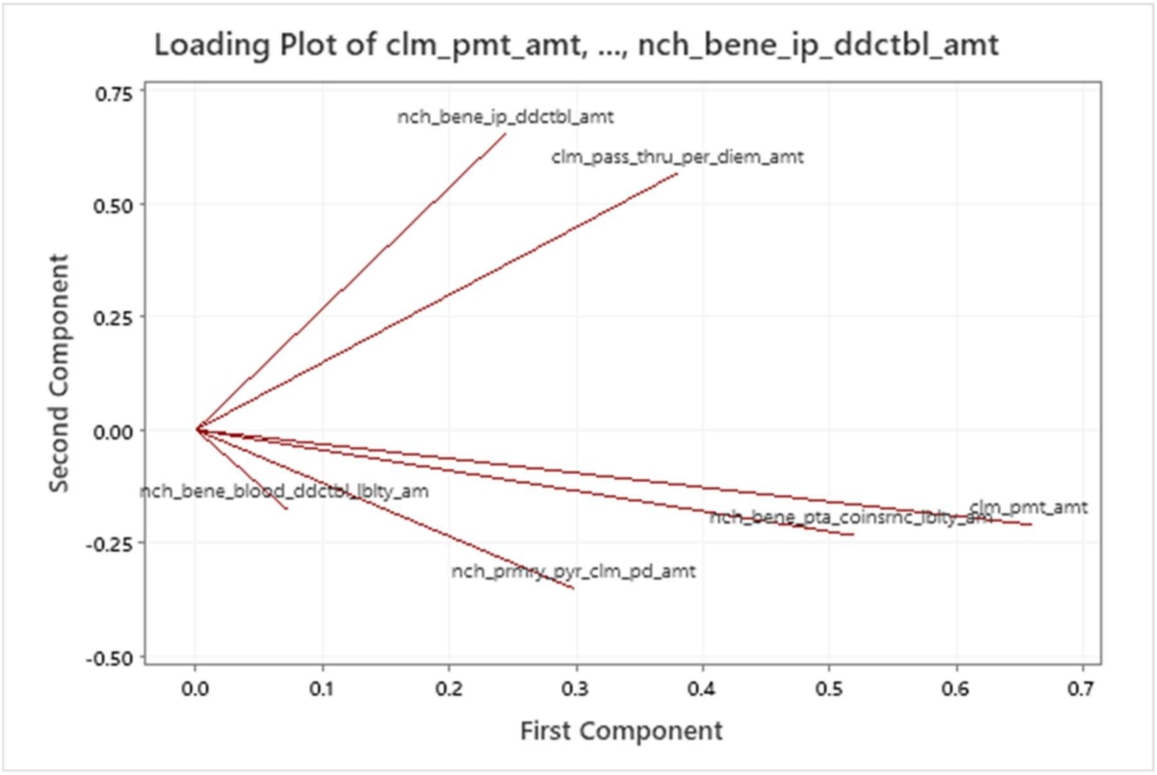
Eigenvectors

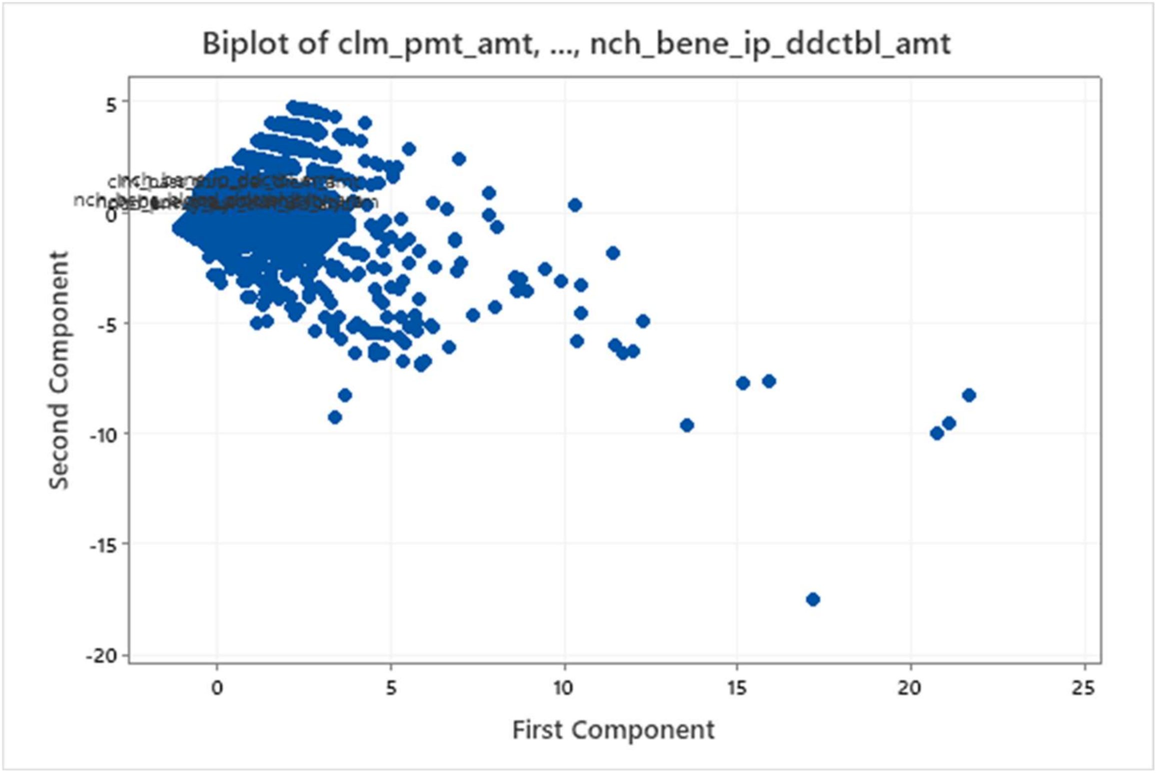
Variable	PC1	PC2	PC3	PC4	PC5	PC6
clm_pmt_amt	0.658	-0.209	-0.057	-0.007	-0.075	0.717
nch_prmry_pyr_clm_pd_amt	0.298	-0.351	0.338	-0.738	-0.019	-0.358
clm_pass_thru_per_diem_amt	0.380	0.567	-0.071	-0.110	0.710	-0.116
nch_bene_pta_coinsrnc_lblty_am	0.518	-0.233	0.059	0.612	-0.079	-0.542
nch_bene_blood_ddctbl_lblty_am	0.071	-0.174	-0.935	-0.219	-0.059	-0.198
nch_bene_ip_ddctbl_amt	0.244	0.654	-0.003	-0.143	-0.693	-0.108


Outlier Plot of clm_pmt_amt, ..., nch_bene_ip_ddctbl_amt
Mahalanobis Distance
Observation
3.549
0
10
20
30
40
0
2000
4000
6000
8000
10000
12000
14000


Scree Plot of clm_pmt_amt, ..., nch_bene_ip_ddctbl_amt
Eigenvalue
Component Number
1.20
1.15
1.10
1.05
1.00
0.95
0.90
0.85
1
2
3
4
5
6


Score Plot of clm_pmt_amt, ..., nch_bene_ip_ddctbl_amt
Second Component
First Component
5
0
-5
-10
-15
-20
0
5
10
15
20
25


Loading Plot of clm_pmt_amt, ..., nch_bene_ip_ddctbl_amt
nch_bene_ip_ddctbl_amt
clm_pass_thru_per_diem_amt
nch_bene_blood_ddctbl_lblty_am
clm_pmt_amt
nch_bene_pta_coinsrnc_lblty_am
nch_prmry_pyr_clm_pd_amt
Second Component
First Component
0.75
0.50
0.25
0.00
-0.25
-0.50
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7


Biplot of clm_pmt_amt, ..., nch_bene_ip_ddctbl_amt
Second Component
First Component
5
0
-5
-10
-15
-20
0
5
10
15
20
25

APPENDIX F: TESTING THE PERFORMANCE OF A 5-FOLD VS. 10-FOLD CROSS VALIDATION ON A DATASET

This appendix provides the detailed result performance of two algorithms on one dataset (outpatient). The result shows there is no significant difference between 5 vs. 10-fold cross-validation for our datasets.

K	Al	Run	Precision					Recall					Accuracy					F Measure					ROC					KS (Measure)	KS (P Value)
			P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4		
5	DT	1	74.50%	97.89%	37.04%	46.94%	28.96%	69.37%	95.36%	37.91%	35.38%	50.48%	88.32%	99.03%	75.28%	72.24%	80.11%	71.85%	96.61%	37.47%	40.35%	36.80%	83.14%	99.39%	60.94%	62.75%	60.93%	8.53%	0.00%
		2	66.58%	93.32%	72.44%	22.64%	38.84%	66.80%	95.15%	46.60%	39.78%	46.88%	86.70%	98.98%	77.89%	77.67%	78.97%	66.69%	94.23%	56.72%	28.86%	42.48%	79.15%	99.36%	75.85%	57.04%	63.92%	12.05%	0.00%
		3	69.19%	94.89%	55.10%	27.40%	38.90%	82.51%	95.20%	42.55%	39.28%	34.78%	90.90%	98.99%	76.14%	77.01%	73.19%	75.26%	95.05%	48.02%	32.28%	36.72%	82.76%	99.37%	68.25%	58.41%	60.33%	3.67%	0.00%
		4	73.52%	95.17%	44.68%	34.50%	41.90%	77.56%	95.18%	40.86%	33.89%	47.08%	90.45%	98.99%	76.00%	73.44%	78.96%	75.48%	95.18%	42.69%	34.20%	44.34%	84.10%	99.37%	64.26%	58.84%	65.06%	2.20%	0.00%
		5	74.09%	95.35%	50.33%	49.63%	34.07%	70.24%	95.20%	46.17%	44.26%	49.87%	88.54%	98.99%	78.33%	77.43%	79.96%	72.11%	95.27%	48.16%	46.79%	40.48%	83.12%	99.37%	67.83%	67.00%	62.75%	6.34%	0.00%
		Averag	**71.57%**	**95.32%**	**51.92%**	**36.23%**	**36.53%**	**73.29%**	**95.22%**	**42.82%**	**38.52%**	**45.82%**	**88.98%**	**99.00%**	**76.73%**	**75.56%**	**78.24%**	**72.28%**	**95.27%**	**46.61%**	**36.50%**	**40.17%**	**82.45%**	**99.37%**	**67.42%**	**60.81%**	**62.60%**	**6.56%**	**0.00%**
10	DT	1	75.53%	94.80%	56.68%	35.11%	46.62%	73.91%	95.80%	41.48%	47.84%	55.85%	89.77%	99.11%	75.35%	79.36%	81.95%	74.71%	95.30%	47.91%	40.50%	50.82%	84.43%	99.41%	68.35%	62.77%	68.70%	8.62%	0.00%
		2	78.32%	93.78%	34.44%	29.33%	45.44%	68.18%	95.80%	43.39%	32.67%	40.49%	88.35%	99.01%	77.90%	73.78%	75.73%	72.89%	94.78%	38.40%	30.91%	42.82%	84.59%	99.15%	61.60%	57.11%	64.37%	3.73%	0.00%
		3	73.60%	91.30%	44.07%	45.02%	34.18%	74.77%	95.56%	41.33%	41.60%	40.08%	89.75%	98.49%	76.30%	76.37%	76.62%	74.18%	93.38%	42.65%	43.24%	36.90%	83.69%	97.91%	64.21%	64.61%	60.71%	2.94%	0.00%
		4	72.01%	95.25%	48.49%	39.44%	38.16%	69.04%	95.13%	39.19%	44.80%	48.42%	87.94%	98.98%	74.65%	78.17%	79.50%	70.49%	95.19%	43.35%	41.95%	42.68%	81.97%	99.36%	64.84%	63.65%	64.00%	6.63%	0.00%
		5	66.16%	94.68%	38.25%	24.83%	36.52%	77.48%	95.43%	28.25%	24.35%	49.41%	89.39%	98.78%	68.22%	69.54%	79.83%	71.38%	95.05%	32.50%	24.59%	42.00%	80.68%	98.72%	56.98%	52.77%	63.58%	5.22%	0.00%
		6	69.62%	96.35%	51.20%	32.97%	33.24%	77.84%	95.12%	36.25%	36.67%	44.76%	89.96%	98.97%	72.23%	75.21%	78.44%	73.50%	95.73%	42.45%	34.72%	38.15%	82.33%	99.36%	64.34%	59.37%	61.49%	7.17%	0.00%
		7	71.98%	95.64%	69.41%	17.28%	30.90%	62.35%	95.39%	40.62%	36.19%	50.52%	85.70%	99.02%	73.59%	77.36%	80.13%	66.82%	95.51%	51.25%	23.39%	38.35%	80.56%	99.37%	72.02%	54.83%	61.67%	18.22%	0.00%
		8	64.42%	95.03%	59.16%	50.63%	31.09%	69.19%	95.04%	51.93%	42.50%	45.32%	87.15%	98.96%	80.88%	76.42%	78.72%	66.72%	95.03%	55.31%	46.21%	36.88%	78.63%	99.35%	72.73%	66.75%	60.86%	6.28%	0.00%
		9	72.17%	95.15%	60.88%	25.06%	35.74%	63.75%	94.71%	40.72%	37.80%	54.66%	86.23%	98.88%	74.45%	76.76%	81.22%	67.70%	94.93%	48.80%	30.14%	43.22%	80.96%	99.30%	69.36%	57.37%	64.16%	13.66%	0.00%
		10	70.50%	93.46%	65.08%	33.81%	21.62%	71.45%	95.41%	44.22%	34.97%	39.23%	88.47%	98.59%	76.59%	74.19%	77.63%	70.97%	94.42%	52.66%	34.38%	27.88%	81.73%	98.23%	72.28%	59.04%	56.62%	9.64%	0.00%
		Averag	**71.43%**	**94.54%**	**52.77%**	**33.35%**	**35.35%**	**70.80%**	**95.34%**	**40.74%**	**37.94%**	**46.87%**	**88.27%**	**98.88%**	**75.02%**	**75.72%**	**78.98%**	**70.94%**	**94.93%**	**45.53%**	**35.00%**	**39.97%**	**81.96%**	**99.02%**	**66.67%**	**59.83%**	**62.62%**	**8.21%**	**0.00%**
5 vs. 10-fold Diff.			0.14%	0.78%	-0.85%	2.88%	1.18%	2.50%	-0.12%	2.08%	0.58%	-1.06%	0.71%	0.12%	1.71%	-0.16%	-0.74%	1.34%	0.33%	1.08%	1.49%	0.20%	0.50%	0.35%	0.75%	0.98%	-0.02%	-1.65%	0.00%
5 vs. 10-fold Average Diff.			**0.83%**					**0.80%**					**0.33%**					**0.89%**					**0.51%**						

K	Al	Run	Precision					Recall					Accuracy					F Measure					ROC					KS (Measure)	KS (P Value)
			P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4	P0	P1	P2	P3	P4		
5	RF	1	81.69%	96.17%	46.01%	21.12%	33.44%	64.58%	95.77%	41.35%	29.72%	38.26%	87.38%	99.00%	76.15%	74.24%	75.90%	72.14%	95.97%	43.55%	24.70%	35.69%	85.25%	99.15%	64.85%	54.32%	59.97%	8.31%	0.00%
		2	82.54%	95.45%	40.33%	18.56%	42.23%	67.88%	95.86%	43.96%	32.06%	33.75%	88.70%	99.01%	77.78%	75.85%	71.87%	74.50%	95.65%	42.07%	23.51%	37.52%	86.39%	99.14%	63.74%	54.36%	60.75%	5.05%	0.00%
		3	80.85%	96.51%	36.17%	14.14%	33.23%	60.03%	95.29%	33.38%	21.81%	38.07%	85.40%	98.99%	72.80%	72.69%	75.83%	68.90%	95.90%	34.72%	17.16%	35.48%	83.69%	99.34%	59.06%	50.73%	59.86%	9.57%	0.00%
		4	80.91%	95.85%	58.74%	16.07%	33.48%	60.85%	95.57%	47.18%	32.70%	37.68%	85.77%	99.05%	78.60%	76.60%	75.62%	69.46%	95.71%	52.33%	21.55%	35.46%	83.95%	99.37%	71.15%	53.90%	59.82%	12.40%	0.00%
		5	81.81%	94.84%	52.08%	14.18%	42.33%	60.24%	95.51%	45.24%	33.47%	41.07%	85.56%	98.87%	77.81%	77.20%	76.32%	69.39%	95.18%	48.42%	19.92%	41.69%	84.16%	98.92%	68.16%	53.57%	63.57%	10.91%	0.00%
		Averag	**81.56%**	**95.76%**	**46.66%**	**16.81%**	**36.94%**	**62.72%**	**95.60%**	**42.22%**	**29.95%**	**37.77%**	**86.56%**	**98.99%**	**76.63%**	**75.31%**	**75.11%**	**70.88%**	**95.68%**	**44.22%**	**21.37%**	**37.17%**	**84.69%**	**99.18%**	**65.39%**	**53.38%**	**60.80%**	**9.25%**	**0.00%**
10	RF	1	82.32%	95.40%	40.49%	20.64%	34.50%	58.43%	96.12%	42.12%	31.56%	36.47%	84.75%	99.00%	76.97%	75.18%	74.88%	68.34%	95.76%	41.29%	24.96%	35.46%	83.84%	98.99%	63.29%	54.73%	59.74%	8.78%	0.00%
		2	81.70%	95.91%	52.87%	23.55%	39.99%	65.38%	95.82%	55.47%	30.16%	40.80%	87.69%	98.99%	82.09%	73.80%	76.39%	72.64%	95.86%	54.14%	26.45%	40.39%	85.44%	99.11%	71.13%	54.96%	62.74%	5.72%	0.00%
		3	83.22%	95.57%	40.82%	18.14%	34.44%	62.60%	95.80%	39.06%	28.20%	36.14%	86.70%	98.86%	75.43%	74.39%	74.72%	71.45%	95.69%	39.92%	22.08%	35.27%	85.39%	98.76%	62.45%	53.30%	59.61%	8.07%	0.00%
		4	82.29%	97.05%	55.80%	23.74%	27.12%	63.80%	95.52%	39.75%	33.36%	49.48%	87.12%	99.05%	74.24%	75.26%	79.89%	71.88%	96.28%	46.43%	27.74%	35.04%	85.31%	99.38%	67.33%	55.94%	60.10%	14.80%	0.00%
		5	82.45%	96.19%	42.72%	26.28%	37.61%	63.38%	94.82%	39.90%	37.96%	42.33%	86.96%	98.78%	75.67%	76.67%	77.28%	71.67%	95.50%	41.26%	31.06%	39.83%	85.27%	98.98%	63.32%	57.77%	62.40%	8.39%	0.00%
		6	81.24%	96.88%	33.21%	15.84%	40.98%	63.34%	95.69%	28.36%	26.72%	45.10%	86.85%	99.09%	69.86%	74.48%	78.22%	71.19%	96.28%	30.59%	19.89%	42.94%	84.74%	99.41%	56.12%	52.49%	64.25%	9.97%	0.00%
		7	81.35%	96.59%	52.73%	14.07%	30.80%	62.15%	95.44%	41.45%	21.26%	43.36%	86.36%	99.02%	75.65%	72.39%	78.11%	70.47%	96.01%	46.41%	16.94%	36.02%	84.48%	99.35%	67.05%	50.52%	60.37%	12.55%	0.00%
		8	81.24%	96.52%	38.38%	28.48%	42.90%	64.56%	96.10%	41.12%	32.51%	48.03%	87.33%	99.16%	76.68%	73.87%	79.30%	71.95%	96.31%	39.70%	30.36%	45.32%	85.04%	99.43%	62.32%	56.85%	65.65%	5.95%	0.00%
		9	82.29%	96.38%	53.08%	18.39%	41.12%	62.34%	95.42%	47.93%	32.57%	42.80%	86.52%	99.02%	79.08%	76.06%	77.23%	70.94%	95.90%	50.38%	23.51%	41.94%	84.93%	99.34%	69.33%	54.44%	63.69%	9.49%	0.00%
		10	81.06%	95.75%	42.93%	9.57%	34.33%	58.91%	95.64%	37.80%	17.80%	37.40%	84.90%	98.85%	74.46%	73.08%	75.37%	68.23%	95.70%	40.20%	12.45%	35.80%	83.46%	98.81%	62.63%	49.26%	59.98%	10.89%	0.00%
		Averag	**81.92%**	**96.23%**	**45.30%**	**19.87%**	**36.38%**	**62.49%**	**95.64%**	**41.29%**	**29.21%**	**42.19%**	**86.52%**	**98.98%**	**76.01%**	**74.52%**	**77.14%**	**70.87%**	**95.93%**	**43.03%**	**23.54%**	**38.80%**	**84.79%**	**99.16%**	**64.50%**	**54.03%**	**61.85%**	**9.46%**	**0.00%**
5 vs. 10-fold Diff.			-0.36%	-0.46%	1.36%	-3.06%	0.56%	0.23%	-0.03%	0.93%	0.74%	-4.43%	0.05%	0.00%	0.61%	0.80%	-2.03%	0.00%	-0.25%	1.19%	-2.18%	-1.63%	-0.10%	0.03%	0.89%	-0.65%	-1.06%	-0.21%	0.00%
5 vs. 10-fold Average Diff.			**-0.39%**					**-0.51%**					**-0.11%**					**-0.57%**					**-0.18%**						

APPENDIX G: DETAIL RESULT OF FEATURE IMPORTANCE CALCULATIONS

This appendix presents Mean Decrease in Impurity (MDI) is used to calculate the feature importance score (Gini importance) presented here.

Outpatient Independent Variable	DTE	DTG	GB	RFE	RFG
clm_pmt_amt	0.764439	0.591168292	0.195357038	0.251641813	0.14127222
nch_prmry_pyr_clm_pd_amt	0.003081	0.004518558	0.005640962	0.004061057	0.003648207
at_physn_npi	0.037157	0.02053663	0.161360039	0.125938191	0.146309041
op_physn_npi	0.004684	0.029549666	0.025856541	0.026117104	0.030777697
ot_physn_npi	0.015246	0.041652799	0.05563106	0.052157129	0.058011679
icd9_dgns_cd_1	0.046217	0.102061247	0.138662911	0.122238481	0.140702307
icd9_dgns_cd_2	0.018545	0.0201475	0.07084581	0.070175518	0.080594996
icd9_dgns_cd_3	0.02204	0.036491813	0.046828383	0.048401513	0.055770272
icd9_prcdr_cd_1	0	0	0	3.65541E-06	2.21377E-05
nch_bene_ptb_ddctbl_amt	0.011737	0.000742858	0.010755506	0.006728204	0.009395545
nch_bene_ptb_coinsrnc_amt	0.015625	0.049211418	0.051435501	0.052257289	0.057007891
admtng_icd9_dgns_cd	0.020557	0.015539273	0.036952278	0.037077828	0.043708748
hcpcs_cd_1	0.019915	0.020086936	0.073544404	0.073292543	0.082820058
hcpcs_cd_2	0.011368	0.036739697	0.064096762	0.060626544	0.069250005
hcpcs_cd_3	0.007935	0.028615234	0.042537076	0.044039248	0.051712673
YEAR	0.001453	0.002938079	0.020495729	0.025243882	0.028996524

Inpatient Independent Variable	DTE	DTG	GB	RFE	RFG
clm_pmt_amt	0.484431	0.341661	0.292049	0.374654	0.307621
nch_prmry_pyr_clm_pd_amt	0.337463	0.524377	0.336782	0.082812	0.167485
at_physn_npi	0.024932	0.004257	0.054808	0.07008	0.06667
op_physn_npi	0.011089	0.002188	0.027201	0.043157	0.043952
ot_physn_npi	0	0	0.011013	0.012117	0.011579
admtng_icd9_dgns_cd	0.004301	0.000556	0.03304	0.058085	0.054768
clm_pass_thru_per_diem_amt	0	0.005087	0.015935	0.020719	0.022672
nch_bene_ip_ddctbl_amt	0	0	0.007959	0.013739	0.014329
nch_bene_pta_coinsrnc_lblty_am	0	0.00065	0.004432	0.004664	0.002334
nch_bene_blood_ddctbl_lblty_am	0	0.005011	0.003224	0.001734	0.000257
clm_utlztn_day_cnt	0.007286	0	0.014681	0.03827	0.037108
clm_drg_cd	0.100582	0.090917	0.07945	0.09042	0.088241
icd9_dgns_cd_1	0.009922	0.010403	0.040799	0.058452	0.056347
icd9_dgns_cd_2	0.009222	0.004184	0.037543	0.057865	0.058257
icd9_dgns_cd_3	0.010772	0.00612	0.032677	0.060532	0.056032
hcpcs_cd_1	0	0	0	0	0
hcpcs_cd_2	0	0	0	0	0
hcpcs_cd_3	0	0	0	0	0
YEAR	0	0.004592	0.008407	0.012701	0.012347

APPENDIX H: DETAIL RESULT OF PERFORMANCE METRICS FOR TESTED ALGORITHMS

This appendix presents performance metrics measured across all 5-fold runs for each algorithm in both with and without oversampling configuration.

