2021

A Methodology for Detecting Credit Card Fraud

Kayode Ayorinde
Minnesota State University, Mankato

A METHODOLOGY FOR DETECTING CREDIT CARD FRAUD

Kayode Ayorinde

Thesis

Master's in Data Science

Minnesota State University

Mankato, MN

August 2021

ENDORSEMENT/SIGNATURE PAGE

THESIS TITLE:	A METHODOLOGY FOR DETECTING CREDIT CARD FRAUD
AUTHOR:	Kayode Olaleye Ayorinde
DATE SUBMITTED:	Summer 2021
	Department of Computer Information Science

Dr. Naseef Mansoor
Advisor and Committee Chair ----------

Dr. Mezbahur Rahman
Committee Member ----------

Dr. Suboh Alkhushayni
Committee Member ----------

ACKNOWLEDGMENT

Firstly, I want to give God almighty all the praise for his mercy and guidance over me and my academic achievement.

The completion of this research work could not have been possible without the expertise of Dr. Mansoor Naseef (thesis Advisor) and Dr. Bukralia Rajeev (thesis proposal Advisor). I would also like to thank Dr. Mezbahur Rahman (thesis Committee), and Dr. Suboh Alkhushayni (thesis committee) for sitting on my panel and taking their time to read my thesis.

A huge debt of appreciation is also owed to my friends Ayodele Olufemi, Emmanuel Larayetan, and Seun Adejugbe for their advice and contributions towards helping to proofread this study.

Lastly, I would like to thank Mr. Gbenga Abioye (my uncle), my siblings, and my parents - Mr. and Mrs. Ayorinde for their moral and financial supports towards achieving my career; without you, none of this would indeed have been possible.

TABLE OF CONTENTS

ENDORSEMENT/SIGNATURE PAGE i
ACKNOWLEDGMENT ii
TABLE OF CONTENTS iii
ABSTRACT v
CHAPTER ONE 1
1.1 INTRODUCTION 1
1.2 CREDIT CARD FRAUD 1
1.3 COMMON TRENDS IN CREDIT CARD FRAUD 2
1.4 SIGNIFICANT OF THE STUDY 5
CHAPTER TWO 6
2.1 REVIEW OF RELEVANT LITERATURE 6
2.2 SOME RELATED WORKS 6
CHAPTER THREE 13
3.1 METHODOLOGY 13
3.2 DATASET 13
3.3 DATA PREPROCESSING 15
3.3.1 DATA CLEANING 16
3.3.2 ENCODING CATEGORICAL VARIABLES 17
3.3.3 FEATURE SCALING 18
3.3.4 DATASET RE-SAMPLING 19
3.3.5 FEATURE CORRELATION AND SELECTION 21
3.4 MACHINE LEARNING MODELS 23
3.4.1 DECISION TREE 23
3.4.2 LOGISTIC CLASSIFICATION 25
3.4.3 RANDOM FOREST 26
3.4.4 XGBOOST CLASSIFIER 27
3.4.5 K-MEANS CLUSTERING 29

3.4.6 AUTOENCODERS NEURAL NETWORKS (ANN) 30
3.5 MODEL CREATION 32
3.6 SPLITTING OF DATA INTO TRAINING AND TEST 33
3.7 HYPERPARAMETER TUNING 33
CHAPTER FOUR 35
4.1 FINDINGS AND RESULTS 35
4.2 METRICS 35
4.2.1 ACCURACY 36
4.2.2 RECALL 36
4.2.3 PRECISION 36
4.2.4 F1-SCORE 37
4.2.5 CONFUSION MATRIX 37
4.2.6 ROC AUC SCORE 38
4.3 MODELLING ORIGINAL DATASET 40
4.4 MODEL RESULT FOR UNDERSAMPLING DATASET 41
4.5 MODEL RESULT FOR OVERSAMPLING DATASET 43
4.6 HYPERPARAMETER TUNNING WITH THE BEST MODEL 44
4.7 COMPARATIVE ANALYSIS 45
CHAPTER FIVE 48
5.1 CONCLUSION 48
5.2 LIMITATION OF THE STUDY 48
5.3 FUTURE RESEARCH 49
REFERENCES 50

ABSTRACT

Fraud detection has appertained to many industries such as banking, retails, financial services, healthcare, etc. As we know, fraud detection is a set of campaigns undertaken to avert the acquisition of illegal means to obtain money or property under false pretense. With an unlimited and growing number of ways fraudsters commit fraud crimes, detecting online fraud was so tricky to achieve. This research work aims to examine feasible ways to identify credit card fraudulent activities that negatively impact financial institutes. In the United States, an average of U.S consumers lost a median of $429 from credit card fraud in 2017, according to "CPO magazine. Almost 79% of consumers who experienced credit card fraud did not suffer any financial impact whatsoever" [35]. One of the questions is, who is paying for these losses if not the consumers? The answer to this question is the financial institutions. According to the Federal Trade Commission report, credit card theft has increased by 44.6% from 2019 to 2020, and the amount of money lost to credit card fraud in the year 2020 is about 149 million in total loss. Without any delay, financial institutes should implement technology safeguards and cybersecurity to decrease the impact of credit card fraud activities. To compare our proposed machine learning algorithms with machine learning techniques that already exist, we carried out a comparative analysis and we were able to determine which algorithm can best predict fraudulent transactions by recognizing a pattern that is different from other patterns. We trained our algorithms over two re-sampling methods (undersampling and oversampling) of the credit card fraud dataset and, the best algorithm is drawn to predict frauds. AUC score and other metrics was used to compare and contrast the results of our algorithms. The following results are concluded based on our study:

1. Our study proposed algorithms such as Random Forest, Decision Trees and Xgboost, K-Means, Logistic Regression and Neural Network have performed better than other machine learning algorithms researchers have used in previous studies to predict credit card frauds.

2. Our ensemble tree algorithms such as Random Forest, Decision Trees and Xgboost came out to be the best model that can predict credit card fraud with AUC score of 1.00%, 0.99% and 0.99% respectively.

3. The best algorithm for this study shows a lot of improvements with the oversampling dataset with overall performance of 1.00% AUC score.

Keywords: Credit Card Fraud**,** Fraud Detection, Machine Learning Algorithms, Banking and Financial Sector, Machine Learning Classifiers, Re-sampling Methods

CHAPTER ONE

1.1 INTRODUCTION

As the events in the world become more digitalized, cybercrimes like credit card or debit card frauds are on the increase. “According to the 2019 report of the Bureau of Consumer Financial Protection on the Consumer Credit Card Market, “fraud remains a constant and costly reality of the credit card market.” This unfortunate situation has adversely affected individuals, public and private organizations globally [4]. The problem is somehow challenging to manage. International transactions on credit cards or running above specific limits have been used to flag some transactions as fraudulent. Still, it has also been discovered that 70 % of such flagged transactions were a false alarm, resulting in a drop in sales for merchants and loss of credibility. This research investigates methods to be adopted in identifying credit card frauds and how our proposed solutions can help solve this fraud.

1.2 CREDIT CARD FRAUD

Credit card fraud happens one of the biggest threats to financial institutions and businesses today. Credit card fraud can be defined as “when an unauthorized person uses a credit card for personal use without the approval or knowledge of the card owner and the card issuer doesn’t have a clue of what the card is being used for.” Different types of systems/models, processes, and preventive measures will help end credit card fraud and help reduce financial risks. Large amounts of credit card account transactions are convened together by financial institutes and companies. A plastic card called a credit card is issued to various

users as one of the methods of carrying out transactions [3]. It allows the card authorized users to purchase goods and services based on the promise made by the holder to pay for them at a later date. Credit cards have become commonplace for individual finance over the past few years; admiration and approval rates are considered clearly in the number of credit cardholders. According to "United States credit card statistics published on Statista website, it is recorded that about 1.1 trillion of credit cards have been issued between 2012 and 2018, this number of credit card issued have surpassed the number of debit cards issued three times. As of 2019, Visa was the largest credit card issuer with more than 300 million credit cards been issued to customers [31]." Secure credit services of financial institutions and development of E-business a reliable fraud detection mode is vital to support safe credit card usage, Fraud detection based on analyzing existing purchase data of cardholder is a promising way for reducing the rate of credit card frauds. Fraud detection systems come into a synopsis when the fraudsters beat the fraud prevention rules and start fraudulent transactions.

1.3 COMMON TRENDS IN CREDIT CARD FRAUD

Most card users are fully aware of the imminent danger from fraudsters; this has made the card thieves advance their operation mode to beat the continuously updated security walls. Therefore, this aspect would briefly discuss some prevalent patterns of credit card fraud [18].

(a) Stolen/Misplaced card: This method is the most prevalent. It has to do with stealing someone's credit card and using it as their own. Indeed, getting information from the front and back of the card without taking the card away is the same as stealing the card. Banks usually inform customers to notify them through the emergency lines anytime their card is stolen or misplaced. The thief can use the information to purchase goods online, and the bank might not notify the owner until the end of the month.

(b) Synthetic Fraud: A synthetic fraud is an act whereby a fraudster applies for a credit card on behalf of someone. The fraudster acquires essential information of their victim like Social Security Number (SSN), date of birth, address, etc., and applies for a credit card on behalf of the victim. This method is also known as the "false application method."

(c) Data Breach: Since people carry out some of their transactions through the internet, their data is vulnerable to hackers. The hacker might adopt several ways to get the victim's data. They can even completely take over someone's phone or computer after visiting some websites. One of the recommended ways to remedy this situation is to avoid saving important information on any device, or better still, to frequently clear data before getting into the wrong hands.

(d) Mail Interception: Fraudsters can also intercept mails intended to go to the user's address. Probably after applying for a new card, the fraudster can manipulate things to get the card before it gets to the owner. The money would have been gone before the card eventually gets to the owner.

(e) Skimming: This kind of fraud is usually swept under the rug because it does not involve much money; it can even be pennies. But when this is done to millions of customers, it becomes a significant figure. Fraudsters can obtain card details like the number and activate them so that whenever the card owner performs any transaction through the card, the thief gets the commission for each transaction.

(f) Merchant Collusion: This is a type of fraud that is usually carried out by an organization. A company owner or its employee can use the customers' credit card or give it to a fraudster. Since card information is occasionally saved with some trusted merchants to make purchasing items easy for the customer, company owners or employees can extract some card information and use it to their destructive ends.

(g) Triangulation: This is another form of fraudulent act that fraudsters use to reap peoples hard earn money. Some goods can be published on a website at a meager price to attract customers. The site owner has the sole aim of obtaining customers' card information. In some cases, the fraudster might not have the goods, but they lure their victims to provide information about their credit cards so that they can use them. The only way to avoid this is to verify every site to be genuine and ensure that they read reviews about it.

There are other fraudulent acts related to credit cards, but this discussion is limited to the ones.

