UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

COMPARING ALGORITHMS FOR PREDICTIVE DATA ANALYTICS

GORAN KIROV

Thesis supervisor: ALFREDO VELLIDO ALCACENA (Department of Computer Science)

Thesis co-supervisor: LILI NEMEC ZLATOLAS

Degree: Master's degree in data science

Thesis report

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC) - BarcelonaTech

26/06/2023

COMPARING ALGORITHMS FOR PREDICTIVE DATA ANALYTICS

Master's degree in data science

Student: Goran Kirov

Thesis supervisor: Dr. Alfredo Vellido Alcacena UPC, FIB

Thesis co-supervisor: Dr. Lili Nemec Zlatolas UM, FERI

Degree: Master's Degree in data science

ACKNOWLEDGEMENTS

Special thanks go to my family, Nikola, Tanja, and Kosta for their constant support and encouragement throughout my studies.

I would like to thank my mentor, Professor Dr. Alfredo Vellido Alcacena, for his help and guidance in completing my master’s thesis. I would also like to thank my co-mentor, Professor Dr. Lili Nemec Zlatolas, for her assistance.

COMPARING ALGORITHMS FOR PREDICTIVE DATA ANALYTICS

Keywords: data analytics, machine learning, classification, evaluation metrics

Abstract

The master's degree thesis is composed of theoretical and practical parts. The theoretical part describes the basics of predictive data analytics and machine learning algorithms for classification such as Logistic Regression, Decision Tree, Random Forest, SVM, and KNN. We also describe different evaluation metrics such as Recall, Precision, Accuracy, F1 Score, Cohen's Kappa, Hamming Loss, and Jaccard Index that are used to measure the performance of these algorithms. Additionally, we record the time taken for the training and prediction processes to provide insights into algorithm scalability.

The key part master's thesis is the practical part that compares these algorithms with a self-implemented tool that shows results for different evaluation metrics on seven datasets. First, we describe the implementation of an application for testing where we measure evaluation metrics scores. We tested these algorithms on all seven datasets using Python libraries such as scikit-learn. Finally, we analyze the results obtained and provide final conclusions.

Contents

List of figures

List of code snippets

1. Introduction

1.1 Background and motivation

Predictive data analytics and machine learning algorithms for classification have picked up significance in later years due to their capacity to analyze gigantic sums of information and deliver exact expectations. These strategies are becoming increasingly noteworthy in advanced times since they have been effectively connected in numerous areas, such as marketing, finance, retail, and healthcare [1].

Based on a set of training data, classification machine learning algorithms are used to predict the class labels of future instances. Logistic Regression, Decision Tree, Random Forest, SVM, and KNN are some of the algorithms that are most frequently used. The data assumptions made by these algorithms and the way they describe the connection between the input attributes and the output labels vary. Evaluation metrics such as Recall, Precision, Accuracy, F1 Score (Harmonic Mean of Precision and Recall), Cohen's Kappa, Hamming Loss, and Jaccard Similarity Score are used to assess the performance of the algorithm. Additionally, the time taken for training and prediction is used as an indicator of the efficiency of the algorithm. The aforementioned metrics and indicators provide a means to compare the accuracy and efficiency of diverse algorithms and decide which one is most appropriate for a specific issue.

The motivation of this thesis is to evaluate these algorithms with a self-implemented tool that displays outcomes for several assessment measures across multiple datasets. Our objective is to carefully look at these algorithms on the datasets to decide which performs the finest under different conditions, and to supply smart recommendations for progressing these algorithms.

1.2 Problem statement

The identification of an optimal algorithmic approach for a given assignment constitutes a critical obstacle within the realm of predictive data analytics and machine learning. The availability of large datasets is rapidly escalating because of the continual collection of user data, which algorithms utilize to assign value to information, thereby emphasizing the significance of accurate and feasible computations [4]. The process of selecting an appropriate algorithm does not conform to a universal, standardized solution. In fact, each algorithm entails inherent advantages and disadvantages, which vary according to the specific nature of the problem in question and the data involved. Consequently, it is fundamental to thoroughly assess and compare diverse algorithms to determine the optimal solution for a specific assignment.

The classification algorithms used in this study are Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). The datasets were selected to encompass a wide range array of issue domains. We aim to discern the algorithm that exhibits optimal performance across multiple scenarios and to present recommendations about the enhancement of algorithm accuracy and effectiveness.

Machine learning requires evaluation metrics because they give us a way to gauge how well our algorithms are performing. The investigation of various evaluation metrics and how well they can be used to gauge the effectiveness of classification algorithms is the second objective of this study.

The master's thesis practical section compares algorithms for predictive data analytics. We will identify the ideal algorithm for each objective by doing a thorough examination. To accomplish this, we will create a tool that will respond to the following research questions.

- RQ1: What is the accuracy, effectiveness, and scalability advantages and disadvantages of the classification algorithms when used to solve different issue domains?
- RQ2: Which evaluation metrics are the best suitable for various kinds of datasets and issue domains, and how do different evaluation metrics compare in their capacity to accurately quantify the performance of classification algorithms?
- RQ3: When used on datasets with different levels of complexity and class imbalance, how do various classification algorithms perform in terms of prediction accuracy?
- RQ4: What effect does dataset size have on the scalability and computational effectiveness of various classification algorithms, and how does it affect their prediction performance?

1.3 Objectives

This central focus of our thesis aims to provide a comprehensive assessment and comparison of several commonly employed algorithms and metrics in the domains of predictive data analytics and machine learning.

The objectives of the master's thesis are:

- To give an in-depth overview of the different classification algorithms used in machine learning and predictive data analytics, such as Logistic Regression, Decision Tree, Random Forest, SVM, and KNN, as well as to define countless assessment criteria for gauging algorithm success.
- To develop a self-designed tool for algorithm testing, measuring evaluation metric scores, and comparing the tool's outcomes for appraising and comparing the performance of classification algorithms. The assessment will take into consideration accuracy, efficiency, and user-friendliness.
- To evaluate and compare the performance of the multiple datasets and the aforementioned classification algorithms, using a variety of assessment criteria

to choose the best method for a given task and offer suggestions for increasing precision and efficiency.

1.4 Assumptions and limitations

Here are some of the assumptions that this study has made along the way:

- It is expected that the datasets used in this study will accurately reflect the population from which they were derived.
- We think the classification algorithms used in the research are appropriate for the problem domains.

This study has the following limitations:

- Because only a few issue domains were covered by the datasets used in this study, it’s possible that the findings won’t apply to other issue domains.
- The study only makes use of open-source and free Python libraries, so there may be certain limitations on how well the algorithms work.
- Due to time constraints, we could only take into account a some of evaluation metrics, so it’s possible that we missed some crucial metrics.

1.5 Structure

The focus of this thesis is on machine learning algorithms for classification and predictive data analytics. The theoretical background section gives an overview of these topics, which include Logistic Regression, Decision Tree, Random Forest, SVM, and KNN. It also discusses different evaluation metrics employed to determine the efficacy of algorithms. The methodology section outlines our study and research process. The application of this thesis compares these algorithms using a self-made program that displays the outcomes for various evaluation criteria on multiple datasets. Using scikit-learn and other Python libraries, we tested these algorithms on both datasets. The results have been scrutinized, prompting the formulation of suggested courses of action. The structure of this thesis consists of several essential components including the introduction, theoretical background, methodology, practical aspects, and conclusion. This thesis makes a contribution by rating and contrasting the effectiveness of different classification algorithms on datasets, depending on the criteria employed to gauge the outcomes. In conclusion, this thesis provides a summary of its notable contribution, delineates the findings, and proffers recommendations for future research.

2. Overview

2.1 Predictive Data Analytics

The growing relevance of Predictive Analytics can largely be attributed to the recent developments in technology and the expansion of Big Data. Organizations are now more inclined towards utilizing data to acquire profound knowledge, thanks to enhanced software, faster and more economical computers, and a greater amount of data on hand. Predictive analytics is a contemporary analytical technique that employs data, algorithms, and machine learning to make predictions concerning future events or outcomes. The objective of this approach is to employ historical data to predict the probability of forthcoming results, thereby providing organizations with informative data and conferring upon them a strategic advantage. Machine learning systems applied to predictive analytics modify their behavior autonomously depending on the patterns detected in the data collection. Using data mining, AI (artificial intelligence), and statistical techniques, this technology collects, analyzes, comprehends, and transforms data. Predictive analytics has a main characteristic, which is that it is solely capable of charting out likelihoods derived from past data, lacking the ability to anticipate the future [2][4].

Predictive models are built using a variety of methods like data mining, statistical modeling and machine learning algorithms. Through the analysis of extensive quantities of data accumulated from diverse data repositories dispersed across the enterprise, predictive models can facilitate businesses in the identification of potential opportunities and threats. There exist numerous approaches for predictive analytics. However, the building of predictive models remains an extensively favored technique. Predictive models can predict several results by analyzing past data, such as the success of a specific product or the possibility of reducing the production cycle by changing suppliers or the consumer’s acceptance of modified packaging [3].

2.1.1 Types of Predictive Models

Within this field, there are three primary models that are frequently used to gain insights into huge and complex datasets. The application of classification models enables the identification of interrelationships within a dataset and the classification of information based on existing data. Logistic regression, decision trees, random forests, neural networks, and Naive Bayes are examples of common classification models. Rather than using labels or categories, clustering models gather information based on comparative traits. Time series models, in contrast to other methods, scrutinize data at specific temporal frequencies to evaluate trends, patterns, and regularity [3][5].

2.1.2 Application of Predictive Analytics

Despite the reality that predictive analytics has been used extensively in many fields for an extensive period, the present era has become synonymous with the practice due to the technology innovation and expanded dependence on the information. The employment of predictive analytics is increasingly gaining traction among businesses as an effective means of augmenting revenues and maximizing profits. This appeal is spurred by numerous factors, among them the proliferation of data in terms of both quantity and diversity, which has prompted the employment of predictive analytics as a means of extracting valuable insights. Also, processing can be done in faster and more affordable ways. The choice of software options is plentiful, with continuous advancements being made toward increasing user accessibility. The competitive environment of expanding the business profitably and the organization's economic circumstances drive them to use predictive analytics. We have compiled a list of common applications [6].

- The banking and financial services industry relies heavily on predictive analytics as a fundamental operation tool. The acquisition of insights from data and the analysis of monetary flows holds a prominent degree of significance in both industries, The use of predictive analytics makes it easier to spot spurious buyers and transactions. It reduces these industries' credit risk when lending money to their clients. It aids in opportunities for cross-selling and up-selling, as well as in retaining and luring valuable customers [6].

- The healthcare sector employs the utilization of predictive analytics to monitor particular infections, such as sepsis, and oversee the treatment of patients suffering from chronic illnesses [7]. This method facilitates effective patient care management, thereby enhancing health outcomes. The application of predictive analytics is embraced by the insurance and pharmaceutical sectors with the intent of optimizing their respective operational functions. On the other side, the insurance industry makes use of predictive analytics models to identify and anticipate instances of client fraud claims [6].
- Supply chain management is one of the primary domains in which predictive analytics has become an integral tool in contemporary business operations. Such analytics perform a vital role in managing product inventories, ascertaining pricing policies, and gauging the cost-benefit analyses surrounding various goods and services over protracted periods. With the utilization of data analysis, enterprises can anticipate how elements like import expenses will impact revenue and fulfill customer needs without excessive inventory in storage [7].
- The retail industry relies on predictive analytics to anticipate customer actions and predict their responses to products. It allows enterprises to establish pricing and develop targeted marketing offers explicitly designed for individual consumers. By identifying and forecasting product demand in particular areas, predictive analytics assists retailers in improving product availability and forecasting the success of products during various seasons [6].
- Sales and marketing departments possess knowledge of business intelligence reports as a means to comprehend previous sales performances. However, the implementation of predictive analytics allows organizations to advance further and take a proactive approach toward customer engagement throughout the entirety of the customer lifecycle [7].

2.2 Machine Learning Algorithms for Classification

The confusion between predictive analytics and machine learning is a common occurrence, although predictive analytics is a subcategory of machine learning. To anticipate future outcomes, numerous statistical methodologies referred to as predictive analytics are used to analyze historical and current data. Contrarily, machine learning involves teaching a computer how to learn from data like humans or animals learn [5]. The application of machine learning algorithms is versatile and can be implemented across diverse domains such as finance, news, investment, marketing, and identification of fraudulent activities [9]. Machine learning focuses on classification, which entails designing the best possible mapping between the data domain and the knowledge base [8].

The area within computer science called machine learning centers on mathematical models and algorithms that are designed to acquire knowledge and gain insight from extensive data sets. It is a process of developing a statistical model from a dataset to tackle real-world problems. The field of machine learning is concerned with accomplishing classification objectives through the development of learning algorithms that can effectively map data domains to knowledge bases. The most popular categorization technique known as supervised learning requires training data sets, validation data sets, and test data sets. The validation data set helps prevent overfitting, the test data set helps determine the model's accuracy, and the training data set helps determine the model's ideal parameter settings [8].

The most standard predictive models include decision trees, regressions (linear and logistic), and neural networks, which are emerging deep learning methods and technologies [5].

2.3 Logistic Regression

Logistic regression is a supervised learning algorithm for binary classification. This concept originates from statistical analyses and anticipates the probability that a particular input can be categorized into a single primary class. The practical application of this approach involves the categorization of resulting outputs as either “primary class” or “not primary class” [10].

The standard probabilistic statistical classification model known as logistic regression has been widely applied in a variety of fields. Given the known values of other variables, logistic regression predicts the unknown variables of a discrete variable. The categorical response variable is limited to a small number of potential values. In binary logistic regression, the response variable is dichotomous, with only two possible outcomes represented by the values of 0 or 1. The dependent variable of the multiple logistic regression model comprises three discrete levels: low, medium, and high [4].