Performance Metrics for all Algorithms on Outpatient Datasets

Algorithm	**Run**	**FP**	**TP**	**FN**	**TN**	**Performance Metrics for Target Group (P1)**			
						Recall	**F1-Score**	**Precision**	**Accuracy**
DTG	1	15	17	290	22452	5.54%	10.03%	53.13%	98.66%
	2	26	17	289	22441	5.56%	9.74%	39.53%	98.62%
	3	37	23	284	22430	7.49%	12.53%	38.33%	98.59%
	4	18	21	285	22449	6.86%	12.17%	53.85%	98.67%
	5	31	18	288	22436	5.88%	10.14%	36.73%	98.60%
Average						**6.27%**	**10.92%**	**44.31%**	**98.63%**
DTG (OS)	1	3365	114	193	19102	37.13%	6.02%	3.28%	84.38%
	2	3337	138	168	19130	45.10%	7.30%	3.97%	84.61%
	3	3219	127	180	19248	41.37%	6.95%	3.80%	85.08%
	4	2981	114	192	19486	37.25%	6.70%	3.68%	86.07%
	5	3980	154	152	18487	50.33%	6.94%	3.73%	81.86%
Average						**42.24%**	**6.78%**	**3.69%**	**84.40%**
DTE	1	21	13	294	22446	4.23%	7.62%	38.24%	98.62%
	2	13	21	285	22454	6.86%	12.35%	61.76%	98.69%
	3	13	17	290	22454	5.54%	10.09%	56.67%	98.67%
	4	23	21	285	22444	6.86%	12.00%	47.73%	98.65%
	5	19	15	291	22448	4.90%	8.82%	44.12%	98.64%
Average						**5.68%**	**10.18%**	**49.70%**	**98.65%**
DTE (OS)	1	3381	143	164	19086	46.58%	7.47%	4.06%	84.43%
	2	3376	152	154	19091	49.67%	7.93%	4.31%	84.50%
	3	3922	146	161	18545	47.56%	6.67%	3.59%	82.07%
	4	3518	152	154	18949	49.67%	7.65%	4.14%	83.88%
	5	3641	127	179	18826	41.50%	6.23%	3.37%	83.23%
Average						**47.00%**	**7.19%**	**3.89%**	**83.62%**
RFG	1	2	11	296	22465	3.58%	6.88%	84.62%	98.69%
	2	0	10	296	22467	3.27%	6.33%	100.00%	98.70%
	3	0	7	300	22467	2.28%	4.46%	100.00%	98.68%

Algorithm	Run	FP	TP	FN	TN	Performance Metrics for Target Group (P1)			
						Recall	F1-Score	Precision	Accuracy
	4	0	7	299	22467	2.29%	4.47%	100.00%	98.69%
	5	2	7	299	22465	2.29%	4.44%	77.78%	98.68%
Average						**2.74%**	**5.32%**	**92.48%**	**98.69%**
RFG (OS)	1	430	45	262	22037	14.66%	11.51%	9.47%	96.96%
	2	436	32	274	22031	10.46%	8.27%	6.84%	96.88%
	3	436	32	275	22031	10.42%	8.26%	6.84%	96.88%
	4	406	35	271	22061	11.44%	9.37%	7.94%	97.03%
	5	428	25	281	22039	8.17%	6.59%	5.52%	96.89%
Average						**11.03%**	**8.80%**	**7.32%**	**96.93%**
RFE	1	1	9	298	22466	2.93%	5.68%	90.00%	98.69%
	2	1	11	295	22466	3.59%	6.92%	91.67%	98.70%
	3	0	5	302	22467	1.63%	3.21%	100.00%	98.67%
	4	1	6	300	22466	1.96%	3.83%	85.71%	98.68%
	5	2	9	297	22465	2.94%	5.68%	81.82%	98.69%
Average						**2.61%**	**5.06%**	**89.84%**	**98.69%**
RFE (OS)	1	420	38	269	22047	12.38%	9.93%	8.30%	96.97%
	2	459	39	267	22008	12.75%	9.70%	7.83%	96.81%
	3	439	28	279	22028	9.12%	7.24%	6.00%	96.85%
	4	445	31	275	22022	10.13%	7.93%	6.51%	96.84%
	5	393	39	267	22074	12.75%	10.57%	9.03%	97.10%
Average						**11.42%**	**9.07%**	**7.53%**	**96.91%**
NB	1	1149	159	148	21318	51.79%	19.69%	12.16%	94.30%
	2	1190	166	140	21277	54.25%	19.98%	12.24%	94.16%
	3	1192	152	155	21275	49.51%	18.41%	11.31%	94.09%
	4	1087	154	152	21380	50.33%	19.91%	12.41%	94.56%
	5	1182	172	134	21285	56.21%	20.72%	12.70%	94.22%
Average						**52.42%**	**19.74%**	**12.16%**	**94.27%**
NB (OS)	1	7031	152	155	15436	49.51%	4.06%	2.12%	68.45%
	2	7162	175	131	15305	57.19%	4.58%	2.39%	67.98%
	3	7151	155	152	15316	50.49%	4.07%	2.12%	67.93%
	4	7165	162	144	15302	52.94%	4.24%	2.21%	67.90%
	5	7049	157	149	15418	51.31%	4.18%	2.18%	68.39%
Average						**52.29%**	**4.23%**	**2.20%**	**68.13%**
kNN	1	91	2	305	22376	0.65%	1.00%	2.15%	98.26%
	2	89	0	306	22378	0.00%	0.00%	0.00%	98.27%
	3	86	1	306	22381	0.33%	0.51%	1.15%	98.28%
	4	86	2	304	22381	0.65%	1.02%	2.27%	98.29%
	5	98	2	304	22369	0.65%	0.99%	2.00%	98.23%
Average						**0.46%**	**0.70%**	**1.51%**	**98.27%**
kNN (OS)	1	3652	60	247	18815	19.54%	2.99%	1.62%	82.88%
	2	3524	45	261	18943	14.71%	2.32%	1.26%	83.38%
	3	3408	46	261	19059	14.98%	2.45%	1.33%	83.89%
	4	3568	49	257	18899	16.01%	2.50%	1.35%	83.20%

Algorithm	Run	FP	TP	FN	TN	Performance Metrics for Target Group (P1)			
						Recall	F1-Score	Precision	Accuracy
	5	3558	51	255	18909	16.67%	2.61%	1.41%	83.26%
Average						**16.38%**	**2.57%**	**1.40%**	**83.32%**
LR		18147	250	57	4320	81.43%	2.67%	1.36%	20.07%
		15747	219	87	6720	71.57%	2.69%	1.37%	30.47%
		9314	140	167	13153	45.60%	2.87%	1.48%	58.37%
		20394	281	25	2073	91.83%	2.68%	1.36%	10.34%
		20619	280	26	1848	91.50%	2.64%	1.34%	9.34%
Average						**76.39%**	**2.71%**	**1.38%**	**25.72%**
LR (OS)		20273	285	22	2194	92.83%	2.73%	1.39%	10.89%
		9584	130	176	12883	42.48%	2.59%	1.34%	57.14%
		19999	271	36	2468	88.27%	2.63%	1.34%	12.03%
		20063	280	26	2404	91.50%	2.71%	1.38%	11.79%
		16940	233	73	5527	76.14%	2.67%	1.36%	25.29%
Average						**78.25%**	**2.67%**	**1.36%**	**23.43%**
NN		1	0	307	22466	0.00%	0.00%	0.00%	98.65%
		0	0	306	22467	0.00%	0.00%	0.00%	98.66%
		749	12	295	21718	3.91%	2.25%	1.58%	95.42%
		0	0	306	22467	0.00%	0.00%	0.00%	98.66%
		0	0	306	22467	0.00%	0.00%	0.00%	98.66%
Average						**0.78%**	**0.45%**	**0.32%**	**98.01%**
NN (OS)		3123	41	266	19344	13.36%	2.36%	1.30%	85.12%
		129	2	304	22338	0.65%	0.92%	1.53%	98.10%
		21071	297	10	1396	96.74%	2.74%	1.39%	7.43%
		8298	99	207	14169	32.35%	2.28%	1.18%	62.65%
		8603	121	185	13864	39.54%	2.68%	1.39%	61.41%
Average						**36.53%**	**2.19%**	**1.36%**	**62.94%**
DA		0	0	307	22467	0.00%	0.00%	0.00%	98.65%
		0	1	305	22467	0.33%	0.65%	100.00%	98.66%
		0	0	307	22467	0.00%	0.00%	0.00%	98.65%
		0	0	306	22467	0.00%	0.00%	0.00%	98.66%
		0	2	304	22467	0.65%	1.30%	100.00%	98.67%
Average						**0.20%**	**0.39%**	**40.00%**	**98.66%**
DA (OS)		10544	154	153	11923	50.16%	2.80%	1.44%	53.03%
		10486	152	154	11981	49.67%	2.78%	1.43%	53.28%
		10387	140	167	12080	45.60%	2.58%	1.33%	53.66%
		10269	146	160	12198	47.71%	2.72%	1.40%	54.20%
		10023	152	154	12444	49.67%	2.90%	1.49%	55.31%
Average						**48.56%**	**2.76%**	**1.42%**	**53.90%**
GB		144	28	279	22323	9.12%	11.69%	16.28%	98.14%
		156	43	263	22311	14.05%	17.03%	21.61%	98.16%
		159	34	273	22308	11.07%	13.60%	17.62%	98.10%
		145	40	266	22322	13.07%	16.29%	21.62%	98.20%
		157	28	278	22310	9.15%	11.41%	15.14%	98.09%

<table>
<tr><th rowspan="2">Algorithm</th><th rowspan="2">Run</th><th rowspan="2">FP</th><th rowspan="2">TP</th><th rowspan="2">FN</th><th rowspan="2">TN</th><th colspan="4">Performance Metrics for Target Group (P1)</th></tr>
<tr><th>Recall</th><th>F1-Score</th><th>Precision</th><th>Accuracy</th></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>11.29%</b></td><td><b>14.00%</b></td><td><b>18.45%</b></td><td><b>98.14%</b></td></tr>
<tr><td rowspan="5"><b>GB (OS)</b></td><td></td><td>375</td><td>77</td><td>230</td><td>22092</td><td>25.08%</td><td>20.29%</td><td>17.04%</td><td>97.34%</td></tr>
<tr><td></td><td>421</td><td>74</td><td>232</td><td>22046</td><td>24.18%</td><td>18.48%</td><td>14.95%</td><td>97.13%</td></tr>
<tr><td></td><td>393</td><td>77</td><td>230</td><td>22074</td><td>25.08%</td><td>19.82%</td><td>16.38%</td><td>97.26%</td></tr>
<tr><td></td><td>455</td><td>76</td><td>230</td><td>22012</td><td>24.84%</td><td>18.16%</td><td>14.31%</td><td>96.99%</td></tr>
<tr><td></td><td>374</td><td>79</td><td>227</td><td>22093</td><td>25.82%</td><td>20.82%</td><td>17.44%</td><td>97.36%</td></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>25.00%</b></td><td><b>19.51%</b></td><td><b>16.02%</b></td><td><b>97.22%</b></td></tr>
</table>

Performance Metrics for all Algorithms on Inpatient Datasets

<table>
<tr><th rowspan="2">Algorithm</th><th rowspan="2">Run</th><th rowspan="2">FP</th><th rowspan="2">TP</th><th rowspan="2">FN</th><th rowspan="2">TN</th><th colspan="4">Performance Metrics for Target Group (P1)</th></tr>
<tr><th>Recall</th><th>F1-Score</th><th>Precision</th><th>Accuracy</th></tr>
<tr><td rowspan="5"><b>DTG</b></td><td>1</td><td>6</td><td>42</td><td>35</td><td>2561</td><td>54.55%</td><td>67.20%</td><td>87.50%</td><td>98.45%</td></tr>
<tr><td>2</td><td>1</td><td>49</td><td>29</td><td>2565</td><td>62.82%</td><td>76.56%</td><td>98.00%</td><td>98.87%</td></tr>
<tr><td>3</td><td>1</td><td>39</td><td>38</td><td>2566</td><td>50.65%</td><td>66.67%</td><td>97.50%</td><td>98.52%</td></tr>
<tr><td>4</td><td>3</td><td>49</td><td>29</td><td>2563</td><td>62.82%</td><td>75.38%</td><td>94.23%</td><td>98.79%</td></tr>
<tr><td>5</td><td>9</td><td>48</td><td>30</td><td>2558</td><td>61.54%</td><td>71.11%</td><td>84.21%</td><td>98.53%</td></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>58.47%</b></td><td><b>71.38%</b></td><td><b>92.29%</b></td><td><b>98.63%</b></td></tr>
<tr><td rowspan="5"><b>DTG (OS)</b></td><td>1</td><td>42</td><td>43</td><td>34</td><td>2525</td><td>55.84%</td><td>53.09%</td><td>50.59%</td><td>97.13%</td></tr>
<tr><td>2</td><td>47</td><td>48</td><td>30</td><td>2519</td><td>61.54%</td><td>55.49%</td><td>50.53%</td><td>97.09%</td></tr>
<tr><td>3</td><td>64</td><td>45</td><td>32</td><td>2503</td><td>58.44%</td><td>48.39%</td><td>41.28%</td><td>96.37%</td></tr>
<tr><td>4</td><td>51</td><td>51</td><td>27</td><td>2515</td><td>65.38%</td><td>56.67%</td><td>50.00%</td><td>97.05%</td></tr>
<tr><td>5</td><td>60</td><td>48</td><td>30</td><td>2507</td><td>61.54%</td><td>51.61%</td><td>44.44%</td><td>96.60%</td></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>60.55%</b></td><td><b>53.05%</b></td><td><b>47.37%</b></td><td><b>96.85%</b></td></tr>
<tr><td rowspan="5"><b>DTE</b></td><td>1</td><td>1</td><td>43</td><td>34</td><td>2566</td><td>55.84%</td><td>71.07%</td><td>97.73%</td><td>98.68%</td></tr>
<tr><td>2</td><td>3</td><td>49</td><td>29</td><td>2563</td><td>62.82%</td><td>75.38%</td><td>94.23%</td><td>98.79%</td></tr>
<tr><td>3</td><td>2</td><td>45</td><td>32</td><td>2565</td><td>58.44%</td><td>72.58%</td><td>95.74%</td><td>98.71%</td></tr>
<tr><td>4</td><td>0</td><td>48</td><td>30</td><td>2566</td><td>61.54%</td><td>76.19%</td><td>100.00%</td><td>98.87%</td></tr>
<tr><td>5</td><td>3</td><td>42</td><td>36</td><td>2564</td><td>53.85%</td><td>68.29%</td><td>93.33%</td><td>98.53%</td></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>58.50%</b></td><td><b>72.70%</b></td><td><b>96.21%</b></td><td><b>98.71%</b></td></tr>
<tr><td rowspan="5"><b>DTE (OS)</b></td><td>1</td><td>32</td><td>47</td><td>30</td><td>2535</td><td>61.04%</td><td>60.26%</td><td>59.49%</td><td>97.66%</td></tr>
<tr><td>2</td><td>48</td><td>46</td><td>32</td><td>2518</td><td>58.97%</td><td>53.49%</td><td>48.94%</td><td>96.97%</td></tr>
<tr><td>3</td><td>42</td><td>47</td><td>30</td><td>2525</td><td>61.04%</td><td>56.63%</td><td>52.81%</td><td>97.28%</td></tr>
<tr><td>4</td><td>51</td><td>47</td><td>31</td><td>2515</td><td>60.26%</td><td>53.41%</td><td>47.96%</td><td>96.90%</td></tr>
<tr><td>5</td><td>43</td><td>41</td><td>37</td><td>2524</td><td>52.56%</td><td>50.62%</td><td>48.81%</td><td>96.98%</td></tr>
<tr><td colspan="6"><b>Average</b></td><td><b>58.77%</b></td><td><b>54.88%</b></td><td><b>51.60%</b></td><td><b>97.16%</b></td></tr>
<tr><td rowspan="3"><b>RFG</b></td><td>1</td><td>0</td><td>46</td><td>31</td><td>2567</td><td>59.74%</td><td>74.80%</td><td>100.00%</td><td>98.83%</td></tr>
<tr><td>2</td><td>0</td><td>42</td><td>36</td><td>2566</td><td>53.85%</td><td>70.00%</td><td>100.00%</td><td>98.64%</td></tr>
<tr><td>3</td><td>0</td><td>35</td><td>42</td><td>2567</td><td>45.45%</td><td>62.50%</td><td>100.00%</td><td>98.41%</td></tr>
</table>

	4	0	49	29	2566	62.82%	77.17%	100.00%	98.90%
	5	0	50	28	2567	64.10%	78.13%	100.00%	98.94%
Average						**57.19%**	**72.52%**	**100.00%**	**98.74%**
	1	39	39	38	2528	50.65%	50.32%	50.00%	97.09%
	2	27	40	38	2539	51.28%	55.17%	59.70%	97.54%
RFG (OS)	3	38	33	44	2529	42.86%	44.59%	46.48%	96.90%
	4	39	40	38	2527	51.28%	50.96%	50.63%	97.09%
	5	40	41	37	2527	52.56%	51.57%	50.62%	97.09%
Average						**49.73%**	**50.52%**	**51.49%**	**97.14%**
	1	0	43	34	2567	55.84%	71.67%	100.00%	98.71%
	2	0	49	29	2566	62.82%	77.17%	100.00%	98.90%
RFE	3	0	41	36	2567	53.25%	69.49%	100.00%	98.64%
	4	0	42	36	2566	53.85%	70.00%	100.00%	98.64%
	5	0	40	38	2567	51.28%	67.80%	100.00%	98.56%
Average						**55.41%**	**71.22%**	**100.00%**	**98.69%**
	1	28	42	35	2539	54.55%	57.14%	60.00%	97.62%
	2	42	42	36	2524	53.85%	51.85%	50.00%	97.05%
RFE (OS)	3	40	39	38	2527	50.65%	50.00%	49.37%	97.05%
	4	38	43	35	2528	55.13%	54.09%	53.09%	97.24%
	5	37	37	41	2530	47.44%	48.68%	50.00%	97.05%
Average						**52.32%**	**52.35%**	**52.49%**	**97.20%**
	1	108	45	32	2459	58.44%	39.13%	29.41%	94.70%
	2	100	48	30	2466	61.54%	42.48%	32.43%	95.08%
NB	3	88	42	35	2479	54.55%	40.58%	32.31%	95.35%
	4	111	36	42	2455	46.15%	32.00%	24.49%	94.21%
	5	98	40	38	2469	51.28%	37.04%	28.99%	94.86%
Average						**54.39%**	**38.25%**	**29.53%**	**94.84%**
	1	171	43	34	2396	55.84%	29.55%	20.09%	92.25%
	2	444	54	24	2122	69.23%	18.75%	10.84%	82.30%
NB (OS)	3	800	48	29	1767	62.34%	10.38%	5.66%	68.65%
	4	225	33	45	2341	42.31%	19.64%	12.79%	89.79%
	5	231	39	39	2336	50.00%	22.41%	14.44%	89.79%
Average						**55.94%**	**20.15%**	**12.77%**	**84.55%**
	1	39	1	76	2528	1.30%	1.71%	2.50%	95.65%
	2	36	0	78	2530	0.00%	0.00%	0.00%	95.69%
kNN	3	32	2	75	2535	2.60%	3.60%	5.88%	95.95%
	4	36	1	77	2530	1.28%	1.74%	2.70%	95.73%
	5	32	2	76	2535	2.56%	3.57%	5.88%	95.92%
Average						**1.55%**	**2.12%**	**3.39%**	**95.79%**
	1	479	9	68	2088	11.69%	3.19%	1.84%	79.31%
	2	465	10	68	2101	12.82%	3.62%	2.11%	79.84%
kNN (OS)	3	470	10	67	2097	12.99%	3.59%	2.08%	79.69%
	4	506	15	63	2060	19.23%	5.01%	2.88%	78.48%
	5	483	9	69	2084	11.54%	3.16%	1.83%	79.13%
Average						**13.65%**	**3.71%**	**2.15%**	**79.29%**
LR		18	1	76	2549	1.30%	2.08%	5.26%	96.44%
		29	2	76	2537	2.56%	3.67%	6.45%	96.03%

		20	1	76	2547	1.30%	2.04%	4.76%	96.37%
		21	1	77	2545	1.28%	2.00%	4.55%	96.29%
		24	0	78	2543	0.00%	0.00%	0.00%	96.14%
Average						**1.29%**	**1.96%**	**4.20%**	**96.26%**
LR (OS)		927	29	48	1640	37.66%	5.61%	3.03%	63.12%
		2284	67	11	282	85.90%	5.52%	2.85%	13.20%
		20	2	75	2547	2.60%	4.04%	9.09%	96.41%
		1317	35	43	1249	44.87%	4.90%	2.59%	48.56%
		1008	19	59	1559	24.36%	3.44%	1.85%	59.66%
Average						**39.08%**	**4.70%**	**3.88%**	**56.19%**
NN		10	0	77	2557	0.00%	0.00%	0.00%	96.71%
		0	0	78	2566	0.00%	0.00%	0.00%	97.05%
		0	0	77	2567	0.00%	0.00%	0.00%	97.09%
		332	8	70	2234	10.26%	3.83%	2.35%	84.80%
		0	0	78	2567	0.00%	0.00%	0.00%	97.05%
Average						**2.05%**	**0.77%**	**0.47%**	**94.54%**
NN (OS)		234	10	67	2333	12.99%	6.23%	4.10%	88.62%
		1339	39	39	1227	50.00%	5.36%	2.83%	47.88%
		1836	50	27	731	64.94%	5.09%	2.65%	29.54%
		1133	33	45	1433	42.31%	5.31%	2.83%	55.45%
		1841	56	22	726	71.79%	5.67%	2.95%	29.57%
Average						**48.40%**	**5.53%**	**3.07%**	**50.21%**
DA		0	0	77	2567	0.00%	0.00%	0.00%	97.09%
		0	0	78	2566	0.00%	0.00%	0.00%	97.05%
		0	0	77	2567	0.00%	0.00%	0.00%	97.09%
		0	0	78	2566	0.00%	0.00%	0.00%	97.05%
		0	0	78	2567	0.00%	0.00%	0.00%	97.05%
Average						**0.00%**	**0.00%**	**0.00%**	**97.07%**
DA (OS)		1152	36	41	1415	46.75%	5.69%	3.03%	54.88%
		1192	41	37	1374	52.56%	6.25%	3.33%	53.52%
		1154	46	31	1413	59.74%	7.20%	3.83%	55.18%
		1214	53	25	1352	67.95%	7.88%	4.18%	53.14%
		1210	57	21	1357	73.08%	8.48%	4.50%	53.46%
Average						**60.02%**	**7.10%**	**3.77%**	**54.04%**
GB		1	42	35	2566	54.55%	70.00%	97.67%	98.64%
		5	51	27	2561	65.38%	76.12%	91.07%	98.79%
		1	40	37	2566	51.95%	67.80%	97.56%	98.56%
		3	52	26	2563	66.67%	78.20%	94.55%	98.90%
		0	47	31	2567	60.26%	75.20%	100.00%	98.83%
Average						**59.76%**	**73.46%**	**96.17%**	**98.74%**
GB (OS)		28	46	31	2539	59.74%	60.93%	62.16%	97.77%
		21	54	24	2545	69.23%	70.59%	72.00%	98.30%
		22	44	33	2545	57.14%	61.54%	66.67%	97.92%
		20	54	24	2546	69.23%	71.05%	72.97%	98.34%
		20	50	28	2547	64.10%	67.57%	71.43%	98.19%
Average						**63.89%**	**66.33%**	**69.05%**	**98.10%**

APPENDIX I: DETAIL RESULTS OF ONE-WAY ANOVA TESTS ON FEATURE IMPORTANCE SCORES

This appendix provides the detailed result of one-way ANOVA conducted on the feature importance scores calculated for inpatient and outpatient features.

OUTPATIENT-FIS 102220

ANOVA Outpatient-feature importance score

* NOTE * Cannot draw the interval plot for the Tukey procedure. Interval plots for comparisons are illegible with more than 45 intervals.

* NOTE * Cannot draw the interval plot for the Fisher procedure. Interval plots for comparisons are illegible with more than 45 intervals.

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.05

Equal variances were assumed for the analysis.

Factor Information

Factor	Levels	Values
Factor	16	clm_pmt_amt, nch_prmry_pyr_clm_pd_amt, at_physn_npi, op_physn_npi, ot_physn_npi, icd9_dgns_cd_1, icd9_dgns_cd_2, icd9_dgns_cd_3, icd9_prcdr_cd_1, nch_bene_ptb_ddctbl_amt, nch_bene_ptb_coinsrnc_amt, admtng_icd9_dgns_cd, hcpcs_cd_1, hcpcs_cd_2, hcpcs_cd_3, YEAR

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	15	0.6359	0.042394	8.02	0.000
Error	64	0.3383	0.005286		
Total	79	0.9742			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
0.0727050	65.27%	57.13%	45.74%

Means

Factor	N	Mean	StDev	95% CI
clm_pmt_amt	5	0.389	0.274	(0.324, 0.454)
nch_prmry_pyr_clm_pd_amt	5	0.004190	0.000969	(-0.060765, 0.069146)
at_physn_npi	5	0.0983	0.0649	(0.0333, 0.1632)
op_physn_npi	5	0.02340	0.01068	(-0.04156, 0.08835)
ot_physn_npi	5	0.04454	0.01753	(-0.02042, 0.10950)
icd9_dgns_cd_1	5	0.1100	0.0389	(0.0450, 0.1749)
icd9_dgns_cd_2	5	0.0521	0.0302	(-0.0129, 0.1170)
icd9_dgns_cd_3	5	0.04191	0.01306	(-0.02305, 0.10686)
icd9_prcdr_cd_1	5	0.000005	0.000010	(-0.064950, 0.064961)
nch_bene_ptb_ddctbl_amt	5	0.00787	0.00441	(-0.05708, 0.07283)
nch_bene_ptb_coinsrnc_amt	5	0.04511	0.01672	(-0.01985, 0.11006)
admtng_icd9_dgns_cd	5	0.03077	0.01206	(-0.03419, 0.09572)
hcpcs_cd_1	5	0.0539	0.0312	(-0.0110, 0.1189)
hcpcs_cd_2	5	0.0484	0.0242	(-0.0165, 0.1134)
hcpcs_cd_3	5	0.03497	0.01726	(-0.02999, 0.09992)
YEAR	5	0.01583	0.01281	(-0.04913, 0.08078)

Pooled StDev = 0.0727050

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Factor	N	Mean	Grouping	
clm_pmt_amt	5	0.389	A	
icd9_dgns_cd_1	5	0.1100		B
at_physn_npi	5	0.0983		B
hcpcs_cd_1	5	0.0539		B
icd9_dgns_cd_2	5	0.0521		B
hcpcs_cd_2	5	0.0484		B
nch_bene_ptb_coinsrnc_amt	5	0.04511		B
ot_physn_npi	5	0.04454		B
icd9_dgns_cd_3	5	0.04191		B
hcpcs_cd_3	5	0.03497		B
admtng_icd9_dgns_cd	5	0.03077		B
op_physn_npi	5	0.02340		B
YEAR	5	0.01583		B
nch_bene_ptb_ddctbl_amt	5	0.00787		B
nch_prmry_pyr_clm_pd_amt	5	0.004190		B
icd9_prcdr_cd_1	5	0.000005		B

Means that do not share a letter are significantly different.