1.4 SIGNIFICANT OF THE STUDY

The benefit of this research paper is to help the financial institutions by improving the existing machine learning algorithms that can predict fraudulent acts with very high accuracy, which will ease them in preventing fraudsters from carrying out transactions that were not approved or authorized by the legitimate owner of various accounts. Despite the extensive range of the problem, relatively some of academic exploration has been done on fraud costs, the root causes, how it occurs, why it occurs, and productive ways to recognize, discourage and avert it. The need for anti-fraud expertise is becoming more urgent as the fraudsters are not reported to public authorities. Organizations must incur significant resources as they strive to protect themselves from fraud and reputational consequences. For smaller organizations, the issue is complicated to deal with due to insufficient resources to set up anti-fraud units. Small businesses must turn to private investigation firms if they want to benefit from specialized expertise in dealing with a fraud problem, but the cost can be pretty substantial. One of the current solutions that helps banks and financial institutions move forward is the machine learning approach.

CHAPTER TWO

2.1 REVIEW OF RELEVANT LITERATURE

Many works have been done related to credit card fraud. In this review, we will synthesize some of the articles to identify works that have already been done. This section discussed machine learning using (supervised methods) such as Logistic Regression, Decision Tree, Random Forest, XGBoost, (unsupervised methods) such as K – Means Clustering, and Autoencoder in Keras. Researchers like Awoyemi et al. (2017), Maniraj et al. (2019), Dornadula (2019), Shirgave et al. (2019), Azhan (2020), Joshi et al. (2020), Sadineni et al. (2020), More et al. (2021), Priya & Saradha (2021), Roy et al. (2021) and Mohari et al. (2021), have identified supervised and unsupervised method of machine learning as the most common methods.

2.2 SOME RELATED WORKS

Maniraj et al. (2019) illustrate the modeling of a data set using machine learning with Credit Card Fraud Detection. The authors try to detect transactions that are 100% fraudulent as they minimize the incorrect fraud classification. The focus was on analyzing and preprocessing datasets and deploying multiple anomaly detection algorithms like the Local Factor Isolation Forest algorithm on the PCA transformed Credit Card Transaction Data. The results show that the algorithm reaches over 99.6% accuracy, but its precision is about 28% when using a tenth of the data set. Nevertheless, as the entire dataset is inputted into

the algorithm, the precision increases to 33%. We expect this rise inaccuracy because of the enormous disparity between valid and genuine transactions [24].

Awoye'mi et al. (2017) identify two problems with credit card fraud detection. The first problem is the constantly changing profiles of standard and fraudulent transactions, and credit card fraud datasets are highly skewed [7]. They further investigate data performance using the naïve Bayes, k-nearest Neighbor, and logistic regression on highly skewed credit card fraud data. 284,807 transactions of the European cardholders were sampled in the research. The researchers applied three techniques to the raw and preprocessed data as the work is implemented in Python. The performance of the methods is assessed based on accuracy, sensitivity, specificity, precision, Matthew's correlation coefficient, and flat classification rate. The findings show optimal accuracy for naïve Bayes, k-nearest neighbor, and logistic regression classifiers as they indicate 97.92%, 97.69%, and 54.86%, respectively. After comparing the methods, it was evident that the k-nearest Neighbor is better than naïve Bayes and logistic regression techniques**.**

Mohari et al. (2021) said those fallacious activities conducted through credit cards could be tackled with Data Science, Machine Learning together with Deep Learning techniques. One advantage of this is that it helps banks and other financial institutions detect frauds as early as possible before it causes excellent damages. On the other hand, the hackers need a minute amount of data to carry out their malicious acts; this makes the victims vulnerable to danger. There are different techniques and methods of unsupervised learning [15]. Mohari et al. (2021) identified ten of them and compared them in their research. They compared Logistics Regression, Random Forest, AdaBoost, Artificial Neural Network,

Genetic Algorithm, Hidden Markov Model (HMM), KNN Classifier, Decision tree, Isolation Forest, and Local Outlier Factor. Out of all the ten methods, their results show that Local Outlier Factor fraud accuracy is greater than the rest of the algorithms [15]. Lebichot et al. (2017) is a graph-based, semi-supervised credit card fraud detection scheme. Globally, it has been recorded those billions of US dollars have been lost to fraudulent activities. To stop these despicable acts, automated Fraud Detection Systems (FDS) can first deny a transaction before it is granted [13]. Lebichot et al. (2017) started from a graph based FDS called APATE, which uses a limited set of confirmed fraudulent transactions to spread evil influence through a network. They further re-designed APATE to be a perfect fit for to e-commerce field reality [13]. These improvements significantly impact accomplishment as it multiplies precision at 100 by three, both on fraudulent credit cards and transaction prediction. This new technique was tested in real life for three months on e-commerce credit card transactions set of data obtained from a large credit card issuer. Feedback was also introduced here, but it does not significantly improve as the impact can be increased if more cards are examined.

Many researchers have worked on credit card fraud detection using the XGBoost model. Some recent ones are Meng et al. (2020) and Parmar et al. (2020). According to Meng et al. (2020), XGBoost is an efficient system implementation of Gradient Boosting and GB algorithm based on CART [19]. Meng and his colleague used accurate online transaction data of an Internet financial institution in researching credit card fraud detection operation. They studied the performance learning algorithm on original the original data set and the Undersampling using and SMOTE and XGBoost [14]. The results show that for optimum

output, SMOKE should be used with XGBoost. In similar research, Parmar et al. (2020) consider multiple techniques, including K-Nearest Neighbor, Support Vector Machine (SVM), Decision Trees, Logistic Regression, Random Forest, and XGBoost to detect credit card frauds [19]. They tested 2,84,808 credit card transactions accrued from an EU financial institution dataset. Although the dataset is relatively imbalanced, it has 0.172% of fraud cases from the actual transactions. The methods are implemented using Python, and the presentation of the methods is classed based on the accuracy and F1 rating, and confusion matrix. The findings show that every set of rules can be used for credit card fraud detection alongside excessive precision.

Shirgave et al. (2019) also reviewed credit card fraud detection using machine learning. They examine different fraud detection techniques using machine learning and compare them using instruments like accuracy, precision, and specificity. They also propose an FDS which uses a supervised Random Forest algorithm. With their proposed system, the precision of detecting fraud in credit cards is increased. Furthermore, the proposed method uses the learning to rank approach, rank the alert, and effectively address the problem concept drift in fraud recognition [28].

Priya et al. (2020) Individuals and financial institutions must be aware of the continuous growth of fraudulent activities. Thus, find an efficient fraud detection algorithm to tackle this problem and separate fraudulent transactions from the real ones since the genuine transactions outnumbered the false ones [20]. That is why Warghade et al. (2020) analyze various machine learning techniques by using multiple metrics for judging multiple classifiers. Their research has been able to improve fraud detection rather than

misclassifying a genuine transaction as fraud. In their model, they recommend synthetic techniques like SMOTE for the conventional oversampling method. And to yield a better result, synthetic sampling methods like SMOTE with advanced boosting methods like Local Outlier factor, Isolation Forest, and SVM can be applied. As a result of the parallel processing model, LOF and Isolation Forest is fast and robust to the outlier. Samples of small records were tested, and the results were terrific [32]. Isolation Forest gives an outstanding 99.74% accuracy score, and Support Vector Machine provides a fair percentage of 45.84% accuracy score. LOF gives an excellent 99.66% accuracy score, making the prediction correct, misclassifying the genuine transaction as fraud.

More et al. (2021) used a Random Forest fraud detection algorithm. This model can help solve fraudulent activities in the real world and has continuously increased the accuracy of detecting fraud in credit card transactions [16]. The dataset used in their research contained 100000 transactions made by cardholders, and the results show that 0.262 % of all transactions are fraud. Although the dataset is highly imbalanced, the unbalanced dataset was processed, which shows 80% of the dataset was used for training the model while 20% of the dataset was used for testing. The performance evaluation was carried out for precision, recall (sensitivity), and accuracy. The accuracy level was 0.9793, which shows that the proposed strategy had shown better accuracy for many training data. Also, 20,000 transactions were identified, of which 19,830 belong to class 0, and 170 transactions belong to a class. The research concluded that despite having an imbalanced dataset, the model works well for credit card fraud detection. The study also showed a comparative analysis of three classifiers - Decision Tree, Naive Bayes, and Random Forest; it was evidenced

that the Random Forest technique performed much better than Decision Tree and Naïve Bayes Technique [5].

Sadineni (2020) also worked on related research using machine learning algorithms. The analysis considers various machine learning techniques such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Decision Trees, Logistic Regression and Random Forest to identify frauds carried out by credit cards. The performance analysis of the techniques is done using accuracy, precision, and false alarm rate metrics, just like other researchers. Precisely 150,000 transactions stored in the Kaggle data repository were analyzed [11]. The researcher reported the database to have numerous fields. The dataset, which contained relevant and irrelevant attributes, was analyzed based on the principal component to extract the relevant details like transaction amount, time of the transaction, etc. The results show that Radom Forest achieved an accuracy of 99.21%, Decision Tree was 98.47%, Logistic Regression was 95.55%, Support Vector Machine (SVM) was 95.16%, and Artificial Neural Network (ANN) was 99.92%. This result, unlike other research, showed that ANN is more accurate than other techniques [25].

Rahmawati et al. (2017) was a fraud detection analysis of event logs of a bank's credit business process using the Hidden Markov Model Algorithm. As stated earlier in this paper, many fraudulent acts are carried out every day using different methods [21]. Therefore, Rahmawati et al. (2017) propose a method for detecting fraud on credit applications. The Hidden Markov Models and activity information recorded in the event log can be used to identify fraudulent activities. The automated system calculates the probability and possibility of fraud based on the event log by identifying the symptoms of

fraudulent activities. The analysis was based on 90 cases, and the results show that HMM method can be used to detect fraud as it has an accuracy of 94%. The model was able to report 10 of the 90 cases as fraudulent and 80 as genuine transactions [21].

Rocha & de Sousa Junior (2010) identified bank frauds by using CRISP-DM and Decision Tree techniques. They evaluate some transactions using decision trees and CRISP-DM to help identify and prevent bank fraud. Like many researchers who came after them, they identify decision trees as an essential concept in artificial intelligence. After the information regarding bank transactions, the analysis identified different fraudulent activities from internet bank transactions [22].

Jisha & Vimal (2020) considered a population-based optimized and condensed fuzzy deep belief network to identify credit card fraudulent acts. Instead of using the common theory deployed for an intellectual way of fraudulent transaction detection, the work adopts an approach of intuitionistic fuzzy theory to determine the significant features that influence the detection process efficiently. The deep fuzzy network exceptionally handles the complex form of credit card transactions with its deep-seated knowledge and stacked restricted Boltzmann machine, the pattern of a dataset is analyzed [10].

CHAPTER THREE

3.1 METHODOLOGY

This chapter discusses the methodology adopted in this study to classify the non-fraudulent transactions from the fraudulent transactions. Figure 1 shows the steps used in this work. However, before we discuss the different steps of the methodology used in this work, we first discussed the dataset.