Logistic regression involves the transformation of odds, representing the likelihood of success relative to failure, through the utilization of the logit formula. This is represented by the logistic equation, where p(X) is the predicted probability, xj is the jth predictor variable, and βj is the coefficient estimate for the jth predictor variable. The coefficient estimates are usually obtained by maximum likelihood estimation [11].

$$log(\frac{p}{1-p}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

Equation 1: Logistic equation

2.4 Decision Tree

The decision tree algorithm is a commonly used algorithm in supervised learning for predicting outcomes and classifying data [10]. Using input variables, the data is divided into subsets based on categories, forming a tree-like structure in which each branch represents a selection between options and each leaf symbolizes a classification or decision [4][6]. The root node of the decision tree poses a specific query to the data and, depending on the response, sends it down a branch. These branches lead to internal nodes that query the data further before sending it, based on the response, to a different branch. Upon encountering an end node, otherwise known as a leaf node, that does not exhibit further branching, the aforementioned process endures [10].

One reason decision trees are popular in the field of machine learning is because they possess the ability to manage extensive datasets with relative simplicity [10]. They are helpful for initial variable selection because they are simple to comprehend and interpret, and they handle missing values well [4][6]. With their tree-like structure relating to decisions and their potential consequences, decision trees can be used for both classification and regression analysis. Decision trees are frequently used in decision analysis due to their ability to offer a clear and visual depiction of the decision-making process [6].

The advantages of decision trees include their adaptability and flexibility, among other benefits. The model has the capability to encompass additional possibilities for outcomes and can be integrated with other decision models when needed. They are limited in their capacity to adjust to changes in the data, though. A small change in the data can have a big impact on the decision tree’s structure. When managing with uncertain data, decision trees can be difficult to calculate and have lower prediction accuracy than other predictive models [6].

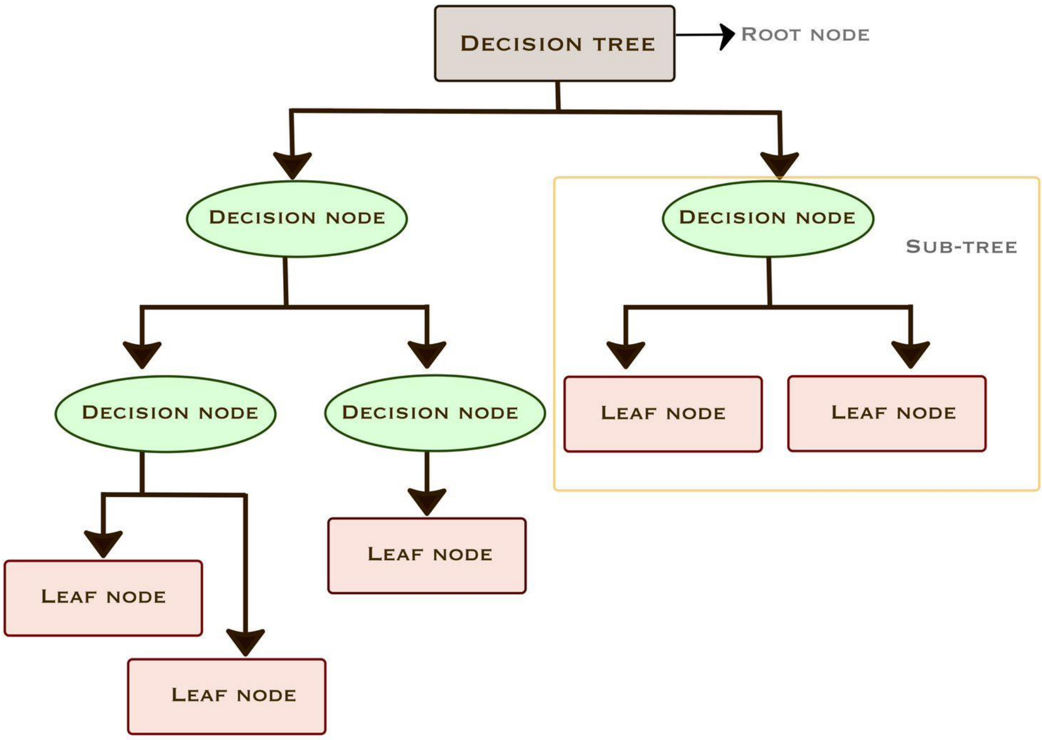


Figure 1: Decision tree

Figure 1 depicts a typical decision tree model, with internal nodes labeled with decision-related questions, branches labeled with potential answers to the question, and leaves labeled with the problem's solution [6].

2.5 Random Forest

The random forest algorithm is an influential technique utilized in classification and predictive modeling, which utilizes an ensemble of decision trees to enhance its performance and mitigate overfitting. The random forest algorithm employs a method termed "bagging", which involves training multiple decision trees on a random subset of the training data [10][14]. The number of decision trees utilized can range from hundreds to thousands. The identical data is inputted into each decision tree after their training, whereby the outcome having the highest frequency is determined as the most probable resolution for the dataset. This approach addresses the issue of "overfitting" that frequently plagues decision trees, where the algorithm becomes overly dependent on its training data set and suffers as a result when exposed to new data [10].

The random forest algorithm integrates bootstrapping with random feature selection to enhance the efficacy of the bagging technique. The phenomenon of tree correlation, commonly referred to as such, has been found to diminish following the implementation of a randomization mechanism in the tree construction process. Moreover, this randomization approach serves to decrease the degree of correlation between predictors and trees, as outlined in [14]. By reducing tree correlation, which is necessary for the technique to function effectively, lessens the reliance of the tree-building process on the original predictors.

2.6 Support Vector Machine

The Support Vector Machine (SVM) is a supervised machine learning algorithm that finds utility in predictive analytics [6]. Data analysis for classification and regression is done using an associative learning algorithm. This mode of operation is predominantly employed for categorization. It is a discriminative classifier that divides examples into categories according to a hyperplane. The objective of SVM is to identify the precise hyperplane that exhibits the maximum margin for the linear classification of distinct classes [15]. A clear gap separates the examples into categories in the representation of examples in a plane, and new examples are then predicted to belong to a class based on which side of the gap they fall [6]. The SVM algorithm draws its basis from the statistical learning theory. To establish equivalence between the character subset classification and the partitioning of the complete dataset, a designated set of characteristic subsets is meticulously selected from the training samples [15].

SVMs are a group of potent modeling techniques that are extremely flexible and are used for both linear and non-linear relations modeling. When compared to alternative classification algorithms, SVM demonstrates superior proficiency in addressing issues related to reduced sample size, nonlinearity, and high-dimensional data [15]. The process of performing regression analysis involves the usage of a particular technique called insensitive regression, which belongs to the category of support vector regression types [14]. Many applications, including intrusion detection, facial expression classification, time series prediction, speech recognition, image recognition, signal processing, gene detection, text classification, font recognition, fault diagnosis, chemical analysis, image recognition, and others, have successfully used SVMs to solve various classification problems. SVM has two significant advantages, including reducing the prediction time and ensuring high precision of the classifier with the optimal solution. While there may be some drawbacks, such as an enduring detection approach and the time and space requirements that increase proportionally with the quantity of data, as noted in reference [15]. SVM is a great method for handling classification issues because it continues to generalize even with little training knowledge.

2.7 K-Nearest Neighbors

The K-nearest neighbor (KNN) algorithm is an instance of supervised learning which is widely employed in both classification and predictive modeling applications. The k-closest training examples in feature space are used in this non-parametric technique. The KNN algorithm produces either the predicted continuous numerical value of an object for regression tasks or predicts the membership of a particular class for classification tasks [6]. The fundamental principle of the KNN algorithm is to classify an observation based on its proximity to other instances plotted on a graph, with the classification being dictated by the nearest cluster [10]. Different techniques, such as the Manhattan distance or the Euclidean distance, can be used to determine the separation between the output and other points. The number of neighbors that will be taken into account for classification or regression is determined by the parameter k in the KNN. It is essential to achieve a balance between overfitting and underfitting when choosing the k value for the KNN algorithm, as it greatly affects its performance [17].

2.8 Evaluation Metrics

The effectiveness of machine learning models or algorithms can be evaluated using evaluation metrics [21]. The choice of metric plays a critical role in evaluating the effectiveness and performance of the model in addressing the current problem [20]. To evaluate machine learning models in various applications, various evaluation metrics have been proposed, and each metric has a specific function [22]. Having a comprehensive knowledge of the available metrics and their appropriate applications is imperative when choosing the most suitable metric for the current problem at hand.

Evaluation metrics play a pivotal role in both the training and testing stages of a conventional data classification problem. The metric is used to test the produced classifier's performance on new data, while also being used to optimize the classification algorithm during the training stage [19]. In machine learning, the development set and test set are frequently used to develop and test the system, respectively. The machine learning algorithm can be tuned by analyzing the errors made by a development test set, a process known as parameter analysis and analysis. In situations where data is limited, K-fold cross-validation can prove to be useful [30].

It is important to note that using a single metric to assess machine learning models may not provide a complete picture of the issue being addressed [20]. As a result, the overall performance of the model can be tested using a combination of different evaluation metrics [21]. For various applications, various evaluation metrics, including accuracy, precision, recall, F1-score, and the area under the curve (AUC), have been proposed [20, 22].

2.8.1 Recall

The recall metric provides valuable insight into the algorithms' performance and is an important measure for Predictive Data Analytics. This measure of the model's ability to identify positive cases takes into account the overall number of positive cases within the dataset. Mathematically, recall is calculated as the ratio of true positive predictions to the sum of true positives and false negatives [23]. The recall may be expressed as the fraction of samples from a given class that the model correctly predicts [20]. The following formula shall be applied to recall:

$$Recall = \frac{TruePositives}{TruePositives + \ FalseNegatives}$$

Equation 2: Recall equation

Recall means figuring out how many of the target class examples our model correctly identified. This measures how often the model correctly predicts a positive result. When we say a high recall value, it means we are good at finding the correct answers. But it doesn't tell us how often we might be wrong. We should be careful when looking at recall values, especially if it could be terrible if we miss something important. We need to remember that precision and recall are opposite to each other. As recall increases, precision tends to decrease, and vice versa. The decision of which error is less costly for the overall goal depends on the problem being addressed [24].

2.8.2 Precision

For the evaluation of algorithms for Predictive Data Analytics, precision is a fundamental metric [23]. The answer to the question "When a model says that an observation belonged to one group, how often has it been?" gives insight into classification correctness. In mathematics, precision means the number of correct positive predictions divided by the total number of predicted positive classifications. A higher precision score means the model is better at recognizing the positive class. It is important to note that preciseness does not tell you how many times something has been missed when it should have been. It is calculated with this equation [24]:

$$Precision = \frac{TruePositives}{TruePositives + \ FalsePositives}$$

Equation 3: Precision equation

The specific problem at hand and the significance of minimizing false positives or false negatives determine whether precision or recall should be chosen as the evaluation metric. Precision is the preferred metric when reducing false positives is the goal. Precision helps reduce false positives in situations like criminal detection, where it would be worse to mistakenly arrest an innocent person than to let a criminal getaway. On the other hand, recall is used when reducing false negatives is the goal [23].

2.8.3 Accuracy

Accuracy is a widely used metric for evaluating the performance of predictive data analytics algorithms [24]. A formula is used to determine this value, which involves dividing the model's predictions by the sum of accurate positive and negative predictions [23]. In datasets with balanced proportions, measuring accuracy is valuable, but in datasets with imbalanced proportions, it can be misleading [24][23]. For instance, a model that classifies every case as non-fraudulent may exhibit a 99% accuracy rate, yet it may fail to effectively recognize instances of fraud in a dataset where the proportion of fraudulent to non-fraudulent cases is 1:99 [24]. It is advisable to consider additional measures such as precision, recall, and F1-score [23] while evaluating the efficiency of an algorithm on imbalanced data. Additionally, it is crucial to evaluate the algorithm's performance against an established system [30]. Accuracy equation calculates the proportion of observations that were correctly predicted to all of the observations.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + TrueNegatives + FalsePositives + FalseNegatives}$$

Equation 4: Accuracy equation

In situations where classification accuracy alone is not a reliable indicator, precision becomes crucial, especially when dealing with imbalanced class distributions. Accuracy can be misleading, in such cases where one classification is more frequent than the next, since a high accuracy rate with no relevant training could occur if all samples are predicted as the most common class. Precision gives you a specific performance metric for each class to allow a more detailed analysis of how the model is capable of anticipating certain classes. The performance of the model to classify each class accurately shall be shown when comparing precision values for different classes [20]. Figure 2 shows us insight into why both metrics accuracy and precision are important.

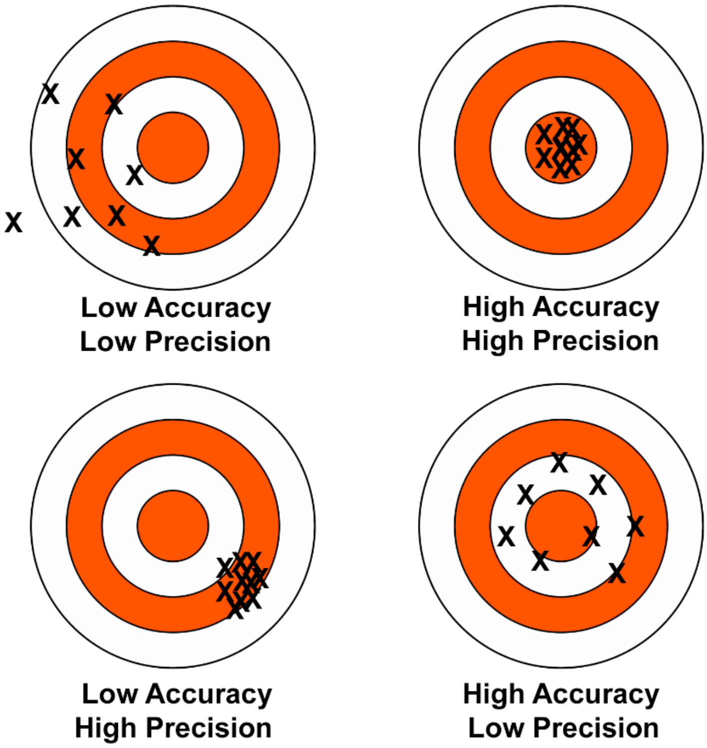


Figure 2: Acurracy and Precision

2.8.4 F1 Score

The F1 Score, which is also known as the F-measure, is a commonly used metric for evaluating the efficacy of predictive data analysis algorithms. This metric is the harmonic mean of the two metrics of precision and recall. F1 score can have a maximum value of 1, which denotes perfect precision and recall, while a value of 0 indicates a complete lack of precision or recall. As a result, F1 Score is helpful where the model's application demands a balance between precision and recall [24].