Tukey Simultaneous Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
nch_prmry_py - clm_pmt_amt	-0.3846	0.0460	(-0.5485, -0.2207)	-8.36	0.000
at_physn_npi - clm_pmt_amt	-0.2905	0.0460	(-0.4544, -0.1266)	-6.32	0.000
op_physn_npi - clm_pmt_amt	-0.3654	0.0460	(-0.5293, -0.2015)	-7.95	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
ot_physn_npi - clm_pmt_amt	-0.3442	0.0460	(-0.5081, -0.1804)	-7.49	0.000
icd9_dgns_cd - clm_pmt_amt	-0.2788	0.0460	(-0.4427, -0.1149)	-6.06	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3367	0.0460	(-0.5006, -0.1728)	-7.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3469	0.0460	(-0.5107, -0.1830)	-7.54	0.000
icd9_prcdr_c - clm_pmt_amt	-0.3888	0.0460	(-0.5526, -0.2249)	-8.45	0.000
nch_bene_ptb - clm_pmt_amt	-0.3809	0.0460	(-0.5448, -0.2170)	-8.28	0.000
nch_bene_ptb - clm_pmt_amt	-0.3437	0.0460	(-0.5075, -0.1798)	-7.47	0.000
admtng_icd9_ - clm_pmt_amt	-0.3580	0.0460	(-0.5219, -0.1941)	-7.79	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3348	0.0460	(-0.4987, -0.1710)	-7.28	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3404	0.0460	(-0.5042, -0.1765)	-7.40	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3538	0.0460	(-0.5177, -0.1899)	-7.69	0.000
YEAR - clm_pmt_amt	-0.3730	0.0460	(-0.5368, -0.2091)	-8.11	0.000
at_physn_npi - nch_prmry_py	0.0941	0.0460	(-0.0698, 0.2579)	2.05	0.790
op_physn_npi - nch_prmry_py	0.0192	0.0460	(-0.1447, 0.1831)	0.42	1.000
ot_physn_npi - nch_prmry_py	0.0403	0.0460	(-0.1235, 0.2042)	0.88	1.000
icd9_dgns_cd - nch_prmry_py	0.1058	0.0460	(-0.0581, 0.2697)	2.30	0.625
icd9_dgns_cd - nch_prmry_py	0.0479	0.0460	(-0.1160, 0.2117)	1.04	1.000
icd9_dgns_cd - nch_prmry_py	0.0377	0.0460	(-0.1262, 0.2016)	0.82	1.000
icd9_prcdr_c - nch_prmry_py	-0.0042	0.0460	(-0.1681, 0.1597)	-0.09	1.000
nch_bene_ptb - nch_prmry_py	0.0037	0.0460	(-0.1602, 0.1676)	0.08	1.000
nch_bene_ptb - nch_prmry_py	0.0409	0.0460	(-0.1230, 0.2048)	0.89	1.000
admtng_icd9_ - nch_prmry_py	0.0266	0.0460	(-0.1373, 0.1905)	0.58	1.000
hcpcs_cd_1 - nch_prmry_py	0.0497	0.0460	(-0.1141, 0.2136)	1.08	0.999
hcpcs_cd_2 - nch_prmry_py	0.0442	0.0460	(-0.1196, 0.2081)	0.96	1.000
hcpcs_cd_3 - nch_prmry_py	0.0308	0.0460	(-0.1331, 0.1947)	0.67	1.000
YEAR - nch_prmry_py	0.0116	0.0460	(-0.1522, 0.1755)	0.25	1.000
op_physn_npi - at_physn_npi	-0.0749	0.0460	(-0.2387, 0.0890)	-1.63	0.957
ot_physn_npi - at_physn_npi	-0.0537	0.0460	(-0.2176, 0.1102)	-1.17	0.998
icd9_dgns_cd - at_physn_npi	0.0117	0.0460	(-0.1522, 0.1756)	0.25	1.000
icd9_dgns_cd - at_physn_npi	-0.0462	0.0460	(-0.2101, 0.1177)	-1.00	1.000
icd9_dgns_cd - at_physn_npi	-0.0564	0.0460	(-0.2202, 0.1075)	-1.23	0.997
icd9_prcdr_c - at_physn_npi	-0.0983	0.0460	(-0.2621, 0.0656)	-2.14	0.735
nch_bene_ptb - at_physn_npi	-0.0904	0.0460	(-0.2543, 0.0735)	-1.97	0.834
nch_bene_ptb - at_physn_npi	-0.0532	0.0460	(-0.2170, 0.1107)	-1.16	0.998
admtng_icd9_ - at_physn_npi	-0.0675	0.0460	(-0.2314, 0.0964)	-1.47	0.982
hcpcs_cd_1 - at_physn_npi	-0.0443	0.0460	(-0.2082, 0.1195)	-0.96	1.000
hcpcs_cd_2 - at_physn_npi	-0.0498	0.0460	(-0.2137, 0.1140)	-1.08	0.999
hcpcs_cd_3 - at_physn_npi	-0.0633	0.0460	(-0.2272, 0.1006)	-1.38	0.990
YEAR - at_physn_npi	-0.0824	0.0460	(-0.2463, 0.0814)	-1.79	0.910
ot_physn_npi - op_physn_npi	0.0211	0.0460	(-0.1427, 0.1850)	0.46	1.000
icd9_dgns_cd - op_physn_npi	0.0866	0.0460	(-0.0773, 0.2505)	1.88	0.874
icd9_dgns_cd - op_physn_npi	0.0287	0.0460	(-0.1352, 0.1925)	0.62	1.000
icd9_dgns_cd - op_physn_npi	0.0185	0.0460	(-0.1454, 0.1824)	0.40	1.000
icd9_prcdr_c - op_physn_npi	-0.0234	0.0460	(-0.1873, 0.1405)	-0.51	1.000
nch_bene_ptb - op_physn_npi	-0.0155	0.0460	(-0.1794, 0.1483)	-0.34	1.000
nch_bene_ptb - op_physn_npi	0.0217	0.0460	(-0.1422, 0.1856)	0.47	1.000
admtng_icd9_ - op_physn_npi	0.0074	0.0460	(-0.1565, 0.1712)	0.16	1.000
hcpcs_cd_1 - op_physn_npi	0.0305	0.0460	(-0.1333, 0.1944)	0.66	1.000
hcpcs_cd_2 - op_physn_npi	0.0250	0.0460	(-0.1389, 0.1889)	0.54	1.000
hcpcs_cd_3 - op_physn_npi	0.0116	0.0460	(-0.1523, 0.1754)	0.25	1.000
YEAR - op_physn_npi	-0.0076	0.0460	(-0.1714, 0.1563)	-0.16	1.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
icd9_dgns_cd - ot_physn_npi	0.0654	0.0460	(-0.0984, 0.2293)	1.42	0.987
icd9_dgns_cd - ot_physn_npi	0.0075	0.0460	(-0.1564, 0.1714)	0.16	1.000
icd9_dgns_cd - ot_physn_npi	-0.0026	0.0460	(-0.1665, 0.1612)	-0.06	1.000
icd9_prcdr_c - ot_physn_npi	-0.0445	0.0460	(-0.2084, 0.1193)	-0.97	1.000
nch_bene_ptb - ot_physn_npi	-0.0367	0.0460	(-0.2005, 0.1272)	-0.80	1.000
nch_bene_ptb - ot_physn_npi	0.0006	0.0460	(-0.1633, 0.1644)	0.01	1.000
admtng_icd9_ - ot_physn_npi	-0.0138	0.0460	(-0.1776, 0.1501)	-0.30	1.000
hcpcs_cd_1 - ot_physn_npi	0.0094	0.0460	(-0.1545, 0.1733)	0.20	1.000
hcpcs_cd_2 - ot_physn_npi	0.0039	0.0460	(-0.1600, 0.1678)	0.08	1.000
hcpcs_cd_3 - ot_physn_npi	-0.0096	0.0460	(-0.1734, 0.1543)	-0.21	1.000
YEAR - ot_physn_npi	-0.0287	0.0460	(-0.1926, 0.1352)	-0.62	1.000
icd9_dgns_cd - icd9_dgns_cd	-0.0579	0.0460	(-0.2218, 0.1060)	-1.26	0.996
icd9_dgns_cd - icd9_dgns_cd	-0.0681	0.0460	(-0.2319, 0.0958)	-1.48	0.981
icd9_prcdr_c - icd9_dgns_cd	-0.1100	0.0460	(-0.2738, 0.0539)	-2.39	0.561
nch_bene_ptb - icd9_dgns_cd	-0.1021	0.0460	(-0.2660, 0.0618)	-2.22	0.680
nch_bene_ptb - icd9_dgns_cd	-0.0649	0.0460	(-0.2287, 0.0990)	-1.41	0.988
admtng_icd9_ - icd9_dgns_cd	-0.0792	0.0460	(-0.2431, 0.0847)	-1.72	0.933
hcpcs_cd_1 - icd9_dgns_cd	-0.0560	0.0460	(-0.2199, 0.1078)	-1.22	0.997
hcpcs_cd_2 - icd9_dgns_cd	-0.0616	0.0460	(-0.2254, 0.1023)	-1.34	0.993
hcpcs_cd_3 - icd9_dgns_cd	-0.0750	0.0460	(-0.2389, 0.0889)	-1.63	0.956
YEAR - icd9_dgns_cd	-0.0942	0.0460	(-0.2580, 0.0697)	-2.05	0.789
icd9_dgns_cd - icd9_dgns_cd	-0.0102	0.0460	(-0.1740, 0.1537)	-0.22	1.000
icd9_prcdr_c - icd9_dgns_cd	-0.0521	0.0460	(-0.2159, 0.1118)	-1.13	0.999
nch_bene_ptb - icd9_dgns_cd	-0.0442	0.0460	(-0.2081, 0.1197)	-0.96	1.000
nch_bene_ptb - icd9_dgns_cd	-0.0070	0.0460	(-0.1708, 0.1569)	-0.15	1.000
admtng_icd9_ - icd9_dgns_cd	-0.0213	0.0460	(-0.1852, 0.1426)	-0.46	1.000
hcpcs_cd_1 - icd9_dgns_cd	0.0019	0.0460	(-0.1620, 0.1657)	0.04	1.000
hcpcs_cd_2 - icd9_dgns_cd	-0.0036	0.0460	(-0.1675, 0.1602)	-0.08	1.000
hcpcs_cd_3 - icd9_dgns_cd	-0.0171	0.0460	(-0.1810, 0.1468)	-0.37	1.000
YEAR - icd9_dgns_cd	-0.0362	0.0460	(-0.2001, 0.1276)	-0.79	1.000
icd9_prcdr_c - icd9_dgns_cd	-0.0419	0.0460	(-0.2058, 0.1220)	-0.91	1.000
nch_bene_ptb - icd9_dgns_cd	-0.0340	0.0460	(-0.1979, 0.1298)	-0.74	1.000
nch_bene_ptb - icd9_dgns_cd	0.0032	0.0460	(-0.1607, 0.1671)	0.07	1.000
admtng_icd9_ - icd9_dgns_cd	-0.0111	0.0460	(-0.1750, 0.1527)	-0.24	1.000
hcpcs_cd_1 - icd9_dgns_cd	0.0120	0.0460	(-0.1518, 0.1759)	0.26	1.000
hcpcs_cd_2 - icd9_dgns_cd	0.0065	0.0460	(-0.1574, 0.1704)	0.14	1.000
hcpcs_cd_3 - icd9_dgns_cd	-0.0069	0.0460	(-0.1708, 0.1569)	-0.15	1.000
YEAR - icd9_dgns_cd	-0.0261	0.0460	(-0.1900, 0.1378)	-0.57	1.000
nch_bene_ptb - icd9_prcdr_c	0.0079	0.0460	(-0.1560, 0.1717)	0.17	1.000
nch_bene_ptb - icd9_prcdr_c	0.0451	0.0460	(-0.1188, 0.2090)	0.98	1.000
admtng_icd9_ - icd9_prcdr_c	0.0308	0.0460	(-0.1331, 0.1946)	0.67	1.000
hcpcs_cd_1 - icd9_prcdr_c	0.0539	0.0460	(-0.1099, 0.2178)	1.17	0.998
hcpcs_cd_2 - icd9_prcdr_c	0.0484	0.0460	(-0.1155, 0.2123)	1.05	0.999
hcpcs_cd_3 - icd9_prcdr_c	0.0350	0.0460	(-0.1289, 0.1988)	0.76	1.000
YEAR - icd9_prcdr_c	0.0158	0.0460	(-0.1481, 0.1797)	0.34	1.000
nch_bene_ptb - nch_bene_ptb	0.0372	0.0460	(-0.1266, 0.2011)	0.81	1.000
admtng_icd9_ - nch_bene_ptb	0.0229	0.0460	(-0.1410, 0.1868)	0.50	1.000
hcpcs_cd_1 - nch_bene_ptb	0.0461	0.0460	(-0.1178, 0.2099)	1.00	1.000
hcpcs_cd_2 - nch_bene_ptb	0.0405	0.0460	(-0.1233, 0.2044)	0.88	1.000
hcpcs_cd_3 - nch_bene_ptb	0.0271	0.0460	(-0.1368, 0.1910)	0.59	1.000
YEAR - nch_bene_ptb	0.0080	0.0460	(-0.1559, 0.1718)	0.17	1.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
admtng_icd9_ - nch_bene_ptb	-0.0143	0.0460	(-0.1782, 0.1495)	-0.31	1.000
hcpcs_cd_1 - nch_bene_ptb	0.0088	0.0460	(-0.1550, 0.1727)	0.19	1.000
hcpcs_cd_2 - nch_bene_ptb	0.0033	0.0460	(-0.1606, 0.1672)	0.07	1.000
hcpcs_cd_3 - nch_bene_ptb	-0.0101	0.0460	(-0.1740, 0.1537)	-0.22	1.000
YEAR - nch_bene_ptb	-0.0293	0.0460	(-0.1932, 0.1346)	-0.64	1.000
hcpcs_cd_1 - admtng_icd9_	0.0232	0.0460	(-0.1407, 0.1870)	0.50	1.000
hcpcs_cd_2 - admtng_icd9_	0.0176	0.0460	(-0.1462, 0.1815)	0.38	1.000
hcpcs_cd_3 - admtng_icd9_	0.0042	0.0460	(-0.1597, 0.1681)	0.09	1.000
YEAR - admtng_icd9_	-0.0149	0.0460	(-0.1788, 0.1489)	-0.32	1.000
hcpcs_cd_2 - hcpcs_cd_1	-0.0055	0.0460	(-0.1694, 0.1584)	-0.12	1.000
hcpcs_cd_3 - hcpcs_cd_1	-0.0190	0.0460	(-0.1828, 0.1449)	-0.41	1.000
YEAR - hcpcs_cd_1	-0.0381	0.0460	(-0.2020, 0.1258)	-0.83	1.000
hcpcs_cd_3 - hcpcs_cd_2	-0.0134	0.0460	(-0.1773, 0.1504)	-0.29	1.000
YEAR - hcpcs_cd_2	-0.0326	0.0460	(-0.1965, 0.1313)	-0.71	1.000
YEAR - hcpcs_cd_3	-0.0191	0.0460	(-0.1830, 0.1447)	-0.42	1.000

Individual confidence level = 99.93%

Fisher Pairwise Comparisons

Grouping Information Using the Fisher LSD Method and 95% Confidence

Factor	N	Mean	Grouping			
clm_pmt_amt	5	0.389	A			
icd9_dgns_cd_1	5	0.1100		B		
at_physn_npi	5	0.0983		B	C	
hcpcs_cd_1	5	0.0539		B	C	D
icd9_dgns_cd_2	5	0.0521		B	C	D
hcpcs_cd_2	5	0.0484		B	C	D
nch_bene_ptb_coinsrnc_amt	5	0.04511		B	C	D
ot_physn_npi	5	0.04454		B	C	D
icd9_dgns_cd_3	5	0.04191		B	C	D
hcpcs_cd_3	5	0.03497		B	C	D
admtng_icd9_dgns_cd	5	0.03077		B	C	D
op_physn_npi	5	0.02340		B	C	D
YEAR	5	0.01583			C	D
nch_bene_ptb_ddctbl_amt	5	0.00787			C	D
nch_prmry_pyr_clm_pd_amt	5	0.004190				D
icd9_prcdr_cd_1	5	0.000005				D

Means that do not share a letter are significantly different.

Fisher Individual Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
nch_prmry_py - clm_pmt_amt	-0.3846	0.0460	(-0.4764, -0.2927)	-8.36	0.000
at_physn_npi - clm_pmt_amt	-0.2905	0.0460	(-0.3824, -0.1987)	-6.32	0.000
op_physn_npi - clm_pmt_amt	-0.3654	0.0460	(-0.4572, -0.2735)	-7.95	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
ot_physn_npi - clm_pmt_amt	-0.3442	0.0460	(-0.4361, -0.2524)	-7.49	0.000
icd9_dgns_cd - clm_pmt_amt	-0.2788	0.0460	(-0.3707, -0.1869)	-6.06	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3367	0.0460	(-0.4286, -0.2449)	-7.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3469	0.0460	(-0.4387, -0.2550)	-7.54	0.000
icd9_prcdr_c - clm_pmt_amt	-0.3888	0.0460	(-0.4806, -0.2969)	-8.45	0.000
nch_bene_ptb - clm_pmt_amt	-0.3809	0.0460	(-0.4728, -0.2890)	-8.28	0.000
nch_bene_ptb - clm_pmt_amt	-0.3437	0.0460	(-0.4355, -0.2518)	-7.47	0.000
admtng_icd9_ - clm_pmt_amt	-0.3580	0.0460	(-0.4499, -0.2661)	-7.79	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3348	0.0460	(-0.4267, -0.2430)	-7.28	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3404	0.0460	(-0.4322, -0.2485)	-7.40	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3538	0.0460	(-0.4457, -0.2619)	-7.69	0.000
YEAR - clm_pmt_amt	-0.3730	0.0460	(-0.4648, -0.2811)	-8.11	0.000
at_physn_npi - nch_prmry_py	0.0941	0.0460	(0.0022, 0.1859)	2.05	0.045
op_physn_npi - nch_prmry_py	0.0192	0.0460	(-0.0727, 0.1111)	0.42	0.678
ot_physn_npi - nch_prmry_py	0.0403	0.0460	(-0.0515, 0.1322)	0.88	0.383
icd9_dgns_cd - nch_prmry_py	0.1058	0.0460	(0.0139, 0.1976)	2.30	0.025
icd9_dgns_cd - nch_prmry_py	0.0479	0.0460	(-0.0440, 0.1397)	1.04	0.302
icd9_dgns_cd - nch_prmry_py	0.0377	0.0460	(-0.0541, 0.1296)	0.82	0.415
icd9_prcdr_c - nch_prmry_py	-0.0042	0.0460	(-0.0960, 0.0877)	-0.09	0.928
nch_bene_ptb - nch_prmry_py	0.0037	0.0460	(-0.0882, 0.0955)	0.08	0.936
nch_bene_ptb - nch_prmry_py	0.0409	0.0460	(-0.0509, 0.1328)	0.89	0.377
admtng_icd9_ - nch_prmry_py	0.0266	0.0460	(-0.0653, 0.1184)	0.58	0.565
hcpcs_cd_1 - nch_prmry_py	0.0497	0.0460	(-0.0421, 0.1416)	1.08	0.283
hcpcs_cd_2 - nch_prmry_py	0.0442	0.0460	(-0.0476, 0.1361)	0.96	0.340
hcpcs_cd_3 - nch_prmry_py	0.0308	0.0460	(-0.0611, 0.1226)	0.67	0.506
YEAR - nch_prmry_py	0.0116	0.0460	(-0.0802, 0.1035)	0.25	0.801
op_physn_npi - at_physn_npi	-0.0749	0.0460	(-0.1667, 0.0170)	-1.63	0.108
ot_physn_npi - at_physn_npi	-0.0537	0.0460	(-0.1456, 0.0381)	-1.17	0.247
icd9_dgns_cd - at_physn_npi	0.0117	0.0460	(-0.0801, 0.1036)	0.25	0.800
icd9_dgns_cd - at_physn_npi	-0.0462	0.0460	(-0.1381, 0.0457)	-1.00	0.319
icd9_dgns_cd - at_physn_npi	-0.0564	0.0460	(-0.1482, 0.0355)	-1.23	0.225
icd9_prcdr_c - at_physn_npi	-0.0983	0.0460	(-0.1901, -0.0064)	-2.14	0.036
nch_bene_ptb - at_physn_npi	-0.0904	0.0460	(-0.1822, 0.0015)	-1.97	0.054
nch_bene_ptb - at_physn_npi	-0.0532	0.0460	(-0.1450, 0.0387)	-1.16	0.252
admtng_icd9_ - at_physn_npi	-0.0675	0.0460	(-0.1594, 0.0244)	-1.47	0.147
hcpcs_cd_1 - at_physn_npi	-0.0443	0.0460	(-0.1362, 0.0475)	-0.96	0.339
hcpcs_cd_2 - at_physn_npi	-0.0498	0.0460	(-0.1417, 0.0420)	-1.08	0.282
hcpcs_cd_3 - at_physn_npi	-0.0633	0.0460	(-0.1552, 0.0286)	-1.38	0.173
YEAR - at_physn_npi	-0.0824	0.0460	(-0.1743, 0.0094)	-1.79	0.078
ot_physn_npi - op_physn_npi	0.0211	0.0460	(-0.0707, 0.1130)	0.46	0.647
icd9_dgns_cd - op_physn_npi	0.0866	0.0460	(-0.0053, 0.1784)	1.88	0.064
icd9_dgns_cd - op_physn_npi	0.0287	0.0460	(-0.0632, 0.1205)	0.62	0.535
icd9_dgns_cd - op_physn_npi	0.0185	0.0460	(-0.0734, 0.1104)	0.40	0.689
icd9_prcdr_c - op_physn_npi	-0.0234	0.0460	(-0.1153, 0.0685)	-0.51	0.613
nch_bene_ptb - op_physn_npi	-0.0155	0.0460	(-0.1074, 0.0763)	-0.34	0.737
nch_bene_ptb - op_physn_npi	0.0217	0.0460	(-0.0702, 0.1136)	0.47	0.638
admtng_icd9_ - op_physn_npi	0.0074	0.0460	(-0.0845, 0.0992)	0.16	0.873
hcpcs_cd_1 - op_physn_npi	0.0305	0.0460	(-0.0613, 0.1224)	0.66	0.509
hcpcs_cd_2 - op_physn_npi	0.0250	0.0460	(-0.0668, 0.1169)	0.54	0.588
hcpcs_cd_3 - op_physn_npi	0.0116	0.0460	(-0.0803, 0.1034)	0.25	0.802
YEAR - op_physn_npi	-0.0076	0.0460	(-0.0994, 0.0843)	-0.16	0.870

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
icd9_dgns_cd - ot_physn_npi	0.0654	0.0460	(-0.0264, 0.1573)	1.42	0.160
icd9_dgns_cd - ot_physn_npi	0.0075	0.0460	(-0.0843, 0.0994)	0.16	0.871
icd9_dgns_cd - ot_physn_npi	-0.0026	0.0460	(-0.0945, 0.0892)	-0.06	0.955
icd9_prcdr_c - ot_physn_npi	-0.0445	0.0460	(-0.1364, 0.0473)	-0.97	0.336
nch_bene_ptb - ot_physn_npi	-0.0367	0.0460	(-0.1285, 0.0552)	-0.80	0.428
nch_bene_ptb - ot_physn_npi	0.0006	0.0460	(-0.0913, 0.0924)	0.01	0.990
admtng_icd9_ - ot_physn_npi	-0.0138	0.0460	(-0.1056, 0.0781)	-0.30	0.766
hcpcs_cd_1 - ot_physn_npi	0.0094	0.0460	(-0.0825, 0.1013)	0.20	0.839
hcpcs_cd_2 - ot_physn_npi	0.0039	0.0460	(-0.0880, 0.0957)	0.08	0.933
hcpcs_cd_3 - ot_physn_npi	-0.0096	0.0460	(-0.1014, 0.0823)	-0.21	0.836
YEAR - ot_physn_npi	-0.0287	0.0460	(-0.1206, 0.0631)	-0.62	0.535
icd9_dgns_cd - icd9_dgns_cd	-0.0579	0.0460	(-0.1498, 0.0339)	-1.26	0.212
icd9_dgns_cd - icd9_dgns_cd	-0.0681	0.0460	(-0.1599, 0.0238)	-1.48	0.144
icd9_prcdr_c - icd9_dgns_cd	-0.1100	0.0460	(-0.2018, -0.0181)	-2.39	0.020
nch_bene_ptb - icd9_dgns_cd	-0.1021	0.0460	(-0.1940, -0.0102)	-2.22	0.030
nch_bene_ptb - icd9_dgns_cd	-0.0649	0.0460	(-0.1567, 0.0270)	-1.41	0.163
admtng_icd9_ - icd9_dgns_cd	-0.0792	0.0460	(-0.1711, 0.0127)	-1.72	0.090
hcpcs_cd_1 - icd9_dgns_cd	-0.0560	0.0460	(-0.1479, 0.0358)	-1.22	0.227
hcpcs_cd_2 - icd9_dgns_cd	-0.0616	0.0460	(-0.1534, 0.0303)	-1.34	0.185
hcpcs_cd_3 - icd9_dgns_cd	-0.0750	0.0460	(-0.1669, 0.0169)	-1.63	0.108
YEAR - icd9_dgns_cd	-0.0942	0.0460	(-0.1860, -0.0023)	-2.05	0.045
icd9_dgns_cd - icd9_dgns_cd	-0.0102	0.0460	(-0.1020, 0.0817)	-0.22	0.826
icd9_prcdr_c - icd9_dgns_cd	-0.0521	0.0460	(-0.1439, 0.0398)	-1.13	0.262
nch_bene_ptb - icd9_dgns_cd	-0.0442	0.0460	(-0.1361, 0.0477)	-0.96	0.340
nch_bene_ptb - icd9_dgns_cd	-0.0070	0.0460	(-0.0988, 0.0849)	-0.15	0.880
admtng_icd9_ - icd9_dgns_cd	-0.0213	0.0460	(-0.1132, 0.0706)	-0.46	0.645
hcpcs_cd_1 - icd9_dgns_cd	0.0019	0.0460	(-0.0900, 0.0937)	0.04	0.968
hcpcs_cd_2 - icd9_dgns_cd	-0.0036	0.0460	(-0.0955, 0.0882)	-0.08	0.937
hcpcs_cd_3 - icd9_dgns_cd	-0.0171	0.0460	(-0.1090, 0.0748)	-0.37	0.711
YEAR - icd9_dgns_cd	-0.0362	0.0460	(-0.1281, 0.0556)	-0.79	0.434
icd9_prcdr_c - icd9_dgns_cd	-0.0419	0.0460	(-0.1338, 0.0500)	-0.91	0.366
nch_bene_ptb - icd9_dgns_cd	-0.0340	0.0460	(-0.1259, 0.0578)	-0.74	0.462
nch_bene_ptb - icd9_dgns_cd	0.0032	0.0460	(-0.0887, 0.0951)	0.07	0.945
admtng_icd9_ - icd9_dgns_cd	-0.0111	0.0460	(-0.1030, 0.0807)	-0.24	0.809
hcpcs_cd_1 - icd9_dgns_cd	0.0120	0.0460	(-0.0798, 0.1039)	0.26	0.795
hcpcs_cd_2 - icd9_dgns_cd	0.0065	0.0460	(-0.0854, 0.0984)	0.14	0.888
hcpcs_cd_3 - icd9_dgns_cd	-0.0069	0.0460	(-0.0988, 0.0849)	-0.15	0.881
YEAR - icd9_dgns_cd	-0.0261	0.0460	(-0.1179, 0.0658)	-0.57	0.573
nch_bene_ptb - icd9_prcdr_c	0.0079	0.0460	(-0.0840, 0.0997)	0.17	0.865
nch_bene_ptb - icd9_prcdr_c	0.0451	0.0460	(-0.0468, 0.1370)	0.98	0.330
admtng_icd9_ - icd9_prcdr_c	0.0308	0.0460	(-0.0611, 0.1226)	0.67	0.506
hcpcs_cd_1 - icd9_prcdr_c	0.0539	0.0460	(-0.0379, 0.1458)	1.17	0.245
hcpcs_cd_2 - icd9_prcdr_c	0.0484	0.0460	(-0.0435, 0.1403)	1.05	0.296
hcpcs_cd_3 - icd9_prcdr_c	0.0350	0.0460	(-0.0569, 0.1268)	0.76	0.450
YEAR - icd9_prcdr_c	0.0158	0.0460	(-0.0760, 0.1077)	0.34	0.732
nch_bene_ptb - nch_bene_ptb	0.0372	0.0460	(-0.0546, 0.1291)	0.81	0.421
admtng_icd9_ - nch_bene_ptb	0.0229	0.0460	(-0.0690, 0.1148)	0.50	0.620
hcpcs_cd_1 - nch_bene_ptb	0.0461	0.0460	(-0.0458, 0.1379)	1.00	0.320
hcpcs_cd_2 - nch_bene_ptb	0.0405	0.0460	(-0.0513, 0.1324)	0.88	0.381
hcpcs_cd_3 - nch_bene_ptb	0.0271	0.0460	(-0.0648, 0.1190)	0.59	0.558
YEAR - nch_bene_ptb	0.0080	0.0460	(-0.0839, 0.0998)	0.17	0.863

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
admtng_icd9_ - nch_bene_ptb	-0.0143	0.0460	(-0.1062, 0.0775)	-0.31	0.756
hcpcs_cd_1 - nch_bene_ptb	0.0088	0.0460	(-0.0830, 0.1007)	0.19	0.848
hcpcs_cd_2 - nch_bene_ptb	0.0033	0.0460	(-0.0886, 0.0952)	0.07	0.943
hcpcs_cd_3 - nch_bene_ptb	-0.0101	0.0460	(-0.1020, 0.0817)	-0.22	0.826
YEAR - nch_bene_ptb	-0.0293	0.0460	(-0.1211, 0.0626)	-0.64	0.527
hcpcs_cd_1 - admtng_icd9_	0.0232	0.0460	(-0.0687, 0.1150)	0.50	0.616
hcpcs_cd_2 - admtng_icd9_	0.0176	0.0460	(-0.0742, 0.1095)	0.38	0.702
hcpcs_cd_3 - admtng_icd9_	0.0042	0.0460	(-0.0877, 0.0961)	0.09	0.927
YEAR - admtng_icd9_	-0.0149	0.0460	(-0.1068, 0.0769)	-0.32	0.746
hcpcs_cd_2 - hcpcs_cd_1	-0.0055	0.0460	(-0.0974, 0.0863)	-0.12	0.905
hcpcs_cd_3 - hcpcs_cd_1	-0.0190	0.0460	(-0.1108, 0.0729)	-0.41	0.681
YEAR - hcpcs_cd_1	-0.0381	0.0460	(-0.1300, 0.0538)	-0.83	0.410
hcpcs_cd_3 - hcpcs_cd_2	-0.0134	0.0460	(-0.1053, 0.0784)	-0.29	0.771
YEAR - hcpcs_cd_2	-0.0326	0.0460	(-0.1245, 0.0593)	-0.71	0.481
YEAR - hcpcs_cd_3	-0.0191	0.0460	(-0.1110, 0.0727)	-0.42	0.679

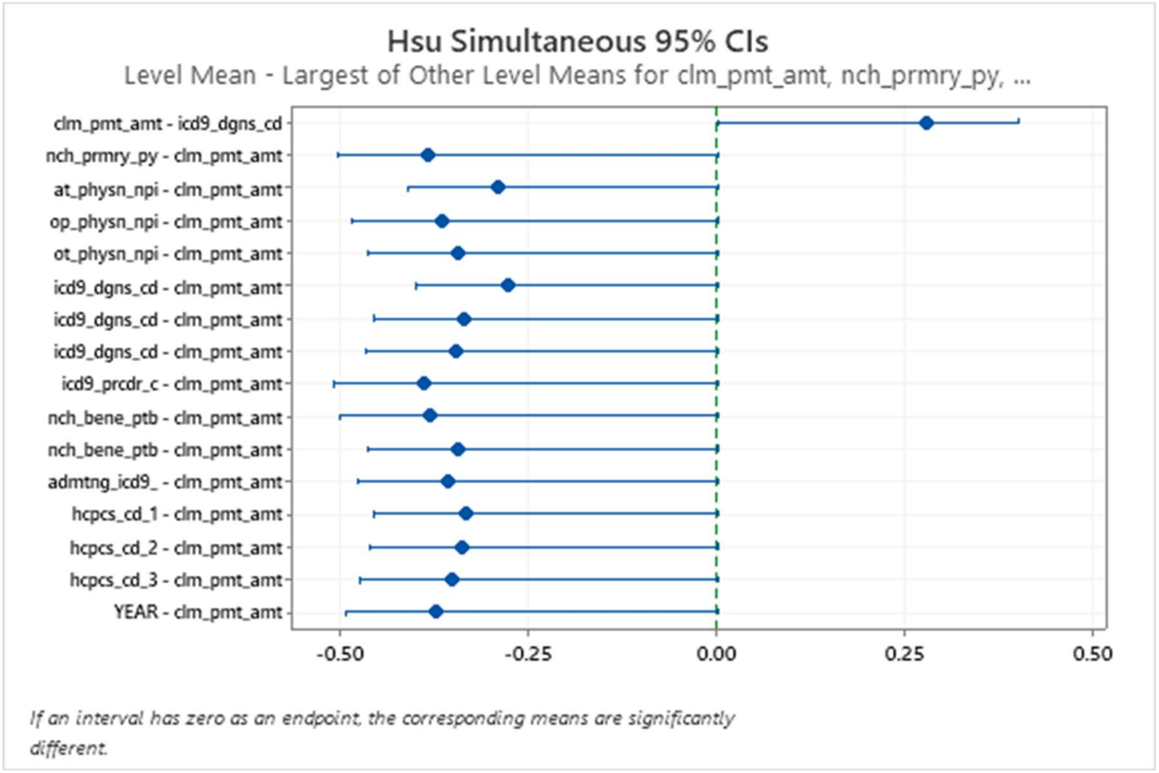
Simultaneous confidence level = 18.26%

Hsu Multiple Comparisons with the Best (MCB)