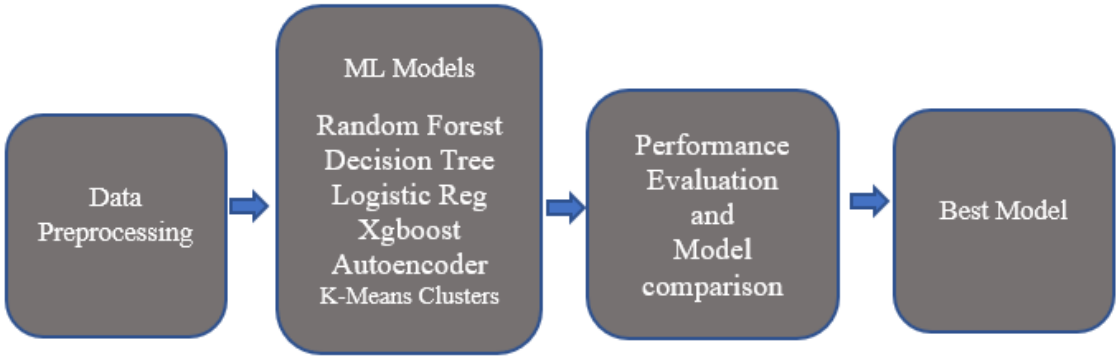


Fig. 1: Classification Methodology

3.2 DATASET

The dataset for this research work is obtained from Kaggle, and it was generated using Sparkov Data Generation, a GitHub tool created by Brandon Harris. The dataset is a simulated credit card transaction containing legitimate and fraudulent transactions. It covers the credit card of 1000 customers doing transactions with a pool of 800 merchants.

The transactions presented by this dataset have 1048575 transactions in total, and the number of fraudulent transactions was recorded to be 6006 out of the total number of transactions. The dataset is highly imbalanced; the positive class (frauds) account for a tiny percentage of about 0.5727 of the complete transactions. The dataset contains 22 features such as" Amount," "Category," "is fraud," and so on, comprising different data types. It also includes both numerical and categorical features. Each transaction recorded per transaction date and time is contained in the feature "trans_date_trans_time" column. The 'Amount' feature column includes the transaction amount carried out, while the last feature in this dataset called "is Fraud" is the response variable that shows whether a transaction is a fraud or not. It takes 1 as a value if it is fraud and 0 if it is not. The dataset is available at

https://www.kaggle.com/kartik2112/fraud-detection

The fig. 2 shows the descriptive statistics between the fraudulent and non-fraudulent transactions for the amount feature. From the output, we can see that the minimum and maximum value of the amount feature for non-fraudulent distribution is 1.00 and 28948.9 respectively while that of fraudulent distribution is 1.18 and 1371.81 respectively. We can also see from the output

	Row Type	Overall Amt Distribution	Non Fraud Amt Distribution	Fraud Amt Distribution
0	count	1048575.000000	1042569.000000	6006.000000
1	mean	70.279095	67.627445	530.573492
2	std	159.951841	153.695606	391.333069
3	min	1.000000	1.000000	1.180000
4	50%	47.450000	47.220000	391.165000
5	95%	196.260000	189.940000	1085.052500
6	99.9%	1496.830880	1502.239520	1289.066100
7	max	28948.900000	28948.900000	1371.810000

Fig. 2: Descriptive statistics of the amount Feature

that the mean of the non-fraudulent distribution for the amount feature which is $67.63 is less than the mean of the fraudulent distribution which is $530.57.

3.3 DATA PREPROCESSING

Preprocessing data is required before implementing a machine learning algorithm, considering various models produce diverse specifications to the predictors, and data training can affect predictive production. Data preprocessing purposes are to clean and prepare the data to a spot that comprises more concise prejudice, checking for missing values, and more variation. Data contains both numerical and categorical, which means encoding the categorical data is necessary before using them for modeling. Outlier detection and removal was performed. We have the independent variables in the same range by performing feature scaling. To reduce feature skewness, a box-cox transformation was carried out. Resampling method such as undersampling and oversampling was performed on the imbalanced original dataset to avoid any form of bias and overfitting in our training model. We have adopted Python data manipulation library pandas and machine learning library sci-kit learn to achieve these preprocessing responsibilities. The steps are shown in fig. 3.

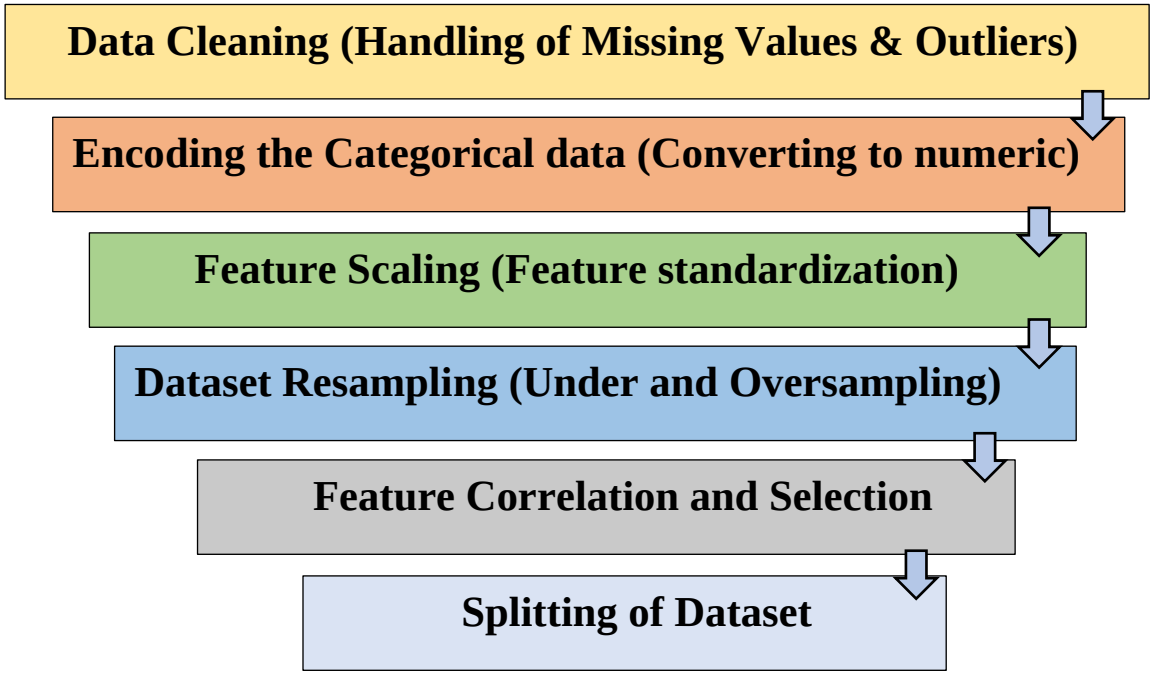


Fig. 3: The Data Preprocessing steps

3.3.1 DATA CLEANING

The credit card dataset was imported using the python import command, and the data cleaning process was done. During data cleaning we perform two tasks; 1. Remove null values and missing values, and 2. Handle outliers.

The dataset contains 1048575 transactions in total. There were no null values in the dataset. Also, our dataset does not have any missing value. Hence, next we look for outliers in the dataset. Outliers are known as the observations that are numerically distant from the rest of the data. The boxplot technique was adopted to detect the presence of outliers in all the independent features. An outlier is a data point located outside the box plot's whiskers. However, for simplicity we only show the box plot for the feature “amount” in fig. 4.

Although the box plots show the presence of outliers in the data, the outliers were removed using the Inter Quantile Range (IQR) technique which is one of the most popular techniques for handling outliers as it is more robust to outliers. In this technique, any value that is outside the Q3 + 1.5 IQR boundary is considered to be an outlier and, any outlier is discarded to make the machine learning models more robust and accurate.

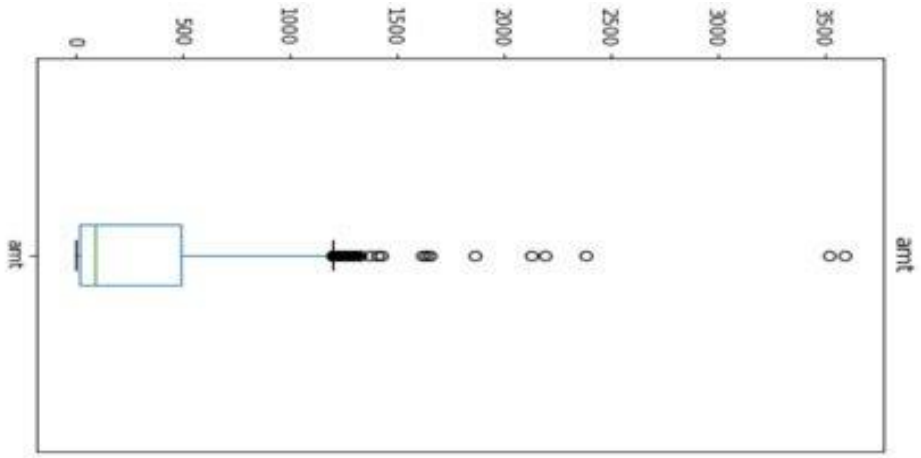


Fig. 4: Boxplot of the amount feature

3.3.2 ENCODING CATEGORICAL VARIABLES

After cleaning the dataset, we convert any categorical features to a numeric value as most machine learning algorithms perform better with numeric inputs. There are few ways to convert categorical values into numeric values with each approach having its own tradeoffs and impact on the feature set. In the study, we have used One-Hot Encoder to convert the categorical variables to numeric values. For a feature with two categories, the categories

are assigned a numeric value of 1 or 0. The fig. 5 shows the results of our categorical variables after conversion.

category_food_dining	category_gas_transport	category_grocery_net	category_grocery_pos	category_health_fitness	category_home	category_kids_pets	categ
0	0	0	0	0	0	0	
0	0	0	1	0	0	0	
0	0	0	0	0	0	0	
0	1	0	0	0	0	0	
0	0	0	0	0	0	0	

Fig. 5: Sample of converted categorical features using One-Hot Encoder

3.3.3 FEATURE SCALING

This is another stage of the data preprocessing method used to normalize the range of independent variables within a dataset. Depending on the adopted scaling technique, it is centered around 0 or in the range of 0 and 1. If input variables have tremendous values applicable to the additional input variables, these large values can overlook or skew some machine learning algorithms. We have performed feature scaling using the Robust Scaler technique, also known as robust standardization. Scaling can be achieved by calculating the median 50th percentile, the 25th, and 75th percentiles. The values of each variable then have their median subtracted and are divided by the interquartile range (IQR), which is the difference between the 75th and 25th percentiles. The fig. 6 below shows our feature scaling process.

```
# Scale "Amount"
from sklearn.preprocessing import StandardScaler, RobustScaler
dataset['scaled_amount'] = RobustScaler().fit_transform(dataset['Amount'].values.reshape(-1,1))
```

Fig. 6: Feature Scaling using RobustScaler()

3.3.4 DATASET RE-SAMPLING

Data resampling is a technique of inexpensively using a data sample to improve the accuracy and measure the unpredictability of a population variable. The nested resampling method has been used to carry out dataset resampling. The dataset used for this study was highly imbalanced; that is why we have carried out resampling methods like Undersampling and Oversampling.