The F1 Score equation can be expressed as a weighted average of precision and recall, with the relative weight of precision and recall being determined by the weight function. When is equal to 1, the F1 Score is a special case of the generalized F-measure [30]. The subsequent equation illustrates the F1 Score:

$$F1 = 2 * \frac{precision * recall}{precision + recall}$$

Equation 5: F1 Score equation

It's crucial to remember that there is always a compromise between recall and precision. For instance, if our goal is to improve precision, the recall rate might go down and the opposite also holds. When evaluating the efficacy of predictive models, it is imperative to consider the relative importance of these two metrics within the given application domain. The F1 Score has received significant criticism for not considering the varying economic consequences of different types of misclassifications, as it gives the same importance to both precision and recall [24]. Hence, it is recommended to consider evaluation metrics that are specific to the domain while evaluating the efficacy of predictive models.

2.8.5 Cohen's Kappa

Cohen's kappa is a widely used metric to assess agreement between raters or evaluate the performance of a classification model [25]. It considers the agreement between two individuals or entities in categorizing items and can be applied to various scenarios, such as comparing credit ratings assigned by bankers [26]. Cohen's kappa is different from overall accuracy because it considers the balance of different groups, so it can be harder to understand. To separate things into two groups, you can find the result by comparing how many are in each group, and counting how many were predicted correctly. Cohen's kappa uses certain values to tell us if predictions are better than just guessing, and tries to fix any unfairness in the evaluation.

Cohen’s kappa coefficient (κ) is considered a robust measure for assessing inter-rater reliability in qualitative items, surpassing simple percent agreement calculation [25]. Cohen's Kappa equation deals with the possibility of agreement, which may be a result of chance, and accounts for disagreement on the issues in question [26]. It is composed of the following formula:

$$\kappa = \frac{p_o - p_e}{1 - p_e}$$

Equation 6: Cohen's Kappa equation

Where p_e represents the expected proportion of agreement between the raters if they were to assign ratings arbitrarily and p_o represents the observed proportion of agreement between the raters. Cohen’s Kappa, which provides a more reliable measure of inter-rater reliability than simple percent agreement, accounts for the possibility of agreement occurring by chance. Perfect agreement is represented by a value of 1, while no agreement beyond what would be predicted by chance is represented by a value of 0 [26].

2.8.6 Hamming Loss

The Hamming Loss metric is used to assess the predictive algorithms’ performance as regards multilabel classification. It would measure less than a fraction of the wrong signs that were predicted using algorithms, compared with the total number of labels. The concept of Hamming Loss was started in information theory and represented the expected distance between actual indications and those described under a multiclass classification scenario. In a multilabel classification the Hamming Loss, which may give insight to the accuracy of algorithms for assigning labels, penalizes only individual labels and not all cases [27].

$$HammingLoss = \frac{1}{N}\sum_{i=1}^{N}\frac{xor(y_i, \widehat{y_i})}{\mid L \mid}$$

Equation 7: Hamming Loss equation

In Equation 7, *L* is the total number of labels, *N* is the number of samples, y_i is the true label set for sample *i*, $\widehat{y_i}$ is the predicted label set for sample *i*, and $xor(y_i, \widehat{y_i})$ is the number of labels that differ between y_i and $\widehat{y_i}$ [28].

The differentiability of the Hamming loss is an important consideration when using it as an actual loss function in machine learning models. The Hamming loss, which counts and normalizes the number of labels for which the predictions are incorrect, is not differentiable, it is important to note4. Certain optimization algorithms that rely on gradients for model training may be affected by this lack of differentiability [27].

2.8.7 Jaccard Index

The measure of a comparison between two sets of data is called the Jaccard similarity index, and it was created by Paul Jaccard. The result ranges from 0 to 1, with a value nearer 1 denoting a greater similarity between the sets. When both datasets have exactly the same members, the Jaccard Similarity Index is 1, while a similarity index of 0 indicates that no common members exist [29]. The Jaccard Index, also called the Jaccard similarity coefficient, measures how similar two sample sets are to one another. It is calculated by dividing the size of the intersection by the sum of the sample sets' sizes [31].

$$J(A, B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid}$$

Equation 8: Jaccard Index equation

In Equation 8, A and B are the two sets being compared, and $|A|$ stands for the size of set A. The Jaccard distance, on the other hand, is determined by subtracting the result value from 1 [31]. And it helps to see how different the two sets are.

In the field of machine learning, the Jaccard Index finds applications in Convolutional Neural Networks (CNNs) for tasks such as image recognition and object detection. When it comes to object detection, the Jaccard Index, for instance, is useful in quantifying the similarities between the objects the computer recognizes and those in the training data [31].

2.9 Comparison of Previous Studies

In previous studies, various algorithms have been compared for predictive data analytics using different datasets. K-Nearest Neighbors, Support Vector Machine, Decision Tree, and Random Forest classifiers were all used in one study [32] to analyze various PIMA datasets. The Random Forest had 100% precision for dataset D3, while the Decision Tree achieved a minimum precision of 64% for dataset D4. Random Forest had the highest recall for datasets D1 and D2, while Decision Trees excelled for D3 and D4. SVM performed best for D4 in terms of accuracy, while Random Forest performed best for D1, D2, and D3. Dataset D3 had the highest F1 score of 75.68%. Based on these findings, D3 was selected as the top dataset. K-Nearest Neighbors improved with more neighbors, and the Decision Tree reached 86.1% accuracy with specific parameters. For D3, Random Forest had an accuracy and precision of 88.61% and 100% respectively. D3 consistently outperformed other datasets in accuracy, precision, sensitivity, and F1 score. Removing rows with missing values improved its performance [32].

Another study [33] compared the performance of the different predictive algorithms KNN, SVM, and Nave Bayes using data from credit cards, employee retention, and housing. SVM performed better than other algorithms in terms of accuracy, but it took much longer to train than different algorithms. Regarding training time, Naive Bayes performed well and had precision comparable to K-Nearest Neighbors.

In alternate research [34], a comparison was made among three distinct algorithms, namely logistic regression, random forest, and K-Nearest Neighbors, concerning their precision, accuracy, F1-score, and support. Upon conducting precision analysis, it was found that logistic regression exhibited a precision of 94% in the business section, random forest yielded a precision of 90%, and K-Nearest Neighbors demonstrated a precision of 96%. Other sections, such as entertainment and politics, were also subjected to similar comparisons.

3. Research Design

3.1 Data Collection

Conducting a thorough analysis and comparison of predictive data analytics algorithms necessitates the acquisition of pertinent and high-quality data. The process of data collection entails a methodical gathering of information or measurements that facilitates researchers in addressing research inquiries, scrutinizing hypotheses, and assessing results. Irrespective of the discipline or nature of data, be it quantitative or qualitative, meticulous data collection is a prerequisite for preserving the veracity of academic inquiry [35].

The data collection process commences with the identification of the requisite data and the subsequent selection of an appropriate sample from a particular population. The two broad categories of data are qualitative and quantitative. Most qualitative data is non-numerical and descriptive or nominal in nature, capturing the subject's feelings or perceptions. Focus groups, interviews, and group discussions are examples of qualitative techniques that are useful for examining the "how" and "why" of a program or phenomenon. Quantitative data, on the other hand, is numerical and can be calculated mathematically. It uses a variety of scales, including nominal, ordinal, interval, and ratio scales, as well as measurements. Quantitative approaches use standardized techniques like surveys and experiments to address the "what" of the program [35].

The gathering of the datasets required for assessing and comparing the performance of various classification algorithms is a crucial step in this research study. In the following section, we delineate the methodology for gathering data, inclusive of the data origins and any pre-processing measures undertaken.

The datasets utilized in this research were procured from the sklearn library, a repository that offers a diverse range of data sets typically employed in the realm of machine learning investigation. These datasets can function as appropriate standards for assessing and contrasting the efficacy of diverse classification algorithms.

- The Iris dataset is a well-known dataset that is widely used for classification tasks. It includes measurements of different characteristics for various species of iris flowers. Sepal length, sepal width, petal length, and petal width make up the dataset's four attributes. The iris flower species, including Setosa, Versicolor, and Virginica, are represented by the target variable. In machine learning research, the Iris dataset is frequently used as a benchmark dataset.
- Digits is a collection of grayscale images of hand-drawn digits. Each sample in the dataset represents an 8x8 image, and the target variable represents a digit ranging from 0 to 9. Digit recognition tasks are frequently performed using the Digits dataset.
- The Breast Cancer dataset contains medical features derived from breast mass samples. The dataset includes attributes like mean radius, mean texture, and mean compactness. The presence or absence of breast cancer is represented by the target variable. The Breast Cancer dataset is widely used in breast cancer research to evaluate classification algorithms.
- The Wine dataset contains wine samples from various regions which have been sampled for the purposes of Chemical Analysis. It contains measurements of 13 attributes such as alcohol content, malic acid concentration, and ash content. The wine's origins are represented by the target variable. In the areas of classification, a wine dataset is commonly used.
- The Olivetti Faces dataset contains grayscale images of people's faces. Each attribute corresponds to a pixel value in a grayscale image. This dataset is frequently used for classification and face recognition tasks. The target variable, however, is not relevant in this situation because the Olivetti Faces dataset is primarily used for unsupervised learning tasks. Research on faces frequently uses the Olivetti Faces dataset.

- The 20 NewsGroups Vectorized dataset is made up of newsgroup posts that have been vectorized into numerical features. It is frequently employed in text classification tasks. The target variable represents the category or topic of the post, and the dataset includes text data from newsgroup posts. The original 20 Newsgroups dataset, which includes information on a variety of subjects such as politics, sports, technology, and more, is provided as a vectorized representation.
- The Covertype dataset has cartographic variables that can be used to determine the type of forest cover based on a variety of geographical features. It is composed of characteristics, such as elevations, slopes and soil type. A forest cover type including seven classes is represented in the target variable. In ecological and environmental research, the Covertype dataset is frequently used to perform multiclassification tasks.

In your research study, you have selected the Iris, Digits, Breast Cancer, Wine, Olivetti Faces, 20 NewsGroups Vectorized, and Covertype datasets for evaluating the performance of the classification algorithms. However, it's worth noting that other datasets are also available in the sklearn library and could be considered for future research or extensions of this study.

The datasets are then loaded into your Python tool using the proper sklearn.datasets module functions after the data collection process. Preprocessing procedures, including how they handle missing values or scaling features, ought to be recorded if used. Upon loading the datasets, the features are subsequently attributed to the variable denoted as X, whilst the target variable is assigned to the variable symbolized as y. In order to ensure impartial assessment, the dataset is subsequently partitioned into separate training and testing subsets. For this, the sklearn.model_selection module's train_test_split() function is utilized. The data is divided by the specific line of code train_test_split(X, y, test_size=0.25, random_state=23), where:

- X: Depicts the dataset's input features.
- Y: Depicts the relevant target variable.

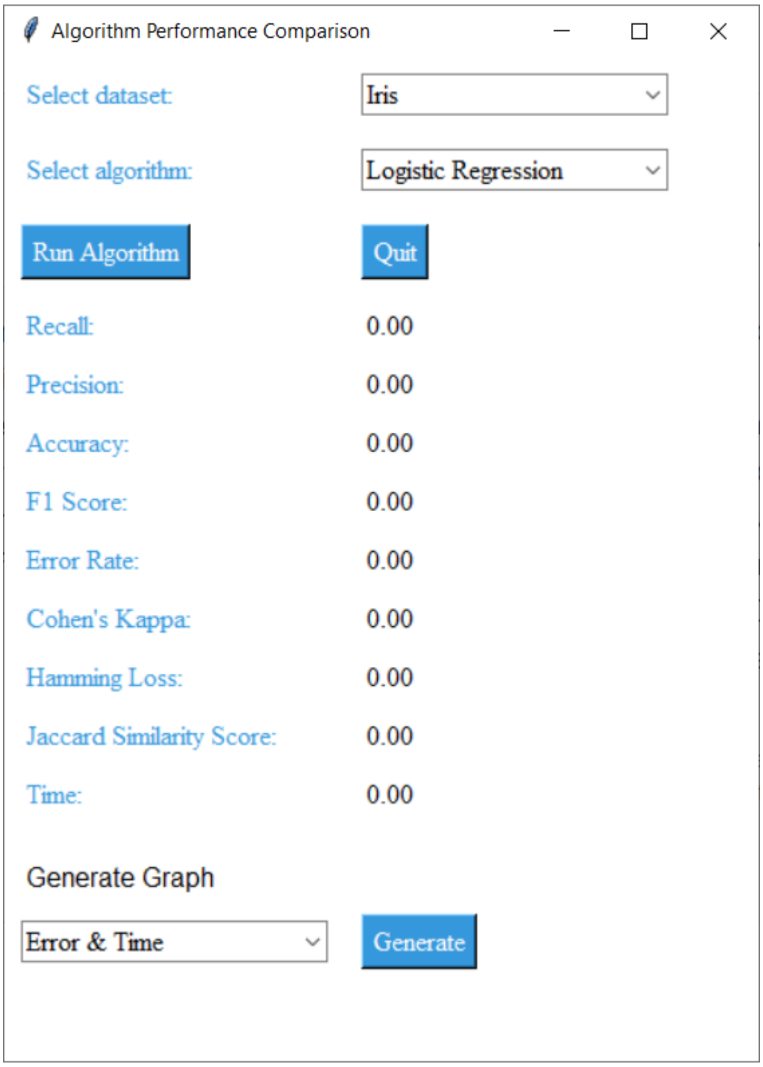
- test_size: Indicates how much of the dataset should be reserved for testing. It is set to 0.25 in this instance, meaning that 25% of the data will be used for testing.
- random_state: Establishes the random seed for consistency. In this instance, 23 is chosen as the value.