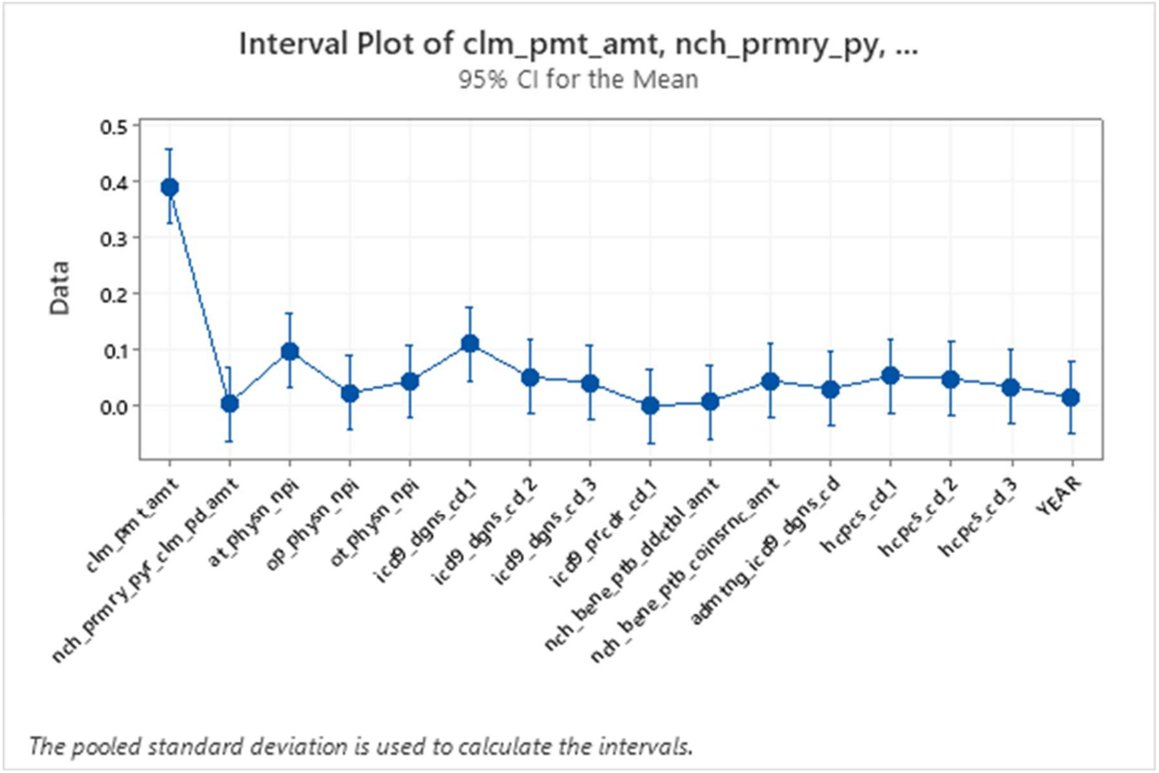
Hsu Simultaneous Tests for Level Mean - Largest of Other Level Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
clm_pmt_amt - icd9_dgns_cd	0.2788	0.0460	(0.0000, 0.3996)	6.06	0.000
nch_prmry_py - clm_pmt_amt	-0.3846	0.0460	(-0.5054, 0.0000)	-8.36	0.000
at_physn_npi - clm_pmt_amt	-0.2905	0.0460	(-0.4114, 0.0000)	-6.32	0.000
op_physn_npi - clm_pmt_amt	-0.3654	0.0460	(-0.4862, 0.0000)	-7.95	0.000
ot_physn_npi - clm_pmt_amt	-0.3442	0.0460	(-0.4651, 0.0000)	-7.49	0.000
icd9_dgns_cd - clm_pmt_amt	-0.2788	0.0460	(-0.3996, 0.0000)	-6.06	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3367	0.0460	(-0.4576, 0.0000)	-7.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3469	0.0460	(-0.4677, 0.0000)	-7.54	0.000
icd9_prcdr_c - clm_pmt_amt	-0.3888	0.0460	(-0.5096, 0.0000)	-8.45	0.000
nch_bene_ptb - clm_pmt_amt	-0.3809	0.0460	(-0.5018, 0.0000)	-8.28	0.000
nch_bene_ptb - clm_pmt_amt	-0.3437	0.0460	(-0.4645, 0.0000)	-7.47	0.000
admtng_icd9_ - clm_pmt_amt	-0.3580	0.0460	(-0.4789, 0.0000)	-7.79	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3348	0.0460	(-0.4557, 0.0000)	-7.28	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3404	0.0460	(-0.4612, 0.0000)	-7.40	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3538	0.0460	(-0.4747, 0.0000)	-7.69	0.000
YEAR - clm_pmt_amt	-0.3730	0.0460	(-0.4938, 0.0000)	-8.11	0.000

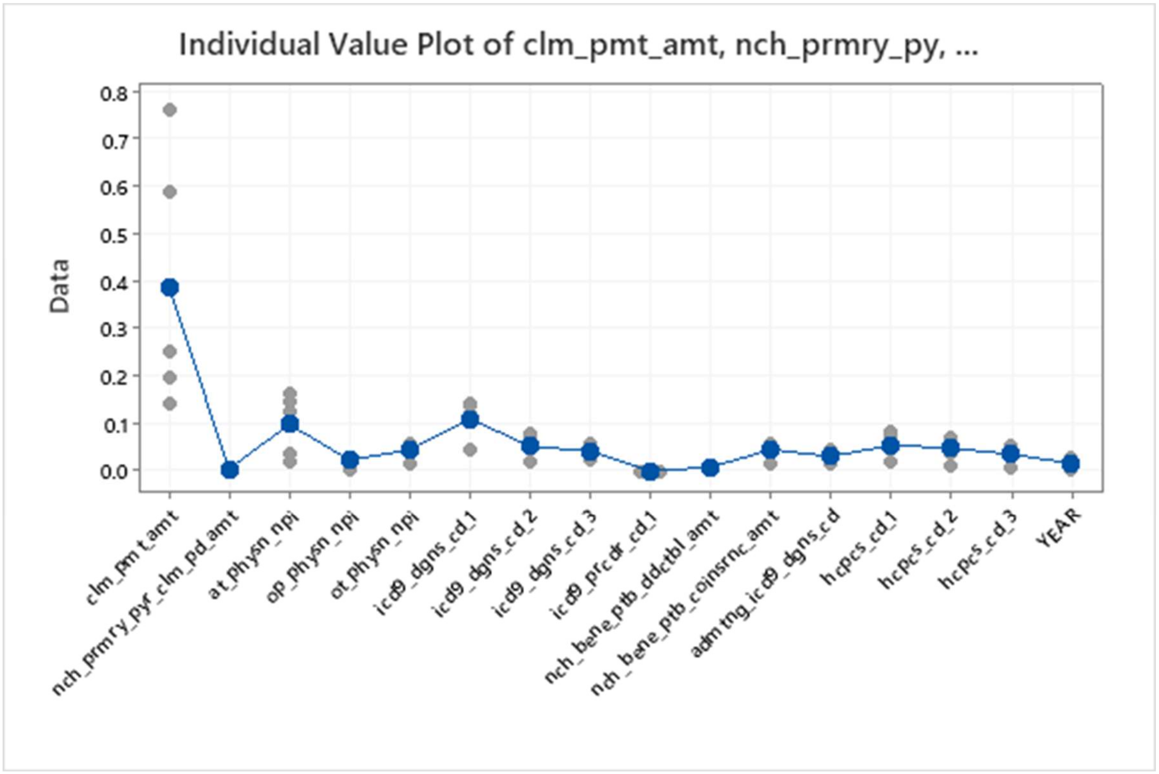
Individual confidence level = 98.93%



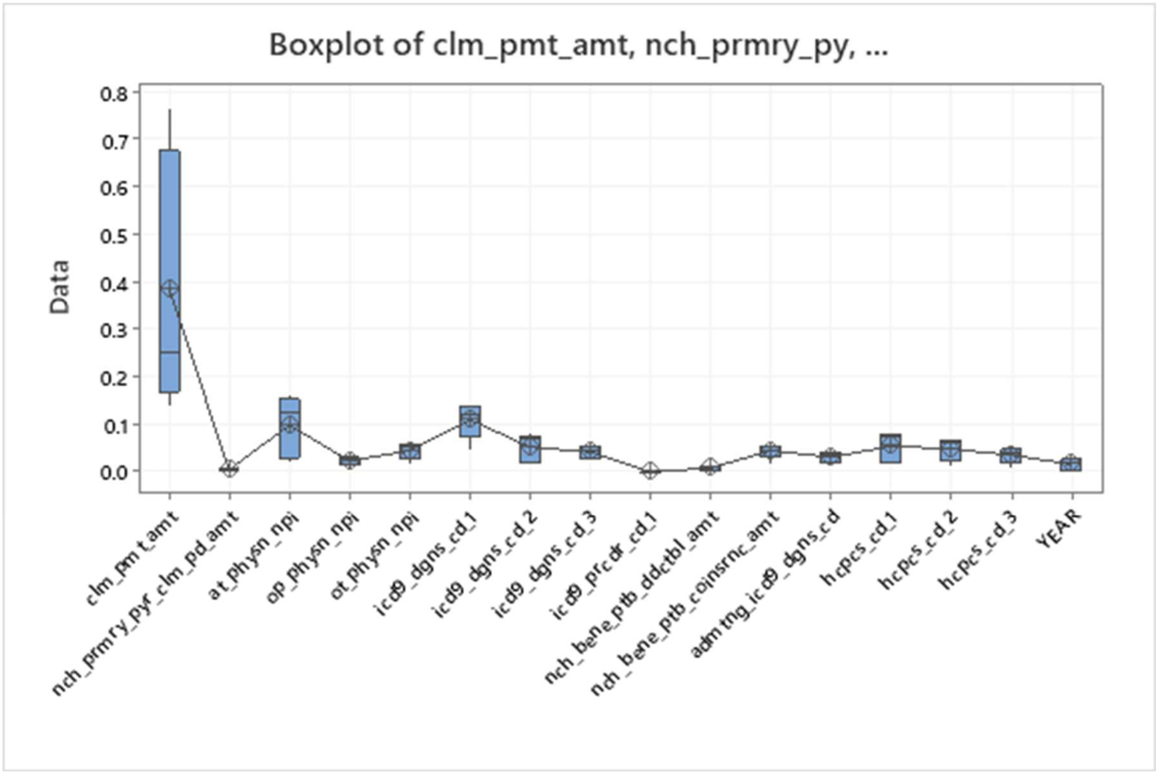
If an interval has zero as an endpoint, the corresponding means are significantly different.



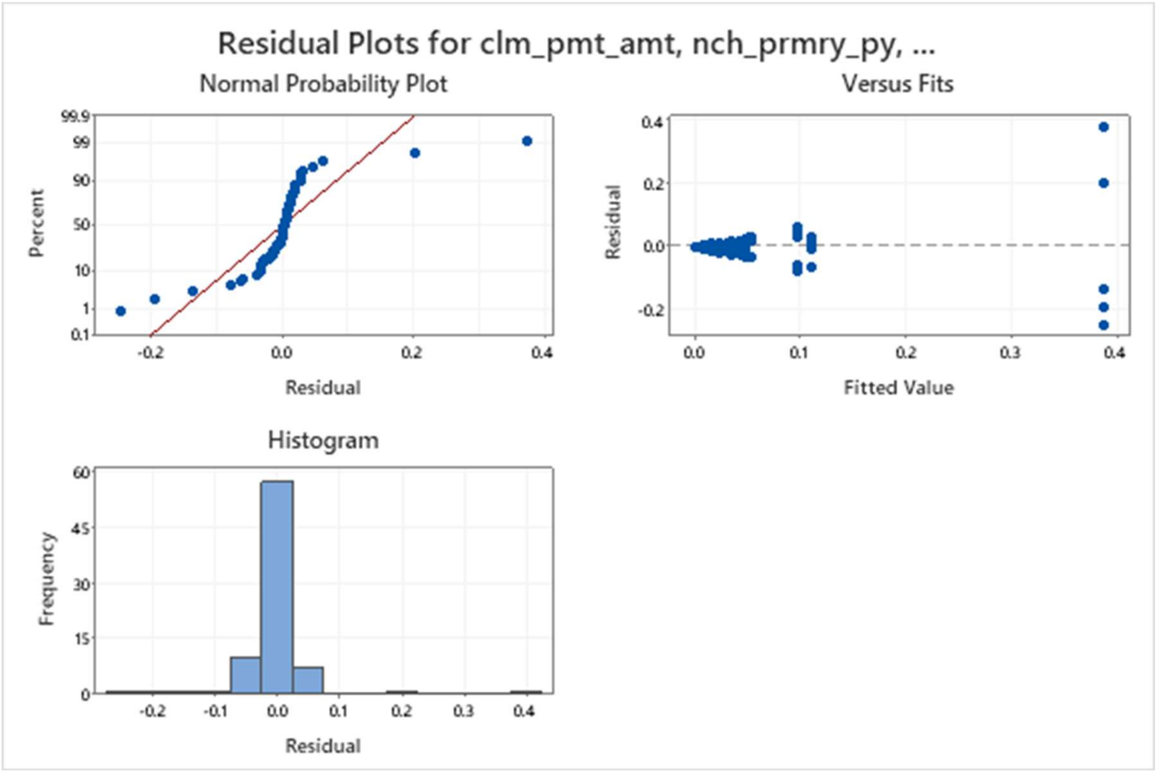
The pooled standard deviation is used to calculate the intervals.



Individual Value Plot of clm_pmt_amt, nch_prmry_py, ...
Data
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
clm_pmt_amt
nch_prmry_pyr_clm_pd_amt
at_physn_npi
op_physn_npi
ot_physn_npi
icd9_dgns_cd_1
icd9_dgns_cd_2
icd9_dgns_cd_3
icd9_prcdr_cd_1
nch_bene_ptb_ddctbl_amt
nch_bene_ptb_coinsrnc_amt
admtng_icd9_dgns_cd
hcpcs_cd_1
hcpcs_cd_2
hcpcs_cd_3
YEAR



Boxplot of clm_pmt_amt, nch_prmry_py, ...
Data
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
clm_pmt_amt
nch_prmry_pyr_clm_pd_amt
at_physn_npi
op_physn_npi
ot_physn_npi
icd9_dgns_cd_1
icd9_dgns_cd_2
icd9_dgns_cd_3
icd9_prcdr_cd_1
nch_bene_ptb_ddctbl_amt
nch_bene_ptb_coinsrnc_amt
admtng_icd9_dgns_cd
hcpcs_cd_1
hcpcs_cd_2
hcpcs_cd_3
YEAR



INPATIENT-FIS 102220

ANOVA Inpatient-feature importance score

* NOTE * Cannot draw the interval plot for the Tukey procedure. Interval plots for comparisons are illegible with more than 45 intervals.

* NOTE * Cannot draw the interval plot for the Fisher procedure. Interval plots for comparisons are illegible with more than 45 intervals.

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.05

Equal variances were assumed for the analysis.

Factor Information

Factor	Levels	Values
Factor	19	clm_pmt_amt, nch_prmry_pyr_clm_pd_amt, at_physn_npi, op_physn_npi, ot_physn_npi, admtng_icd9_dgns_cd, clm_pass_thru_per_diem_amt, nch_bene_ip_ddctbl_amt, nch_bene_pta_coinsrnc_lblty_am, nch_bene_blood_ddctbl_lblty_am, clm_utlztn_day_cnt, clm_drg_cd,

icd9_dgns_cd_1, icd9_dgns_cd_2, icd9_dgns_cd_3, hcpcs_cd_1, hcpcs_cd_2, hcpcs_cd_3, YEAR

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	18	0.8838	0.049103	23.60	0.000
Error	76	0.1581	0.002080		
Total	94	1.0419			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
0.0456100	84.83%	81.23%	76.29%

Means

Factor	N	Mean	StDev	95% CI
clm_pmt_amt	5	0.3601	0.0765	(0.3195, 0.4007)
nch_prmry_pyr_clm_pd_amt	5	0.2898	0.1713	(0.2492, 0.3304)
at_physn_npi	5	0.0441	0.0285	(0.0035, 0.0848)
op_physn_npi	5	0.02552	0.01875	(-0.01511, 0.06614)
ot_physn_npi	5	0.00694	0.00635	(-0.03368, 0.04757)
admtng_icd9_dgns_cd	5	0.0302	0.0271	(-0.0105, 0.0708)
clm_pass_thru_per_diem_amt	5	0.01288	0.00992	(-0.02774, 0.05351)
nch_bene_ip_ddctbl_amt	5	0.00721	0.00703	(-0.03342, 0.04783)
nch_bene_pta_coinsrnc_lblty_am	5	0.002416	0.002126	(-0.038209, 0.043041)
nch_bene_blood_ddctbl_lblty_am	5	0.002045	0.002101	(-0.038580, 0.042670)
clm_utlztn_day_cnt	5	0.01947	0.01743	(-0.02116, 0.06009)
clm_drg_cd	5	0.08992	0.00754	(0.04930, 0.13055)
icd9_dgns_cd_1	5	0.0352	0.0238	(-0.0054, 0.0758)
icd9_dgns_cd_2	5	0.0334	0.0258	(-0.0072, 0.0740)
icd9_dgns_cd_3	5	0.0332	0.0250	(-0.0074, 0.0739)
hcpcs_cd_1	5	0.000000	0.000000	(-0.040625, 0.040625)
hcpcs_cd_2	5	0.000000	0.000000	(-0.040625, 0.040625)
hcpcs_cd_3	5	0.000000	0.000000	(-0.040625, 0.040625)
YEAR	5	0.00761	0.00539	(-0.03302, 0.04823)

Pooled StDev = 0.0456100

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Factor	N	Mean	Grouping	
clm_pmt_amt	5	0.3601	A	
nch_prmry_pyr_clm_pd_amt	5	0.2898	A	
clm_drg_cd	5	0.08992		B
at_physn_npi	5	0.0441		B
icd9_dgns_cd_1	5	0.0352		B
icd9_dgns_cd_2	5	0.0334		B
icd9_dgns_cd_3	5	0.0332		B
admtng_icd9_dgns_cd	5	0.0302		B
op_physn_npi	5	0.02552		B
clm_utlztn_day_cnt	5	0.01947		B
clm_pass_thru_per_diem_amt	5	0.01288		B

Factor	N	Mean	Grouping
YEAR	5	0.00761	B
nch_bene_ip_ddctbl_amt	5	0.00721	B
ot_physn_npi	5	0.00694	B
nch_bene_pta_coinsrnc_lblty_am	5	0.002416	B
nch_bene_blood_ddctbl_lblty_am	5	0.002045	B
hcpcs_cd_3	5	0.000000	B
hcpcs_cd_2	5	0.000000	B
hcpcs_cd_1	5	0.000000	B

Means that do not share a letter are significantly different.

Tukey Simultaneous Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
nch_prmry_py - clm_pmt_amt	-0.0703	0.0288	(-0.1753, 0.0347)	-2.44	0.609
at_physn_npi - clm_pmt_amt	-0.3159	0.0288	(-0.4210, -0.2109)	-10.95	0.000
op_physn_npi - clm_pmt_amt	-0.3346	0.0288	(-0.4396, -0.2295)	-11.60	0.000
ot_physn_npi - clm_pmt_amt	-0.3531	0.0288	(-0.4582, -0.2481)	-12.24	0.000
admtng_icd9_ - clm_pmt_amt	-0.3299	0.0288	(-0.4350, -0.2249)	-11.44	0.000
clm_pass_thr - clm_pmt_amt	-0.3472	0.0288	(-0.4522, -0.2422)	-12.04	0.000
nch_bene_ip_ - clm_pmt_amt	-0.3529	0.0288	(-0.4579, -0.2478)	-12.23	0.000
nch_bene_pta - clm_pmt_amt	-0.3577	0.0288	(-0.4627, -0.2526)	-12.40	0.000
nch_bene_blo - clm_pmt_amt	-0.3580	0.0288	(-0.4631, -0.2530)	-12.41	0.000
clm_utlztn_d - clm_pmt_amt	-0.3406	0.0288	(-0.4457, -0.2356)	-11.81	0.000
clm_drg_cd - clm_pmt_amt	-0.2702	0.0288	(-0.3752, -0.1651)	-9.37	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3249	0.0288	(-0.4299, -0.2199)	-11.26	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3267	0.0288	(-0.4317, -0.2216)	-11.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3269	0.0288	(-0.4319, -0.2218)	-11.33	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3601	0.0288	(-0.4651, -0.2550)	-12.48	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3601	0.0288	(-0.4651, -0.2550)	-12.48	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3601	0.0288	(-0.4651, -0.2550)	-12.48	0.000
YEAR - clm_pmt_amt	-0.3525	0.0288	(-0.4575, -0.2474)	-12.22	0.000
at_physn_npi - nch_prmry_py	-0.2456	0.0288	(-0.3507, -0.1406)	-8.52	0.000
op_physn_npi - nch_prmry_py	-0.2643	0.0288	(-0.3693, -0.1592)	-9.16	0.000
ot_physn_npi - nch_prmry_py	-0.2828	0.0288	(-0.3879, -0.1778)	-9.81	0.000
admtng_icd9_ - nch_prmry_py	-0.2596	0.0288	(-0.3647, -0.1546)	-9.00	0.000
clm_pass_thr - nch_prmry_py	-0.2769	0.0288	(-0.3819, -0.1719)	-9.60	0.000
nch_bene_ip_ - nch_prmry_py	-0.2826	0.0288	(-0.3876, -0.1775)	-9.80	0.000
nch_bene_pta - nch_prmry_py	-0.2874	0.0288	(-0.3924, -0.1823)	-9.96	0.000
nch_bene_blo - nch_prmry_py	-0.2877	0.0288	(-0.3928, -0.1827)	-9.97	0.000
clm_utlztn_d - nch_prmry_py	-0.2703	0.0288	(-0.3754, -0.1653)	-9.37	0.000
clm_drg_cd - nch_prmry_py	-0.1999	0.0288	(-0.3049, -0.0948)	-6.93	0.000
icd9_dgns_cd - nch_prmry_py	-0.2546	0.0288	(-0.3596, -0.1496)	-8.83	0.000
icd9_dgns_cd - nch_prmry_py	-0.2564	0.0288	(-0.3614, -0.1513)	-8.89	0.000
icd9_dgns_cd - nch_prmry_py	-0.2566	0.0288	(-0.3616, -0.1515)	-8.89	0.000
hcpcs_cd_1 - nch_prmry_py	-0.2898	0.0288	(-0.3948, -0.1847)	-10.05	0.000
hcpcs_cd_2 - nch_prmry_py	-0.2898	0.0288	(-0.3948, -0.1847)	-10.05	0.000
hcpcs_cd_3 - nch_prmry_py	-0.2898	0.0288	(-0.3948, -0.1847)	-10.05	0.000
YEAR - nch_prmry_py	-0.2822	0.0288	(-0.3872, -0.1771)	-9.78	0.000
op_physn_npi - at_physn_npi	-0.0186	0.0288	(-0.1237, 0.0864)	-0.65	1.000
ot_physn_npi - at_physn_npi	-0.0372	0.0288	(-0.1423, 0.0678)	-1.29	0.998

