Credit Card Fraud Detection Using Machine Learning Techniques

by

Tasmia Kabir
17301015
Tahnin Nishat
17301231
Saria Bulbul Tory
17301039

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

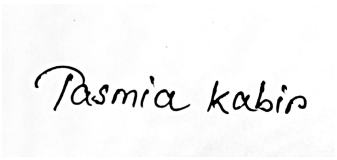
Department of Computer Science and Engineering
Brac University
September 2021

Declaration

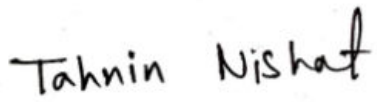
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Tasmia Kabir
17301015



Tahnin Nishat
17301231

Saria Bulbul Tory
17301039

Approval

The thesis titled "Credit Card Fraud Detection Using Machine Learning Techniques" submitted by

1. Tasmia Kabir (17301015)
2. Tahnin Nishat (17301231)
3. Saria Bulbul Tory (17301039)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 25, 2021.

Examining Committee:

Supervisor:
(Member)

Faisal Bin Ashraf
Lecturer
Department of Computer Science & Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science & Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science & Engineering
BRAC University

Abstract

The extensive use of the internet is perpetually drifting businesses to incorporate their administrations in the online environment. As a result of the development of e-commerce websites, people and monetary corporations count on online administrations to carry out their transactions. The ever-expanding utilization of internet banking associated with vast variety of online transactions has led to an exponential increase in credit card frauds. The fraudsters can likewise utilize anything to influence the systematic operation of the current fraud detection system (FDS). Therefore, we have taken up the challenge to upgrade the existing FDS with the most potential exactness. This research intends to develop an efficient FDS using machine learning (ML) techniques that are adaptive to consumer behavior changes and tends to diminish fraud manipulation, by distinguishing and filtering fraud in real-time. The ML techniques include Logistic Regression, Support Vector Machine, naïve Bayes, K-nearest neighbor, Random Forest, and Decision tree. According to this study, the Decision Tree classifier has emerged as the most useful algorithm among the wide range of various strategies.

Keywords: Machine Learning, Random Forest, Decision tree, Support Vector Machine, confusion matrix, outlier.

Dedication

Dedicated to every single soul who works hard to bring a change to the world of science.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor Faisal Bin Ashraf sir for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CM Confusion Matrix

DT Decision Tree

FN False Negative

FP False Positive

FPR False Positive Rate

KNN K-Nearest Neighbour

LR Logistic Regression

NB Naive Bayes

RF Random Forest

ROC Receiving Operating Characteristic

SVM Support Vector Machine

TN True Negative

TP True Positive

TPR True Positive Rate

Chapter 1

Introduction

Most organizations, enterprises, and government agencies have implemented electronic commerce to maximize their productivity in trading services and goods, like credit cards, telecommunications and so on. Since, e-commerce networks are used by both authorized customers and fraudsters, they are increasingly prone to large-scale fraud. Fraud is defined as a crime committed with the intent of obtaining money through deception.

The goal of a fraud control mechanism is to keep advanced technologies secure from fraud by preventing it from happening in the first place. Nevertheless, this approach is insufficient to prevent fraud. Fraud detection is often recommended as a way to improve the security of technical systems. Credit Card Fraud Detection identifies and recognizes fraudulent transactions as they occur in the system and alerts the system administrator. This is a delicate and intriguing process that requires proper identification and detection potentials. For a machine or a system to detect credit card fraud is a challenging phenomenon. For, accomplishing the process of fraudulent detection, a system needs to be highly trained with relevant data. Machine learning is learning procedure of the system or a machine where a machine learns through statistical methods of classification, regression and clustering. The reasons for which we have chosen machine learning based algorithms for identifying credit card fraudulent activities are:

- It detects frauds unfailingly.
- It can do the real-time streaming.
- It needs less time in authentication methods.
- It can pick out hidden interconnection in data.

As we know, machine learning depends deliberately on the data. The data is the essential feature that makes algorithm training feasible and explains why machine learning has grown so popular in recent years. It would be a matter of great concern, if we are unable to properly maintain data records, as this would render the system nearly useless and lead to waste of time and money. Nonetheless, if we apply dataset methodology, we may train the machine to operate with a large number of examples, and our work will be more relaxing.