Through the segregation of the dataset into distinct training and testing sets, it becomes possible to evaluate the efficacy of classification algorithms on unobserved data.

This study involves a rigorous data collection process that aims to facilitate a robust comparison of the effectiveness of various algorithms, including Logistic Regression, Decision Tree, Random Forest, SVM, and KNN, with respect to their performance over the Diabetes and Iris datasets. The following sections aim to elucidate the research methodology used to assess the algorithms and provide answers to the research questions.

3.2 Development of the Comparison Tool

This section aims to elucidate the tool design and functionality employed to compare algorithms in the domain of predictive data analytics. The stated tool has been developed utilizing the programming language Python and incorporates an assortment of traits to assess and contrast diverse classification algorithms over pre-selected datasets. A detailed overview of the design and functionality of the aforementioned tool is provided below.



Algorithm Performance Comparison

Select dataset: Iris

Select algorithm: Logistic Regression

Recall: 0.00

Precision: 0.00

Accuracy: 0.00

F1 Score: 0.00

Error Rate: 0.00

Cohen's Kappa: 0.00

Hamming Loss: 0.00

Jaccard Similarity Score: 0.00

Time: 0.00

Generate Graph

Error & Time

Figure 3: Tool layout

3.2.1 Tool Design

The software tool uses the Tkinter library to develop a graphical user interface (GUI). The design of the tool adheres to a layout based on a grid system. The interface comprises diverse elements, involving dropdown menus to select the dataset and algorithm. A viable option for the acquisition of a dataset is to select one among the collection available in the sklearn library, such as the Diabetes or the Iris datasets. In like manner, one has the ability to choose from an array of classification algorithms, such as Logistic Regression, Decision Tree, Random Forest, SVM, or KNN.

The “Run Algorithm” button on the tool starts the execution of the chosen algorithm on the selected dataset. A “Quit” button is also present for shutting down the tool. After the algorithm has been run, the tool shows a variety of performance metrics, including Recall, Precision, Accuracy, F1 Score, Cohen’s Kappa, Hamming Loss, Jaccard Index, and training time. We can evaluate the algorithms’ efficacy and accuracy using these metrics.

The tool has a “Generate Graph” feature that makes it easier to compare algorithms. By selecting a specific metric from the drop-down menu and clicking the “Generate Graph” button, the matplotlib library creates a graph for us. The graph provides a visual representation of the performance of the algorithms for the chosen metric, which helps with the process of analysis and comparison of the results.

3.2.2 Tool Functionality

The present tool is characterized by a set of essential functionalities:

- Dataset Selection: We can choose between seven datasets that are easily accessible in the sklearn library, namely Diabetes, and Iris. A dropdown menu is used to make the selection, offering a simple way to change between datasets.
- Algorithm Selection: The tool includes a dropdown menu that enables us to choose one algorithm from a list of options, enabling effective comparison. Several algorithms, namely Logistic Regression, Decision Tree, Random Forest, SVM, and KNN can be compared. To compare the performance of different algorithms on the chosen dataset, we can switch between them with ease.
- Metrics Evaluation: Once the algorithm and dataset have been selected, we can commence the assessment process by activating a “Run Algorithm” button for metrics evaluation. The tool then runs the selected algorithm and utilizes the Sklearn library to calculate various performance metrics. The metrics include Recall, Precision, Accuracy, F1 Score, Cohen’s Kappa, Hamming Loss

and Jaccard Index. A wide range of metrics is used to obtain a comprehensive understanding of the predictive capabilities of the algorithm.

- Display of Results: The tool makes sure that the calculated metric scores are presented in a way that is both clear and informative. Since the results are presented as labeled components, we can quickly understand and evaluate how well various algorithms perform. The results' precise labeling makes it simple to comprehend the evaluation's findings.
- Result Storage: To save the metric scores for later analysis and visualization, the comparison tool uses an object-oriented approach. The outcomes are saved in an object along with the dataset name, algorithm name, metric name, and corresponding score. With this method, evaluation data can be handled and retrieved quickly for the creation of graphs and more thorough comparisons.
- Graph Generation: The tool includes a feature to produce visual graphs, which improves the comparison process. These graphs give a clear, visual representation of how the algorithm performed for the chosen metric. The graphs are made using the matplotlib library, which makes it easier to see how the results compare.

The combination of these design decisions and functionalities guarantees that we are able to effortlessly investigate and contrast different algorithms for predictive data analytics. The comparison tool's usability and effectiveness are enhanced by a combination of an interactive user interface, flexible algorithm selection, comprehensive metric evaluation, informative display of results, and visual generation of graphs.

3.3 Algorithm Selection and Configuration

In this section, we will be discussing both the algorithm selection process and the detailed configurations of the tool. The objective of the tool is to assess the accuracy, efficiency, and scalability of classification algorithms in diverse problem areas. By utilizing the tool, we can select a dataset and an algorithm from the drop-down menus and evaluate their performance using a variety of performance metrics.

The algorithms taken into account for evaluation in this study are as follows:

- Logistic Regression (configured with max_iter=100000000): For binary classification tasks, Logistic Regression is a popular linear classification algorithm. To ensure convergence, the maximum number of iterations for the Logistic Regression algorithm in this study is set to 100,000,000.
- Decision Tree: For classification tasks, Decision Trees are non-parametric machine learning algorithms that produce a model that resembles a tree. In this study, the Decision Tree algorithm is used without any special configuration.
- Random Forest (configured with n_estimators=100): This ensemble learning technique combines several Decision Trees to increase accuracy and decrease overfitting. The Random Forest algorithm is set up with 100 decision trees (n_estimators=100) to create the ensemble in this study.
- Support Vector Classification (SVC): The SVC is a supervised learning algorithm that is utilized for binary classification tasks. It finds the best hyperplane to separate the classes in the feature space. In this study, the SVC algorithm is used without any particular configuration.
- K-Nearest Neighbors (configured with n_neighbors=5): This straightforward but efficient algorithm categorizes new instances according to the consensus of their k nearest neighbors. In this study, the K-Nearest Neighbors algorithm is set up with k set to 5 (n_neighbors=5).

3.4 Metrics and Evaluation Criteria

We outline the metrics and evaluation standards used to compare the precision, efficacy, and scalability of the classification algorithms in our tool in this section.

Making defensible choices about the algorithms’ suitability for various issue domains is made easier with the help of these metrics, which offer insightful information about their performance.

3.4.1 Performance Metrics

We utilize a range of performance metrics to objectively measure the predictive proficiency of the classification algorithms. The following metrics were chosen on the basis of their relevance to the objectives of our study. Every metric includes two parameters: "y_test," which represents the true values of the target variable obtained from the test set, and "y_pred," which represents the predicted values of the target variable.

1. Recall Score: The recall score, also known as the true positive rate, assesses the algorithms' ability to correctly identify positive instances. Our tool's recall score configuration is calculated as follows:

   Code snippet 1: Recall Score code

   - Parameters:
     - average='macro': This parameter specifies that the recall score should be computed as the unweighted mean across all classes.
     - zero_division=1: This parameter specifies the value to use for recall when there are no predicted positive instances.

2. Precision Score: The precision score, also known as the positive predictive value, assesses the algorithms' ability to correctly identify positive instances among predicted positive instances. Our tool's precision score configuration is calculated as follows:

   Code snippet 2: Precision Score code

   - Parameters:

        - average='macro': This parameter specifies that the precision score should be computed as the unweighted mean across all classes.
        - zero_division=1: This parameter specifies the value to use for precision when there are no predicted positive instances.

3. Accuracy: Accuracy is the percentage of instances that are correctly classified out of all instances. It is a typical metric used to assess how well algorithms perform in classification tasks. The following formula is used to calculate the accuracy configuration in our tool:

Code snippet 3: Accuracy Score code

4. F1 Score: The F1 score combines precision and recall to offer a fair evaluation of how well algorithms perform in the presence of unbalanced datasets. Both false positives and false negatives are taken into account. The following formula is used to calculate the F1 score configuration in our tool:

Code snippet 4: F1 Score code

    - Parameters:
        - average=‘macro’: This parameter specifies that the F1 score should be computed as the unweighted mean across all classes.

5. Cohen's Kappa Score: Cohen's kappa score assesses the agreement between predicted and true labels while accounting for the possibility of coincidental agreement. It is especially useful when dealing with skewed datasets. Our tool's Cohen's kappa score configuration is calculated as follows:

Code snippet 5: Cohen's Kappa code

6. Hamming Loss: The hamming loss is calculated by averaging the fraction of incorrectly predicted labels across all instances and labels. It is especially useful for multi-label classification problems. Our tool's hamming loss configuration is calculated as follows:

Code snippet 6: Hamming Loss code

7. Jaccard Score: The Jaccard score, also known as the Jaccard index or intersection over union, calculates the degree of similarity between predicted and true labels. It is especially useful for assessing algorithm performance in multi-label classification tasks. Our tool's Jaccard score configuration is calculated as follows:

Code snippet 7: Jaccard Score code

- Parameters:
    - average='macro': This parameter specifies that the Jaccard score should be computed as the unweighted mean across all classes.

3.4.2 Time as a Performance Indicator

We also think of the algorithm execution time as a crucial evaluation criterion in addition to the performance metrics already mentioned. In real-world applications, where efficiency and scalability are crucial considerations, time is a crucial factor. To evaluate each algorithm’s computational efficiency, the time it took for it to complete the classification task was recorded and compared.

We aim to provide a thorough evaluation of the algorithms’ accuracy, efficacy, and scalability in various issue domains by taking into account these performance metrics and the execution time. The combination of these metrics enables a comprehensive assessment, enabling our tool’s users to decide on the best course of action based on their unique needs.

3.5 Libraries Used

The tool’s implementation was created in the Python programming language with the aid of several libraries and frameworks. The major libraries used in the creation of the tool will be covered in this section, along with each one’s function.

1. scikit-learn (sklearn): Python’s Scikit-learn is a well-known machine learning library. For the preprocessing of data, the choice of models, and the evaluation process, it offers effective implementations of numerous algorithms and tools. Accessing datasets, dividing data into training and testing sets, and running various classification algorithms are all made possible by this tool’s use of sklearn.
2. tkinter: Tkinter is a Python library that is typically used to build graphical user interfaces (GUI). It makes it possible to create interactive components like dropdown menus, buttons, and text displays. The tool’s user interface is made easier by tkinter, which also displays the calculated performance metrics and generated graphs while letting users choose datasets, algorithms, and metrics.
3. matplotlib: Matplotlib is a popular Python plotting library. It offers a flexible set of functions for producing various graph and visualization types. This tool uses matplotlib to produce graphs that compare the effectiveness of various algorithms based on the chosen metric. The produced graphs assist in visualizing algorithmic variations and give us information about the algorithms’ propensity for prediction.
4. time: The time module, which is a commonplace library in the Python language, presents a set of functions that enable the quantification of temporal operations. The time library is employed within the tool to capture and document the runtime of the selected algorithms. This information holds immense significance in assessing the scalability and efficacy of the algorithms for practical applications.

The efficient data handling, algorithmic execution, result visualization, and performance assessment made possible by these libraries are crucial to the tool’s implementation.

4. Measurements and analyzing results

Throughout our study, we examined numerous metrics to assess the performance of algorithms and determine their effectiveness. The evaluation involves Logistic Regression, Decision Tree, Random Forest, SVM, and KNN. Our analysis included a thorough evaluation of algorithm performance, which was measured using several metrics. These include recall, precision, accuracy, F1 Score, Cohen’s Kappa, Hamming loss, Jaccard index, and training time.

4.1 Comparison of Algorithm Performance on Dataset 1

The performance of various machine learning algorithms for predictive data analytics on Dataset Iris is examined in this section. The goal of this analysis is to compare the classification algorithms, namely Logistic Regression, Decision Tree, Random Forest, SVM, and KNN, in terms of accuracy, effectiveness, and scalability.

4.1.1 Error Metrics and Training Time

Figure 4 compares the selected algorithms on Dataset Iris in terms of recall, precision, F1 score, and execution time. According to the results, Logistic Regression and KNN achieved perfect performances for the metrics, with an execution time of 0.041 seconds and 0.002 seconds. Decision Tree achieved a recall of 0.970, precision of 0.972, and F1 score of 0.970, with an execution time of 0.0001 seconds. Random Forest and SVM also achieved a recall, precision, and F1 score with the same result as Decision Tree, respectively, with execution times of 0.264 and 0.003 seconds. These findings indicate that Logistic Regression and KNN achieved perfect scores for recall, precision, and F1 score, demonstrating their exceptional performance. Decision Tree, Random Forest, and SVM also exhibited high performance with similar scores for these metrics, with Decision Tree showing the fastest execution time and Random Forest having the longest execution time.