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
admtng_icd9_ - at_physn_npi	-0.0140	0.0288	(-0.1190, 0.0910)	-0.49	1.000
clm_pass_thr - at_physn_npi	-0.0313	0.0288	(-0.1363, 0.0738)	-1.08	1.000
nch_bene_ip_ - at_physn_npi	-0.0369	0.0288	(-0.1420, 0.0681)	-1.28	0.999
nch_bene_pta - at_physn_npi	-0.0417	0.0288	(-0.1468, 0.0633)	-1.45	0.994
nch_bene_blo - at_physn_npi	-0.0421	0.0288	(-0.1472, 0.0629)	-1.46	0.993
clm_utlztn_d - at_physn_npi	-0.0247	0.0288	(-0.1297, 0.0804)	-0.86	1.000
clm_drg_cd - at_physn_npi	0.0458	0.0288	(-0.0593, 0.1508)	1.59	0.984
icd9_dgns_cd - at_physn_npi	-0.0090	0.0288	(-0.1140, 0.0961)	-0.31	1.000
icd9_dgns_cd - at_physn_npi	-0.0107	0.0288	(-0.1158, 0.0943)	-0.37	1.000
icd9_dgns_cd - at_physn_npi	-0.0109	0.0288	(-0.1160, 0.0941)	-0.38	1.000
hcpcs_cd_1 - at_physn_npi	-0.0441	0.0288	(-0.1492, 0.0609)	-1.53	0.989
hcpcs_cd_2 - at_physn_npi	-0.0441	0.0288	(-0.1492, 0.0609)	-1.53	0.989
hcpcs_cd_3 - at_physn_npi	-0.0441	0.0288	(-0.1492, 0.0609)	-1.53	0.989
YEAR - at_physn_npi	-0.0365	0.0288	(-0.1416, 0.0685)	-1.27	0.999
ot_physn_npi - op_physn_npi	-0.0186	0.0288	(-0.1236, 0.0865)	-0.64	1.000
admtng_icd9_ - op_physn_npi	0.0046	0.0288	(-0.1004, 0.1097)	0.16	1.000
clm_pass_thr - op_physn_npi	-0.0126	0.0288	(-0.1177, 0.0924)	-0.44	1.000
nch_bene_ip_ - op_physn_npi	-0.0183	0.0288	(-0.1234, 0.0867)	-0.63	1.000
nch_bene_pta - op_physn_npi	-0.0231	0.0288	(-0.1281, 0.0819)	-0.80	1.000
nch_bene_blo - op_physn_npi	-0.0235	0.0288	(-0.1285, 0.0816)	-0.81	1.000
clm_utlztn_d - op_physn_npi	-0.0060	0.0288	(-0.1111, 0.0990)	-0.21	1.000
clm_drg_cd - op_physn_npi	0.0644	0.0288	(-0.0406, 0.1695)	2.23	0.750
icd9_dgns_cd - op_physn_npi	0.0097	0.0288	(-0.0954, 0.1147)	0.34	1.000
icd9_dgns_cd - op_physn_npi	0.0079	0.0288	(-0.0971, 0.1129)	0.27	1.000
icd9_dgns_cd - op_physn_npi	0.0077	0.0288	(-0.0973, 0.1128)	0.27	1.000
hcpcs_cd_1 - op_physn_npi	-0.0255	0.0288	(-0.1306, 0.0795)	-0.88	1.000
hcpcs_cd_2 - op_physn_npi	-0.0255	0.0288	(-0.1306, 0.0795)	-0.88	1.000
hcpcs_cd_3 - op_physn_npi	-0.0255	0.0288	(-0.1306, 0.0795)	-0.88	1.000
YEAR - op_physn_npi	-0.0179	0.0288	(-0.1230, 0.0871)	-0.62	1.000
admtng_icd9_ - ot_physn_npi	0.0232	0.0288	(-0.0818, 0.1283)	0.80	1.000
clm_pass_thr - ot_physn_npi	0.0059	0.0288	(-0.0991, 0.1110)	0.21	1.000
nch_bene_ip_ - ot_physn_npi	0.0003	0.0288	(-0.1048, 0.1053)	0.01	1.000
nch_bene_pta - ot_physn_npi	-0.0045	0.0288	(-0.1096, 0.1005)	-0.16	1.000
nch_bene_blo - ot_physn_npi	-0.0049	0.0288	(-0.1099, 0.1001)	-0.17	1.000
clm_utlztn_d - ot_physn_npi	0.0125	0.0288	(-0.0925, 0.1176)	0.43	1.000
clm_drg_cd - ot_physn_npi	0.0830	0.0288	(-0.0221, 0.1880)	2.88	0.310
icd9_dgns_cd - ot_physn_npi	0.0282	0.0288	(-0.0768, 0.1333)	0.98	1.000
icd9_dgns_cd - ot_physn_npi	0.0265	0.0288	(-0.0786, 0.1315)	0.92	1.000
icd9_dgns_cd - ot_physn_npi	0.0263	0.0288	(-0.0788, 0.1313)	0.91	1.000
hcpcs_cd_1 - ot_physn_npi	-0.0069	0.0288	(-0.1120, 0.0981)	-0.24	1.000
hcpcs_cd_2 - ot_physn_npi	-0.0069	0.0288	(-0.1120, 0.0981)	-0.24	1.000
hcpcs_cd_3 - ot_physn_npi	-0.0069	0.0288	(-0.1120, 0.0981)	-0.24	1.000
YEAR - ot_physn_npi	0.0007	0.0288	(-0.1044, 0.1057)	0.02	1.000
clm_pass_thr - admtng_icd9_	-0.0173	0.0288	(-0.1223, 0.0878)	-0.60	1.000
nch_bene_ip_ - admtng_icd9_	-0.0229	0.0288	(-0.1280, 0.0821)	-0.80	1.000
nch_bene_pta - admtng_icd9_	-0.0277	0.0288	(-0.1328, 0.0773)	-0.96	1.000
nch_bene_blo - admtng_icd9_	-0.0281	0.0288	(-0.1332, 0.0769)	-0.97	1.000
clm_utlztn_d - admtng_icd9_	-0.0107	0.0288	(-0.1157, 0.0944)	-0.37	1.000
clm_drg_cd - admtng_icd9_	0.0598	0.0288	(-0.0453, 0.1648)	2.07	0.843
icd9_dgns_cd - admtng_icd9_	0.0050	0.0288	(-0.1000, 0.1101)	0.17	1.000
icd9_dgns_cd - admtng_icd9_	0.0033	0.0288	(-0.1018, 0.1083)	0.11	1.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
icd9_dgns_cd - admtng_icd9_	0.0031	0.0288	(-0.1020, 0.1081)	0.11	1.000
hcpcs_cd_1 - admtng_icd9_	-0.0302	0.0288	(-0.1352, 0.0749)	-1.05	1.000
hcpcs_cd_2 - admtng_icd9_	-0.0302	0.0288	(-0.1352, 0.0749)	-1.05	1.000
hcpcs_cd_3 - admtng_icd9_	-0.0302	0.0288	(-0.1352, 0.0749)	-1.05	1.000
YEAR - admtng_icd9_	-0.0225	0.0288	(-0.1276, 0.0825)	-0.78	1.000
nch_bene_ip_ - clm_pass_thr	-0.0057	0.0288	(-0.1107, 0.0994)	-0.20	1.000
nch_bene_pta - clm_pass_thr	-0.0105	0.0288	(-0.1155, 0.0946)	-0.36	1.000
nch_bene_blo - clm_pass_thr	-0.0108	0.0288	(-0.1159, 0.0942)	-0.38	1.000
clm_utlztn_d - clm_pass_thr	0.0066	0.0288	(-0.0985, 0.1116)	0.23	1.000
clm_drg_cd - clm_pass_thr	0.0770	0.0288	(-0.0280, 0.1821)	2.67	0.442
icd9_dgns_cd - clm_pass_thr	0.0223	0.0288	(-0.0827, 0.1273)	0.77	1.000
icd9_dgns_cd - clm_pass_thr	0.0205	0.0288	(-0.0845, 0.1256)	0.71	1.000
icd9_dgns_cd - clm_pass_thr	0.0203	0.0288	(-0.0847, 0.1254)	0.71	1.000
hcpcs_cd_1 - clm_pass_thr	-0.0129	0.0288	(-0.1179, 0.0922)	-0.45	1.000
hcpcs_cd_2 - clm_pass_thr	-0.0129	0.0288	(-0.1179, 0.0922)	-0.45	1.000
hcpcs_cd_3 - clm_pass_thr	-0.0129	0.0288	(-0.1179, 0.0922)	-0.45	1.000
YEAR - clm_pass_thr	-0.0053	0.0288	(-0.1103, 0.0998)	-0.18	1.000
nch_bene_pta - nch_bene_ip_	-0.0048	0.0288	(-0.1098, 0.1003)	-0.17	1.000
nch_bene_blo - nch_bene_ip_	-0.0052	0.0288	(-0.1102, 0.0999)	-0.18	1.000
clm_utlztn_d - nch_bene_ip_	0.0123	0.0288	(-0.0928, 0.1173)	0.43	1.000
clm_drg_cd - nch_bene_ip_	0.0827	0.0288	(-0.0223, 0.1878)	2.87	0.315
icd9_dgns_cd - nch_bene_ip_	0.0280	0.0288	(-0.0771, 0.1330)	0.97	1.000
icd9_dgns_cd - nch_bene_ip_	0.0262	0.0288	(-0.0788, 0.1313)	0.91	1.000
icd9_dgns_cd - nch_bene_ip_	0.0260	0.0288	(-0.0790, 0.1311)	0.90	1.000
hcpcs_cd_1 - nch_bene_ip_	-0.0072	0.0288	(-0.1123, 0.0978)	-0.25	1.000
hcpcs_cd_2 - nch_bene_ip_	-0.0072	0.0288	(-0.1123, 0.0978)	-0.25	1.000
hcpcs_cd_3 - nch_bene_ip_	-0.0072	0.0288	(-0.1123, 0.0978)	-0.25	1.000
YEAR - nch_bene_ip_	0.0004	0.0288	(-0.1046, 0.1055)	0.01	1.000
nch_bene_blo - nch_bene_pta	-0.0004	0.0288	(-0.1054, 0.1047)	-0.01	1.000
clm_utlztn_d - nch_bene_pta	0.0171	0.0288	(-0.0880, 0.1221)	0.59	1.000
clm_drg_cd - nch_bene_pta	0.0875	0.0288	(-0.0175, 0.1926)	3.03	0.227
icd9_dgns_cd - nch_bene_pta	0.0328	0.0288	(-0.0723, 0.1378)	1.14	1.000
icd9_dgns_cd - nch_bene_pta	0.0310	0.0288	(-0.0740, 0.1360)	1.07	1.000
icd9_dgns_cd - nch_bene_pta	0.0308	0.0288	(-0.0742, 0.1359)	1.07	1.000
hcpcs_cd_1 - nch_bene_pta	-0.0024	0.0288	(-0.1075, 0.1026)	-0.08	1.000
hcpcs_cd_2 - nch_bene_pta	-0.0024	0.0288	(-0.1075, 0.1026)	-0.08	1.000
hcpcs_cd_3 - nch_bene_pta	-0.0024	0.0288	(-0.1075, 0.1026)	-0.08	1.000
YEAR - nch_bene_pta	0.0052	0.0288	(-0.0999, 0.1102)	0.18	1.000
clm_utlztn_d - nch_bene_blo	0.0174	0.0288	(-0.0876, 0.1225)	0.60	1.000
clm_drg_cd - nch_bene_blo	0.0879	0.0288	(-0.0172, 0.1929)	3.05	0.220
icd9_dgns_cd - nch_bene_blo	0.0331	0.0288	(-0.0719, 0.1382)	1.15	1.000
icd9_dgns_cd - nch_bene_blo	0.0314	0.0288	(-0.0737, 0.1364)	1.09	1.000
icd9_dgns_cd - nch_bene_blo	0.0312	0.0288	(-0.0739, 0.1362)	1.08	1.000
hcpcs_cd_1 - nch_bene_blo	-0.0020	0.0288	(-0.1071, 0.1030)	-0.07	1.000
hcpcs_cd_2 - nch_bene_blo	-0.0020	0.0288	(-0.1071, 0.1030)	-0.07	1.000
hcpcs_cd_3 - nch_bene_blo	-0.0020	0.0288	(-0.1071, 0.1030)	-0.07	1.000
YEAR - nch_bene_blo	0.0056	0.0288	(-0.0995, 0.1106)	0.19	1.000
clm_drg_cd - clm_utlztn_d	0.0705	0.0288	(-0.0346, 0.1755)	2.44	0.605
icd9_dgns_cd - clm_utlztn_d	0.0157	0.0288	(-0.0893, 0.1208)	0.54	1.000
icd9_dgns_cd - clm_utlztn_d	0.0139	0.0288	(-0.0911, 0.1190)	0.48	1.000
icd9_dgns_cd - clm_utlztn_d	0.0138	0.0288	(-0.0913, 0.1188)	0.48	1.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
hcpcs_cd_1 - clm_utlztn_d	-0.0195	0.0288	(-0.1245, 0.0856)	-0.67	1.000
hcpcs_cd_2 - clm_utlztn_d	-0.0195	0.0288	(-0.1245, 0.0856)	-0.67	1.000
hcpcs_cd_3 - clm_utlztn_d	-0.0195	0.0288	(-0.1245, 0.0856)	-0.67	1.000
YEAR - clm_utlztn_d	-0.0119	0.0288	(-0.1169, 0.0932)	-0.41	1.000
icd9_dgns_cd - clm_drg_cd	-0.0547	0.0288	(-0.1598, 0.0503)	-1.90	0.918
icd9_dgns_cd - clm_drg_cd	-0.0565	0.0288	(-0.1616, 0.0485)	-1.96	0.895
icd9_dgns_cd - clm_drg_cd	-0.0567	0.0288	(-0.1617, 0.0484)	-1.97	0.892
hcpcs_cd_1 - clm_drg_cd	-0.0899	0.0288	(-0.1950, 0.0151)	-3.12	0.189
hcpcs_cd_2 - clm_drg_cd	-0.0899	0.0288	(-0.1950, 0.0151)	-3.12	0.189
hcpcs_cd_3 - clm_drg_cd	-0.0899	0.0288	(-0.1950, 0.0151)	-3.12	0.189
YEAR - clm_drg_cd	-0.0823	0.0288	(-0.1874, 0.0227)	-2.85	0.323
icd9_dgns_cd - icd9_dgns_cd	-0.0018	0.0288	(-0.1068, 0.1033)	-0.06	1.000
icd9_dgns_cd - icd9_dgns_cd	-0.0020	0.0288	(-0.1070, 0.1031)	-0.07	1.000
hcpcs_cd_1 - icd9_dgns_cd	-0.0352	0.0288	(-0.1402, 0.0699)	-1.22	0.999
hcpcs_cd_2 - icd9_dgns_cd	-0.0352	0.0288	(-0.1402, 0.0699)	-1.22	0.999
hcpcs_cd_3 - icd9_dgns_cd	-0.0352	0.0288	(-0.1402, 0.0699)	-1.22	0.999
YEAR - icd9_dgns_cd	-0.0276	0.0288	(-0.1326, 0.0775)	-0.96	1.000
icd9_dgns_cd - icd9_dgns_cd	-0.0002	0.0288	(-0.1052, 0.1049)	-0.01	1.000
hcpcs_cd_1 - icd9_dgns_cd	-0.0334	0.0288	(-0.1385, 0.0716)	-1.16	1.000
hcpcs_cd_2 - icd9_dgns_cd	-0.0334	0.0288	(-0.1385, 0.0716)	-1.16	1.000
hcpcs_cd_3 - icd9_dgns_cd	-0.0334	0.0288	(-0.1385, 0.0716)	-1.16	1.000
YEAR - icd9_dgns_cd	-0.0258	0.0288	(-0.1309, 0.0792)	-0.89	1.000
hcpcs_cd_1 - icd9_dgns_cd	-0.0332	0.0288	(-0.1383, 0.0718)	-1.15	1.000
hcpcs_cd_2 - icd9_dgns_cd	-0.0332	0.0288	(-0.1383, 0.0718)	-1.15	1.000
hcpcs_cd_3 - icd9_dgns_cd	-0.0332	0.0288	(-0.1383, 0.0718)	-1.15	1.000
YEAR - icd9_dgns_cd	-0.0256	0.0288	(-0.1307, 0.0794)	-0.89	1.000
hcpcs_cd_2 - hcpcs_cd_1	0.0000	0.0288	(-0.1050, 0.1050)	0.00	1.000
hcpcs_cd_3 - hcpcs_cd_1	0.0000	0.0288	(-0.1050, 0.1050)	0.00	1.000
YEAR - hcpcs_cd_1	0.0076	0.0288	(-0.0974, 0.1127)	0.26	1.000
hcpcs_cd_3 - hcpcs_cd_2	0.0000	0.0288	(-0.1050, 0.1050)	0.00	1.000
YEAR - hcpcs_cd_2	0.0076	0.0288	(-0.0974, 0.1127)	0.26	1.000
YEAR - hcpcs_cd_3	0.0076	0.0288	(-0.0974, 0.1127)	0.26	1.000

Individual confidence level = 99.95%

Fisher Pairwise Comparisons

Grouping Information Using the Fisher LSD Method and 95% Confidence

Factor	N	Mean	Grouping			
clm_pmt_amt	5	0.3601	A			
nch_prmry_pyr_clm_pd_amt	5	0.2898		B		
clm_drg_cd	5	0.08992			C	
at_physn_npi	5	0.0441			C	D
icd9_dgns_cd_1	5	0.0352			C	D
icd9_dgns_cd_2	5	0.0334			C	D
icd9_dgns_cd_3	5	0.0332			C	D
admtng_icd9_dgns_cd	5	0.0302				D
op_physn_npi	5	0.02552				D

Factor	N	Mean	Grouping
clm_utlztn_day_cnt	5	0.01947	D
clm_pass_thru_per_diem_amt	5	0.01288	D
YEAR	5	0.00761	D
nch_bene_ip_ddctbl_amt	5	0.00721	D
ot_physn_npi	5	0.00694	D
nch_bene_pta_coinsrnc_lblty_am	5	0.002416	D
nch_bene_blood_ddctbl_lblty_am	5	0.002045	D
hcpcs_cd_3	5	0.000000	D
hcpcs_cd_2	5	0.000000	D
hcpcs_cd_1	5	0.000000	D

Means that do not share a letter are significantly different.

Fisher Individual Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
nch_prmry_py - clm_pmt_amt	-0.0703	0.0288	(-0.1278, -0.0128)	-2.44	0.017
at_physn_npi - clm_pmt_amt	-0.3159	0.0288	(-0.3734, -0.2585)	-10.95	0.000
op_physn_npi - clm_pmt_amt	-0.3346	0.0288	(-0.3920, -0.2771)	-11.60	0.000
ot_physn_npi - clm_pmt_amt	-0.3531	0.0288	(-0.4106, -0.2957)	-12.24	0.000
admtng_icd9_ - clm_pmt_amt	-0.3299	0.0288	(-0.3874, -0.2725)	-11.44	0.000
clm_pass_thr - clm_pmt_amt	-0.3472	0.0288	(-0.4047, -0.2897)	-12.04	0.000
nch_bene_ip_ - clm_pmt_amt	-0.3529	0.0288	(-0.4103, -0.2954)	-12.23	0.000
nch_bene_pta - clm_pmt_amt	-0.3577	0.0288	(-0.4151, -0.3002)	-12.40	0.000
nch_bene_blo - clm_pmt_amt	-0.3580	0.0288	(-0.4155, -0.3006)	-12.41	0.000
clm_utlztn_d - clm_pmt_amt	-0.3406	0.0288	(-0.3981, -0.2832)	-11.81	0.000
clm_drg_cd - clm_pmt_amt	-0.2702	0.0288	(-0.3276, -0.2127)	-9.37	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3249	0.0288	(-0.3824, -0.2674)	-11.26	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3267	0.0288	(-0.3841, -0.2692)	-11.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3269	0.0288	(-0.3843, -0.2694)	-11.33	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3601	0.0288	(-0.4175, -0.3026)	-12.48	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3601	0.0288	(-0.4175, -0.3026)	-12.48	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3601	0.0288	(-0.4175, -0.3026)	-12.48	0.000
YEAR - clm_pmt_amt	-0.3525	0.0288	(-0.4099, -0.2950)	-12.22	0.000
at_physn_npi - nch_prmry_py	-0.2456	0.0288	(-0.3031, -0.1882)	-8.52	0.000
op_physn_npi - nch_prmry_py	-0.2643	0.0288	(-0.3217, -0.2068)	-9.16	0.000
ot_physn_npi - nch_prmry_py	-0.2828	0.0288	(-0.3403, -0.2254)	-9.81	0.000
admtng_icd9_ - nch_prmry_py	-0.2596	0.0288	(-0.3171, -0.2022)	-9.00	0.000
clm_pass_thr - nch_prmry_py	-0.2769	0.0288	(-0.3344, -0.2194)	-9.60	0.000
nch_bene_ip_ - nch_prmry_py	-0.2826	0.0288	(-0.3400, -0.2251)	-9.80	0.000
nch_bene_pta - nch_prmry_py	-0.2874	0.0288	(-0.3448, -0.2299)	-9.96	0.000
nch_bene_blo - nch_prmry_py	-0.2877	0.0288	(-0.3452, -0.2303)	-9.97	0.000
clm_utlztn_d - nch_prmry_py	-0.2703	0.0288	(-0.3278, -0.2129)	-9.37	0.000
clm_drg_cd - nch_prmry_py	-0.1999	0.0288	(-0.2573, -0.1424)	-6.93	0.000
icd9_dgns_cd - nch_prmry_py	-0.2546	0.0288	(-0.3121, -0.1971)	-8.83	0.000
icd9_dgns_cd - nch_prmry_py	-0.2564	0.0288	(-0.3138, -0.1989)	-8.89	0.000
icd9_dgns_cd - nch_prmry_py	-0.2566	0.0288	(-0.3140, -0.1991)	-8.89	0.000
hcpcs_cd_1 - nch_prmry_py	-0.2898	0.0288	(-0.3472, -0.2323)	-10.05	0.000
hcpcs_cd_2 - nch_prmry_py	-0.2898	0.0288	(-0.3472, -0.2323)	-10.05	0.000
hcpcs_cd_3 - nch_prmry_py	-0.2898	0.0288	(-0.3472, -0.2323)	-10.05	0.000
YEAR - nch_prmry_py	-0.2822	0.0288	(-0.3396, -0.2247)	-9.78	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
op_physn_npi - at_physn_npi	-0.0186	0.0288	(-0.0761, 0.0388)	-0.65	0.520
ot_physn_npi - at_physn_npi	-0.0372	0.0288	(-0.0947, 0.0202)	-1.29	0.201
admtng_icd9_ - at_physn_npi	-0.0140	0.0288	(-0.0715, 0.0435)	-0.49	0.629
clm_pass_thr - at_physn_npi	-0.0313	0.0288	(-0.0887, 0.0262)	-1.08	0.282
nch_bene_ip_ - at_physn_npi	-0.0369	0.0288	(-0.0944, 0.0205)	-1.28	0.204
nch_bene_pta - at_physn_npi	-0.0417	0.0288	(-0.0992, 0.0157)	-1.45	0.152
nch_bene_blo - at_physn_npi	-0.0421	0.0288	(-0.0996, 0.0153)	-1.46	0.149
clm_utlztn_d - at_physn_npi	-0.0247	0.0288	(-0.0821, 0.0328)	-0.86	0.395
clm_drg_cd - at_physn_npi	0.0458	0.0288	(-0.0117, 0.1032)	1.59	0.117
icd9_dgns_cd - at_physn_npi	-0.0090	0.0288	(-0.0664, 0.0485)	-0.31	0.757
icd9_dgns_cd - at_physn_npi	-0.0107	0.0288	(-0.0682, 0.0467)	-0.37	0.711
icd9_dgns_cd - at_physn_npi	-0.0109	0.0288	(-0.0684, 0.0465)	-0.38	0.706
hcpcs_cd_1 - at_physn_npi	-0.0441	0.0288	(-0.1016, 0.0133)	-1.53	0.130
hcpcs_cd_2 - at_physn_npi	-0.0441	0.0288	(-0.1016, 0.0133)	-1.53	0.130
hcpcs_cd_3 - at_physn_npi	-0.0441	0.0288	(-0.1016, 0.0133)	-1.53	0.130
YEAR - at_physn_npi	-0.0365	0.0288	(-0.0940, 0.0209)	-1.27	0.209
ot_physn_npi - op_physn_npi	-0.0186	0.0288	(-0.0760, 0.0389)	-0.64	0.522
admtng_icd9_ - op_physn_npi	0.0046	0.0288	(-0.0528, 0.0621)	0.16	0.873
clm_pass_thr - op_physn_npi	-0.0126	0.0288	(-0.0701, 0.0448)	-0.44	0.663
nch_bene_ip_ - op_physn_npi	-0.0183	0.0288	(-0.0758, 0.0391)	-0.63	0.527
nch_bene_pta - op_physn_npi	-0.0231	0.0288	(-0.0806, 0.0344)	-0.80	0.426
nch_bene_blo - op_physn_npi	-0.0235	0.0288	(-0.0809, 0.0340)	-0.81	0.418
clm_utlztn_d - op_physn_npi	-0.0060	0.0288	(-0.0635, 0.0514)	-0.21	0.834
clm_drg_cd - op_physn_npi	0.0644	0.0288	(0.0070, 0.1219)	2.23	0.029
icd9_dgns_cd - op_physn_npi	0.0097	0.0288	(-0.0478, 0.0671)	0.34	0.738
icd9_dgns_cd - op_physn_npi	0.0079	0.0288	(-0.0496, 0.0653)	0.27	0.785
icd9_dgns_cd - op_physn_npi	0.0077	0.0288	(-0.0497, 0.0652)	0.27	0.790
hcpcs_cd_1 - op_physn_npi	-0.0255	0.0288	(-0.0830, 0.0319)	-0.88	0.379
hcpcs_cd_2 - op_physn_npi	-0.0255	0.0288	(-0.0830, 0.0319)	-0.88	0.379
hcpcs_cd_3 - op_physn_npi	-0.0255	0.0288	(-0.0830, 0.0319)	-0.88	0.379
YEAR - op_physn_npi	-0.0179	0.0288	(-0.0754, 0.0395)	-0.62	0.537
admtng_icd9_ - ot_physn_npi	0.0232	0.0288	(-0.0342, 0.0807)	0.80	0.424
clm_pass_thr - ot_physn_npi	0.0059	0.0288	(-0.0515, 0.0634)	0.21	0.837
nch_bene_ip_ - ot_physn_npi	0.0003	0.0288	(-0.0572, 0.0577)	0.01	0.993
nch_bene_pta - ot_physn_npi	-0.0045	0.0288	(-0.0620, 0.0529)	-0.16	0.876
nch_bene_blo - ot_physn_npi	-0.0049	0.0288	(-0.0623, 0.0526)	-0.17	0.866
clm_utlztn_d - ot_physn_npi	0.0125	0.0288	(-0.0449, 0.0700)	0.43	0.665
clm_drg_cd - ot_physn_npi	0.0830	0.0288	(0.0255, 0.1404)	2.88	0.005
icd9_dgns_cd - ot_physn_npi	0.0282	0.0288	(-0.0292, 0.0857)	0.98	0.331
icd9_dgns_cd - ot_physn_npi	0.0265	0.0288	(-0.0310, 0.0839)	0.92	0.362
icd9_dgns_cd - ot_physn_npi	0.0263	0.0288	(-0.0312, 0.0837)	0.91	0.365
hcpcs_cd_1 - ot_physn_npi	-0.0069	0.0288	(-0.0644, 0.0505)	-0.24	0.810
hcpcs_cd_2 - ot_physn_npi	-0.0069	0.0288	(-0.0644, 0.0505)	-0.24	0.810
hcpcs_cd_3 - ot_physn_npi	-0.0069	0.0288	(-0.0644, 0.0505)	-0.24	0.810
YEAR - ot_physn_npi	0.0007	0.0288	(-0.0568, 0.0581)	0.02	0.982
clm_pass_thr - admtng_icd9_	-0.0173	0.0288	(-0.0747, 0.0402)	-0.60	0.551
nch_bene_ip_ - admtng_icd9_	-0.0229	0.0288	(-0.0804, 0.0345)	-0.80	0.429
nch_bene_pta - admtng_icd9_	-0.0277	0.0288	(-0.0852, 0.0297)	-0.96	0.339
nch_bene_blo - admtng_icd9_	-0.0281	0.0288	(-0.0856, 0.0293)	-0.97	0.333
clm_utlztn_d - admtng_icd9_	-0.0107	0.0288	(-0.0681, 0.0468)	-0.37	0.712
clm_drg_cd - admtng_icd9_	0.0598	0.0288	(0.0023, 0.1172)	2.07	0.042

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
icd9_dgns_cd - admtng_icd9_	0.0050	0.0288	(-0.0524, 0.0625)	0.17	0.862
icd9_dgns_cd - admtng_icd9_	0.0033	0.0288	(-0.0542, 0.0607)	0.11	0.910
icd9_dgns_cd - admtng_icd9_	0.0031	0.0288	(-0.0544, 0.0605)	0.11	0.915
hcpcs_cd_1 - admtng_icd9_	-0.0302	0.0288	(-0.0876, 0.0273)	-1.05	0.299
hcpcs_cd_2 - admtng_icd9_	-0.0302	0.0288	(-0.0876, 0.0273)	-1.05	0.299
hcpcs_cd_3 - admtng_icd9_	-0.0302	0.0288	(-0.0876, 0.0273)	-1.05	0.299
YEAR - admtng_icd9_	-0.0225	0.0288	(-0.0800, 0.0349)	-0.78	0.437
nch_bene_ip_ - clm_pass_thr	-0.0057	0.0288	(-0.0631, 0.0518)	-0.20	0.845
nch_bene_pta - clm_pass_thr	-0.0105	0.0288	(-0.0679, 0.0470)	-0.36	0.718
nch_bene_blo - clm_pass_thr	-0.0108	0.0288	(-0.0683, 0.0466)	-0.38	0.708
clm_utlztn_d - clm_pass_thr	0.0066	0.0288	(-0.0509, 0.0640)	0.23	0.820
clm_drg_cd - clm_pass_thr	0.0770	0.0288	(0.0196, 0.1345)	2.67	0.009
icd9_dgns_cd - clm_pass_thr	0.0223	0.0288	(-0.0352, 0.0798)	0.77	0.442
icd9_dgns_cd - clm_pass_thr	0.0205	0.0288	(-0.0369, 0.0780)	0.71	0.479
icd9_dgns_cd - clm_pass_thr	0.0203	0.0288	(-0.0371, 0.0778)	0.71	0.483
hcpcs_cd_1 - clm_pass_thr	-0.0129	0.0288	(-0.0703, 0.0446)	-0.45	0.656
hcpcs_cd_2 - clm_pass_thr	-0.0129	0.0288	(-0.0703, 0.0446)	-0.45	0.656
hcpcs_cd_3 - clm_pass_thr	-0.0129	0.0288	(-0.0703, 0.0446)	-0.45	0.656
YEAR - clm_pass_thr	-0.0053	0.0288	(-0.0627, 0.0522)	-0.18	0.855
nch_bene_pta - nch_bene_ip_	-0.0048	0.0288	(-0.0622, 0.0527)	-0.17	0.869
nch_bene_blo - nch_bene_ip_	-0.0052	0.0288	(-0.0626, 0.0523)	-0.18	0.858
clm_utlztn_d - nch_bene_ip_	0.0123	0.0288	(-0.0452, 0.0697)	0.43	0.672
clm_drg_cd - nch_bene_ip_	0.0827	0.0288	(0.0253, 0.1402)	2.87	0.005
icd9_dgns_cd - nch_bene_ip_	0.0280	0.0288	(-0.0295, 0.0854)	0.97	0.335
icd9_dgns_cd - nch_bene_ip_	0.0262	0.0288	(-0.0312, 0.0837)	0.91	0.366
icd9_dgns_cd - nch_bene_ip_	0.0260	0.0288	(-0.0314, 0.0835)	0.90	0.370
hcpcs_cd_1 - nch_bene_ip_	-0.0072	0.0288	(-0.0647, 0.0502)	-0.25	0.803
hcpcs_cd_2 - nch_bene_ip_	-0.0072	0.0288	(-0.0647, 0.0502)	-0.25	0.803
hcpcs_cd_3 - nch_bene_ip_	-0.0072	0.0288	(-0.0647, 0.0502)	-0.25	0.803
YEAR - nch_bene_ip_	0.0004	0.0288	(-0.0570, 0.0579)	0.01	0.989
nch_bene_blo - nch_bene_pta	-0.0004	0.0288	(-0.0578, 0.0571)	-0.01	0.990
clm_utlztn_d - nch_bene_pta	0.0171	0.0288	(-0.0404, 0.0745)	0.59	0.556
clm_drg_cd - nch_bene_pta	0.0875	0.0288	(0.0301, 0.1450)	3.03	0.003
icd9_dgns_cd - nch_bene_pta	0.0328	0.0288	(-0.0247, 0.0902)	1.14	0.260
icd9_dgns_cd - nch_bene_pta	0.0310	0.0288	(-0.0265, 0.0885)	1.07	0.286
icd9_dgns_cd - nch_bene_pta	0.0308	0.0288	(-0.0266, 0.0883)	1.07	0.289
hcpcs_cd_1 - nch_bene_pta	-0.0024	0.0288	(-0.0599, 0.0550)	-0.08	0.933
hcpcs_cd_2 - nch_bene_pta	-0.0024	0.0288	(-0.0599, 0.0550)	-0.08	0.933
hcpcs_cd_3 - nch_bene_pta	-0.0024	0.0288	(-0.0599, 0.0550)	-0.08	0.933
YEAR - nch_bene_pta	0.0052	0.0288	(-0.0523, 0.0626)	0.18	0.858
clm_utlztn_d - nch_bene_blo	0.0174	0.0288	(-0.0400, 0.0749)	0.60	0.548
clm_drg_cd - nch_bene_blo	0.0879	0.0288	(0.0304, 0.1453)	3.05	0.003
icd9_dgns_cd - nch_bene_blo	0.0331	0.0288	(-0.0243, 0.0906)	1.15	0.254
icd9_dgns_cd - nch_bene_blo	0.0314	0.0288	(-0.0261, 0.0888)	1.09	0.280
icd9_dgns_cd - nch_bene_blo	0.0312	0.0288	(-0.0263, 0.0886)	1.08	0.283
hcpcs_cd_1 - nch_bene_blo	-0.0020	0.0288	(-0.0595, 0.0554)	-0.07	0.944
hcpcs_cd_2 - nch_bene_blo	-0.0020	0.0288	(-0.0595, 0.0554)	-0.07	0.944
hcpcs_cd_3 - nch_bene_blo	-0.0020	0.0288	(-0.0595, 0.0554)	-0.07	0.944
YEAR - nch_bene_blo	0.0056	0.0288	(-0.0519, 0.0630)	0.19	0.848
clm_drg_cd - clm_utlztn_d	0.0705	0.0288	(0.0130, 0.1279)	2.44	0.017
icd9_dgns_cd - clm_utlztn_d	0.0157	0.0288	(-0.0417, 0.0732)	0.54	0.587