On the contrary, all datasets are not precise; this is why data processing is a crucial step in the machine learning procedure. In short, processing a dataset will make

it more useable for machine learning. Moreover, data processing encompasses the development of an appropriate data collection mechanism. Therefore, the dataset method is the standard method that we have chosen. The reason for this is that it is the most prevalent and the simplest method of applying machine learning algorithms. It is preferred by everyone since it is convenient, easy and reliable. As we previously stated, no data is completely error-free or precise. Our dataset also contains some flaws that we can work on. We may need to prune and shape the data in the dataset we are using.

The thesis contains the subsequent sections. The literature review in chapter 2 provides several related approaches researchers used at times to address this similar problem. Chapter 3 discusses the methodology and the machine learning models that we simulated and in chapter 4 we have elaborately discussed about our data collection, manipulation & handling procedure. The experimental results and relevant discussions are presented in chapters 5 and 6 respectively. Lastly, chapter 7 contains the summary of the report and conclusion, as well as some future work plans.

1.1 Problem Statement

A couple of issues make this fraud detection system troublesome for researchers to execute. The absence of the literature providing test results and of genuine data is among one of the biggest issues. The explanation behind this is often the sensitive financial information related to the fraud that needs to be unbroken confidential for the aim of customer's privacy. Now, here we have a tendency to enumerate completely different properties a fraud detection system ought to have so as to get proper results:

- The system ought to be able to deal with slanted dispersions since just a small proportion of all credit card transaction is fraudulent.
- The system ought to have a correct means to handle the noise. Noise is that the errors that are present within the information, for instance, incorrect dates. This noise in actual information limits the accuracy of generalization that may be achieved, irrespective of how extensive the training set is.
- Another downside associated with this field is overlapping information. Some transactions might be counted as fallacious transactions whereas, they are real transactions and vice versa.
- The system should be prepared to adjust to new sorts of fraud. Proficient fraudsters persistently find new and quick methods of carrying out their work since flourishing fraud techniques become inefficient as they become standards after a while.
- There is a need for proper measurements to evaluate the classifier system. The overall precision cannot be evaluated on a slanted dispersion because even with high accuracy; most fallacious transactions are often classified as genuine.
- The system should deal with the amount of money that is being lost because of fraud and therefore the amount of cash that is required to detect that fraud.

1.2 Research Objectives

We intend to solve the credit card fraudulent detection using supervised machine learning algorithms. To do such, we came up with a technique of feature extraction of dataset and use the dataset to train models with the sole purpose of detecting fraud transactions. Our proposed system will accomplish following objectives:

- To detect fraud transactions of credit cards with high accuracy and efficiency
- To train the high performance model by utilizing label and features of a given dataset
- To use the trained model to check if a particular transaction is fraud or not.
- To introduce cost-saving methods to detect fraudulent transactions.

In this research, we focused on data driven models and methods that are accessible at the least expense in the market.

Chapter 2

Literature Review

Starting from the very beginning of Machine Learning, engineers have been utilizing this innovation in Credit Card Fraudulence Discovery. At various times, researchers experimented with novel ideas and attempted to consolidate algorithms in collaboration with Machine Learning. In our background study, we came across such research where the authors effectively conducted experiments utilizing different ML strategies such as Random Forest Algorithm, K-Nearest Neighbor, and others.

In this paper, the authors S.P. Maniraj, A. Saini and S. D. Sarkar utilized the Local Original Factor and Isolation Forest Algorithms to identify card fraudulence [14]. They pointed that this process detection includes coordinating the activities of the clients to estimate, perceive or avoid questionable behavior, consisting of fraud, intrusion, and defaulting. According to the authors, the fraudulence issue can be identified from the relevant and related parameters as it is potentialized by its different components. Also, the transaction designs frequently alter their statistical properties over time. From time to time, numerous researchers have utilized algorithms like Outlier Mining, K-Nearest Neighbor, Logistic Regression, Adversarial Detection and more to find credit card extortion.