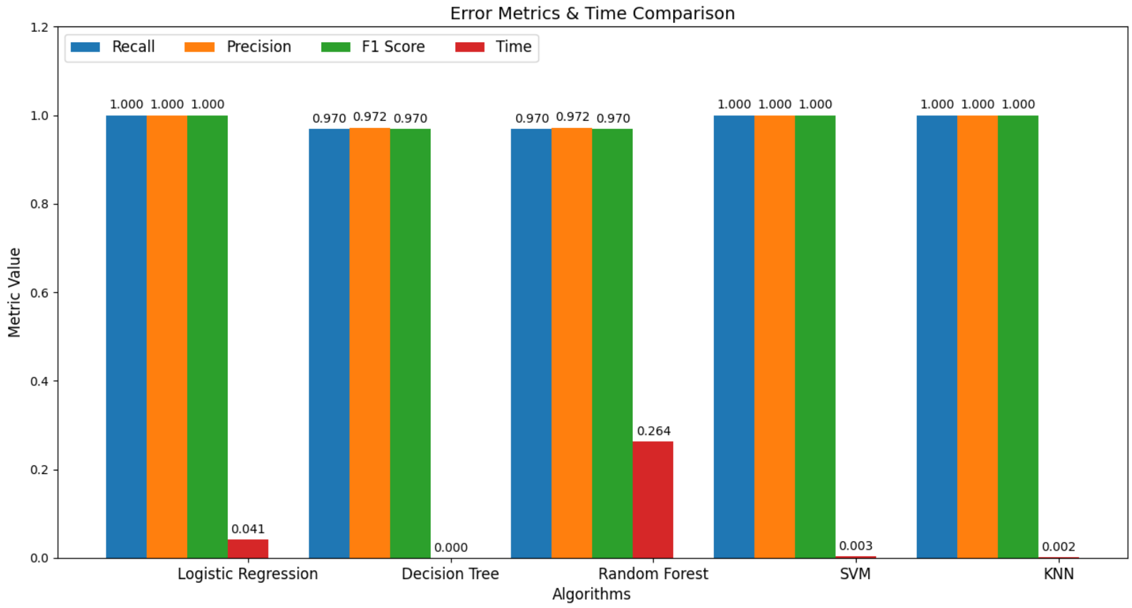


Figure 4: Error Metrics and Time for 'Iris Dataset

4.1.2 Correctness and Agreement Metrics

Figure 5 compares the selected algorithms on Dataset Iris in terms of accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score. According to the results, all algorithms achieved high accuracy values of 1.0, indicating accurate predictions. The Cohen's Kappa values ranged from 0.960 to 1.0, suggesting substantial agreement between predicted and actual classes. The Hamming Loss values ranged from 0.0 to 0.026, indicating low error rates in the classification. Additionally, the Jaccard Score values ranged from 0.942 to 1.0, signifying strong similarity between predicted and actual class sets. These findings demonstrate that all the selected algorithms perform exceptionally well in terms of accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score on Dataset Iris.

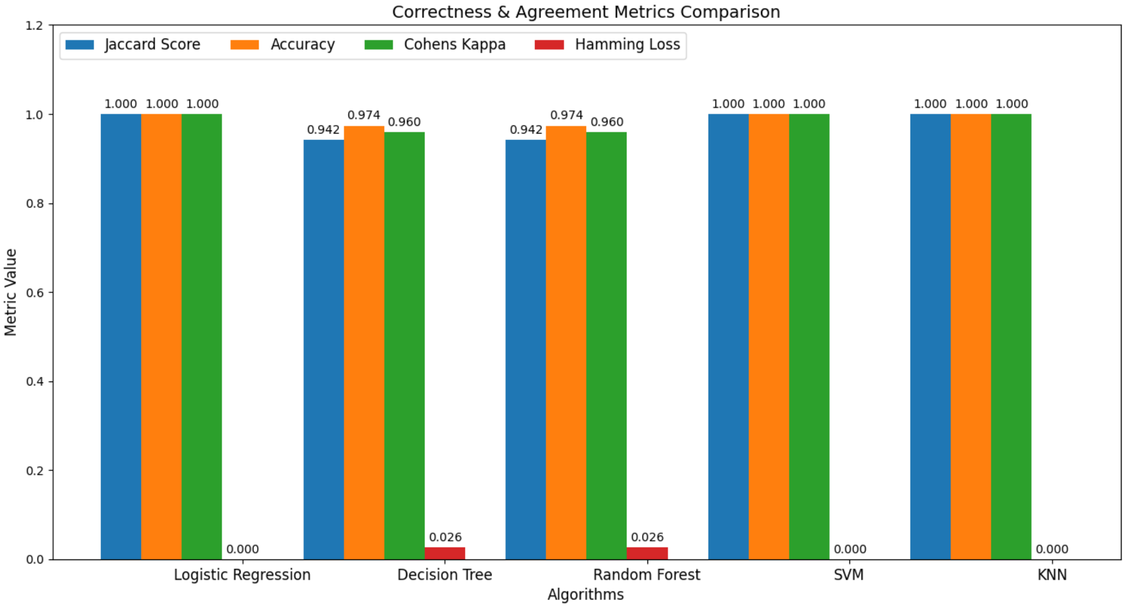


Figure 5: Correctness and Agreement Metrics for 'Iris' Dataset

4.2 Comparison of Algorithm Performance on Dataset 2

We analyzed the performance of classification algorithms on the widely acknowledged Digit dataset.

4.2.1 Error Metrics and Training Time

Figure 6 presents the comparison of the selected algorithms in terms of Recall, Precision, F1 Score, and Time on Dataset Digits. The results indicate that Logistic Regression achieved the highest Recall, Precision, and F1 Score among the algorithms. It demonstrated a Recall of 0.962, Precision of 0.962, and F1 Score of 0.962, with a time of 4.542 seconds. Decision Tree obtained relatively lower values, with a Recall of 0.867, Precision of 0.866, and F1 Score of 0.866, but exhibited a significantly lower execution time of 0.031 seconds. Random Forest, SVM, and KNN achieved Recall, Precision, and F1 Score values ranging from 0.983 to 0.991, indicating their effectiveness in correctly classifying instances. However, they exhibited varying execution times, with Logistic Regression having the longest time of 4.542 seconds.

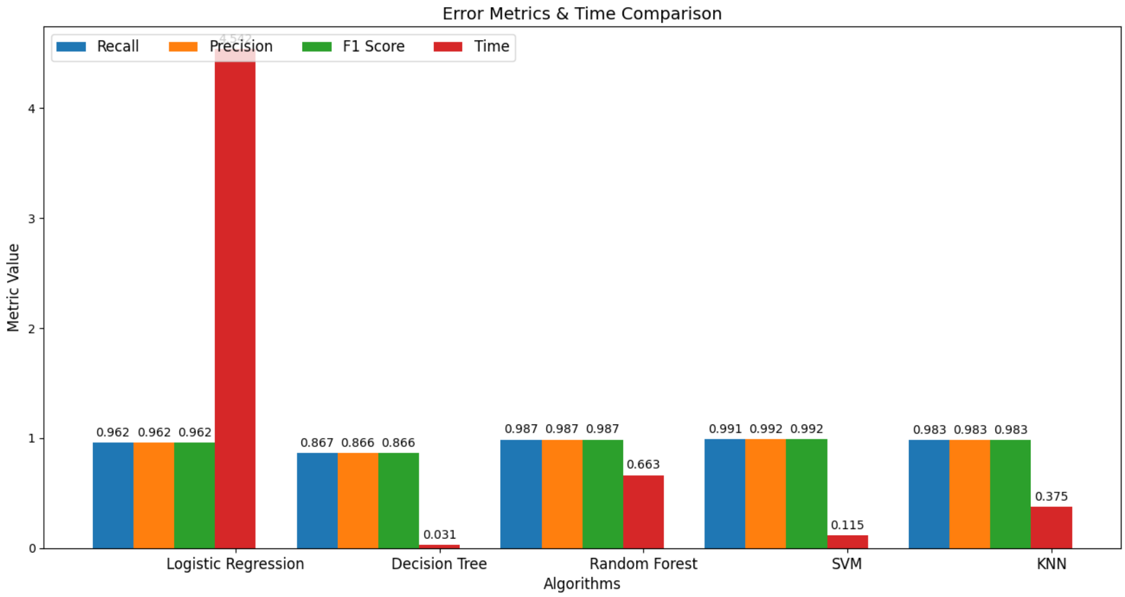


Figure 6: Error Metrics and Time for 'Digit Dataset

4.2.2 Correctness and Agreement Metrics

Figure 7 depicts the performance metrics Accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score for the various machine learning algorithms on Dataset Digits. The accuracy of Logistic Regression is 0.96 and Cohen's Kappa is 0.955, indicating a high overall correctness rate and agreement between predicted and actual labels. In terms of Accuracy (0.987), Cohen's Kappa (0.985), and Jaccard Score (0.975), Random Forest outperforms the other algorithms, demonstrating its effectiveness in making accurate predictions and capturing agreement. When compared to the other algorithms, Decision Tree performs relatively poorly in terms of Accuracy (0.864) and Cohen's Kappa (0.849), implying a lower overall correctness rate and agreement. SVM and KNN also achieve high Accuracy, Cohen's Kappa, and Jaccard Score values, indicating their ability to predict and agree with true labels. However, Decision Tree showed performed the worst regarding Hamming Loss metric.

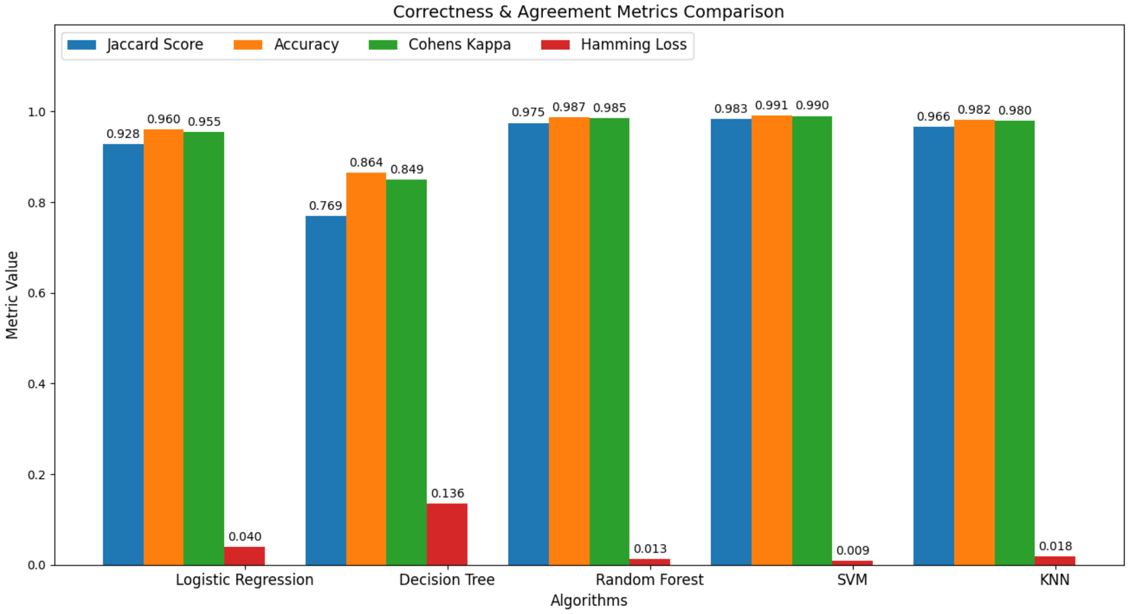


Figure 7: Correctness and Agreement Metrics for 'Digit' Dataset

4.3 Comparison of Algorithm Performance on Dataset 3

In addition, we undertook an analysis of Dataset 3, namely the Wine dataset, by implementing classification algorithms and measuring their efficacy through a range of evaluative metrics. The present research sought to offer a comprehensive comprehension of the algorithms' capabilities and their pertinence to the Wine dataset.

4.3.1 Error Metrics and Training Time

Figure 8 presents the comparison of the selected algorithms in terms of recall, precision, F1 score, and time on the Wine dataset. The results indicate that Logistic Regression achieved perfect recall, precision, F1 score, and had a time of 0.650 seconds, indicating excellent performance. Decision Tree obtained high recall (0.967), precision (0.984), F1 score (0.974), and a time of 0.001 seconds, demonstrating competitive performance. Random Forest also achieved perfect recall, precision, F1 score, and had a time of 0.255 seconds, similar to Logistic Regression. SVM and KNN exhibited relatively lower recall, precision, and F1 score values, indicating comparatively weaker performance.

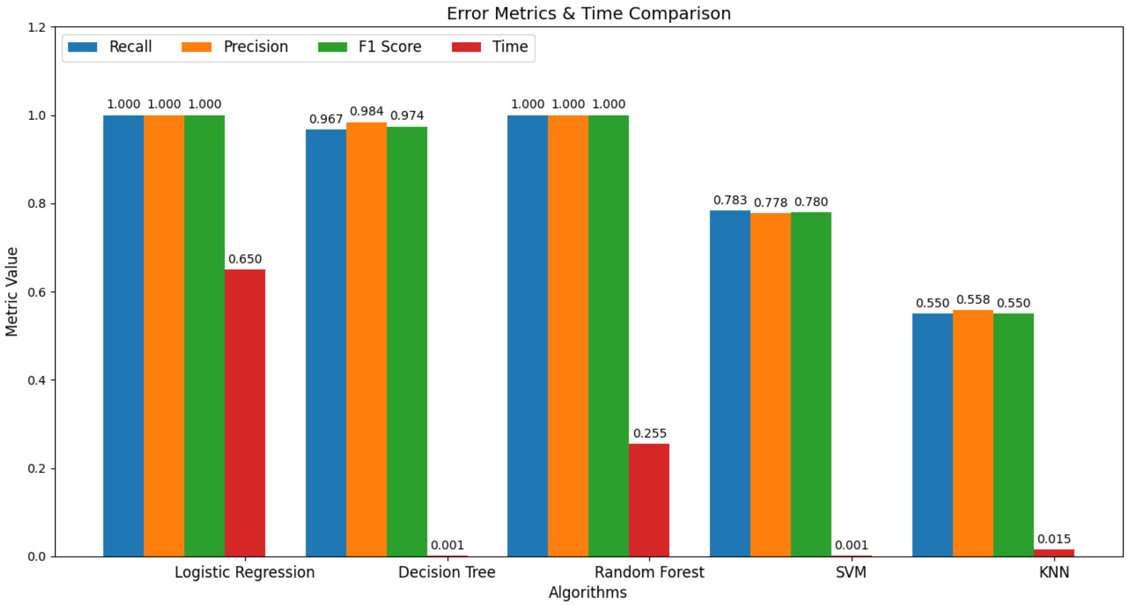


Figure 8: Error Metrics and Time for 'Wine' Dataset

4.3.2 Correctness and Agreement Metrics

Figure 9 presents the comparison of the selected algorithms in terms of accuracy, Cohen's kappa, hamming loss, and Jaccard score on the Wine dataset. The results indicate that Logistic Regression achieved perfect accuracy, Cohen's kappa, and Jaccard score, with a hamming loss of 0.0. Decision Tree obtained high accuracy (0.978), Cohen's kappa (0.965), Jaccard score (0.951), and a hamming loss of 0.022. Random Forest also achieved perfect accuracy, Cohen's kappa, and Jaccard score, with a hamming loss of 0. SVM and KNN exhibited relatively lower accuracy, Cohen's kappa, Jaccard score values, and higher hamming loss values, indicating comparatively weaker performance. The findings from Figure 9 on the Wine dataset show that Logistic Regression and Random Forest achieved perfect accuracy, Cohen's kappa, and Jaccard score, indicating accurate predictions and high agreement with the true class labels. In contrast, SVM and KNN exhibited lower accuracy, agreement, and similarity scores, along with higher hamming loss values, suggesting weaker performance and a higher number of misclassifications.