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
icd9_dgns_cd - clm_utlztn_d	0.0139	0.0288	(-0.0435, 0.0714)	0.48	0.630
icd9_dgns_cd - clm_utlztn_d	0.0138	0.0288	(-0.0437, 0.0712)	0.48	0.635
hcpcs_cd_1 - clm_utlztn_d	-0.0195	0.0288	(-0.0769, 0.0380)	-0.67	0.502
hcpcs_cd_2 - clm_utlztn_d	-0.0195	0.0288	(-0.0769, 0.0380)	-0.67	0.502
hcpcs_cd_3 - clm_utlztn_d	-0.0195	0.0288	(-0.0769, 0.0380)	-0.67	0.502
YEAR - clm_utlztn_d	-0.0119	0.0288	(-0.0693, 0.0456)	-0.41	0.682
icd9_dgns_cd - clm_drg_cd	-0.0547	0.0288	(-0.1122, 0.0027)	-1.90	0.062
icd9_dgns_cd - clm_drg_cd	-0.0565	0.0288	(-0.1140, 0.0009)	-1.96	0.054
icd9_dgns_cd - clm_drg_cd	-0.0567	0.0288	(-0.1141, 0.0008)	-1.97	0.053
hcpcs_cd_1 - clm_drg_cd	-0.0899	0.0288	(-0.1474, -0.0325)	-3.12	0.003
hcpcs_cd_2 - clm_drg_cd	-0.0899	0.0288	(-0.1474, -0.0325)	-3.12	0.003
hcpcs_cd_3 - clm_drg_cd	-0.0899	0.0288	(-0.1474, -0.0325)	-3.12	0.003
YEAR - clm_drg_cd	-0.0823	0.0288	(-0.1398, -0.0249)	-2.85	0.006
icd9_dgns_cd - icd9_dgns_cd	-0.0018	0.0288	(-0.0592, 0.0557)	-0.06	0.951
icd9_dgns_cd - icd9_dgns_cd	-0.0020	0.0288	(-0.0594, 0.0555)	-0.07	0.946
hcpcs_cd_1 - icd9_dgns_cd	-0.0352	0.0288	(-0.0926, 0.0223)	-1.22	0.226
hcpcs_cd_2 - icd9_dgns_cd	-0.0352	0.0288	(-0.0926, 0.0223)	-1.22	0.226
hcpcs_cd_3 - icd9_dgns_cd	-0.0352	0.0288	(-0.0926, 0.0223)	-1.22	0.226
YEAR - icd9_dgns_cd	-0.0276	0.0288	(-0.0850, 0.0299)	-0.96	0.342
icd9_dgns_cd - icd9_dgns_cd	-0.0002	0.0288	(-0.0576, 0.0573)	-0.01	0.995
hcpcs_cd_1 - icd9_dgns_cd	-0.0334	0.0288	(-0.0909, 0.0240)	-1.16	0.250
hcpcs_cd_2 - icd9_dgns_cd	-0.0334	0.0288	(-0.0909, 0.0240)	-1.16	0.250
hcpcs_cd_3 - icd9_dgns_cd	-0.0334	0.0288	(-0.0909, 0.0240)	-1.16	0.250
YEAR - icd9_dgns_cd	-0.0258	0.0288	(-0.0833, 0.0316)	-0.89	0.374
hcpcs_cd_1 - icd9_dgns_cd	-0.0332	0.0288	(-0.0907, 0.0242)	-1.15	0.253
hcpcs_cd_2 - icd9_dgns_cd	-0.0332	0.0288	(-0.0907, 0.0242)	-1.15	0.253
hcpcs_cd_3 - icd9_dgns_cd	-0.0332	0.0288	(-0.0907, 0.0242)	-1.15	0.253
YEAR - icd9_dgns_cd	-0.0256	0.0288	(-0.0831, 0.0318)	-0.89	0.377
hcpcs_cd_2 - hcpcs_cd_1	0.0000	0.0288	(-0.0575, 0.0575)	0.00	1.000
hcpcs_cd_3 - hcpcs_cd_1	0.0000	0.0288	(-0.0575, 0.0575)	0.00	1.000
YEAR - hcpcs_cd_1	0.0076	0.0288	(-0.0498, 0.0651)	0.26	0.793
hcpcs_cd_3 - hcpcs_cd_2	0.0000	0.0288	(-0.0575, 0.0575)	0.00	1.000
YEAR - hcpcs_cd_2	0.0076	0.0288	(-0.0498, 0.0651)	0.26	0.793
YEAR - hcpcs_cd_3	0.0076	0.0288	(-0.0498, 0.0651)	0.26	0.793

Simultaneous confidence level = 11.92%

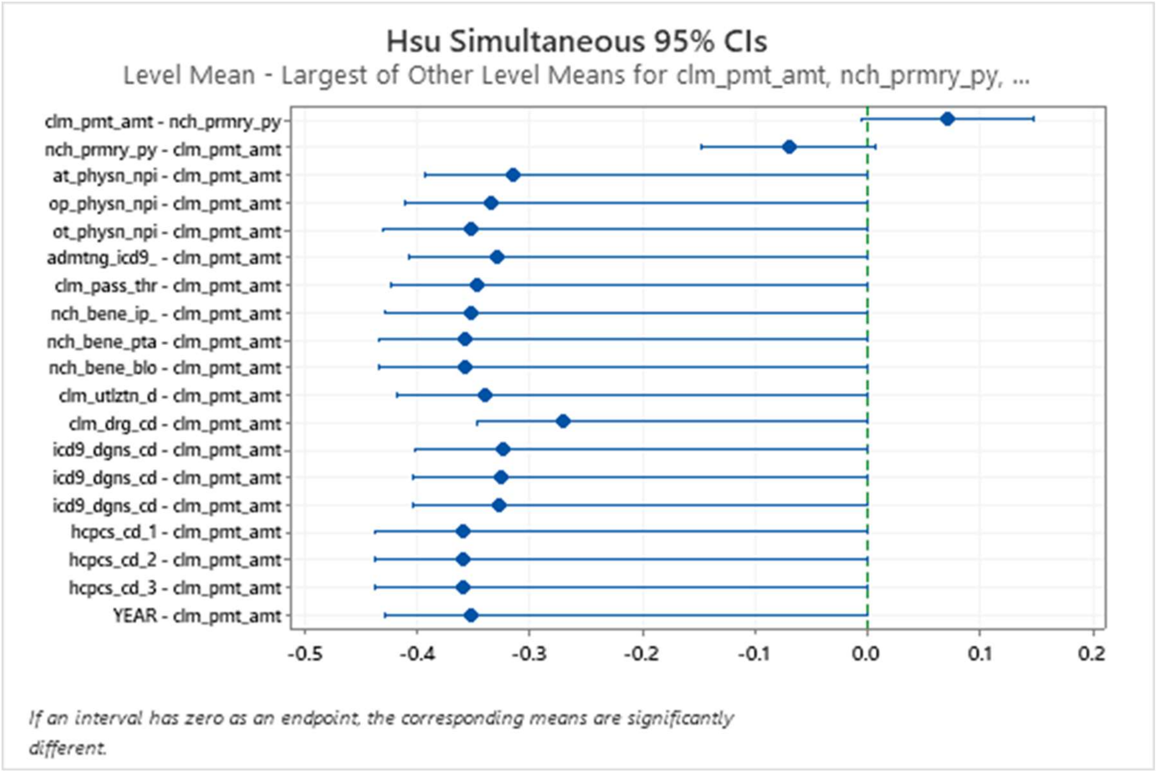
Hsu Multiple Comparisons with the Best (MCB)

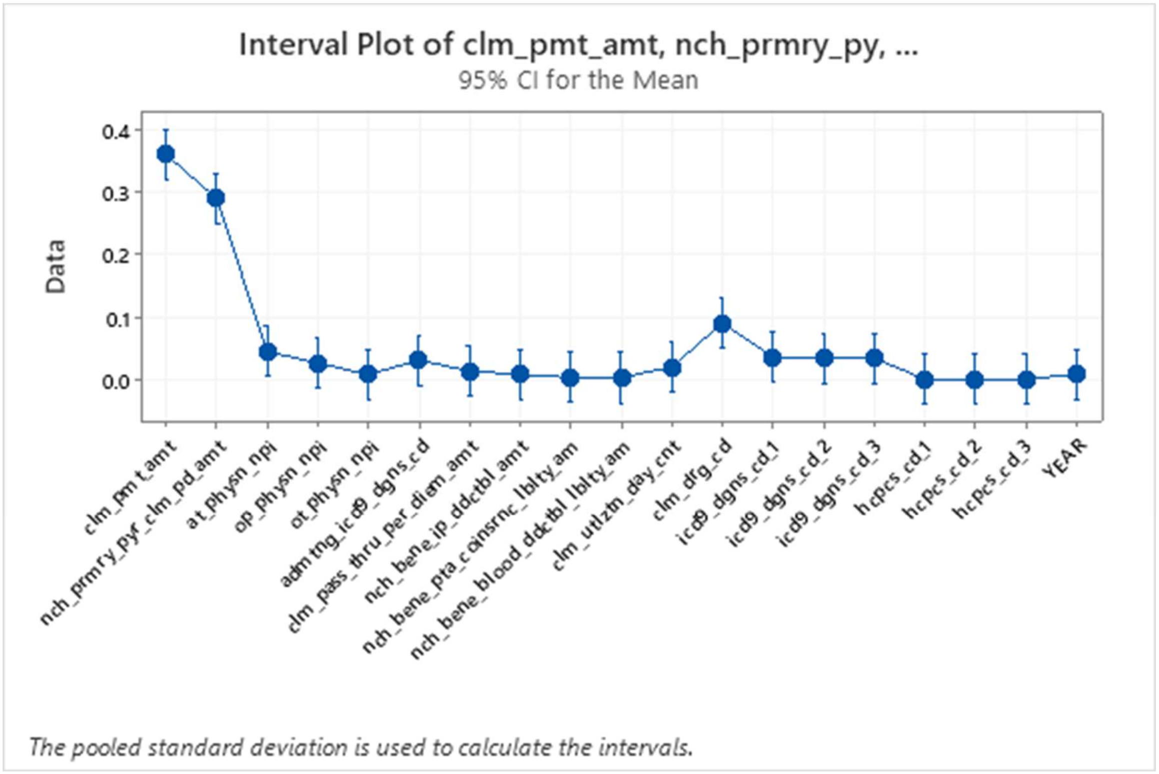
Hsu Simultaneous Tests for Level Mean - Largest of Other Level Means

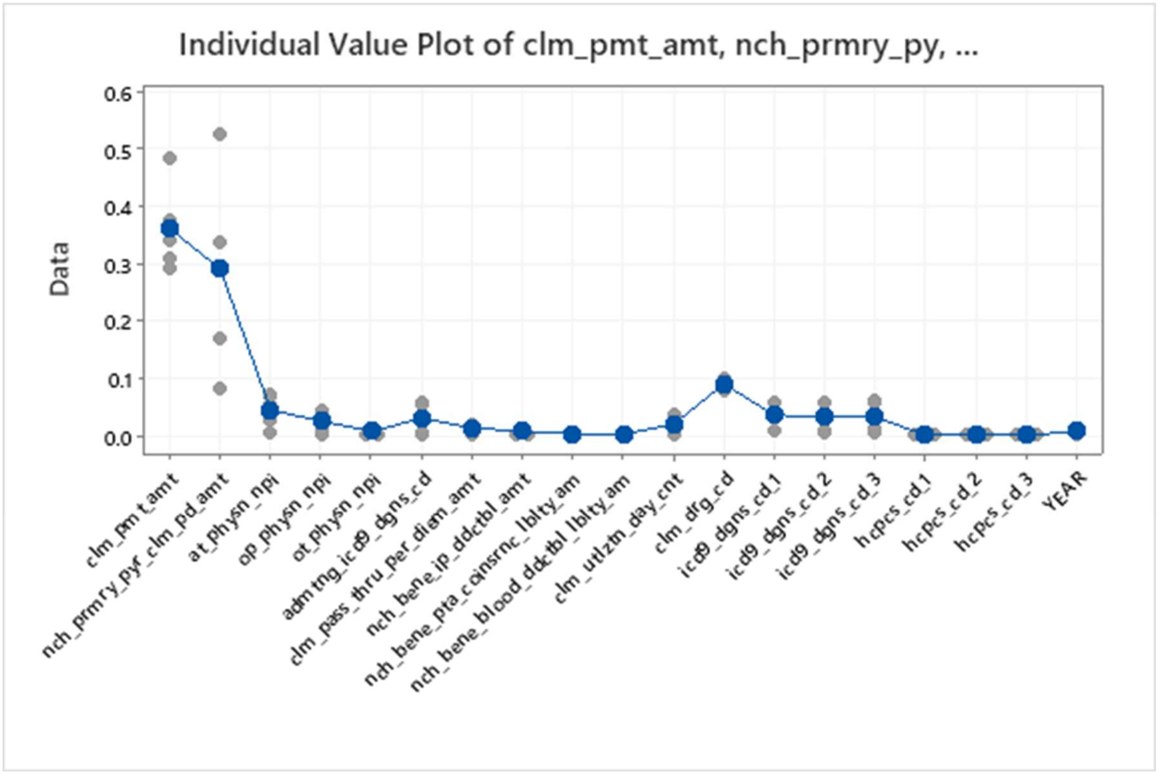
Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
clm_pmt_amt - nch_prmry_py	0.0703	0.0288	(-0.0068, 0.1473)	2.44	0.084
nch_prmry_py - clm_pmt_amt	-0.0703	0.0288	(-0.1473, 0.0068)	-2.44	0.084
at_physn_npi - clm_pmt_amt	-0.3159	0.0288	(-0.3930, 0.0000)	-10.95	0.000
op_physn_npi - clm_pmt_amt	-0.3346	0.0288	(-0.4116, 0.0000)	-11.60	0.000
ot_physn_npi - clm_pmt_amt	-0.3531	0.0288	(-0.4302, 0.0000)	-12.24	0.000
admtng_icd9_ - clm_pmt_amt	-0.3299	0.0288	(-0.4070, 0.0000)	-11.44	0.000

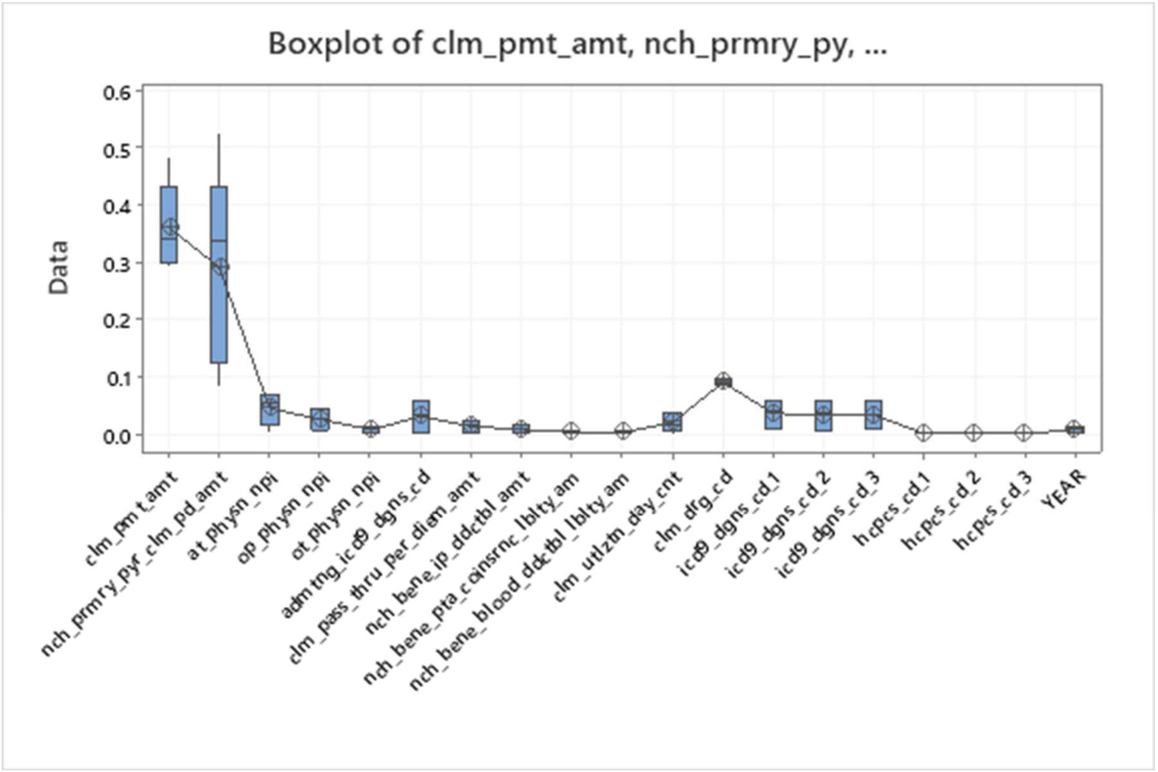
Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
clm_pass_thr - clm_pmt_amt	-0.3472	0.0288	(-0.4243, 0.0000)	-12.04	0.000
nch_bene_ip_ - clm_pmt_amt	-0.3529	0.0288	(-0.4299, 0.0000)	-12.23	0.000
nch_bene_pta - clm_pmt_amt	-0.3577	0.0288	(-0.4347, 0.0000)	-12.40	0.000
nch_bene_blo - clm_pmt_amt	-0.3580	0.0288	(-0.4351, 0.0000)	-12.41	0.000
clm_utlztn_d - clm_pmt_amt	-0.3406	0.0288	(-0.4177, 0.0000)	-11.81	0.000
clm_drg_cd - clm_pmt_amt	-0.2702	0.0288	(-0.3472, 0.0000)	-9.37	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3249	0.0288	(-0.4019, 0.0000)	-11.26	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3267	0.0288	(-0.4037, 0.0000)	-11.32	0.000
icd9_dgns_cd - clm_pmt_amt	-0.3269	0.0288	(-0.4039, 0.0000)	-11.33	0.000
hcpcs_cd_1 - clm_pmt_amt	-0.3601	0.0288	(-0.4371, 0.0000)	-12.48	0.000
hcpcs_cd_2 - clm_pmt_amt	-0.3601	0.0288	(-0.4371, 0.0000)	-12.48	0.000
hcpcs_cd_3 - clm_pmt_amt	-0.3601	0.0288	(-0.4371, 0.0000)	-12.48	0.000
YEAR - clm_pmt_amt	-0.3525	0.0288	(-0.4295, 0.0000)	-12.22	0.000

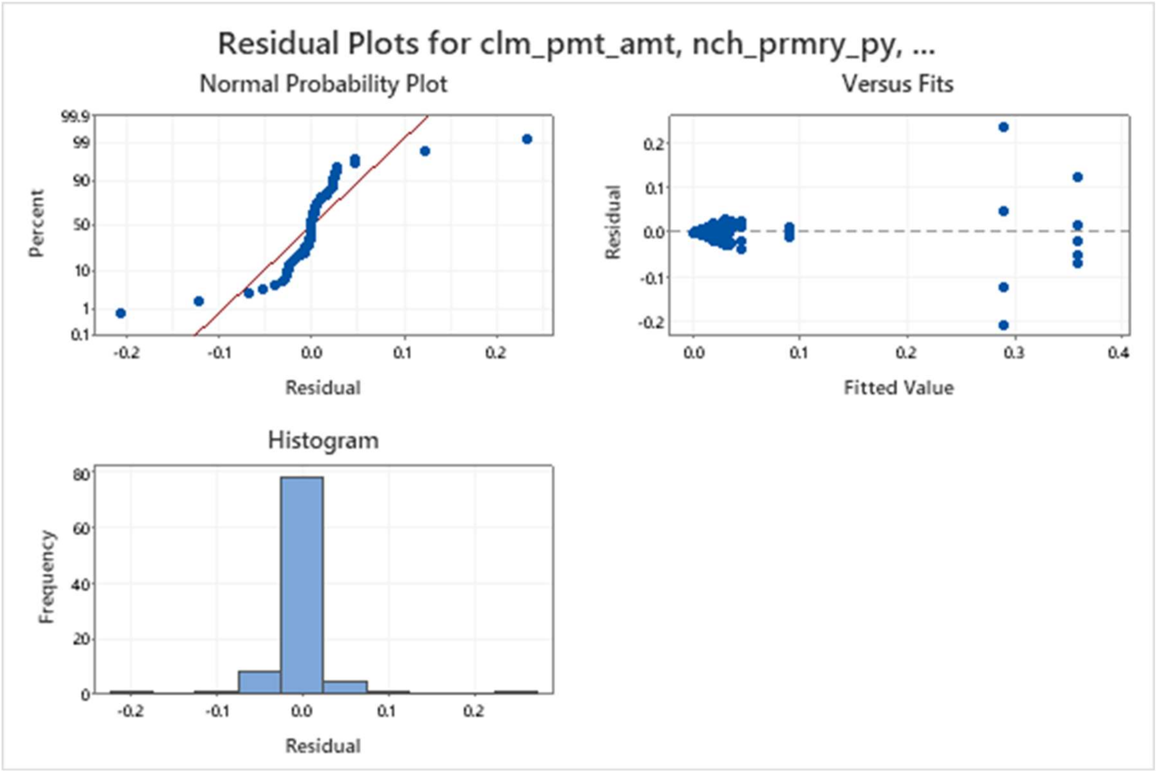
Individual confidence level = 99.08%




Interval Plot of clm_pmt_amt, nch_prmry_py, ...
95% CI for the Mean
Data
0.4
0.3
0.2
0.1
0.0
clm_pmt_amt
nch_prmry_pyr_clm_pd_amt
at_physn_npi
op_physn_npi
ot_physn_npi
admtng_icd9_dgns_cd
clm_pass_thru_per_diem_amt
nch_bene_ip_ddctbl_amt
nch_bene_pta_coinsrnc_lblty_am
nch_bene_blood_ddctbl_lblty_am
clm_utlztn_day_cnt
clm_drg_cd
icd9_dgns_cd_1
icd9_dgns_cd_2
icd9_dgns_cd_3
hcpcs_cd_1
hcpcs_cd_2
hcpcs_cd_3
YEAR
The pooled standard deviation is used to calculate the intervals.


Individual Value Plot of clm_pmt_amt, nch_prmry_py, ...
Data
0.6
0.5
0.4
0.3
0.2
0.1
0.0
clm_pmt_amt
nch_prmry_pyr_clm_pd_amt
at_physn_npi
op_physn_npi
ot_physn_npi
admtng_icd9_dgns_cd
clm_pass_thru_per_diem_amt
nch_bene_ip_ddctbl_amt
nch_bene_pta_coinsrnc_lblty_am
nch_bene_blood_ddctbl_lblty_am
clm_utlztn_day_cnt
clm_drg_cd
icd9_dgns_cd_1
icd9_dgns_cd_2
icd9_dgns_cd_3
hcpcs_cd_1
hcpcs_cd_2
hcpcs_cd_3
YEAR


Boxplot of clm_pmt_amt, nch_prmry_py, ...
Data
0.6
0.5
0.4
0.3
0.2
0.1
0.0
clm_pmt_amt
nch_prmry_pyr_clm_pd_amt
at_physn_npi
op_physn_npi
ot_physn_npi
admtng_icd9_dgns_cd
clm_pass_thru_per_diem_amt
nch_bene_ip_ddctbl_amt
nch_bene_pta_coinsrnc_lblty_am
nch_bene_blood_ddctbl_lblty_am
clm_utlztn_day_cnt
clm_drg_cd
icd9_dgns_cd_1
icd9_dgns_cd_2
icd9_dgns_cd_3
hcpcs_cd_1
hcpcs_cd_2
hcpcs_cd_3
YEAR


Residual Plots for clm_pmt_amt, nch_prmry_py, ...
Normal Probability Plot
Percent
99.9
99
90
50
10
1
0.1
-0.2
-0.1
0.0
0.1
0.2
Residual
Versus Fits
Residual
0.2
0.1
0.0
-0.1
-0.2
0.0
0.1
0.2
0.3
0.4
Fitted Value
Histogram
Frequency
80
60
40
20
0
-0.2
-0.1
0.0
0.1
0.2
Residual

APPENDIX J: DETAIL RESULTS OF ONE=WAY ANOVA TEST ON TESTED ALGORITHMS' PERFORMANCE METRICS

This appendix provides the detail result of the one-way ANOVA test run on the recall and F1-score for tested algorithms on both outpatient and inpatient claims.

Inpatient F1 ANOVA 102220

* NOTE * Cannot draw the interval plot for the Tukey procedure. Interval plots for comparisons are illegible with more than 45 intervals.

* NOTE * Cannot draw the interval plot for the Fisher procedure. Interval plots for comparisons are illegible with more than 45 intervals.

Method

Null hypothesis	All means are equal
Alternative hypothesis	Not all means are equal
Significance level	α = 0.05

Equal variances were assumed for the analysis.

Factor Information

Factor	Levels	Values
Factor	20	DTG, DTG (OS), DTE, DTE (OS), RFG, RFG (OS), RFE, RFE (OS), NB, NB (OS), kNN, kNN (OS), LR, LR (OS), NN, NN (OS), DA, DA (OS), GB, GB (OS)

Analysis of Variance

Source	DF	Adj SS	Adj MS	F-Value	P-Value
Factor	19	8.2962	0.436642	237.27	0.000
Error	80	0.1472	0.001840		
Total	99	8.4434			

Model Summary

S	R-sq	R-sq(adj)	R-sq(pred)
0.0428983	98.26%	97.84%	97.28%

Means

Factor	N	Mean	StDev	95% CI
DTG	5	0.7138	0.0455	(0.6757, 0.7520)
DTG (OS)	5	0.5305	0.0327	(0.4923, 0.5687)

Factor	N	Mean	StDev	95% CI
DTE	5	0.7270	0.0322	(0.6889, 0.7652)
DTE (OS)	5	0.5488	0.0368	(0.5106, 0.5870)
RFG	5	0.7252	0.0642	(0.6870, 0.7634)
RFG (OS)	5	0.5052	0.0381	(0.4671, 0.5434)
RFE	5	0.7122	0.0360	(0.6741, 0.7504)
RFE (OS)	5	0.5235	0.0336	(0.4854, 0.5617)
NB	5	0.3825	0.0402	(0.3443, 0.4206)
NB (OS)	5	0.2015	0.0691	(0.1633, 0.2397)
kNN	5	0.02125	0.01510	(-0.01693, 0.05943)
kNN (OS)	5	0.03712	0.00756	(-0.00106, 0.07530)
LR	5	0.01959	0.01303	(-0.01859, 0.05777)
LR (OS)	5	0.04701	0.00944	(0.00883, 0.08519)
NN	5	0.00766	0.01712	(-0.03052, 0.04583)
NN (OS)	5	0.05532	0.00442	(0.01714, 0.09350)
DA	5	0.6002	0.1078	(0.5620, 0.6383)
DA (OS)	5	0.07102	0.01141	(0.03284, 0.10919)
GB	5	0.7346	0.0438	(0.6964, 0.7728)
GB (OS)	5	0.6633	0.0485	(0.6252, 0.7015)

Pooled StDev = 0.0428983

Tukey Pairwise Comparisons

Grouping Information Using the Tukey Method and 95% Confidence

Factor	N	Mean	Grouping					
GB	5	0.7346	A					
DTE	5	0.7270	A					
RFG	5	0.7252	A					
DTG	5	0.7138	A					
RFE	5	0.7122	A					
GB (OS)	5	0.6633	A	B				
DA	5	0.6002		B	C			
DTE (OS)	5	0.5488			C			
DTG (OS)	5	0.5305			C			
RFE (OS)	5	0.5235			C			
RFG (OS)	5	0.5052			C			
NB	5	0.3825				D		
NB (OS)	5	0.2015					E	
DA (OS)	5	0.07102						F
NN (OS)	5	0.05532						F
LR (OS)	5	0.04701						F
kNN (OS)	5	0.03712						F
kNN	5	0.02125						F
LR	5	0.01959						F
NN	5	0.00766						F

Means that do not share a letter are significantly different.