3.3.4.1 UNDERSAMPLING

Since most of the instances in the dataset belong to the majority class, the dataset was under-sampled randomly, by reducing the numbers of instances of the majority class, which means that some essential data instances are not captured for training purposes in the data. The result of our undersampling is shown in fig. 7.

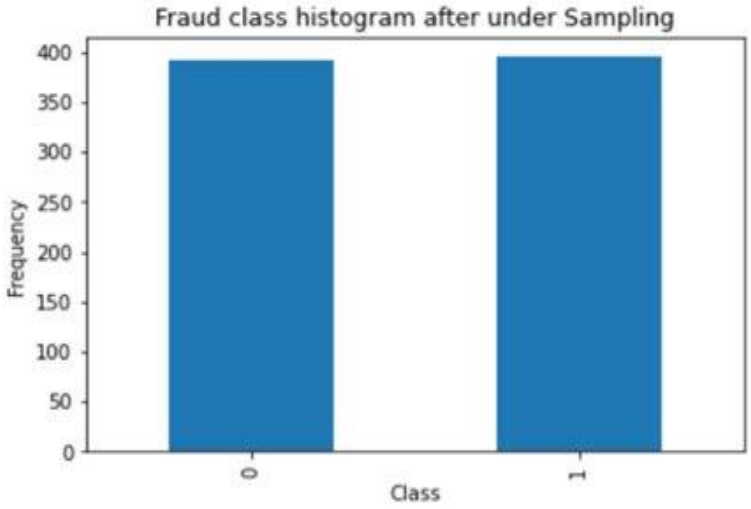


Fig. 7: Distribution of the classes after Undersampling

3.3.4.2 OVERSAMPLING

This method duplicates new or sometimes simulates examples in the minority class. It increases the instances, which makes the training of the model to perform better. The result of our oversampling is shown in fig. 8.

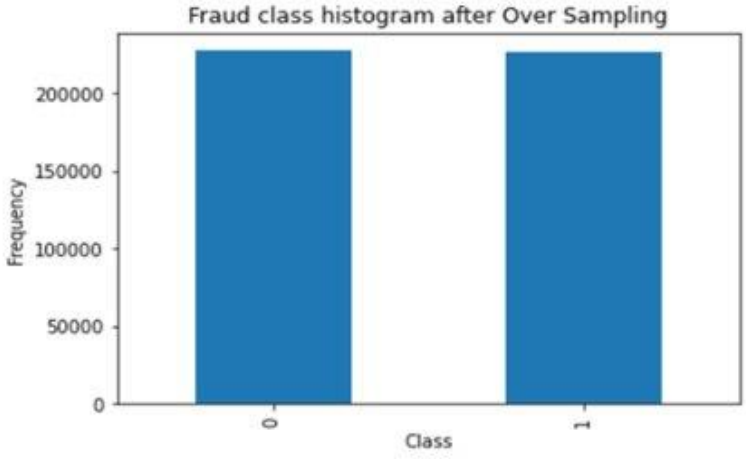


Fig. 8: Distribution of the classes after Oversampling

3.3.5 FEATURE CORRELATION AND SELECTION

Each of the features we obtain in the dataset might not be beneficial in building a machine learning model to execute the necessary prediction. Using some of the features might improve the prediction accuracy. So, feature correlation performs a tremendous purpose in creating a better machine learning model. Features with high correlation are more likely to be linearly dependent and have almost the same impact on the dependent variable. Therefore, when two features produce a high correlation, we can drop one of the two features. The heatmap for the correlation of the original dataset, and resampled dataset (both undersampled, and the oversampled) is shown in Fig. 9, and 10. It can be observed that the heatmap is not revealing too much information because it's a huge dataset, and that is why we performed feature selection to help select the important features. Feature selection is one of the important stages in data preprocessing, and it is known as a path to capture relevant features for use in the implementation of the machine learning model to

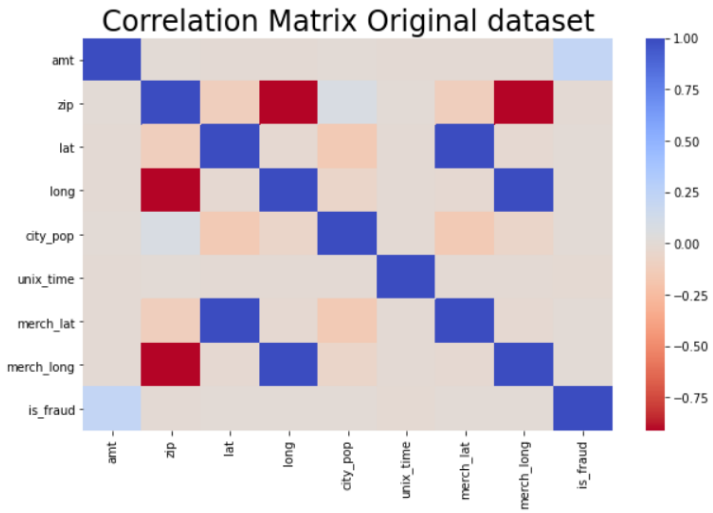


Fig. 9: Heatmap for the Original dataset

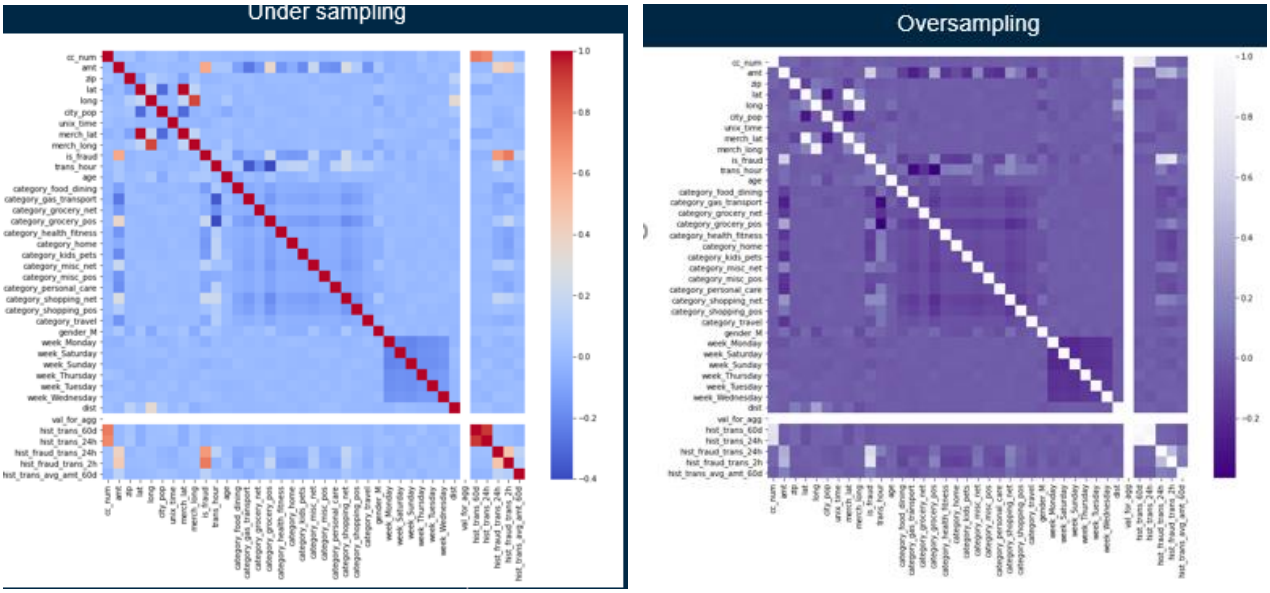


Fig. 10: Heatmap for Undersampling and Oversampling

expedite the training period and improve the learning interpretability and decrease the model over-fitting when there are many unnecessary features contributing no more helpful information than the current subset of variables. The excessive and verbose information in the dataset may hugely influence the performance of our model.

In this study, we have performed feature selection using the lasso technique, which is a tool that helps minimize the cost function. Lasso regression will automatically choose the features that are beneficial to our model, discarding the redundant features. So, the purpose of using Lasso regression for feature selection goals is straightforward: we apply a Lasso regression on our scaled dataset, and we admit only those features that produce a coefficient different from 0. In the output of our feature selections using lasso, as shown in fig. 11, we

```
#Feature Selection using LassoCV

from sklearn.linear_model import LassoCV

#Feature Selection
X = df2[X_cols]  #Feature Matrix
y = df2[Y_cols]         #Target Variable

reg = LassoCV()
reg.fit(X, y)
print("Best alpha using built-in LassoCV: %f" % reg.alpha_)
print("Best score using built-in LassoCV: %f" %reg.score(X,y))
coef = pd.Series(reg.coef_, index = X.columns)

print("Lasso picked " + str(sum(coef != 0)) + " variables and eliminated the other " +
      str(sum(coef == 0)) + " variables")

imp_coef = coef.sort_values()
import matplotlib
matplotlib.rcParams['figure.figsize'] = (8.0, 10.0)
imp_coef.plot(kind = "barh")
plt.title("Feature importance using Lasso Model")
```

```
Best alpha using built-in LassoCV: 0.043864
Best score using built-in LassoCV: 0.057553
Lasso picked 4 variables and eliminated the other 25 variables

Text(0.5, 1.0, 'Feature importance using Lasso Model')
```

Fig. 11: Feature Selection using Lasso

observed that 4 important variables were chosen which will be used for modeling, and the technique eliminated the remaining 25 variables.

3.4 MACHINE LEARNING MODELS

In this study, we have experimented with both supervised, and unsupervised machine learning model to classify the fraudulent transactions. The machine learning models used in this study is discussed in the next subsection. We also discuss the process of model creation and selecting the values of the hyperparameters for the best model.

3.4.1 DECISION TREE

Machine Learning technologies use advanced data analysis algorithms. The most popular algorithm used in Machine Learning applications is called the decision tree model. Decision trees work very quickly and smartly, mainly when used to mine and analyze large

amounts of data. The decision tree model works simply by directing a transaction in a specific direction based on the features generated from the data. It follows a fundamental root question and branches in which the details are used to form particular components that finally culminate in endpoints or the leaves of the tree. Decision trees are non-parametric supervised learning methods that can be used for classification and regression purposes where continuous splitting of data is based on a specific parameter. It consists of Nodes, Edges, and Leaf nodes. An example can be seen below.