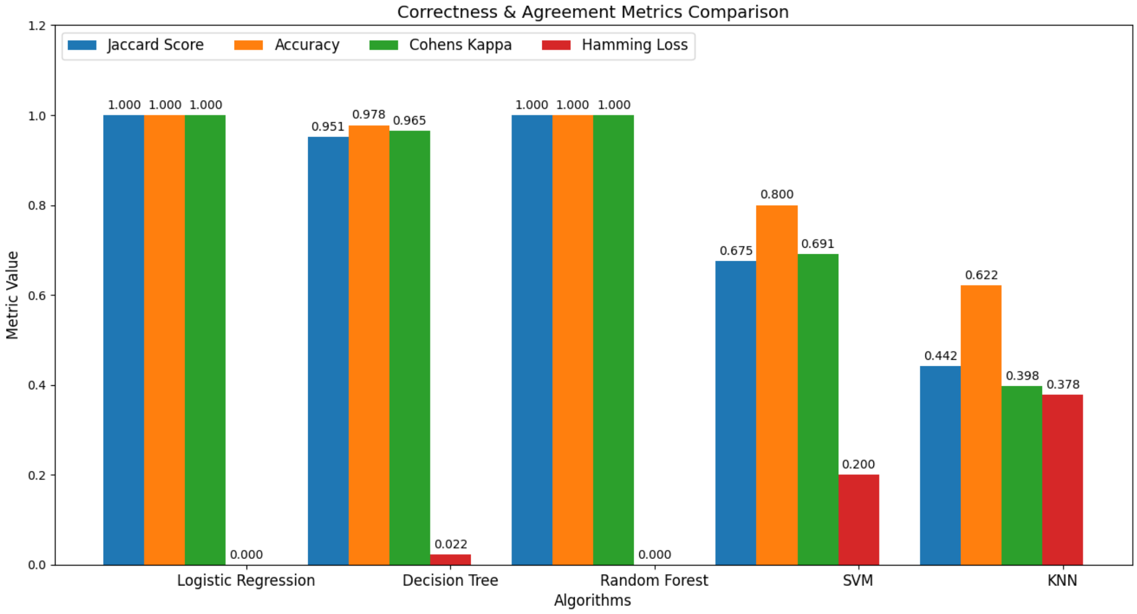


Figure 9: Correctness and Agreement Metrics for 'Wine' Dataset

These findings indicate that Logistic Regression and Random Forest performed exceptionally well across multiple evaluation metrics, including Recall, Precision, Accuracy, F1 Score, Cohen's Kappa, Hamming Loss, and Jaccard Score. Decision Tree also exhibited strong performance in terms of Recall, Precision, Accuracy, and F1 Score but had lower scores in explaining the variance. SVM and KNN had comparatively lower scores across most metrics.

4.4 Comparison of Algorithm Performance on Dataset 4

The performance of the algorithms in Dataset 4, specifically in the Breast Cancer dataset, was scrutinized using classification metrics. Furthermore, we evaluated the training duration of the algorithms, thus affording a comprehensive appraisal of their proficiency on the Breast Cancer dataset.

4.4.1 Error Metrics and Training Time

The execution times for the various machine learning algorithms on the Dataset Breast Cancer are shown in Figure 10 along with the performance metrics Recall, Precision, and F1 Score. With a Recall of 0.948, Precision of 0.958, and F1 Score of 0.952, Logistic Regression exhibits strong performance, demonstrating its ability to accurately identify positive instances. The most accurate classification of instances is achieved by Random Forest, which has the highest Recall (0.958), Precision (0.963), and F1 Score (0.961). Recall (0.953), Precision (0.953), and F1 Score (0.953) all perform similarly to Decision Tree.

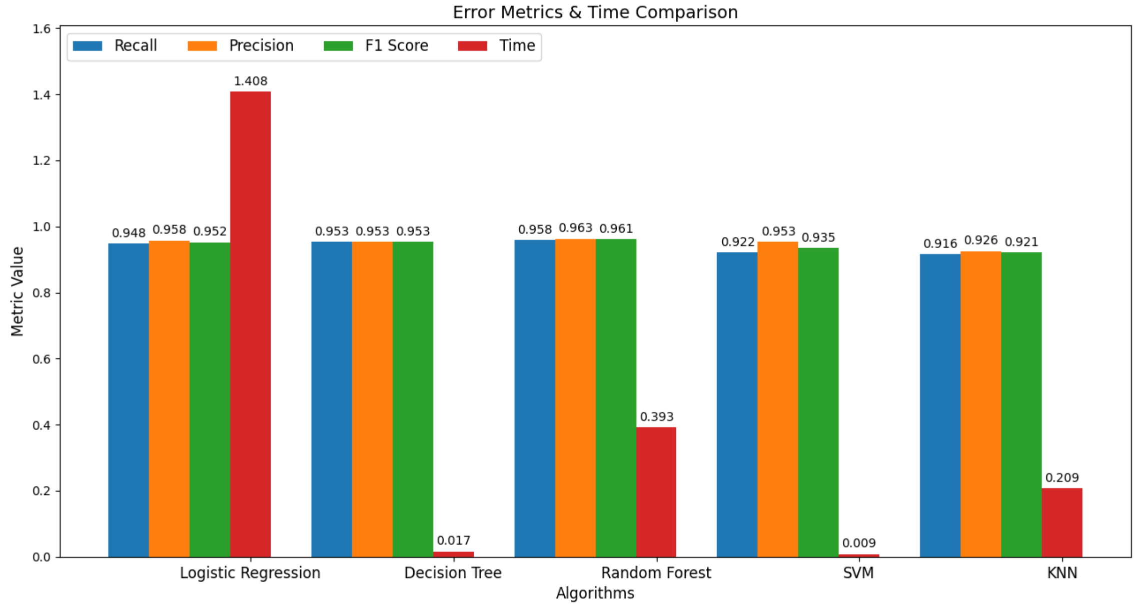


Figure 10: Error Metrics and Time for 'Breast Cancer' Dataset

SVM and KNN also have respectable Recall, Precision, and F1 Score values, indicating their ability to correctly classify instances. There are notable differences in execution times, with Decision Tree being the fastest and Logistic Regression taking the longest.

4.4.2 Correctness and Agreement Metrics

Figure 11 depicts the performance metrics Accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score for the various machine learning algorithms on Dataset Breast

Cancer. The accuracy of Logistic Regression is 0.958 and the Cohen's Kappa is 0.905, indicating a high overall correctness rate and agreement between predicted and actual labels. In terms of Accuracy (0.965), Cohen's Kappa (0.921), and Jaccard Score (0.924), Random Forest outperforms the other algorithms, demonstrating its effectiveness in making accurate predictions and capturing agreement. Decision Tree performs similarly to Accuracy (0.958) and Cohen's Kappa (0.906). SVM and KNN also achieve high Accuracy, Cohen's Kappa, and Jaccard Score values, indicating their ability to predict and agree with true labels.

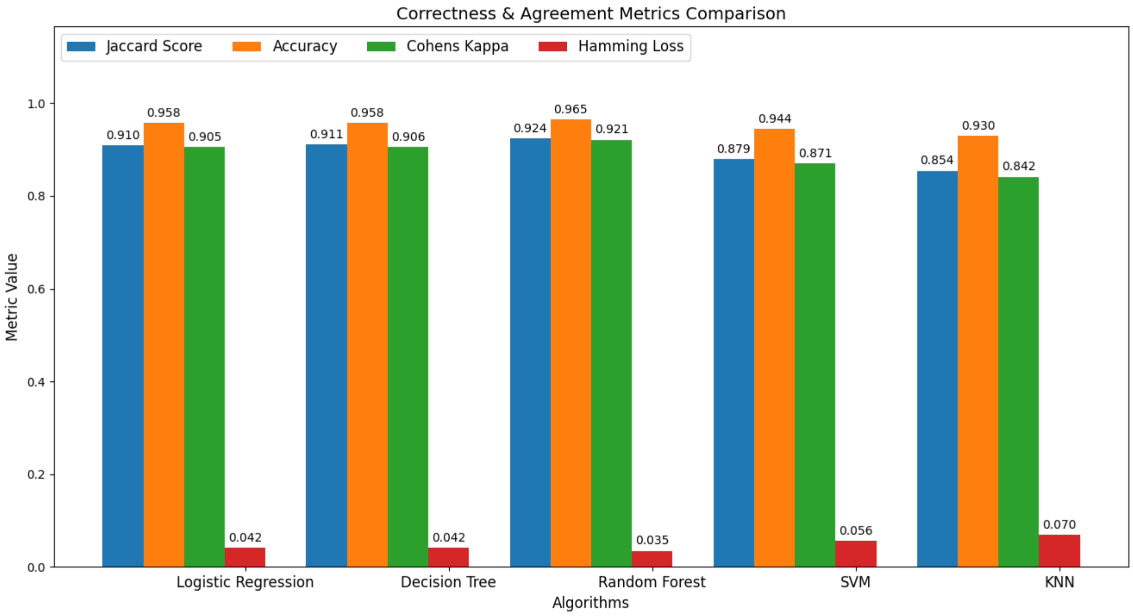


Figure 11: Correctness and Agreement Metrics for 'Breast Cancer' Dataset

4.5 Comparison of Algorithm Performance on Dataset 5

Through an in-depth exploration of Dataset 5, which is the Olivetti faces dataset, we conducted a comprehensive analysis that involved assessing the efficacy of various machine learning algorithms through performance evaluation metrics. This study has yielded valuable insights into the predictive efficacy of the aforementioned algorithms.

4.5.1 Error Metrics and Training Time

Figure 12 displays the recall, precision, F1 score, and execution time of the algorithms. According to the results, Logistic Regression demonstrates the highest values for Recall, Precision, and F1 Score, indicating its effectiveness in classifying the images in the Olivetti Faces dataset. Decision Tree exhibits lower performance compared to other algorithms, with lower values for Recall, Precision, and F1 Score. Random Forest, SVM, and KNN show moderate performance, with varying levels of Recall, Precision, and F1 Score.

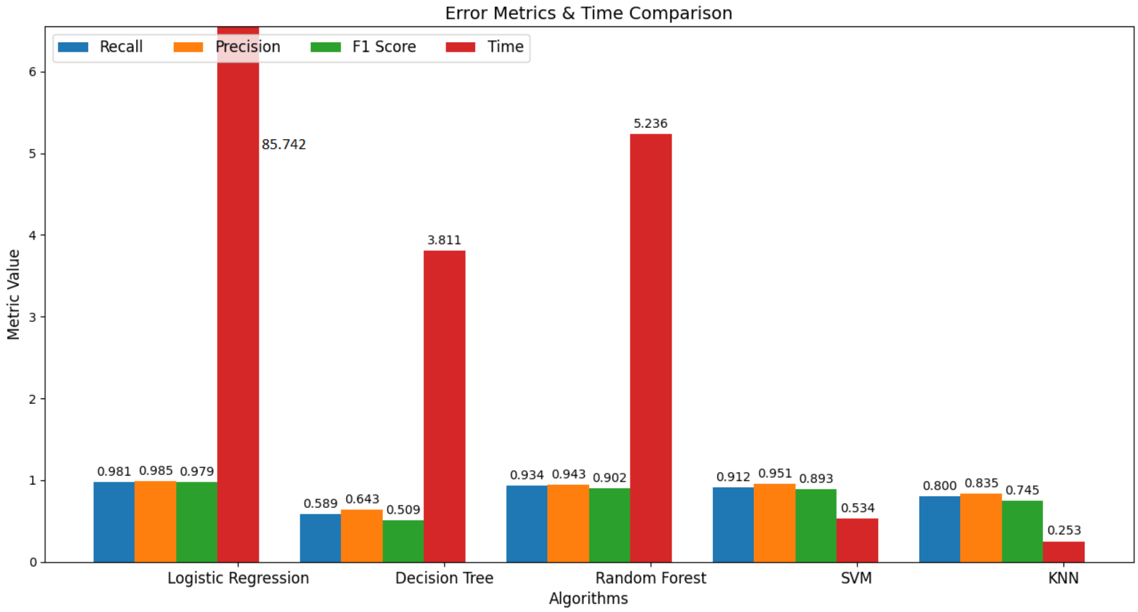


Figure 12: Error Metrics and Time for 'Olivetti Faces' Dataset

The execution time for the algorithms is also provided in Figure 12. Logistic Regression took 85.742 seconds, Decision Tree took 3.811 seconds, Random Forest took 5.236 seconds, SVM took 0.534 seconds, and KNN took 0.253 seconds. These results show significant variations in the execution times, with Logistic Regression having the longest execution time and KNN having the shortest.