Tukey Simultaneous Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DTG (OS) - DTG	-0.1834	0.0271	(-0.2827, -0.0840)	-6.76	0.000
DTE - DTG	0.0132	0.0271	(-0.0862, 0.1126)	0.49	1.000
DTE (OS) - DTG	-0.1651	0.0271	(-0.2644, -0.0657)	-6.08	0.000
RFG - DTG	0.0113	0.0271	(-0.0881, 0.1107)	0.42	1.000
RFG (OS) - DTG	-0.2086	0.0271	(-0.3080, -0.1092)	-7.69	0.000
RFE - DTG	-0.0016	0.0271	(-0.1010, 0.0978)	-0.06	1.000
RFE (OS) - DTG	-0.1903	0.0271	(-0.2897, -0.0909)	-7.01	0.000
NB - DTG	-0.3314	0.0271	(-0.4308, -0.2320)	-12.21	0.000
NB (OS) - DTG	-0.5124	0.0271	(-0.6117, -0.4130)	-18.88	0.000
kNN - DTG	-0.6926	0.0271	(-0.7920, -0.5932)	-25.53	0.000
kNN (OS) - DTG	-0.6767	0.0271	(-0.7761, -0.5774)	-24.94	0.000
LR - DTG	-0.6943	0.0271	(-0.7936, -0.5949)	-25.59	0.000
LR (OS) - DTG	-0.6668	0.0271	(-0.7662, -0.5675)	-24.58	0.000
NN - DTG	-0.7062	0.0271	(-0.8056, -0.6068)	-26.03	0.000
NN (OS) - DTG	-0.6585	0.0271	(-0.7579, -0.5592)	-24.27	0.000
DA - DTG	-0.1137	0.0271	(-0.2131, -0.0143)	-4.19	0.010
DA (OS) - DTG	-0.6428	0.0271	(-0.7422, -0.5435)	-23.69	0.000
GB - DTG	0.0208	0.0271	(-0.0786, 0.1201)	0.77	1.000
GB (OS) - DTG	-0.0505	0.0271	(-0.1499, 0.0489)	-1.86	0.940
DTE - DTG (OS)	0.1966	0.0271	(0.0972, 0.2959)	7.24	0.000
DTE (OS) - DTG (OS)	0.0183	0.0271	(-0.0811, 0.1177)	0.67	1.000
RFG - DTG (OS)	0.1947	0.0271	(0.0953, 0.2941)	7.18	0.000
RFG (OS) - DTG (OS)	-0.0253	0.0271	(-0.1246, 0.0741)	-0.93	1.000
RFE - DTG (OS)	0.1818	0.0271	(0.0824, 0.2811)	6.70	0.000
RFE (OS) - DTG (OS)	-0.0070	0.0271	(-0.1063, 0.0924)	-0.26	1.000
NB - DTG (OS)	-0.1480	0.0271	(-0.2474, -0.0487)	-5.46	0.000
NB (OS) - DTG (OS)	-0.3290	0.0271	(-0.4284, -0.2296)	-12.13	0.000
kNN - DTG (OS)	-0.5092	0.0271	(-0.6086, -0.4099)	-18.77	0.000
kNN (OS) - DTG (OS)	-0.4934	0.0271	(-0.5927, -0.3940)	-18.18	0.000
LR - DTG (OS)	-0.5109	0.0271	(-0.6103, -0.4115)	-18.83	0.000
LR (OS) - DTG (OS)	-0.4835	0.0271	(-0.5829, -0.3841)	-17.82	0.000
NN - DTG (OS)	-0.5228	0.0271	(-0.6222, -0.4235)	-19.27	0.000
NN (OS) - DTG (OS)	-0.4752	0.0271	(-0.5745, -0.3758)	-17.51	0.000
DA - DTG (OS)	0.0697	0.0271	(-0.0297, 0.1691)	2.57	0.538
DA (OS) - DTG (OS)	-0.4595	0.0271	(-0.5589, -0.3601)	-16.94	0.000
GB - DTG (OS)	0.2041	0.0271	(0.1048, 0.3035)	7.52	0.000
GB (OS) - DTG (OS)	0.1329	0.0271	(0.0335, 0.2322)	4.90	0.001
DTE (OS) - DTE	-0.1783	0.0271	(-0.2776, -0.0789)	-6.57	0.000
RFG - DTE	-0.0019	0.0271	(-0.1012, 0.0975)	-0.07	1.000
RFG (OS) - DTE	-0.2218	0.0271	(-0.3212, -0.1224)	-8.18	0.000
RFE - DTE	-0.0148	0.0271	(-0.1142, 0.0846)	-0.55	1.000
RFE (OS) - DTE	-0.2035	0.0271	(-0.3029, -0.1041)	-7.50	0.000
NB - DTE	-0.3446	0.0271	(-0.4440, -0.2452)	-12.70	0.000
NB (OS) - DTE	-0.5256	0.0271	(-0.6249, -0.4262)	-19.37	0.000
kNN - DTE	-0.7058	0.0271	(-0.8052, -0.6064)	-26.01	0.000
kNN (OS) - DTE	-0.6899	0.0271	(-0.7893, -0.5906)	-25.43	0.000
LR - DTE	-0.7075	0.0271	(-0.8068, -0.6081)	-26.08	0.000
LR (OS) - DTE	-0.6800	0.0271	(-0.7794, -0.5807)	-25.06	0.000
NN - DTE	-0.7194	0.0271	(-0.8188, -0.6200)	-26.52	0.000
NN (OS) - DTE	-0.6717	0.0271	(-0.7711, -0.5724)	-24.76	0.000
DA - DTE	-0.1269	0.0271	(-0.2263, -0.0275)	-4.68	0.002

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DA (OS) - DTE	-0.6560	0.0271	(-0.7554, -0.5567)	-24.18	0.000
GB - DTE	0.0076	0.0271	(-0.0918, 0.1070)	0.28	1.000
GB (OS) - DTE	-0.0637	0.0271	(-0.1631, 0.0357)	-2.35	0.696
RFG - DTE (OS)	0.1764	0.0271	(0.0770, 0.2758)	6.50	0.000
RFG (OS) - DTE (OS)	-0.0436	0.0271	(-0.1429, 0.0558)	-1.61	0.986
RFE - DTE (OS)	0.1634	0.0271	(0.0641, 0.2628)	6.02	0.000
RFE (OS) - DTE (OS)	-0.0253	0.0271	(-0.1246, 0.0741)	-0.93	1.000
NB - DTE (OS)	-0.1663	0.0271	(-0.2657, -0.0670)	-6.13	0.000
NB (OS) - DTE (OS)	-0.3473	0.0271	(-0.4467, -0.2479)	-12.80	0.000
kNN - DTE (OS)	-0.5275	0.0271	(-0.6269, -0.4282)	-19.44	0.000
kNN (OS) - DTE (OS)	-0.5117	0.0271	(-0.6111, -0.4123)	-18.86	0.000
LR - DTE (OS)	-0.5292	0.0271	(-0.6286, -0.4298)	-19.51	0.000
LR (OS) - DTE (OS)	-0.5018	0.0271	(-0.6012, -0.4024)	-18.49	0.000
NN - DTE (OS)	-0.5411	0.0271	(-0.6405, -0.4418)	-19.95	0.000
NN (OS) - DTE (OS)	-0.4935	0.0271	(-0.5929, -0.3941)	-18.19	0.000
DA - DTE (OS)	0.0514	0.0271	(-0.0480, 0.1507)	1.89	0.931
DA (OS) - DTE (OS)	-0.4778	0.0271	(-0.5772, -0.3784)	-17.61	0.000
GB - DTE (OS)	0.1858	0.0271	(0.0865, 0.2852)	6.85	0.000
GB (OS) - DTE (OS)	0.1146	0.0271	(0.0152, 0.2139)	4.22	0.009
RFG (OS) - RFG	-0.2199	0.0271	(-0.3193, -0.1206)	-8.11	0.000
RFE - RFG	-0.0129	0.0271	(-0.1123, 0.0864)	-0.48	1.000
RFE (OS) - RFG	-0.2016	0.0271	(-0.3010, -0.1023)	-7.43	0.000
NB - RFG	-0.3427	0.0271	(-0.4421, -0.2433)	-12.63	0.000
NB (OS) - RFG	-0.5237	0.0271	(-0.6231, -0.4243)	-19.30	0.000
kNN - RFG	-0.7039	0.0271	(-0.8033, -0.6046)	-25.95	0.000
kNN (OS) - RFG	-0.6881	0.0271	(-0.7874, -0.5887)	-25.36	0.000
LR - RFG	-0.7056	0.0271	(-0.8050, -0.6062)	-26.01	0.000
LR (OS) - RFG	-0.6782	0.0271	(-0.7775, -0.5788)	-25.00	0.000
NN - RFG	-0.7175	0.0271	(-0.8169, -0.6181)	-26.45	0.000
NN (OS) - RFG	-0.6699	0.0271	(-0.7692, -0.5705)	-24.69	0.000
DA - RFG	-0.1250	0.0271	(-0.2244, -0.0256)	-4.61	0.002
DA (OS) - RFG	-0.6542	0.0271	(-0.7535, -0.5548)	-24.11	0.000
GB - RFG	0.0094	0.0271	(-0.0899, 0.1088)	0.35	1.000
GB (OS) - RFG	-0.0618	0.0271	(-0.1612, 0.0376)	-2.28	0.743
RFE - RFG (OS)	0.2070	0.0271	(0.1076, 0.3064)	7.63	0.000
RFE (OS) - RFG (OS)	0.0183	0.0271	(-0.0811, 0.1177)	0.67	1.000
NB - RFG (OS)	-0.1228	0.0271	(-0.2222, -0.0234)	-4.53	0.003
NB (OS) - RFG (OS)	-0.3038	0.0271	(-0.4031, -0.2044)	-11.20	0.000
kNN - RFG (OS)	-0.4840	0.0271	(-0.5834, -0.3846)	-17.84	0.000
kNN (OS) - RFG (OS)	-0.4681	0.0271	(-0.5675, -0.3687)	-17.25	0.000
LR - RFG (OS)	-0.4856	0.0271	(-0.5850, -0.3863)	-17.90	0.000
LR (OS) - RFG (OS)	-0.4582	0.0271	(-0.5576, -0.3588)	-16.89	0.000
NN - RFG (OS)	-0.4976	0.0271	(-0.5970, -0.3982)	-18.34	0.000
NN (OS) - RFG (OS)	-0.4499	0.0271	(-0.5493, -0.3505)	-16.58	0.000
DA - RFG (OS)	0.0949	0.0271	(-0.0044, 0.1943)	3.50	0.079
DA (OS) - RFG (OS)	-0.4342	0.0271	(-0.5336, -0.3348)	-16.00	0.000
GB - RFG (OS)	0.2294	0.0271	(0.1300, 0.3288)	8.45	0.000
GB (OS) - RFG (OS)	0.1581	0.0271	(0.0587, 0.2575)	5.83	0.000
RFE (OS) - RFE	-0.1887	0.0271	(-0.2881, -0.0893)	-6.96	0.000
NB - RFE	-0.3298	0.0271	(-0.4292, -0.2304)	-12.16	0.000
NB (OS) - RFE	-0.5108	0.0271	(-0.6101, -0.4114)	-18.83	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
kNN - RFE	-0.6910	0.0271	(-0.7904, -0.5916)	-25.47	0.000
kNN (OS) - RFE	-0.6751	0.0271	(-0.7745, -0.5757)	-24.88	0.000
LR - RFE	-0.6927	0.0271	(-0.7920, -0.5933)	-25.53	0.000
LR (OS) - RFE	-0.6652	0.0271	(-0.7646, -0.5659)	-24.52	0.000
NN - RFE	-0.7046	0.0271	(-0.8040, -0.6052)	-25.97	0.000
NN (OS) - RFE	-0.6569	0.0271	(-0.7563, -0.5575)	-24.21	0.000
DA - RFE	-0.1121	0.0271	(-0.2115, -0.0127)	-4.13	0.012
DA (OS) - RFE	-0.6412	0.0271	(-0.7406, -0.5418)	-23.63	0.000
GB - RFE	0.0224	0.0271	(-0.0770, 0.1218)	0.82	1.000
GB (OS) - RFE	-0.0489	0.0271	(-0.1483, 0.0505)	-1.80	0.955
NB - RFE (OS)	-0.1411	0.0271	(-0.2405, -0.0417)	-5.20	0.000
NB (OS) - RFE (OS)	-0.3221	0.0271	(-0.4214, -0.2227)	-11.87	0.000
kNN - RFE (OS)	-0.5023	0.0271	(-0.6017, -0.4029)	-18.51	0.000
kNN (OS) - RFE (OS)	-0.4864	0.0271	(-0.5858, -0.3870)	-17.93	0.000
LR - RFE (OS)	-0.5039	0.0271	(-0.6033, -0.4046)	-18.57	0.000
LR (OS) - RFE (OS)	-0.4765	0.0271	(-0.5759, -0.3771)	-17.56	0.000
NN - RFE (OS)	-0.5159	0.0271	(-0.6153, -0.4165)	-19.01	0.000
NN (OS) - RFE (OS)	-0.4682	0.0271	(-0.5676, -0.3688)	-17.26	0.000
DA - RFE (OS)	0.0766	0.0271	(-0.0227, 0.1760)	2.82	0.360
DA (OS) - RFE (OS)	-0.4525	0.0271	(-0.5519, -0.3531)	-16.68	0.000
GB - RFE (OS)	0.2111	0.0271	(0.1117, 0.3105)	7.78	0.000
GB (OS) - RFE (OS)	0.1398	0.0271	(0.0404, 0.2392)	5.15	0.000
NB (OS) - NB	-0.1810	0.0271	(-0.2804, -0.0816)	-6.67	0.000
kNN - NB	-0.3612	0.0271	(-0.4606, -0.2618)	-13.31	0.000
kNN (OS) - NB	-0.3453	0.0271	(-0.4447, -0.2460)	-12.73	0.000
LR - NB	-0.3629	0.0271	(-0.4622, -0.2635)	-13.37	0.000
LR (OS) - NB	-0.3354	0.0271	(-0.4348, -0.2361)	-12.36	0.000
NN - NB	-0.3748	0.0271	(-0.4742, -0.2754)	-13.81	0.000
NN (OS) - NB	-0.3271	0.0271	(-0.4265, -0.2278)	-12.06	0.000
DA - NB	0.2177	0.0271	(0.1183, 0.3171)	8.02	0.000
DA (OS) - NB	-0.3114	0.0271	(-0.4108, -0.2121)	-11.48	0.000
GB - NB	0.3522	0.0271	(0.2528, 0.4515)	12.98	0.000
GB (OS) - NB	0.2809	0.0271	(0.1815, 0.3803)	10.35	0.000
kNN - NB (OS)	-0.1802	0.0271	(-0.2796, -0.0809)	-6.64	0.000
kNN (OS) - NB (OS)	-0.1644	0.0271	(-0.2637, -0.0650)	-6.06	0.000
LR - NB (OS)	-0.1819	0.0271	(-0.2813, -0.0825)	-6.70	0.000
LR (OS) - NB (OS)	-0.1545	0.0271	(-0.2538, -0.0551)	-5.69	0.000
NN - NB (OS)	-0.1938	0.0271	(-0.2932, -0.0944)	-7.14	0.000
NN (OS) - NB (OS)	-0.1462	0.0271	(-0.2455, -0.0468)	-5.39	0.000
DA - NB (OS)	0.3987	0.0271	(0.2993, 0.4981)	14.69	0.000
DA (OS) - NB (OS)	-0.1305	0.0271	(-0.2298, -0.0311)	-4.81	0.001
GB - NB (OS)	0.5331	0.0271	(0.4338, 0.6325)	19.65	0.000
GB (OS) - NB (OS)	0.4619	0.0271	(0.3625, 0.5612)	17.02	0.000
kNN (OS) - kNN	0.0159	0.0271	(-0.0835, 0.1152)	0.58	1.000
LR - kNN	-0.0017	0.0271	(-0.1010, 0.0977)	-0.06	1.000
LR (OS) - kNN	0.0258	0.0271	(-0.0736, 0.1251)	0.95	1.000
NN - kNN	-0.0136	0.0271	(-0.1130, 0.0858)	-0.50	1.000
NN (OS) - kNN	0.0341	0.0271	(-0.0653, 0.1334)	1.26	0.999
DA - kNN	0.5789	0.0271	(0.4795, 0.6783)	21.34	0.000
DA (OS) - kNN	0.0498	0.0271	(-0.0496, 0.1491)	1.83	0.948
GB - kNN	0.7134	0.0271	(0.6140, 0.8128)	26.29	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
GB (OS) - kNN	0.6421	0.0271	(0.5427, 0.7415)	23.67	0.000
LR - kNN (OS)	-0.0175	0.0271	(-0.1169, 0.0818)	-0.65	1.000
LR (OS) - kNN (OS)	0.0099	0.0271	(-0.0895, 0.1093)	0.36	1.000
NN - kNN (OS)	-0.0295	0.0271	(-0.1288, 0.0699)	-1.09	1.000
NN (OS) - kNN (OS)	0.0182	0.0271	(-0.0812, 0.1176)	0.67	1.000
DA - kNN (OS)	0.5630	0.0271	(0.4637, 0.6624)	20.75	0.000
DA (OS) - kNN (OS)	0.0339	0.0271	(-0.0655, 0.1333)	1.25	0.999
GB - kNN (OS)	0.6975	0.0271	(0.5981, 0.7969)	25.71	0.000
GB (OS) - kNN (OS)	0.6262	0.0271	(0.5269, 0.7256)	23.08	0.000
LR (OS) - LR	0.0274	0.0271	(-0.0720, 0.1268)	1.01	1.000
NN - LR	-0.0119	0.0271	(-0.1113, 0.0874)	-0.44	1.000
NN (OS) - LR	0.0357	0.0271	(-0.0636, 0.1351)	1.32	0.999
DA - LR	0.5806	0.0271	(0.4812, 0.6800)	21.40	0.000
DA (OS) - LR	0.0514	0.0271	(-0.0479, 0.1508)	1.90	0.930
GB - LR	0.7150	0.0271	(0.6157, 0.8144)	26.35	0.000
GB (OS) - LR	0.6438	0.0271	(0.5444, 0.7431)	23.73	0.000
NN - LR (OS)	-0.0394	0.0271	(-0.1387, 0.0600)	-1.45	0.995
NN (OS) - LR (OS)	0.0083	0.0271	(-0.0911, 0.1077)	0.31	1.000
DA - LR (OS)	0.5532	0.0271	(0.4538, 0.6525)	20.39	0.000
DA (OS) - LR (OS)	0.0240	0.0271	(-0.0754, 0.1234)	0.88	1.000
GB - LR (OS)	0.6876	0.0271	(0.5882, 0.7870)	25.34	0.000
GB (OS) - LR (OS)	0.6163	0.0271	(0.5170, 0.7157)	22.72	0.000
NN (OS) - NN	0.0477	0.0271	(-0.0517, 0.1470)	1.76	0.965
DA - NN	0.5925	0.0271	(0.4931, 0.6919)	21.84	0.000
DA (OS) - NN	0.0634	0.0271	(-0.0360, 0.1627)	2.34	0.705
GB - NN	0.7270	0.0271	(0.6276, 0.8263)	26.79	0.000
GB (OS) - NN	0.6557	0.0271	(0.5563, 0.7551)	24.17	0.000
DA - NN (OS)	0.5448	0.0271	(0.4455, 0.6442)	20.08	0.000
DA (OS) - NN (OS)	0.0157	0.0271	(-0.0837, 0.1151)	0.58	1.000
GB - NN (OS)	0.6793	0.0271	(0.5799, 0.7787)	25.04	0.000
GB (OS) - NN (OS)	0.6080	0.0271	(0.5087, 0.7074)	22.41	0.000
DA (OS) - DA	-0.5292	0.0271	(-0.6285, -0.4298)	-19.50	0.000
GB - DA	0.1345	0.0271	(0.0351, 0.2338)	4.96	0.001
GB (OS) - DA	0.0632	0.0271	(-0.0362, 0.1626)	2.33	0.709
GB - DA (OS)	0.6636	0.0271	(0.5642, 0.7630)	24.46	0.000
GB (OS) - DA (OS)	0.5923	0.0271	(0.4930, 0.6917)	21.83	0.000
GB (OS) - GB	-0.0713	0.0271	(-0.1707, 0.0281)	-2.63	0.495

Individual confidence level = 99.96%

Fisher Pairwise Comparisons

Grouping Information Using the Fisher LSD Method and 95% Confidence

Factor	N	Mean	Grouping	
GB	5	0.7346	A	
DTE	5	0.7270	A	
RFG	5	0.7252	A	
DTG	5	0.7138	A	B

RFE	5	0.7122	A	B						
GB (OS)	5	0.6633		B						
DA	5	0.6002			C					
DTE (OS)	5	0.5488			C	D				
DTG (OS)	5	0.5305				D				
RFE (OS)	5	0.5235				D				
RFG (OS)	5	0.5052				D				
NB	5	0.3825					E			
NB (OS)	5	0.2015						F		
DA (OS)	5	0.07102							G	
NN (OS)	5	0.05532							G	H
LR (OS)	5	0.04701							G	H
kNN (OS)	5	0.03712							G	H
kNN	5	0.02125							G	H
LR	5	0.01959							G	H
NN	5	0.00766								H

Means that do not share a letter are significantly different.