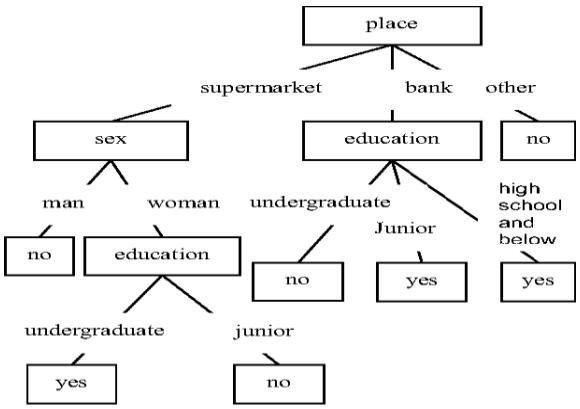


Figure 2. Fraud detection classification decision tree

One of the decision tree objectives is to design a model for training that can be used to predict the class of the response variable. This technique is one of the methods used to make predictions that classify transactions. It is a collection of branches/nodes connected through the edges. Interior nodes of a tree make an assessment, and edges represent the result of the evaluation. The terminal nodes signify a class label. Its function is about using the Depth-first Breadth method to recursively divide the given dataset until all the elements in a set are assigned to a specific class. The advantage of this technique is that no feature scaling is needed, and it is robust to outliers and automatedly and automatically handles

missing values. It spends less time for the training phase and very good at handling classification and regression problems. One major pitfall it has is the single tree may raise complexity and lead to overfitting when the size of the dataset increases. According to Wikipedia “Classification Tree, (Yes/No types) analysis is when the predicted outcome is the class (discrete) to which the data belongs. It is a tree in which an internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with input are labeled with each of the possible values of the response variable that leads to a subordinate decision node on a different input feature [33].” It uses different algorithms to determine whether to divide a node into two or many sub-nodes. The decision tree divides the nodes on every accessible variable and afterward chooses the split, which brings about the most homogeneous sub-nodes. To achieve the quality of splitting into two or more nodes, the decision tree applies the following metrics on the possible subset: Gini impurity, Information gain, Variance reduction, and Measure of goodness. Advantages of Decision Tree are that it can analyze both categorical and numeric data, is straightforward to understand and interpret, does not need too many data preparations, and modeling with large datasets is not a problem. The disadvantages are the non-robustness of the tree, which means any slit change in the data for training can cause a tremendous difference in the tree culminating predictions[33].

3.4.2 LOGISTIC CLASSIFICATION

This is the most uncomplicated technique used to resolve classification and regression problems. It can be used in classifying tumors, emails, spam detection, among others. It

establishes the probability of an output that can be either binomial or multinomial. It adopts the sigmoid function in describing data and the relationship between dependent and independent variables. It can also be used in the current research work to classify a transaction as fraud or not. It is very efficient, although it can overfit high-dimensional datasets. It offers better accuracy and makes no assumptions about the scattering of classes in feature space as some other techniques do. The weakness is that it uses the assumption of linearity between the dependent and independent variables. Classification and regression functions are the two sorts of supervised learning, yet the yield factors of the two assignments are unique. In a regression task, the yield variable is a numeric worth that $\ell = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2$ exists on a constant scale, or to put that a different way, the yield of a regression task is a whole number or drifting point esteem. In other words, the classification task deals with when the result of the algorithm gives one of the different pre-chosen categories with several input variables given and placing them into the exact category they belong to. It can also be called a logic model, a binary classification in which conditional probability of one of the two possible perceptions of the response variable is deduced to match a linear combination of two or more input variables modified by the logistic function. In Binary Classification, the model ought to have the option to predict the response variable as one of the two likely classes, which could be 0 or 1. The Logistic Regression can be clarified with Logistic function, otherwise called Sigmoid function that takes any genuine input x and yields likelihood esteem somewhere in the range of 0 and 1 [34].

3.4.3 RANDOM FOREST

One popularly used machine learning algorithm is Random Forest. It is a technique adopted in solving both classification and regression problems. It is a pool of an enormous number of separate decision trees that are called "forest". Each different tree makes a class prediction. Any class that has the maximum votes is considered for prediction. Thus, the technique adopts a bagging approach in creating a group of decision trees that will build a forest. The strength of this technique is that a feature selection is not needed, and it runs the model quickly and balances the errors smartly. The con of this technique is that it is sensitive to data with diverse values and attributes with more values and can easily flag them as fraud. The 'forest' that this algorithm builds is known as decision tree ensemble, which is usually trained with a method called bagging, an application of Bootstrap strategy to a high variance algorithm used in machine learning [9]. Bagging and Random forests are algorithms that combine multiple models into one package. Both algorithms are very effective in different types of predictive modeling problems. It is one of the best algorithms used in the banking system for fraud detection. Advantage of random forest is that we can use it to solve both classification and regression issues. When the Random algorithm starts to build the tree, it always attaches the randomness, making it more essential to find the topmost feature among all features for modeling, especially during the splitting of the node [17]. The random forest hyperparameter improved the predictive ability of the model or boosted the speed of the model. Overfitting problem is one of the issues we face in machine learning modeling. Still, a random forest classifier helps because of its ability to create many trees in the forest, and the classifier will not overfit the model.

3.4.4 XGBOOST CLASSIFIER

XGBoost means eXtreme Gradient Boosting. An ensemble method algorithm that implements the gradient boosted decision tree is designed for high momentum and outstanding performance [12]. Tianqi Chen creates this algorithm. It is a highly scalable machine learning algorithm that can be used to tackle data science problems. It supports interfaces like C++, Python, R, Julia, Java, Scala, and Command Line Interface (CLI). It is an exceptionally adaptable and flexible apparatus that can work through most regression, Classification, and issues that deal with ranking. The execution of XGBoost offers a few progressed highlights for model tuning, processing conditions, and algorithm upgrades. It can execute all the gradients boosting such as Stochastic gradient boosting, Regularized gradient boosting, etc. Because of its robustness, adding more regularized parameters helps boost the hyperparameters tuning and avoid over-fitting. One of the crucial characteristics of XGBoost is its ability to lessen the time for computation successfully [8]. At the same time, it is also capable of handling missing values through “Sparse Aware, “Block structuring," which enhances parallelization when performing tree assembling, and “Continued Training” with its capacity to fit trained model well even if new data are being added. There are several main types of parameters that we need to run on XGBoost, such as 'General parameters,' which deals with which booster to use while boosting, 'Booster Parameter' which shows the chosen booster, 'Learning task Parameters' rules on the learning layout, and 'Command line Parameters' which investigate the conduct of CLI version explaining the idea of boosting. This ensemble strategy tries to make a solid classifier (model) given "powerless" classifiers. XGBoost explicitly carries out this

algorithm for decision tree boosting with an additional custom regularization term in the goal work.

3.4.5 K-MEANS CLUSTERING

The k-means is an unsupervised clustering algorithm used for the significant clustering of data. It groups unlabeled points into several k – clusters. It is classified as unsupervised because the points have no external classification. The clustering study strategy is one of the principle insightful strategies in data mining, the technique for clustering algorithm will impact the results of the grouping straightforwardly. This paper talks about the standard k-means grouping analysis and algorithm, the inadequacies of standard k-means algorithm; for example, the k-means clustering algorithm needs to ascertain the distance between every data object and all cluster centers in every iteration, which makes the effectiveness of clustering is not high. According to Shi Na et al. 2010, k-means was proposed in 1967 by MacQueen, and it serves as one of the most simple, non-supervised machine learning algorithms that can solve the well-known cluster's problem. K-means algorithm can partition clustering of data. This method calcifies the given data points into several k clusters through the iterative and minimum local convergent [27]. Therefore, the output of the groups that are generated through this process is compact and independent. K-means algorithm contains two different stages. The first stage is where the k centers are selected randomly, where there is an advanced fixed value of k Stage two, where each data point is assigned to the nearest center. The distance connecting each data point and the center of the clusters is generally calculated using the Euclidean distance. The first stage mentioned

earlier is completed when all the data points are grouped in clusters and recalculates the average of the early formed clusters [6]. The process of iteration will continue until the function of the criterion becomes the minimum. The Euclidean distance can also be known as the criterion function, which can calculate the distance between each data point and

$$\left[\sum_{i=1}^{n}(x_i - y_i)^2\right]^{1/2}$$

cluster center. The Euclidean distance connecting two vectors x and y can be denoted as follows $\mathrm{x} = (\mathrm{x_1}, \mathrm{x_2}, \ldots \mathrm{x_n})$ and $\mathrm{y} = (\mathrm{y_1}, \mathrm{y_2}, \cdots \mathrm{y_n})$ while the distance $\mathrm{d}(\mathrm{x_i}, \ \mathrm{y_i})$ of the Euclidean can be given as

3.4.6 AUTOENCODERS NEURAL NETWORKS (ANN)

An Artificial Neural Network (ANN) is an interconnected gathering of processing nodes, for example, "neurons," that together play out an (ordinarily nonlinear) change of contributions to specific ideal outputs. This technique uses a set of neurons connected, and the neurons contribute to the decision-making [2]. ANN uses human thoughts and processing techniques and also capabilities of computers to make predictions for some transactions as fraudulent. It takes it bearing from the previous patterns of operations from the datasets and uses the same design to predict if an existing transaction is fraudulent or not. An autoencoder (an unsupervised machine learning technique that does not need an explicit label to train on) is an extraordinary kind of neural network whose goal is to recreate the contributions instead of anticipating some response variables. An autoencoder

attempts to learn with a consolidated representation of the input data by recreating inputs - an interaction otherwise called "encoding" [Tie Luo et al. 2018]. This technique is suitable for detecting an anomaly in a model. Anomaly detection in data mining is how data points or observations digress from a normal distribution of others [30]. It can also be called an outlier detector. During the modeling phase, anomalous data can designate a captious incident such as equipment faults, technical malfunction, or a change in consumer behavior. Autoencoder consists of an input layer, output layer, one or more hidden layers, and activation function and hyperparameters. [Sai G. Nagarajan et al. 2018] (1) An input layer – this is an M-dimension vector that can denote the input indicator, and it can be represented as $\mathbf{x} = (x_1,\ x_2,\ \ldots,\ x_M)$.