4.5.2 Correctness and Agreement Metrics

Figure 13 presents the comparison of the selected algorithms in terms of Accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score on the Olivetti Faces dataset. The results indicate that Logistic Regression demonstrates the highest values for Accuracy, Cohen's Kappa, and Jaccard Score, indicating its superior performance in classification accuracy and agreement with the true labels. Decision Tree exhibits lower performance compared to other algorithms, with lower values for Accuracy, Cohen's Kappa, and Jaccard Score. Random Forest, SVM, and KNN show moderate performance, with varying levels of Accuracy, Cohen's Kappa, and Jaccard Score.

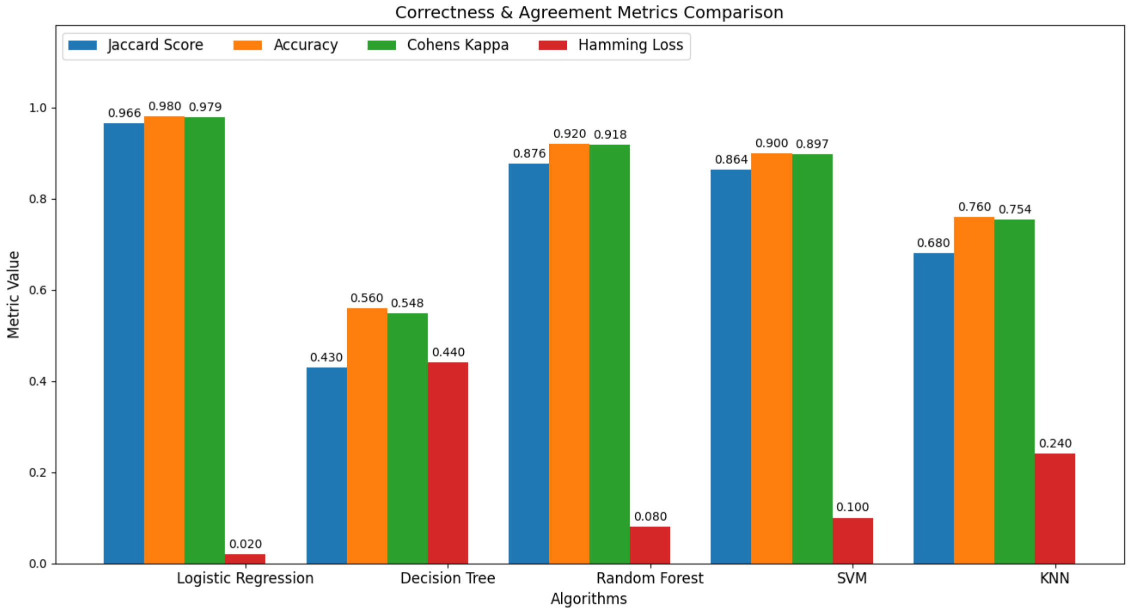


Figure 13: Correctness and Agreement Metrics for 'Olivetti faces' Dataset

The findings from the Olivetti Faces dataset suggest that Logistic Regression is the most effective algorithm in terms of classification accuracy, precision, and agreement with the true labels. Decision Tree, Random Forest, SVM, and KNN provide alternative options with varying levels of performance.

4.6 Comparison of Algorithm Performance on Dataset 6

We explored the performance of different machine learning algorithms on Dataset 6, the 20 Newsgroups Vectorized dataset. The objective was to provide a thorough assessment of the algorithms’ suitability for predictive data analytics on the 20 Newsgroups Vectorized dataset.

4.6.1 Error Metrics and Training Time

Figure 14 presents a comparison of the performance metrics Recall, Precision, and F1 Score, along with the execution time, for the different machine learning algorithms on Dataset 20 Newsgroups Vectorized. Logistic Regression demonstrates competitive performance with a Recall of 0.776, Precision of 0.795, and F1 Score of 0.779, indicating its ability to accurately identify positive instances. Random Forest outperforms the other algorithms, achieving the highest Recall (0.810), Precision (0.829), and F1 Score (0.814), demonstrating its effectiveness in correctly classifying instances. Decision Tree shows moderate performance in terms of Recall (0.611), Precision (0.614), and F1 Score (0.612), while KNN exhibits relatively lower values.

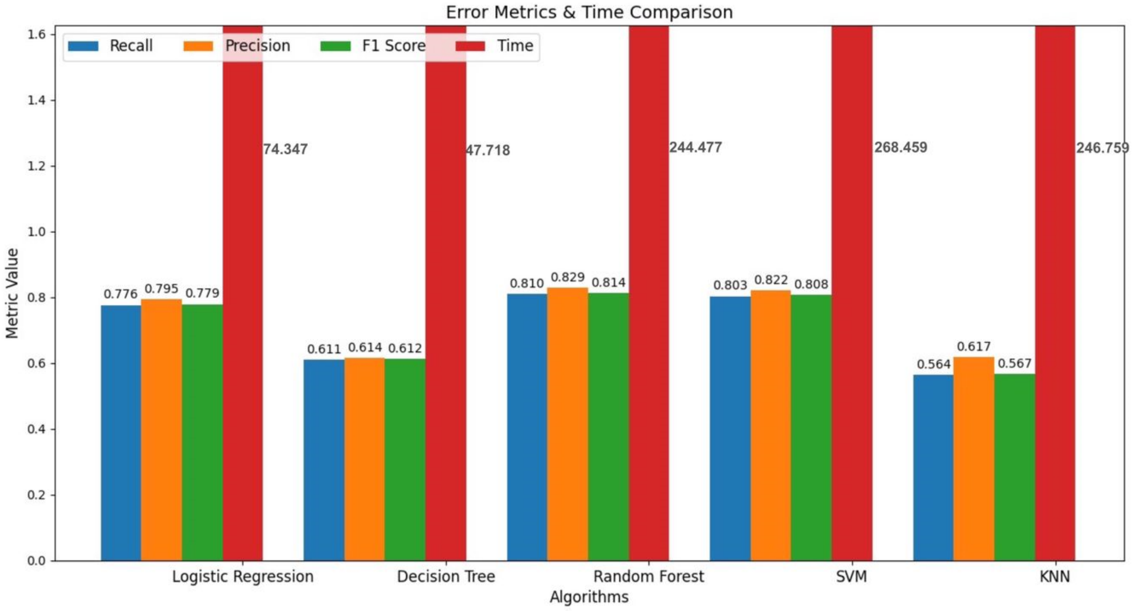


Figure 14: Error Metrics and Time for ‘20 News Groups Vectorized’ Dataset

Additionally, there are notable differences in the execution times, with Decision Tree being the fastest algorithm and SVM having the longest execution time.

4.6.2 Correctness and Agreement Metrics

Figure 15 provides an overview of the performance metrics Accuracy, Cohen's Kappa, Hamming Loss, and Jaccard Score for the different machine learning algorithms on Dataset 20 Newsgroups Vectorized. Logistic Regression achieves an Accuracy of 0.786 and a Cohen's Kappa of 0.775, indicating a high overall correctness rate and agreement between predicted and actual labels. Random Forest outperforms the other algorithms in terms of Accuracy (0.821), Cohen's Kappa (0.811), and Jaccard Score (0.696), demonstrating its effectiveness in achieving accurate predictions and capturing agreement. Decision Tree exhibits a moderate level of correctness but has the highest Hamming Loss, implying a higher number of incorrect predictions. KNN shows relatively lower performance in terms of Accuracy, Cohen's Kappa, and Jaccard Score, indicating a lower correctness rate and agreement compared to the other algorithms.

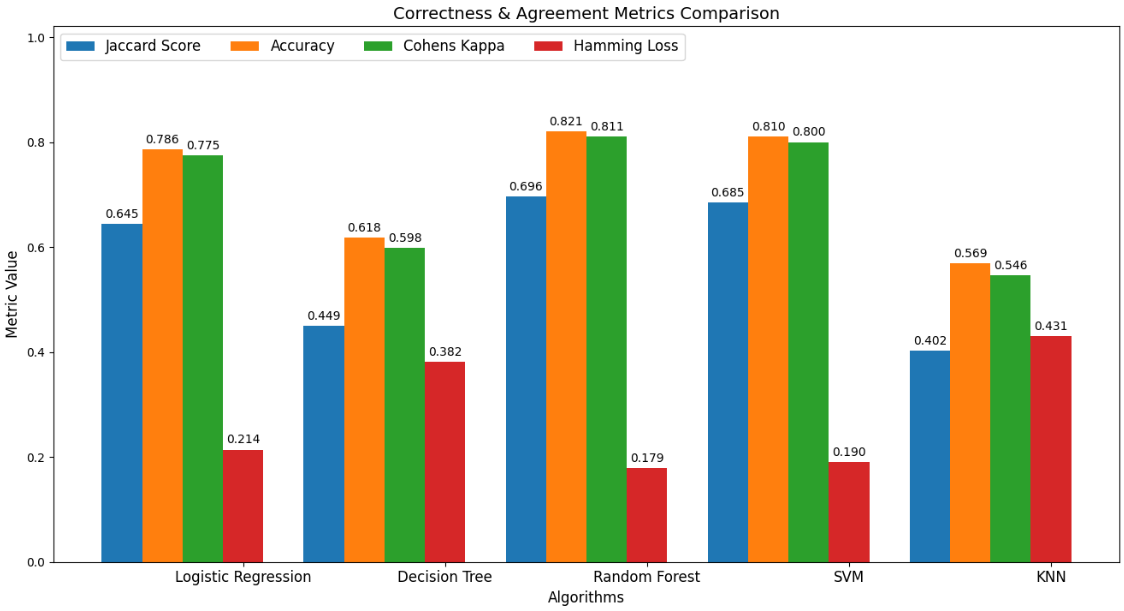


Figure 15: Correctness and Agreement Metrics for '20 News Groups Vectorized' Dataset

4.7 Comparison of Algorithm Performance on Dataset 7

Finally, we explored the performance of different machine learning algorithms on Covertype dataset. The objective was to conduct a comprehensive evaluation of the algorithms' applicability for predictive data analytics on a substantial and expansive dataset.

4.7.1 Error Metrics and Training Time

Figure 16 displays the recall, precision, F1 score, and execution time of the algorithms on the Covertype dataset. According to the results, Logistic Regression demonstrates a moderate recall, precision, and F1 score. Decision Tree exhibits the highest recall, precision, and F1 score among the algorithms, indicating its effectiveness in classifying the data in the dataset. Random Forest, SVM, and KNN show relatively high performance, with varying levels of recall, precision, and F1 score.

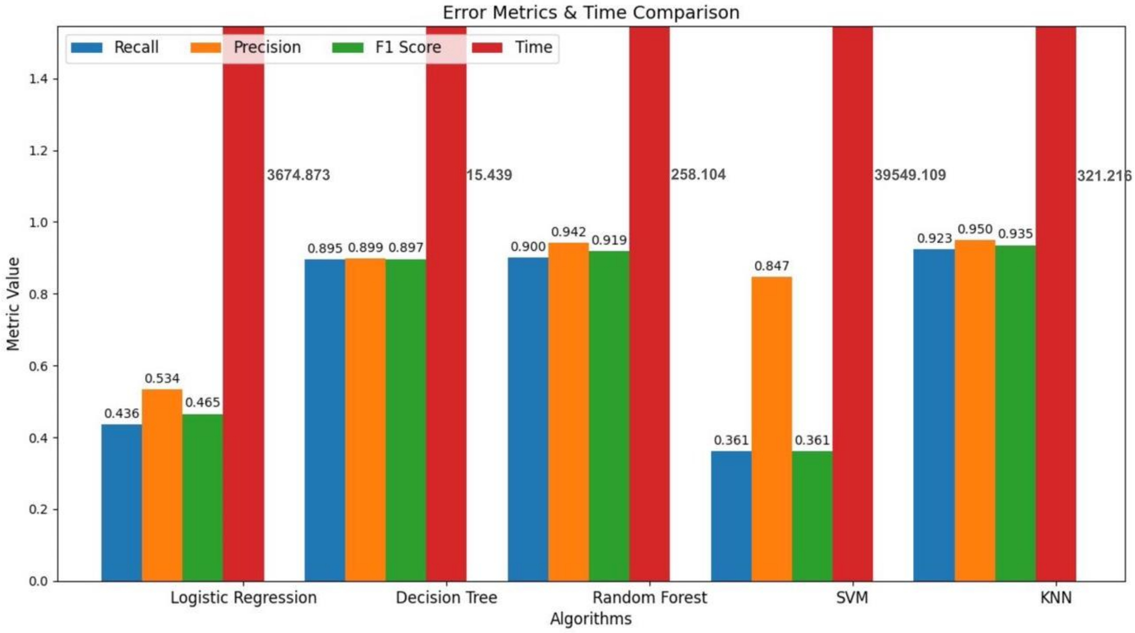


Figure 16: Error Metrics and Time for 'Covertype' Dataset

The execution time for the algorithms is also provided in Figure 16. Logistic Regression took 3674.873 seconds, Decision Tree took 15.439 seconds, Random Forest took 258.104 seconds, SVM took 39549.109 seconds, and KNN took 321.216 seconds. These results show significant variations in the execution times, with SVM having the longest execution time and Decision Tree having the shortest.

4.7.2 Correctness and Agreement Metrics

Figure 17 presents the comparison of the selected algorithms in terms of accuracy, Cohen's Kappa, Hamming loss, and Jaccard score on the Covertype dataset. The results indicate that Logistic Regression demonstrates a relatively high accuracy, Cohen's Kappa, and Jaccard score, indicating its superior performance in classification accuracy and agreement with the true labels. Decision Tree exhibits the highest accuracy, Cohen's Kappa, and Jaccard score among the algorithms, indicating its effectiveness in classifying the data in the dataset. Random Forest, SVM, and KNN show relatively high performance, with varying levels of accuracy, Cohen's Kappa, and Jaccard score.