Fisher Individual Tests for Differences of Means

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DTG (OS) - DTG	-0.1834	0.0271	(-0.2374, -0.1294)	-6.76	0.000
DTE - DTG	0.0132	0.0271	(-0.0408, 0.0672)	0.49	0.628
DTE (OS) - DTG	-0.1651	0.0271	(-0.2190, -0.1111)	-6.08	0.000
RFG - DTG	0.0113	0.0271	(-0.0427, 0.0653)	0.42	0.678
RFG (OS) - DTG	-0.2086	0.0271	(-0.2626, -0.1546)	-7.69	0.000
RFE - DTG	-0.0016	0.0271	(-0.0556, 0.0524)	-0.06	0.953
RFE (OS) - DTG	-0.1903	0.0271	(-0.2443, -0.1363)	-7.01	0.000
NB - DTG	-0.3314	0.0271	(-0.3854, -0.2774)	-12.21	0.000
NB (OS) - DTG	-0.5124	0.0271	(-0.5664, -0.4584)	-18.88	0.000
kNN - DTG	-0.6926	0.0271	(-0.7466, -0.6386)	-25.53	0.000
kNN (OS) - DTG	-0.6767	0.0271	(-0.7307, -0.6227)	-24.94	0.000
LR - DTG	-0.6943	0.0271	(-0.7483, -0.6403)	-25.59	0.000
LR (OS) - DTG	-0.6668	0.0271	(-0.7208, -0.6128)	-24.58	0.000
NN - DTG	-0.7062	0.0271	(-0.7602, -0.6522)	-26.03	0.000
NN (OS) - DTG	-0.6585	0.0271	(-0.7125, -0.6045)	-24.27	0.000
DA - DTG	-0.1137	0.0271	(-0.1677, -0.0597)	-4.19	0.000
DA (OS) - DTG	-0.6428	0.0271	(-0.6968, -0.5888)	-23.69	0.000
GB - DTG	0.0208	0.0271	(-0.0332, 0.0748)	0.77	0.446
GB (OS) - DTG	-0.0505	0.0271	(-0.1045, 0.0035)	-1.86	0.066
DTE - DTG (OS)	0.1966	0.0271	(0.1426, 0.2505)	7.24	0.000
DTE (OS) - DTG (OS)	0.0183	0.0271	(-0.0357, 0.0723)	0.67	0.502
RFG - DTG (OS)	0.1947	0.0271	(0.1407, 0.2487)	7.18	0.000
RFG (OS) - DTG (OS)	-0.0253	0.0271	(-0.0792, 0.0287)	-0.93	0.355
RFE - DTG (OS)	0.1818	0.0271	(0.1278, 0.2357)	6.70	0.000
RFE (OS) - DTG (OS)	-0.0070	0.0271	(-0.0609, 0.0470)	-0.26	0.798
NB - DTG (OS)	-0.1480	0.0271	(-0.2020, -0.0940)	-5.46	0.000
NB (OS) - DTG (OS)	-0.3290	0.0271	(-0.3830, -0.2750)	-12.13	0.000
kNN - DTG (OS)	-0.5092	0.0271	(-0.5632, -0.4552)	-18.77	0.000
kNN (OS) - DTG (OS)	-0.4934	0.0271	(-0.5474, -0.4394)	-18.18	0.000
LR - DTG (OS)	-0.5109	0.0271	(-0.5649, -0.4569)	-18.83	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
LR (OS) - DTG (OS)	-0.4835	0.0271	(-0.5375, -0.4295)	-17.82	0.000
NN - DTG (OS)	-0.5228	0.0271	(-0.5768, -0.4688)	-19.27	0.000
NN (OS) - DTG (OS)	-0.4752	0.0271	(-0.5292, -0.4212)	-17.51	0.000
DA - DTG (OS)	0.0697	0.0271	(0.0157, 0.1237)	2.57	0.012
DA (OS) - DTG (OS)	-0.4595	0.0271	(-0.5135, -0.4055)	-16.94	0.000
GB - DTG (OS)	0.2041	0.0271	(0.1501, 0.2581)	7.52	0.000
GB (OS) - DTG (OS)	0.1329	0.0271	(0.0789, 0.1869)	4.90	0.000
DTE (OS) - DTE	-0.1783	0.0271	(-0.2322, -0.1243)	-6.57	0.000
RFG - DTE	-0.0019	0.0271	(-0.0559, 0.0521)	-0.07	0.945
RFG (OS) - DTE	-0.2218	0.0271	(-0.2758, -0.1678)	-8.18	0.000
RFE - DTE	-0.0148	0.0271	(-0.0688, 0.0392)	-0.55	0.587
RFE (OS) - DTE	-0.2035	0.0271	(-0.2575, -0.1495)	-7.50	0.000
NB - DTE	-0.3446	0.0271	(-0.3986, -0.2906)	-12.70	0.000
NB (OS) - DTE	-0.5256	0.0271	(-0.5796, -0.4716)	-19.37	0.000
kNN - DTE	-0.7058	0.0271	(-0.7598, -0.6518)	-26.01	0.000
kNN (OS) - DTE	-0.6899	0.0271	(-0.7439, -0.6359)	-25.43	0.000
LR - DTE	-0.7075	0.0271	(-0.7615, -0.6535)	-26.08	0.000
LR (OS) - DTE	-0.6800	0.0271	(-0.7340, -0.6260)	-25.06	0.000
NN - DTE	-0.7194	0.0271	(-0.7734, -0.6654)	-26.52	0.000
NN (OS) - DTE	-0.6717	0.0271	(-0.7257, -0.6177)	-24.76	0.000
DA - DTE	-0.1269	0.0271	(-0.1809, -0.0729)	-4.68	0.000
DA (OS) - DTE	-0.6560	0.0271	(-0.7100, -0.6020)	-24.18	0.000
GB - DTE	0.0076	0.0271	(-0.0464, 0.0616)	0.28	0.781
GB (OS) - DTE	-0.0637	0.0271	(-0.1177, -0.0097)	-2.35	0.021
RFG - DTE (OS)	0.1764	0.0271	(0.1224, 0.2304)	6.50	0.000
RFG (OS) - DTE (OS)	-0.0436	0.0271	(-0.0976, 0.0104)	-1.61	0.112
RFE - DTE (OS)	0.1634	0.0271	(0.1095, 0.2174)	6.02	0.000
RFE (OS) - DTE (OS)	-0.0253	0.0271	(-0.0793, 0.0287)	-0.93	0.355
NB - DTE (OS)	-0.1663	0.0271	(-0.2203, -0.1124)	-6.13	0.000
NB (OS) - DTE (OS)	-0.3473	0.0271	(-0.4013, -0.2933)	-12.80	0.000
kNN - DTE (OS)	-0.5275	0.0271	(-0.5815, -0.4736)	-19.44	0.000
kNN (OS) - DTE (OS)	-0.5117	0.0271	(-0.5657, -0.4577)	-18.86	0.000
LR - DTE (OS)	-0.5292	0.0271	(-0.5832, -0.4752)	-19.51	0.000
LR (OS) - DTE (OS)	-0.5018	0.0271	(-0.5558, -0.4478)	-18.49	0.000
NN - DTE (OS)	-0.5411	0.0271	(-0.5951, -0.4871)	-19.95	0.000
NN (OS) - DTE (OS)	-0.4935	0.0271	(-0.5475, -0.4395)	-18.19	0.000
DA - DTE (OS)	0.0514	0.0271	(-0.0026, 0.1054)	1.89	0.062
DA (OS) - DTE (OS)	-0.4778	0.0271	(-0.5318, -0.4238)	-17.61	0.000
GB - DTE (OS)	0.1858	0.0271	(0.1318, 0.2398)	6.85	0.000
GB (OS) - DTE (OS)	0.1146	0.0271	(0.0606, 0.1685)	4.22	0.000
RFG (OS) - RFG	-0.2199	0.0271	(-0.2739, -0.1659)	-8.11	0.000
RFE - RFG	-0.0129	0.0271	(-0.0669, 0.0411)	-0.48	0.635
RFE (OS) - RFG	-0.2016	0.0271	(-0.2556, -0.1476)	-7.43	0.000
NB - RFG	-0.3427	0.0271	(-0.3967, -0.2887)	-12.63	0.000
NB (OS) - RFG	-0.5237	0.0271	(-0.5777, -0.4697)	-19.30	0.000
kNN - RFG	-0.7039	0.0271	(-0.7579, -0.6499)	-25.95	0.000
kNN (OS) - RFG	-0.6881	0.0271	(-0.7420, -0.6341)	-25.36	0.000
LR - RFG	-0.7056	0.0271	(-0.7596, -0.6516)	-26.01	0.000
LR (OS) - RFG	-0.6782	0.0271	(-0.7322, -0.6242)	-25.00	0.000
NN - RFG	-0.7175	0.0271	(-0.7715, -0.6635)	-26.45	0.000
NN (OS) - RFG	-0.6699	0.0271	(-0.7239, -0.6159)	-24.69	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DA - RFG	-0.1250	0.0271	(-0.1790, -0.0710)	-4.61	0.000
DA (OS) - RFG	-0.6542	0.0271	(-0.7082, -0.6002)	-24.11	0.000
GB - RFG	0.0094	0.0271	(-0.0445, 0.0634)	0.35	0.729
GB (OS) - RFG	-0.0618	0.0271	(-0.1158, -0.0078)	-2.28	0.025
RFE - RFG (OS)	0.2070	0.0271	(0.1530, 0.2610)	7.63	0.000
RFE (OS) - RFG (OS)	0.0183	0.0271	(-0.0357, 0.0723)	0.67	0.502
NB - RFG (OS)	-0.1228	0.0271	(-0.1768, -0.0688)	-4.53	0.000
NB (OS) - RFG (OS)	-0.3038	0.0271	(-0.3578, -0.2498)	-11.20	0.000
kNN - RFG (OS)	-0.4840	0.0271	(-0.5380, -0.4300)	-17.84	0.000
kNN (OS) - RFG (OS)	-0.4681	0.0271	(-0.5221, -0.4141)	-17.25	0.000
LR - RFG (OS)	-0.4856	0.0271	(-0.5396, -0.4317)	-17.90	0.000
LR (OS) - RFG (OS)	-0.4582	0.0271	(-0.5122, -0.4042)	-16.89	0.000
NN - RFG (OS)	-0.4976	0.0271	(-0.5516, -0.4436)	-18.34	0.000
NN (OS) - RFG (OS)	-0.4499	0.0271	(-0.5039, -0.3959)	-16.58	0.000
DA - RFG (OS)	0.0949	0.0271	(0.0409, 0.1489)	3.50	0.001
DA (OS) - RFG (OS)	-0.4342	0.0271	(-0.4882, -0.3802)	-16.00	0.000
GB - RFG (OS)	0.2294	0.0271	(0.1754, 0.2834)	8.45	0.000
GB (OS) - RFG (OS)	0.1581	0.0271	(0.1041, 0.2121)	5.83	0.000
RFE (OS) - RFE	-0.1887	0.0271	(-0.2427, -0.1347)	-6.96	0.000
NB - RFE	-0.3298	0.0271	(-0.3838, -0.2758)	-12.16	0.000
NB (OS) - RFE	-0.5108	0.0271	(-0.5648, -0.4568)	-18.83	0.000
kNN - RFE	-0.6910	0.0271	(-0.7450, -0.6370)	-25.47	0.000
kNN (OS) - RFE	-0.6751	0.0271	(-0.7291, -0.6211)	-24.88	0.000
LR - RFE	-0.6927	0.0271	(-0.7466, -0.6387)	-25.53	0.000
LR (OS) - RFE	-0.6652	0.0271	(-0.7192, -0.6112)	-24.52	0.000
NN - RFE	-0.7046	0.0271	(-0.7586, -0.6506)	-25.97	0.000
NN (OS) - RFE	-0.6569	0.0271	(-0.7109, -0.6029)	-24.21	0.000
DA - RFE	-0.1121	0.0271	(-0.1661, -0.0581)	-4.13	0.000
DA (OS) - RFE	-0.6412	0.0271	(-0.6952, -0.5872)	-23.63	0.000
GB - RFE	0.0224	0.0271	(-0.0316, 0.0764)	0.82	0.412
GB (OS) - RFE	-0.0489	0.0271	(-0.1029, 0.0051)	-1.80	0.075
NB - RFE (OS)	-0.1411	0.0271	(-0.1951, -0.0871)	-5.20	0.000
NB (OS) - RFE (OS)	-0.3221	0.0271	(-0.3761, -0.2681)	-11.87	0.000
kNN - RFE (OS)	-0.5023	0.0271	(-0.5563, -0.4483)	-18.51	0.000
kNN (OS) - RFE (OS)	-0.4864	0.0271	(-0.5404, -0.4324)	-17.93	0.000
LR - RFE (OS)	-0.5039	0.0271	(-0.5579, -0.4500)	-18.57	0.000
LR (OS) - RFE (OS)	-0.4765	0.0271	(-0.5305, -0.4225)	-17.56	0.000
NN - RFE (OS)	-0.5159	0.0271	(-0.5699, -0.4619)	-19.01	0.000
NN (OS) - RFE (OS)	-0.4682	0.0271	(-0.5222, -0.4142)	-17.26	0.000
DA - RFE (OS)	0.0766	0.0271	(0.0226, 0.1306)	2.82	0.006
DA (OS) - RFE (OS)	-0.4525	0.0271	(-0.5065, -0.3985)	-16.68	0.000
GB - RFE (OS)	0.2111	0.0271	(0.1571, 0.2651)	7.78	0.000
GB (OS) - RFE (OS)	0.1398	0.0271	(0.0858, 0.1938)	5.15	0.000
NB (OS) - NB	-0.1810	0.0271	(-0.2350, -0.1270)	-6.67	0.000
kNN - NB	-0.3612	0.0271	(-0.4152, -0.3072)	-13.31	0.000
kNN (OS) - NB	-0.3453	0.0271	(-0.3993, -0.2913)	-12.73	0.000
LR - NB	-0.3629	0.0271	(-0.4169, -0.3089)	-13.37	0.000
LR (OS) - NB	-0.3354	0.0271	(-0.3894, -0.2814)	-12.36	0.000
NN - NB	-0.3748	0.0271	(-0.4288, -0.3208)	-13.81	0.000
NN (OS) - NB	-0.3271	0.0271	(-0.3811, -0.2731)	-12.06	0.000
DA - NB	0.2177	0.0271	(0.1637, 0.2717)	8.02	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DA (OS) - NB	-0.3114	0.0271	(-0.3654, -0.2574)	-11.48	0.000
GB - NB	0.3522	0.0271	(0.2982, 0.4062)	12.98	0.000
GB (OS) - NB	0.2809	0.0271	(0.2269, 0.3349)	10.35	0.000
kNN - NB (OS)	-0.1802	0.0271	(-0.2342, -0.1262)	-6.64	0.000
kNN (OS) - NB (OS)	-0.1644	0.0271	(-0.2184, -0.1104)	-6.06	0.000
LR - NB (OS)	-0.1819	0.0271	(-0.2359, -0.1279)	-6.70	0.000
LR (OS) - NB (OS)	-0.1545	0.0271	(-0.2085, -0.1005)	-5.69	0.000
NN - NB (OS)	-0.1938	0.0271	(-0.2478, -0.1398)	-7.14	0.000
NN (OS) - NB (OS)	-0.1462	0.0271	(-0.2002, -0.0922)	-5.39	0.000
DA - NB (OS)	0.3987	0.0271	(0.3447, 0.4527)	14.69	0.000
DA (OS) - NB (OS)	-0.1305	0.0271	(-0.1845, -0.0765)	-4.81	0.000
GB - NB (OS)	0.5331	0.0271	(0.4792, 0.5871)	19.65	0.000
GB (OS) - NB (OS)	0.4619	0.0271	(0.4079, 0.5159)	17.02	0.000
kNN (OS) - kNN	0.0159	0.0271	(-0.0381, 0.0699)	0.58	0.560
LR - kNN	-0.0017	0.0271	(-0.0557, 0.0523)	-0.06	0.951
LR (OS) - kNN	0.0258	0.0271	(-0.0282, 0.0798)	0.95	0.345
NN - kNN	-0.0136	0.0271	(-0.0676, 0.0404)	-0.50	0.618
NN (OS) - kNN	0.0341	0.0271	(-0.0199, 0.0881)	1.26	0.213
DA - kNN	0.5789	0.0271	(0.5249, 0.6329)	21.34	0.000
DA (OS) - kNN	0.0498	0.0271	(-0.0042, 0.1038)	1.83	0.070
GB - kNN	0.7134	0.0271	(0.6594, 0.7674)	26.29	0.000
GB (OS) - kNN	0.6421	0.0271	(0.5881, 0.6961)	23.67	0.000
LR - kNN (OS)	-0.0175	0.0271	(-0.0715, 0.0365)	-0.65	0.520
LR (OS) - kNN (OS)	0.0099	0.0271	(-0.0441, 0.0639)	0.36	0.716
NN - kNN (OS)	-0.0295	0.0271	(-0.0835, 0.0245)	-1.09	0.281
NN (OS) - kNN (OS)	0.0182	0.0271	(-0.0358, 0.0722)	0.67	0.504
DA - kNN (OS)	0.5630	0.0271	(0.5091, 0.6170)	20.75	0.000
DA (OS) - kNN (OS)	0.0339	0.0271	(-0.0201, 0.0879)	1.25	0.215
GB - kNN (OS)	0.6975	0.0271	(0.6435, 0.7515)	25.71	0.000
GB (OS) - kNN (OS)	0.6262	0.0271	(0.5722, 0.6802)	23.08	0.000
LR (OS) - LR	0.0274	0.0271	(-0.0266, 0.0814)	1.01	0.315
NN - LR	-0.0119	0.0271	(-0.0659, 0.0421)	-0.44	0.661
NN (OS) - LR	0.0357	0.0271	(-0.0183, 0.0897)	1.32	0.192
DA - LR	0.5806	0.0271	(0.5266, 0.6346)	21.40	0.000
DA (OS) - LR	0.0514	0.0271	(-0.0026, 0.1054)	1.90	0.062
GB - LR	0.7150	0.0271	(0.6610, 0.7690)	26.35	0.000
GB (OS) - LR	0.6438	0.0271	(0.5898, 0.6978)	23.73	0.000
NN - LR (OS)	-0.0394	0.0271	(-0.0933, 0.0146)	-1.45	0.151
NN (OS) - LR (OS)	0.0083	0.0271	(-0.0457, 0.0623)	0.31	0.760
DA - LR (OS)	0.5532	0.0271	(0.4992, 0.6071)	20.39	0.000
DA (OS) - LR (OS)	0.0240	0.0271	(-0.0300, 0.0780)	0.88	0.379
GB - LR (OS)	0.6876	0.0271	(0.6336, 0.7416)	25.34	0.000
GB (OS) - LR (OS)	0.6163	0.0271	(0.5623, 0.6703)	22.72	0.000
NN (OS) - NN	0.0477	0.0271	(-0.0063, 0.1017)	1.76	0.083
DA - NN	0.5925	0.0271	(0.5385, 0.6465)	21.84	0.000
DA (OS) - NN	0.0634	0.0271	(0.0094, 0.1174)	2.34	0.022
GB - NN	0.7270	0.0271	(0.6730, 0.7810)	26.79	0.000
GB (OS) - NN	0.6557	0.0271	(0.6017, 0.7097)	24.17	0.000
DA - NN (OS)	0.5448	0.0271	(0.4909, 0.5988)	20.08	0.000
DA (OS) - NN (OS)	0.0157	0.0271	(-0.0383, 0.0697)	0.58	0.564
GB - NN (OS)	0.6793	0.0271	(0.6253, 0.7333)	25.04	0.000

Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
GB (OS) - NN (OS)	0.6080	0.0271	(0.5540, 0.6620)	22.41	0.000
DA (OS) - DA	-0.5292	0.0271	(-0.5831, -0.4752)	-19.50	0.000
GB - DA	0.1345	0.0271	(0.0805, 0.1884)	4.96	0.000
GB (OS) - DA	0.0632	0.0271	(0.0092, 0.1172)	2.33	0.022
GB - DA (OS)	0.6636	0.0271	(0.6096, 0.7176)	24.46	0.000
GB (OS) - DA (OS)	0.5923	0.0271	(0.5383, 0.6463)	21.83	0.000
GB (OS) - GB	-0.0713	0.0271	(-0.1253, -0.0173)	-2.63	0.010

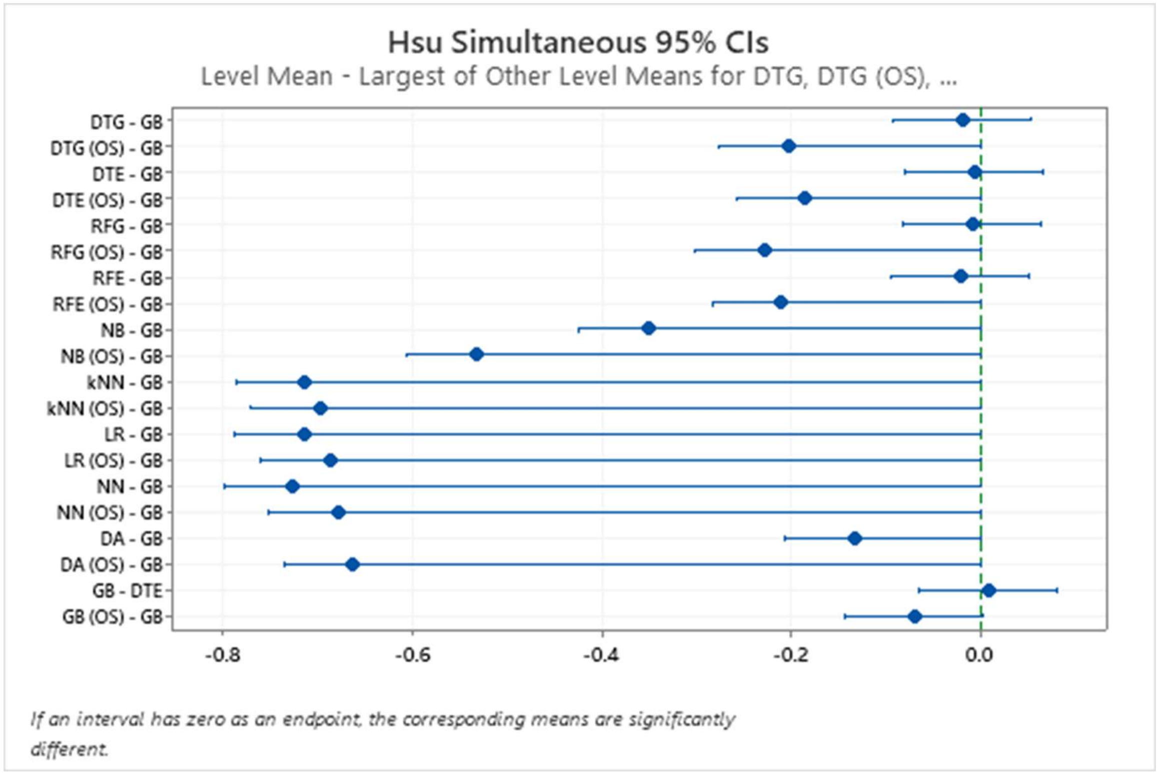
Simultaneous confidence level = 10.32%

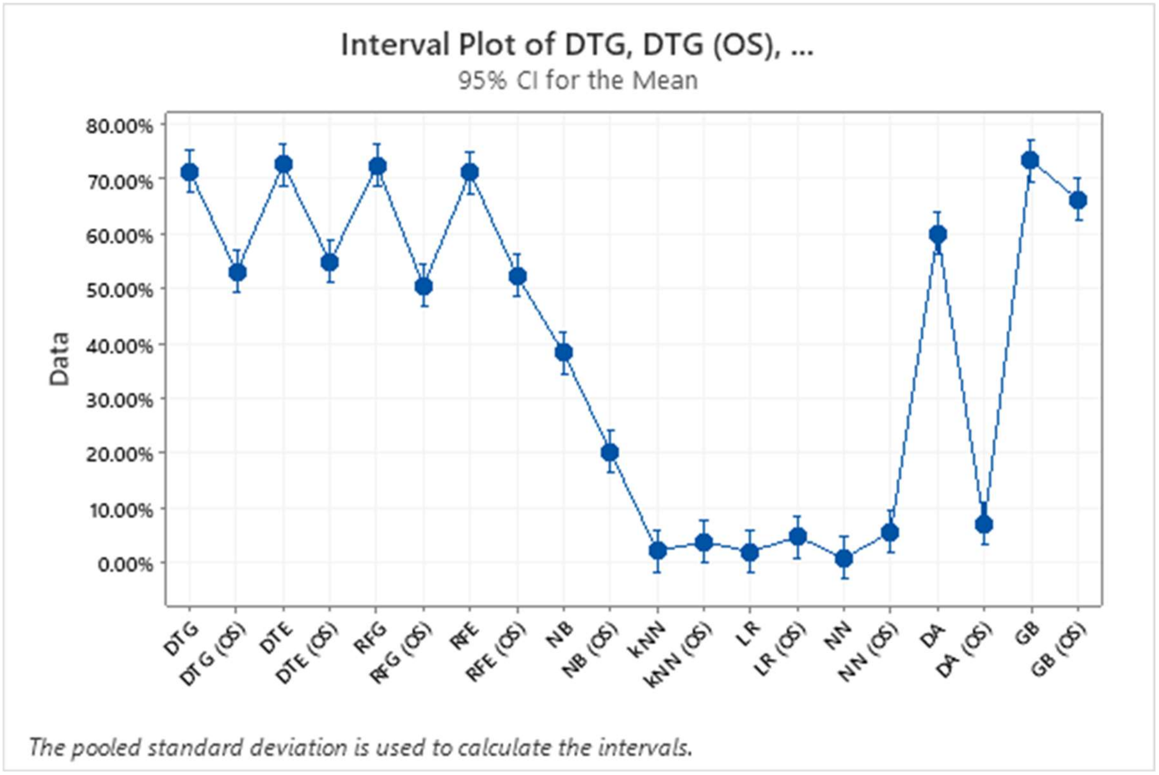
Hsu Multiple Comparisons with the Best (MCB)

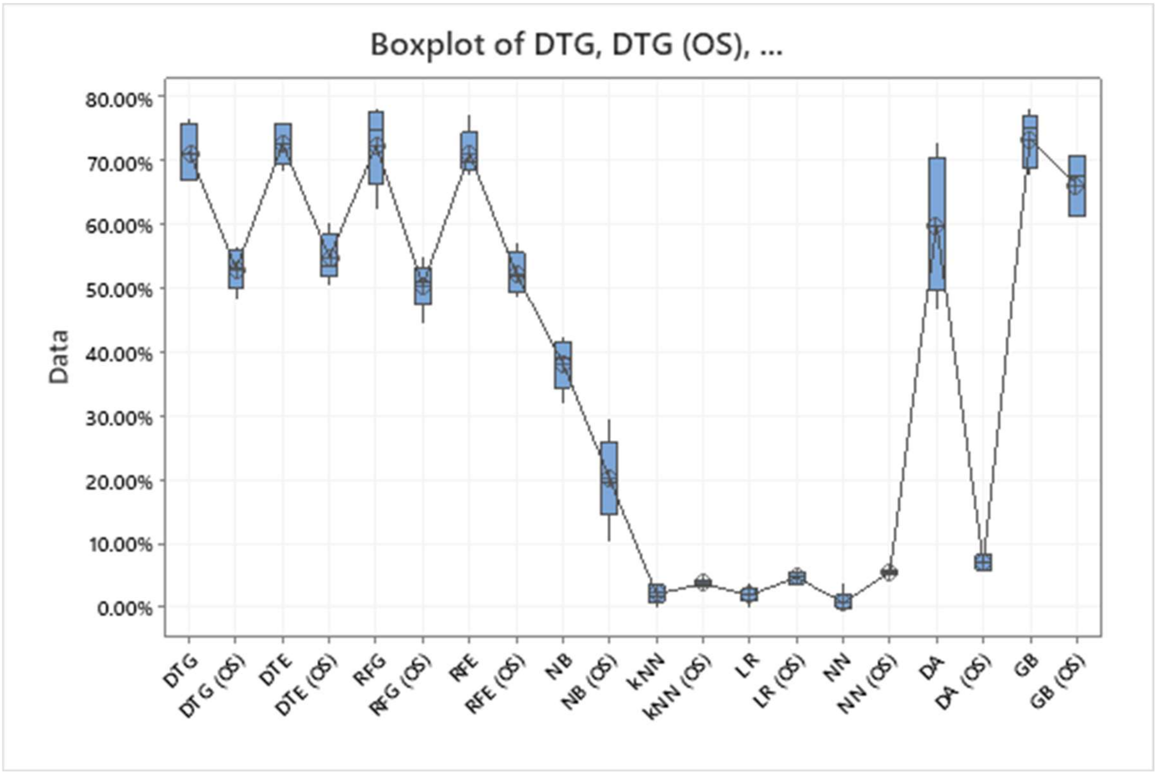
Hsu Simultaneous Tests for Level Mean - Largest of Other Level Means

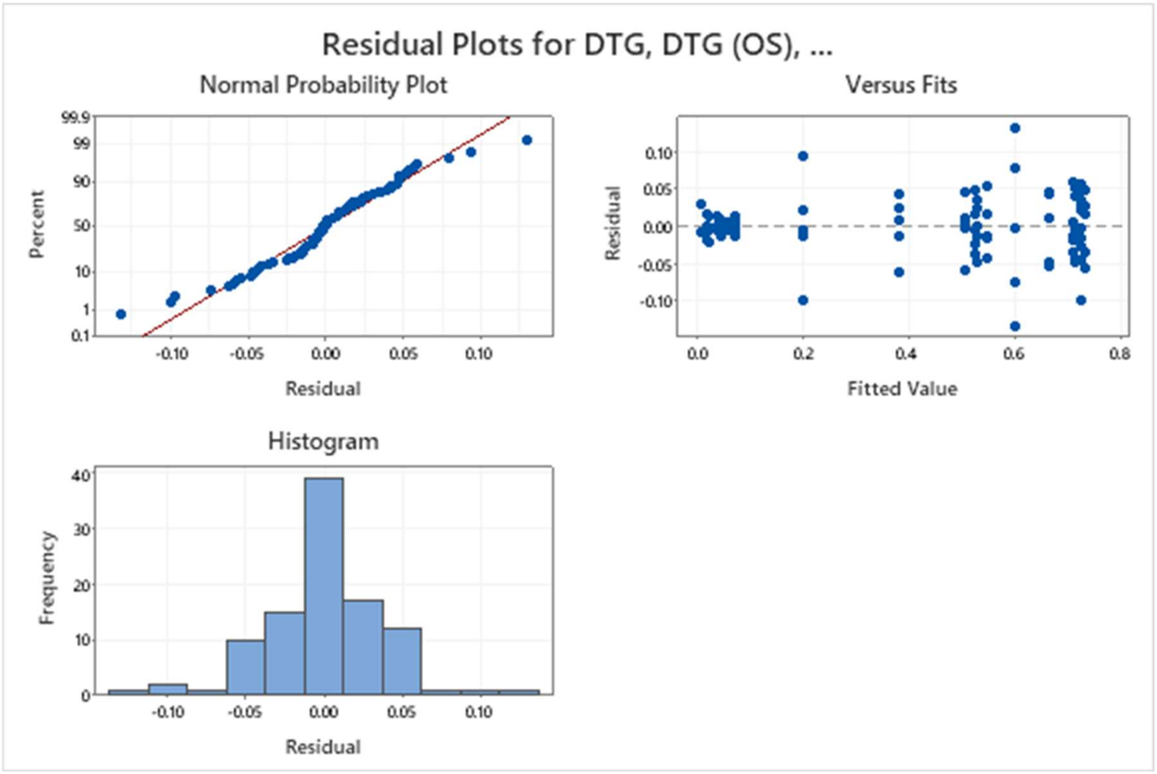
Difference of Levels	Difference of Means	SE of Difference	95% CI	T-Value	Adjusted P-Value
DTG - GB	-0.0208	0.0271	(-0.0936, 0.0520)	-0.77	0.749
DTG (OS) - GB	-0.2041	0.0271	(-0.2769, 0.0000)	-7.52	0.000
DTE - GB	-0.0076	0.0271	(-0.0804, 0.0652)	-0.28	0.901
DTE (OS) - GB	-0.1858	0.0271	(-0.2586, 0.0000)	-6.85	0.000
RFG - GB	-0.0094	0.0271	(-0.0823, 0.0634)	-0.35	0.885
RFG (OS) - GB	-0.2294	0.0271	(-0.3022, 0.0000)	-8.45	0.000
RFE - GB	-0.0224	0.0271	(-0.0952, 0.0504)	-0.82	0.724
RFE (OS) - GB	-0.2111	0.0271	(-0.2839, 0.0000)	-7.78	0.000
NB - GB	-0.3522	0.0271	(-0.4250, 0.0000)	-12.98	0.000
NB (OS) - GB	-0.5331	0.0271	(-0.6060, 0.0000)	-19.65	0.000
kNN - GB	-0.7134	0.0271	(-0.7862, 0.0000)	-26.29	0.000
kNN (OS) - GB	-0.6975	0.0271	(-0.7703, 0.0000)	-25.71	0.000
LR - GB	-0.7150	0.0271	(-0.7878, 0.0000)	-26.35	0.000
LR (OS) - GB	-0.6876	0.0271	(-0.7604, 0.0000)	-25.34	0.000
NN - GB	-0.7270	0.0271	(-0.7998, 0.0000)	-26.79	0.000
NN (OS) - GB	-0.6793	0.0271	(-0.7521, 0.0000)	-25.04	0.000
DA - GB	-0.1345	0.0271	(-0.2073, 0.0000)	-4.96	0.000
DA (OS) - GB	-0.6636	0.0271	(-0.7364, 0.0000)	-24.46	0.000
GB - DTE	0.0076	0.0271	(-0.0652, 0.0804)	0.28	0.901
GB (OS) - GB	-0.0713	0.0271	(-0.1441, 0.0015)	-2.63	0.057

Individual confidence level = 99.12%


Hsu Simultaneous 95% CIs
Level Mean - Largest of Other Level Means for DTG, DTG (OS), ...
DTG - GB
DTG (OS) - GB
DTE - GB
DTE (OS) - GB
RFG - GB
RFG (OS) - GB
RFE - GB
RFE (OS) - GB
NB - GB
NB (OS) - GB
kNN - GB
kNN (OS) - GB
LR - GB
LR (OS) - GB
NN - GB
NN (OS) - GB
DA - GB
DA (OS) - GB
GB - DTE
GB (OS) - GB
-0.8
-0.6
-0.4
-0.2
0.0
If an interval has zero as an endpoint, the corresponding means are significantly different.


Interval Plot of DTG, DTG (OS), ...
95% CI for the Mean
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
Data
DTG
DTG (OS)
DTE
DTE (OS)
RFG
RFG (OS)
RFE
RFE (OS)
NB
NB (OS)
kNN
kNN (OS)
LR
LR (OS)
NN
NN (OS)
DA
DA (OS)
GB
GB (OS)
The pooled standard deviation is used to calculate the intervals.


Boxplot of DTG, DTG (OS), ...
Data
80.00%
70.00%
60.00%
50.00%
40.00%
30.00%
20.00%
10.00%
0.00%
DTG
DTG (OS)
DTE
DTE (OS)
RFG
RFG (OS)
RFE
RFE (OS)
NB
NB (OS)
kNN
kNN (OS)
LR
LR (OS)
NN
NN (OS)
DA
DA (OS)
GB
GB (OS)


Residual Plots for DTG, DTG (OS), ...
Normal Probability Plot
Percent
Residual
Versus Fits
Residual
Fitted Value
Histogram
Frequency
Residual