(2) An output layer – a vector denoted by $\hat{\mathbf{x}} = (\hat{x}_1, \hat{x}_2,\ \ldots, \hat{x}_M)$. We should realize that this is different from the standard way of neural network where the output layer is denoted as $\mathbf{y} = (y_1,\ y_2,\ \ldots,\ y_{|\mathbf{y}|})$

On account of autoencoders, the output layer has a similar measurement as the input layer. We might want the output to be equivalent to the contribution to reproduce the first input. Henceforth we naturally get our training samples when we set y = x, which is why autoencoders are known as unsupervised learning models. (3) One or more hidden layers – this layer is between the input layer and output layer, and its objective is to learn the pattern in the input layer and encode valuable facts. Overall, the autoencoder does usually has hidden layers in multiple forms. (4) Activation function and Hyperparameter –

‘sigmoid’ function is always used as the activation function, and it can be represented as

$$f(z) = \frac{1}{1 - e^{-z}}.$$

Several hyperparameters need to be put in place before training can be done in autoencoder, such as (1) code size (nodes numbers), (2) Number of layers (number of layers can be as much as we like), (3) Number of nodes per layer (if it is stacked in nature, it means layers can be stacked one after the other), and (4) Loss function (binary cross-entropy if the value of the input layer is between [0,1] and mean square error can be used if not). [Arden Dertat. 2017] About implementing this algorithm, there is a callable layer called output of 'Dense,' which uses the provided API function in the input and saves the output. The current output layer will become the next input layer for the next layer. In deep learning, the standard activation function used by layers is known as the 'relu' activation function. Still, the last layer uses the sigmoid function because the output has to be between [0,1] while the input is also in the value range. The pros of this technique are its capability to work with incomplete knowledge [2]. It can also store data on the entire network, fault-tolerant, distributed memory, and parallel processes. Nevertheless, it is not with its weakness. Some of the limitations identified are hardware-dependent on the determination of appropriate network structure, and the duration of the network is unfamiliar. It also has some unexplained behavior of the network.

3.5 MODEL CREATION

In this section, we present the specifications on model creation. Following preprocessing the dataset, data are split into training and test. The training data is used to define the parameters for the models while the test set is used to evaluate our models.

3.6 SPLITTING OF DATA INTO TRAINING AND TEST

The main objective of the machine learning model is to learn from previous experience and its ability to make use of the information to generate new instances. Performance evaluation of the model is usually done on the subset of the whole dataset by training on it, and the remaining dataset can be used to evaluate the model's performance. In this study, our dataset was split into a 70:30 ratio; that is, 70% of the dataset is used for training the model and the remaining 30% to evaluate the model's performance. Parameters, often called hyperparameters of the model, are determined during model training, and these hyperparameters also helped find the best model fit for a machine learning model. More advantage of hyperparameters will be explained in the next subsection of this study.

3.7 HYPERPARAMETER TUNING

When creating a machine learning model is done, we will be given design options as to how to determine our model architecture. Often, we don't instantly know what the optimal model architecture should be for the assigned model, and thus we'd like to be able to examine a range of chances. In the proper machine learning method, we will ideally ask the machine to achieve this exploration and automatically decide the optimal model architecture. Parameters that determine the model architecture are called hyperparameters**,**

and therefore this process of exploring for the perfect model architecture is referred to as hyperparameter tuning. The hyperparameter addresses the model design questions such as what degree of a polynomial can be used for linear models, what is the minimum and maximum depth allowed for a decision tree, how many trees should be created in a random forest, how many layers of neurons should we have in case of neural network layer creation and what should the learning for the gradient descent be? In this study, we have performed hyperparameter tuning on our best model and to ensure that our model is not overfitting, 3 – fold cross validation was carried out during the grid search. We use used a python function called GridsearchCV throughout the whole process of hyperparameter tuning. We set our n_estimator to be 100, learning rate to 0.1 maximum depth is between 4 and 8, and our minimum sample leaf's is set between the range of 4 and 6.

CHAPTER FOUR

4.1 FINDINGS AND RESULTS

In this chapter, we present the results of our finding with the machine learning models that we discussed in the previous chapter. To evaluate the performance of our model, we adopted the use of a metric called AUC score and other metrics to evaluate the performance of our model. The Metrics of each model will be shown based on how they have performed with our original, undersampling and oversampling dataset, then we present a comparative study to determine which of our model is the best for predicting of credit card fraud.

4.2 METRICS

Evaluating the performance of the machine learning algorithms is an essential part of any research work. This will show how each of the algorithms performed and to know which gives satisfactory or unsatisfactory results. We often use accuracy to weigh the model performance in classification algorithms, although it is not the only true way to judge the model. In this study, evaluation metrics like F1-Score, Precision, Recall, Confusion matrix, Accuracy, and ROC AUC Score (which happens to be the primary metric we have used to evaluate our model) [1]. We have made AUC score the primary evaluation metric because it is the most widely used metric among all metrics, it shows the score and likewise the plot that shows how each model have performed.

4.2.1 ACCURACY

Accuracy is the ratio of the correct prediction number to the total number of input samples. It functions admirably just if there are an equivalent number of samples having a place with each class. For instance, consider 98% examples of class A and 2% examples of class B in our training set. Then, at that point, our model can undoubtedly get 98% accuracy by basically anticipating each training sample to be allied to class A. When a similar model is tried on a test set with 60% examples of class A and 40% examples of class B, then, at that point, the test accuracy would be reduced to 60%. Classification Accuracy is extraordinary; however, it gives us the misguided feeling of accomplishing high precision.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

4.2.2 RECALL

Recall can be calculated when the correct positive number results are divided by the number of all samples, which should have been recognized as a positive value.

$$Recall = \frac{True\ Positives}{True\ Positives+False\ Negatives}$$

4.2.3 PRECISION

Precision is dividing the correct positive number results by the number of positive results that the classifier predicted.

$$Precision = \frac{True\ Positives}{True\ Positives+False\ Positives}$$

4.2.4 F1-SCORE

F1-score is used to evaluate the test's accuracy. It is the consonant mean between recall and precision. It allows a report on how precise the Classification is and how strong it can be. If a result gives high precision but low recall, it means we have incredibly high accuracy but note; it may miss a very high number of possibilities that are hard to classify. In short, it means the higher the F1 score, the best the model performed. It can be calculated using

$$F1 = 2 \times \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

4.2.5 CONFUSION MATRIX

Confusion Matrix gives us a complete breakdown of the model performance in terms of matrix output. It evaluates well, especially when working with a binary classification where we have samples that belong to two classes: TRUE or False, YES or NO.

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Image source: https://glassboxmedicine.com/2019/02/17/measuring-performance-the-confusion-matrix/

The four important terms we have are True Positives, True Negatives, False Positives, and False Negatives.

- True Positives: this is the case where the algorithm predicted YES, and the true output came out YES.
- True Negatives: this is the case where the algorithm predicted NO, and the true output came out NO.
- False Positives: this is the case where the algorithm predicted YES, and the true output came out NO.
- False Negatives: this is the case where the algorithm predicted NO, and the true output came out YES. The accuracy of the confusion matrix can be calculated by

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample}$$

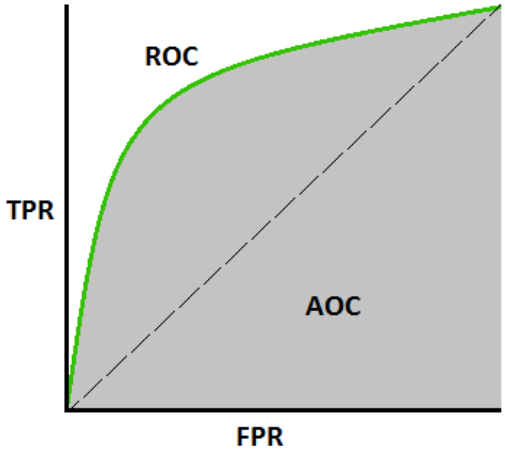
4.2.6 ROC AUC SCORE

ROC AUC Score: ROC (Receiver Operating Characteristics) AUC (Area Under Curve) is a widely used metric for model evaluation. AUC is the degree of measurement for separability, which reports how the model can differentiate between classes. Classification problems should measure performance with different thresholds been set. A better model can predict 0 classes as 0 and 1 classes as 1, while this can be confirmed if the AUC score is high. ROC is the curve probability [26]. This ROC curve plots the TPR (True Positive Rate) y-axis against the FPR (False Positive Rate) x-axis.

TPR (True Positive Rate) / Recall /Sensitivity = $\frac{TP}{TP+FN}$

Specificity = $\frac{TN}{TN+FP}$

$FPR = 1 - Specificity = \frac{FP}{TN+FP}$



AUC-ROC Curve Image source: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

The results of different model used for this study based on different dataset such as undersampling and oversampling are shown. Comparison was made to choose the best predictive model using the AUC score as metric and comparing the metric with other metrics to further established how good each model has performed. The area under the curve known as (AUC) is the same as the probability that a model will rank a randomly chosen positive instance higher than a randomly chosen negative example. The higher the score of AUC, the better the model is when predicting fraudulent and non-fraudulent transactions. If we are trying to identify the strength of a model to differentiate between two outcomes AUC is a metric that can help identify such because it creates a clear

boundary between the positive class and the negative class. The result of each classifier is given below.

4.3 MODELLING ORIGINAL DATASET

	Logistic Reg.	Decision Tree	Random Forest	XGBoost	K-means Clustering	Autoencoders
Original dataset	0.86	0.99	0.99	0.99	0.50	0.99

Table 1: The AUC Score modeling with the original dataset

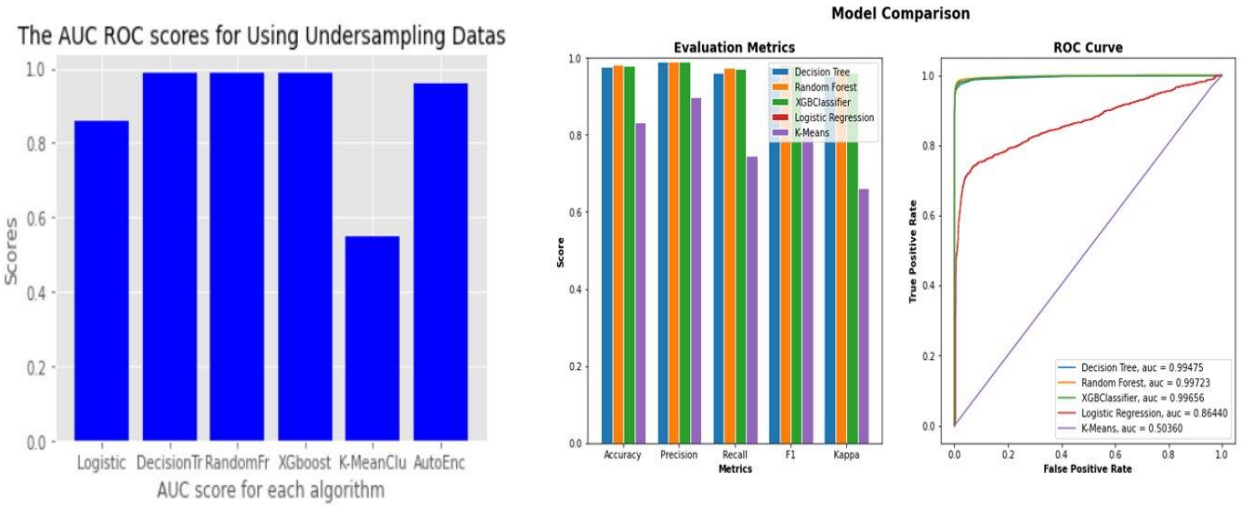


Fig. 12. The bar chart of the AUC score for each classifier and the plot of the AUC score comparing to other metrics.