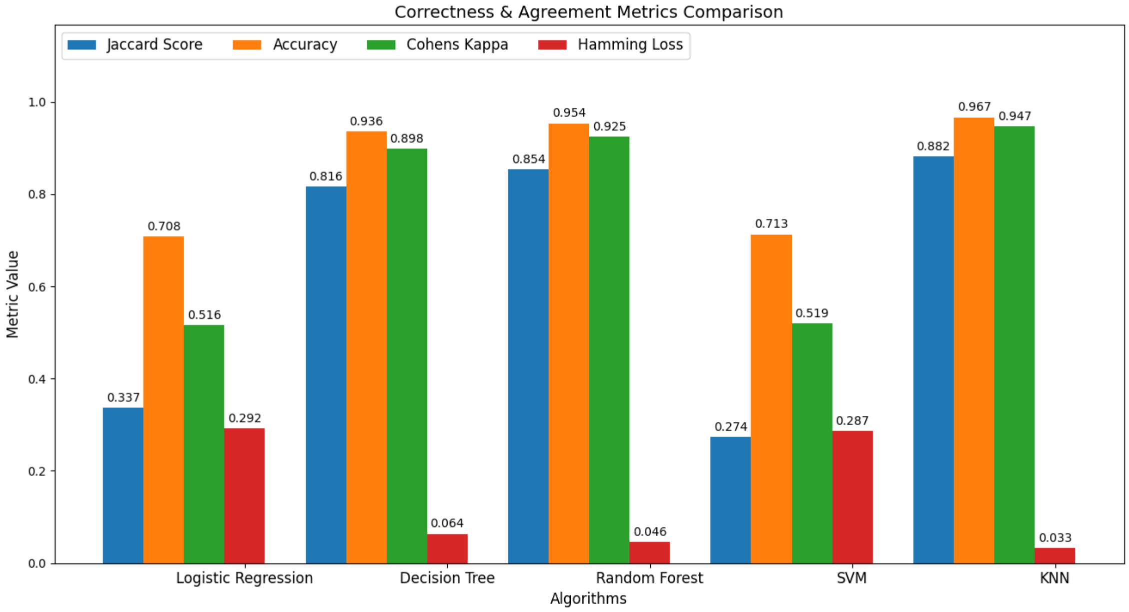


Figure 17: Correctness and Agreement Metrics for 'Covertype' Dataset

Finally, the findings suggest that the choice of algorithm for predictive data analytics on the Covertype dataset depends on the specific objectives and requirements of the application. Random Forest, SVM, and KNN provide alternative options with higher recall, precision, F1 Score, accuracy, Cohen's Kappa, and Jaccard Score, while Logistic

Regression and Decision Tree offer a balance between performance and execution time.

4.8 Summary of Results

On various datasets, the performance of several algorithms was examined. For the Iris dataset, all models scored good recall, precision, F1 score, accuracy, Cohen's Kappa, Hamming loss, and Jaccard score, suggesting their classification efficacy. Similarly, the models performed admirably on the Digits dataset, earning high marks across multiple evaluation metrics. The Wine dataset produced outstanding results for logistic regression and random forest, but significantly lower scores for SVM and KNN. All models performed consistently well on the Breast Cancer dataset, with logistic regression, decision tree, and random forest reaching remarkable results. However, SVM and KNN scored slightly worse than the other models. Furthermore, the Olivetti Faces dataset showcased superior performance for logistic regression, random forest, and SVM, while decision tree and KNN showed lower scores. Following that, the 20 Newsgroups Vectorized dataset produced impressive results for logistic regression, random forest, and SVM, with good recall, precision, accuracy, F1 score, error rate, Cohen's Kappa, Hamming loss, and Jaccard score. Decision trees and KNN, on the other hand, scored lower across multiple evaluation metrics. On the Covertype dataset, Logistic Regression performed moderately, whereas Decision Tree, Random Forest, and KNN performed well across various evaluation metrics. In comparison to the other models, SVM performed quite poorly.

5. Conclusion

In conclusion, the goal of this master's thesis was to evaluate and compare classification algorithms and evaluation criteria in the disciplines of predictive data analytics and machine learning. We attempted to answer the research questions and achieve the study's objectives by developing a self-designed tool and evaluating numerous datasets. Our findings provide important insights into the accuracy, effectiveness, and scalability advantages and disadvantages of various classification algorithms when applied across different issue domains. Additionally, we determined which evaluation metrics were most appropriate for various datasets and problem domains and contrasted their ability to precisely measure algorithm performance.

On various datasets, the evaluation of various machine learning models produced some intriguing performance patterns and insights. We noticed that across the datasets under evaluation, logistic regression, decision tree, random forest, and SVM consistently scored highly across a variety of evaluation metrics, demonstrating their efficacy in classification tasks. KNN, however, displayed slightly lower scores in some circumstances. In particular, the Iris and Digits datasets illustrated the efficacy of all models, whereas the Wine, Breast Cancer, Olivetti Faces, 20 Newsgroups Vectorized, and Covertype datasets illustrated variations in algorithm performance. These results underline how crucial it is to take dataset characteristics and problem domains into account when choosing the best classification algorithm. Our analysis also helped practitioners make wise decisions by highlighting the advantages and disadvantages of various algorithms.

While the understanding of algorithm performance and evaluation in the context of predictive data analytics and machine learning has greatly benefited from this research, it is important to recognize its limitations and pinpoint areas for future research. Multiple datasets used for the evaluation and comparison of classification algorithms have yielded useful insights into their advantages and disadvantages. These results underline how crucial it is to take dataset properties and problem domains into account when choosing the best algorithm. It is crucial to remember that the evaluation was constrained to a particular set of algorithms, datasets, and evaluation metrics. Future studies can broaden the analysis to incorporate more algorithms, bigger datasets, and different evaluation metrics, enabling a more in-depth comprehension of algorithm behavior. Additionally, examining particular issues like class imbalance and the impact of dataset size can shed more light on algorithm performance and direct the creation of more reliable predictive models.

6. References

[1] Datacy. (2023, February 21). “Predictive Analytics: An Overview.” Datacy Blog. Retrieved March 14, 2023, from https://datacy.com/blog/predictive-analytics-an-overview.

[2] Ribeiro, J. (2020, December 4). “What is Predictive Analytics and how can you use it today?” Towards Data Science. Retrieved April 23, 2023, from https://towardsdatascience.com/what-is-predictive-analytics-dc6db9759936.

[3] IBM. (n.d.). “What is predictive analytics?” IBM. Retrieved April 23, 2023, from https://www.ibm.com/topics/predictive-analytics.

[4] SAS. (n.d.). “Predictive Analytics: What it is and why it matters”. SAS. Retrieved April 23, 2023, from https://www.sas.com/en_us/insights/analytics/predictive-analytics.html.

[5] Halton, C. (2023, January 30). “Predictive Analytics: Definition, Model Types, and Uses”. Investopedia. Retrieved April 23, 2023, from https://www.investopedia.com/terms/p/predictive-analytics.asp.

[6] Kumar, V. (2018). “Predictive Analytics: A Review of Trends and Techniques”. ResearchGate. Retrieved May 10, 2023, from https://www.researchgate.net/publication/326435728_Predictive_Analytics_A_Review_of_Trends_and_Techniques.

[7] Abdullahi, A. (2022, October 19). “What are the different types of predictive modeling?” TechRepublic. Retrieved April 23, 2023, from https://www.techrepublic.com/article/types-of-predictive-modeling.

[8] Suthaharan, S. (2016). “Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning”. Springer. Retrieved April 25, 2023, from https://link.springer.com/book/10.1007/978-1-4899-7641-3.

[9] Frankenfield, J. (2022, January 16). “Understanding Machine Learning: Uses, Example”. Investopedia. Retrieved April 25, 2023 from https://www.investopedia.com/terms/m/machine-learning.asp.

[10] Coursera. (2023). “7 Machine Learning Algorithms to Know: A Beginner’s Guide”. Retrieved May 10, 2023, from https://www.coursera.org/articles/machine-learning-algorithms.

[11] IBM. (n.d.). “What is Logistic regression?” IBM. Retrieved April 19, 2023, from https://www.ibm.com/topics/logistic-regression.

[12] Jurafsky, D., & Martin, J. H. (2023). “Speech and Language Processing (3rd ed.) [PDF file]”. Retrieved May 10, 2023, from https://web.stanford.edu/~jurafsky/slp3/5.pdf.

[13] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). “Generative Adversarial Nets. In Advances in Neural Information Processing Systems 27 (NIPS 2014) [PDF file]”. Retrieved May 10, 2023, from https://proceedings.neurips.cc/paper_files/paper/2014/file/6cdd60ea0045eb7a6ec44c54d29ed402-Paper.pdf.

[14] Kuhn, M., & Johnson, K. (2013). “Applied Predictive Modeling”. Springer Science+Business Media. Retrieved May 10, 2023, from https://www.academia.edu/43332156/Applied_Predictive_Modeling.

[15] Abdulazeez, A. M. (2021, May). “Machine Learning Applications based on SVM Classification: A Review.” ResearchGate. Retrieved June 19, 2023, from https://www.researchgate.net/publication/351275238_Machine_Learning_Applications_based_on_SVM_Classification_A_Review.

[16] Sarker, I. H. (2021). “Machine Learning: Algorithms, Real-World Applications and Research Directions”. SN Computer Science, 2(160). doi:10.1007/s42979-021-00592-x.

[17] Zhang, Z. (2016). “Introduction to machine learning: K-nearest neighbors”. ResearchGate. Retrieved May 10, 2023, from https://www.researchgate.net/publication/303958989_Introduction_to_machine_learning_K-nearest_neighbors.

[18] Khan, A. S. (2017). “Medicinally Important Trees”. Springer International Publishing. Retrieved on May 10, 2023, from https://link.springer.com/book/10.1007/978-3-319-78503-5.

[19] Hossin, M., & Sulaiman, M. N. (2015). “A Review on Evaluation Metrics for Data Classification Evaluations”. International Journal of Data Mining & Knowledge Management Process, 5(2). Retrieved on May 10, 2023, from https://www.researchgate.net/publication/275224157_A_Review_on_Evaluation_Metrics_for_Data_Classification_Evaluations.

[20] Minaee, S. (2020, September 22). “20 Popular Machine Learning Metrics. Part 1: Classification & Regression Evaluation Metrics.” Towards Data Science. Retrieved May 10, 2023, from https://towardsdatascience.com/20-popular-machine-learning-metrics-part-1-classification-regression-evaluation-metrics-1ca3e282a2ce.

[21] DeepAI. (n.d.). “Evaluation Metrics Definition”. DeepAI. Retrieved on May 10, 2023, from https://deepai.org/machine-learning-glossary-and-terms/evaluation-metrics.

[22] Bajaj, A. (2023, May 9). “Performance Metrics in Machine Learning [Complete Guide]”. Neptune.ai. Retrieved on May 10, 2023, from https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide.

[23] Mankad, S. (2020, November 24). “A Tour of Evaluation Metrics for Machine Learning”. Analytics Vidhya. Retrieved on May 10, 2023, from https://www.analyticsvidhya.com/blog/2020/11/a-tour-of-evaluation-metrics-for-machine-learning/.

[24] C. Jiji (2021, February 8). “Evaluation Metrics 101: Regression, MSE, RMSE, R-squared, Precision, Recall, F1 score, ROC and AUC.” Medium. Retrieved May 10, 2023, from https://medium.datadriveninvestor.com/evaluation-metrics-101-7c8b4c3421c2.

[25] Widmann, M. (2020, September 21). Cohen’s Kappa: Learn It, Use It, Judge It. KNIME. Retrieved May 13, 2023, from https://www.knime.com/blog/cohens-kappa-an-overview.

[26] Wikipedia. (n.d.). Cohen’s kappa. Retrieved May 13, 2023, from https://en.wikipedia.org/wiki/Cohen%27s_kappa.

[27] idswater. (2020, January 19). What is hamming loss? Retrieved May 13, 2023, from https://ids-water.com/2020/01/19/what-is-hamming-loss/.

[28] Wu, G., & Zhu, J. (2020, November 16). “Multi-label classification: do Hamming loss and subset accuracy really conflict with each other?” arXiv preprint arXiv:2011.07805. Retrieved May 19, 2023, from https://arxiv.org/abs/2011.07805.

[29] Zach. (2020, December 23). A Simple Explanation of the Jaccard Similarity Index. Statology. Retrieved May 13, 2023, from https://www.statology.org/jaccard-similarity/.

[30] Dalianis, H. (2018). “Characteristics of Patient Records and Clinical Corpora.” In: Clinical Text Mining. Springer, Cham. https://doi.org/10.1007/978-3-319-78503-5_4.

[31] DeepAI. (n.d.). Jaccard Index Definition. Retrieved May 13, 2023, from https://deepai.org/machine-learning-glossary-and-terms/jaccard-index.

[32] Gupta, S. C., & Goel, N. (2023). “Predictive Modeling and Analytics for Diabetes using Hyperparameter tuned Machine Learning Techniques.” Procedia Computer Science, 218, 1257-1269. Retrieved May 09, 2023, from https://doi.org/10.1016/j.procs.2023.01.104.

[33] Theng, D., & Theng, M. (2020, July). “Machine Learning Algorithms for Predictive Analytics: A Review and New Perspectives.” ResearchGate. Retrieved May 09, 2023, from https://www.researchgate.net/publication/342976767_Machine_Learning_Algorithms_for_Predictive_Analytics_A_Review_and_New_Perspectives.

[34] Shah, K., Patel, H., Sanghvi, D., & Shah, M. (2020). “A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification.” Augmented Human Research, 5(12). Retrieved May 09, 2023, from https://doi.org/10.1007/s41133-020-00032-0.

[35] Kabir, S. M. S. (2016, July.). “METHODS OF DATA COLLECTION.” ResearchGate. Retrieved May 11, 2023, from

https://www.researchgate.net/publication/325846997_METHODS_OF_DATA_COLLECTION.