From the above table 1 and fig. 12, the performance of each algorithm was shown based on the AUC Score, and we can see that all the trees' algorithms and the autoencoder having the highest AUC Score of about 0.99 % and the least performing algorithm based on using

the original dataset for modeling is k-means clustering with a score of 0.50 %. For a model to be considered as the best performing model, they have to have a high AUC Score. We cannot trust these results because we are modeling with the original dataset directly without any form of pre-processing. There might be some kind of biasness in our result as the dataset is highly imbalanced and some of the classifiers cannot perform very well with imbalanced dataset. To further improved the performance of our classifiers since we cannot depend on the results of our classifiers based on the original dataset, we have carried out re-sampling techniques (undersampling and oversampling) to balance the dataset and the results are given below.

4.4 MODEL RESULT FOR UNDERSAMPLING DATASET

Since most of the instances in the dataset belong to the majority class, the dataset was under-sampled randomly and this was achieved by reducing the numbers of instances of the majority class, which means that some essential data instances are not captured for training purposes in the data.

	Logistic Reg.	Decision Tree	Random Forest	XGBoost	K-means Clustering	Autoencoders
Undersampling Dataset	0.74	0.99	0.99	0.99	0.50	0.96

Table 2: The AUC Score of the Undersampling data model.

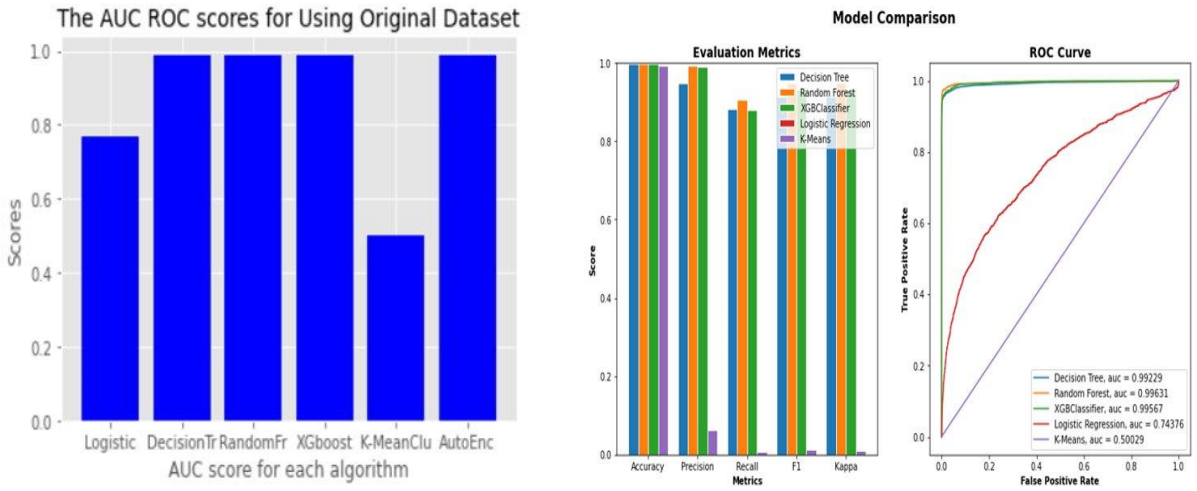


Fig. 13: The bar chart of the AUC score for each classifier and the plot of the AUC score comparing to other metrics.

Based on the output in table 2 and fig. 13, using the Undersampling data for modeling it shows that the ensemble trees' algorithms still perform better with AUC Score of about 0.99 %. The performances of Random Forest, decision trees, and Xgboost are similar to that of model with original dataset. Their learning rates was set at 0.1, max depth at 5, this determines the maximum depth of a tree. The subsampling is 0.5 which means that the algorithms would sample half of the training data randomly which will prevent overfitting. Comparing the AUC score of the models with other metrics shows that the tree algorithms have a high precision rate and low recall. The lowest performing model for Undersampling is k-mean which belongs to unsupervised classification clustering. Unsupervised classification is also called clusterization that groups objects into k groups based on common characteristics. The k-means model has an AUC score of 0.50 % shows that it might not be the best algorithm to use since the problem at hand is a classification problem

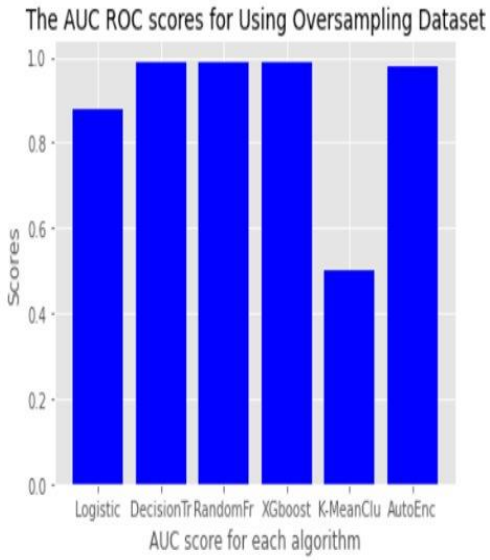
and the dataset already have a predefined classes in which objects are assigned, this means our dataset is already predefined and labeled but clustering only look into similarities among objects grouped together according to common characteristics that separate them from others.

4.5 MODEL RESULT FOR OVERSAMPLING DATASET

This method duplicates new or sometimes simulates examples in the minority class. It increases the instances, which makes the training of the model to be better.

	Logistic Reg.	Decision Tree	Random Forest	XGBoost	K-means Clustering	Autoencoders
Oversampling Dataset	0.87	0.99	1.00	0.99	0.50	0.98

Table 3: The AUC Score of the Oversampling data model.



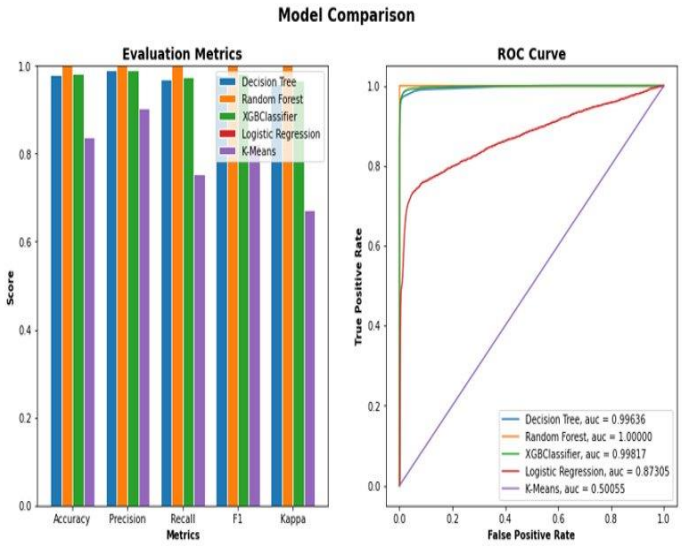


Fig. 14: The bar chart of the AUC score for each classifier and the plot of the AUC score comparing to other metrics.

From the result of the model performance in table 3 and fig. 14, it is shown that all the algorithms are performing very well with the oversampling dataset comparing to undersampling dataset and we can see how the trees' algorithms outperformed other algorithms especially the random forest which shows an AUC score of 1.00 %. Decision tree and Xgboost shows a 0.99 % AUC score. We also noticed the improvement in the Score of other algorithms which indicated that most algorithm works well with oversampling dataset rather than undersampling. Comparing the AUC metric with other metrics shows that Random Forest algorithm is still the best among the trees' algorithms. Accuracy has %high precision and low recall which means we can as well rely on its prediction results for the credit card fraud.

4.6 HYPERPARAMETER TUNNING WITH THE BEST MODEL

To further evaluate how some of our best algorithms has performed, we carried out hyperparameter tuning on the algorithms. Hyperparameter helps in choosing a set of optimal parameters for a learning algorithm because the key to machine learning classifiers.

	Decision Tree	Random Forest	XGBoost	Autoencoders
Hyperparameter Tunning	0.99	1.00	0.99	0.98

Table 4: The AUC Score of the hyperparameter tuning.

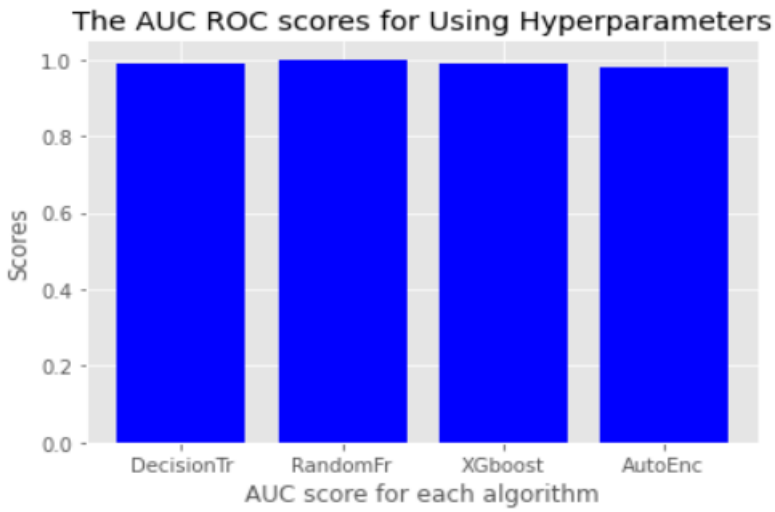


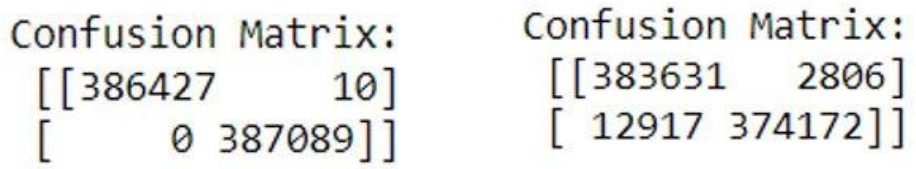
Fig. 15: The bar chart for each algorithm.

Table 4 and fig. 15, shows that with hyperparameter tuning, our best model is still the ensemble tree algorithms such as Random Forest, decision Tree and Xgboost with the AUC Score of 1.00 %, 0.99%, and 0.99% respectively. The neural network model also shows an improvement with of 0.98 %.

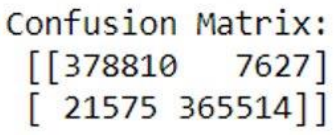
4.7 COMPARATIVE ANALYSIS

In this section, a comparative analysis of our model was made based on the types of datasets and the result of the metrics used to measure how each algorithm has performed. Based on the performance of our model with a different dataset that we have explored for this study using the AUC score to evaluate the performances and pick the best overall model, we observed that with the original dataset, undersampling, and oversampling dataset; the ensemble tree model performed very well rather than other model using the AUC score, the accuracy, precision, recall, and F1-score to compare between them. Fig. 16 shows the

confusion matrix for the ensemble tree model, that is, Random Forest, Xgboost, and Decision Tree; we can see from the output of the confusion matrix for Random Forest, the true positive result is 387089, which means out of all the total number of transactions of 386427 that was used for testing, Random Forest was able to predict correctly that 387089 transactions can be flagged as fraudulent transactions. Its false negative shows a value of 0 which means Random Forest did not incorrectly identified any fraudulent transactions as genuine transactions. In this case, the algorithm did not flag any transaction as fraudulent or genuine. 10 genuine transactions incorrectly identified as fraud.



Confusion Matrix for Random Forest and Xgboost model



Confusion Matrix for Xgboost classifier.

Fig. 16: Confusion matrix of the ensemble tree models

Comparing the confusion matrix's result of Random Forest with Xgboost and Decision Tree, we can see that Xgboost, and Decision tree have a true positive value of 374172 and 365514, respectively, which means the two models were able to correctly predict these cases of fraudulent transactions as fraud and the false-negative shows that they both incorrectly predicted 12917 and 21575 genuine transactions as a fraudulent transaction.

Methods	Accuracy	Precision	AUC score
Random Forest	1.0000	1.0000	1.0000
Decision Tree	0.9522	0.9881	0.9982
Xgboost	0.9853	0.9822	0.9996
Logistic Regression	0.8179	0.7562	0.8761
Autoencoder	0.9821	0.9011	0.9889
K-means cluster	0.5001	0.7312	0.5005

Table 5: Accuracy, Precision and AUC score after hyperparameter tuning

Comparing all these results of how each ensemble tree algorithm has performed based on the metric, Random Forest has the highest AUC score, accuracy, and precision, as shown in table 5. Hence, we have selected the Random Forest algorithm as the best model for predicting credit card fraud.

CHAPTER FIVE

In this chapter, we present the summary result along with the limitation and future research direction.

5.1 CONCLUSION

The technology change influenced several improvements. We are talking about online transactions done through credit cards, which leads to credit card frauds, and this study is about improving machine learning algorithms for fraud detection. In this study, we put forth fraud detection methods based on supervised learning such as Random Forest, Decision Tree, Xgboost, and logistic regression, unsupervised learning such as K-means clusters, and one deep learning algorithm known as Autoencoder Neural Network. We compared all the algorithms with different datasets by first using the original dataset itself; we then use resampling techniques such as undersampling and oversampling because our dataset is highly imbalanced. Finally, we concluded that Random Forest would be the perfect fit for our model. It can be inferred that oversampling works better because the smaller number of observations helps in training our model efficiently. Oversampling will be an ideal sampling technique in the real-world scenario as the information containing a pattern is not lost.

5.2 LIMITATION OF THE STUDY

Credit card fraud detection is exceedingly difficult but also a general problem for solution. As there is an inadequate amount of data with the transactions entrusted, current fraud

solutions have limited data visibility and, therefore, produce significant false positives. For example, precise information about the amount, merchant categories or locations, date and time of transactions are withheld from the general public, which means researchers must work with limited information provided.

5.3 FUTURE RESEARCH

Based on the conclusion we have above, we can further improve our approach in this study. Future work can be done on this topic:

The current study agreed with the result of the oversampling dataset, which duplicates. Instead of using this oversampling method that duplicates, we can use another sampling technique called interpolation, where redundant observations are not added to our dataset.

REFERENCES

[1] Aditya Mishra. (2018). Metrics to Evaluate your Machine Learning Algorithm. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234

[2] Arden Dertat. (2017). Applied Deep Learning – autoencoder. https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

[3] Azhan, Mohd. (2020). Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques. 10.1109/ICISS49785.2020.9316002.

[4] Consumer Financial protection Bureau. (2019). https://www.consumerfinance.gov/data-research/research-reports/the-consumer-credit-market-2019/

[5] Dornadula, Vaishnavi & Geetha, S. (2019). Credit Card Fraud Detection using Machine Learning Algorithms. Procedia Computer Science. 165. 631-641. 10.1016/j.procs.2020.01.057.

[6] Guo G., Wang H., Bell D., Bi Y., Greer K. (2003). KNN Model-Based Approach in Classification. In: Meersman R., Tari Z., Schmidt D.C. (eds) On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE. OTM 2003. Lecture Notes in Computer Science, vol 2888. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-39964-3_62

[7] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. (2017). "Credit card fraud detection using machine learning techniques: A comparative analysis," International Conference on Computing Networking and Informatics (ICCNI), 2017, pp. 1-9, Doi: 10.1109/ICCNI.2017.8123782.

[8] Jason Brownlee. (2021). A Gentle introduction to XGBoost for Applied Machine Learning. https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

[9] Jason Brownlee. (2021). Bagging and Random Forest for imbalanced Classification. https://machinelearningmastery.com/bagging-and-random-forest-for-imbalanced-classification/

[10] Jisha, M.V. & Vimal, D. (2020). Population-based Optimized and Condensed Fuzzy Deep Belief Network for Credit Card Fraudulent Detection. International Journal of Advanced Computer Science and Applications. 11. 10.14569/IJACSA.2020.0110970.

[11] Joshi, Aruna & Shirol, Vikram & Jogar, Shrikanth & Naik, Pavankumar & Yaligar, Annapoorna. (2020). Credit Card Fraud Detection Using Machine Learning Techniques. International Journal of Scientific Research in Computer Science, Engineering, and Information Technology. 436-442. 10.32628/CSEIT2063114.

[12] KDnuggets. (2017). XGBoost, a Top Machine Learning Method. https://www.kdnuggets.com/2017/10/xgboost-top-machine-learning-method-kaggle-explained.html

[13] Lebichot, Bertrand & Braun, Fabian & Caelen, Olivier & Saerens, Marco. (2017). A graph- based, semi-supervised, credit card fraud detection system. 693. 721-733. 10.1007/978-3-319-50901-3_57.

[14] Meng, Cuizhu & Zhou, Li & Liu, Bisong. (2020). A Case Study in Credit Fraud Detection with SMOTE and XGBoost. Journal of Physics: Conference Series. 1601. 052016. 10.1088/1742-6596/1601/5/052016.

[15]]Mohari, Ankit & Dowerah, Joyeeta & Das, Kashyavee & Koucher, Faiyaz & Bora, Dibya & Bora. (2021). A COMPARATIVE STUDY ON CLASSIFICATION ALGORITHMS FOR CREDIT CARD FRAUD DETECTION.

[16] More, Rashmi & Awati, Chetan & Shirgave, Suresh & Deshmukh, Rashmi & Patil, Sonam. (2021). Credit Card Fraud Detection Using Supervised Learning Approach. International Journal of Scientific & Technology Research. 9. 216-219.

[17] Niklas Donges. (2021). A complete guide to the Random Forest algorithm. https://builtin.com/data-science/random-forest-algorithm

[18] Paige Schaffer. (2018). Reducing the Impact of New Account and Credit Card Fraud on Financial Institution. https://www.cpomagazine.com/cyber-security/reducing-the-impact-of-new-account-and-credit-card-fraud-on-financial-institutions/

[19] Parmar, Jasmin & Patel, Achyut & Savsani, Mayur. (2020). Credit Card Fraud Detection Framework - A Machine Learning Perspective. International Journal of Scientific Research in Science and Technology. 431-435. 10.32628/IJSRST207671.

[20] Priya, G. & Saradha, S. (2021). Fraud Detection and Prevention Using Machine Learning Algorithms: A Review. 564-568. 10.1109/ICEES51510.2021.9383631.

[21] Rahmawati, Dewi & Sarno, Riyanarto & Sunaryono, Dwi & Fatichah, Chastine. (2017). Fraud Detection on Event Log of Bank Financial Credit Business Process using Hidden Markov Model Algorithm. 10.1109/ICSITech.2017.8257082.

[22] Rocha, Bruno & de Sousa Junior, Rafael. (2010). Identifying Bank Frauds Using CRISP-DM and Decision Trees. International Journal of Computer Science & Information Technology. 2. 10.5121/ijcsit.2010.2512.

[23] Roy, Parth & Rao, Prateek & Gajre, Jay & Katake, Kanchan & Jagtap, Arvind & Gajmal, Yogesh. (2021). Comprehensive Analysis for Fraud Detection of Credit Card through Machine Learning. 765-769. 10.1109/ESCI50559.2021.9397029.

[24] S P, Maniraj & Saini, Aditya & Ahmed, Shadab & Sarkar, Swarna. (2019). Credit Card Fraud Detection using Machine Learning and Data Science. International Journal of Engineering Research and. 08. 10.17577/IJERTV8IS090031.

[25] Sadineni, Praveen Kumar. (2020). Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms. 659-660. 10.1109/I-SMAC49090.2020.9243545.

[26] Sarang Narkhede. (2018). Understanding AUC – ROC Curve. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

[27] Shi Na, Liu Xumin, Guan Yong. (2020) Research on k-means Clustering Algorithm: An improved k-Means Clustering Algorithm. Third International Symposium on Intelligent Information Technology and Security Informatics. DOI: 10.1109/IITSI.2010.74

[28] Shirgave, Suresh & Awati, Chetan & More, Rashmi & Patil, Sonam. (2019). A Review on Credit Card Fraud Detection Using Machine Learning. International Journal of Scientific & Technology Research. 8. 1217-1220.

[29] Singh, Ajeet & Jain, Anurag. (2020). Cost-sensitive metaheuristic technique for credit card fraud detection. Journal of Information and Optimization Sciences. 41. 1-13. DOI:10.1080/02522667.2020.1809090.

[30] Tie Luo, Sai G. Nagarajan. (2018). Distributed Anomaly Detection Using Autoencoder Neural Networks in WSN for IoT. 2018 IEEE International Conference on Communications (ICC). DOI: 10.1109/ICC.2018.8422402

[31] Raynor de Best. U.S. Credit cards – statistic and fact. (2020). https://www.statista.com/topics/1118/credit-cards-in-the-united-states/

[32] Warghade, Swati & Desai, Shubhada & Patil, Vijaykumar. (2020). Credit Card Fraud Detection from Imbalanced Dataset Using Machine Learning Algorithm. International Journal of Computer Trends and Technology. 68. 22-28. 10.14445/22312803/IJCTT-V68I3P105.

[33] Wikipedia contributors. (2021). Decision tree learning. In *Wikipedia, The Free Encyclopedia*. Retrieved June 11, 2021, from https://en.wikipedia.org/w/index.php?title=Decision_tree_learning&oldid=1027552736

[34] Wikipedia contributors. (2021). Logistic regression. In *Wikipedia, The Free Encyclopedia*. Retrieved June 12, 2021, from https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=1024548320

[35] Matt Tatham. Identity theft Statistics. (2018). https://www.experian.com/blogs/ask-experian/identity-theft-statistics/