The authors in this paper, A. Saini and S. D. Sarkar have focused on utilizing the Outlier Factor and forest algorithm to address this issue [14]. They got significant datasets from Kaggle where 28 were named as v1-v28 to protect delicate information out of 31 columns. The other columns are named time, amount and class. Time represents the exact time of the transaction. Amount is the total cost of the products purchased or money transferred. Class 0 shows a substantial transaction and 1 distinguishes a false one. They plotted different charts like histogram, heat maps and more to look for unpredictability within the dataset. After understanding, organizing, and handling the dataset, the authors applied outlier detection modules utilizing open-source Python libraries (NumPy, SciPy, matplotlib). Local Original factor is an unsupervised Outlier detection algorithm. It mentions the result of each test. It calculates the portion which the parameter deviated locally in comparison to its neighbors. The algorithm K-Nearest Neighbors calculates the locality, in which the local data is determined from the distance. On the other hand, Isolation forest subjectively chooses a feature to isolate the observations and then randomly selects a split esteem to compare the minimum and maximum assigned feature values. A tree can display the recursive partitioning where the splits necessary to separate samples can be calculated from the path length between the terminating node and the root node. A. Bhanusri and K. R. Sree outlined a modern comparison measure

representing the benefit and losses for the proposal of fraud detection [12]. In their paper, a Bayes based minimum risk is displayed which ultimately utilizes the recommended cost measure. When this method and advanced algorithms are analyzed, updates up to 23% are obtained. The data set of this paper was based on real-life transactional information by a tremendous European company and individual properties in data are kept private. According to P. Jyothi, G. V. Sai and R. R. Sai the significance of this paper was to discover an algorithm and to reduce the cost proportion. The result obtained was 23% and the algorithm they found was Bayes minimum risk [18]. Here, from the source and cleaning and approval the credit dataset is taken which is performed on the dataset which incorporates to evacuate redundancy and changing over necessary variables into classes. At that point data is separated into 2 parts, for preparing 70 percent is utilized whereas 30% is set for validating and testing. Now K fold cross-validation is done which is the genuine example and is randomly separated into k equivalent measured subsamples. In the k subsamples, an individual subsample is kept as the approval data for testing the model, and the remaining k1 subsamples are utilized for planning information for Logistic regression, Decision tree, SVM, Random Forest, and after that proficiency, sensitivity, specificity, precision are decided, and a relation is made. From the tests, the result that has been gotten is that, Logistic regression includes a accuracy of 97.7% whereas SVM appears a accuracy of 97.5% and Decision tree appears a accuracy of 95.5% but the leading comes about are found by Random forest with exact accuracy of 98.6%. So the results conclude that random forest shows the foremost definite and high accuracy of 98.6% in a problem of credit card fraud detection with dataset applied by machine learning.

As discussed by S. Lakshmi and S. D. Kavila, in final few years the utilization of credit cards has significantly expanded due to the noteworthy progression of the e-commerce administrations. As credit cards are gaining popularity in large portion, in the meantime the numbers of fraud cases are expanding as well [2]. Each year, millions of individuals fall victim to those fraud cases that costs the national economy with a huge amount of money. Here, in our paper we are going to use machine learning algorithms to form an appropriate framework that can detect the frauds. As said by mentioned authors, the method of finding or recognizing these frauds is very challenging. In addition, there are different models which are proposed models for fraud detection of credit card and now this detection has been executed by lots of strategies such as data mining, statistics, and artificial intelligence. We can use Genetic Algorithms and other algorithms for making the system. This permits the data to be classified into either non-suspicious or suspicious activity. This credit card fraud detection strategy is additionally valuable for home insurance data. It is a vital strategy when handling credit card fraud and uses a lot of strategies that highlight suspicious transactions. We can also use decision tree method. This strategy works by using a similarity tree, created by using decision tree logic. A similarity tree is made with nodes and leaves which have attributes and factors. Clustering methods is another great strategy for solving this issue.

After going through comparable works and research in the credit card fraud detection sector, we have chosen to apply a collaborated algorithm that will have the fundamental features of pattern detection, anomaly and duplicate prediction utilizing machine learning. Our motive is to come with effective and best solution for

detecting these frauds. Our work helps to a better understanding of what can be considered relevant knowledge for a credit card fraud detection work and how to incorporate it within the classification task for simpler way of fraud detection.

As indicated by the 2019 Federal Reserve Payments Study [1], an estimate of total card installment has arrived at the mark of 131.2 billion with a value of $7.08 trillion in 2018. There is a growth in credit card payments at a rate of 8.9% per year from 2015 to 2018. With this, there is an expansion in credit card fraud losses too and has arrived at the mark of $27.85 billion in 2018, as per the Nilson Report 2019 [2]. In the current framework, we researched when the credit card fraud issue first occurred, the fundamental prevention solution was spinning around "how can we protect our credit card from being stolen". To diminish the rate of credit card fraud, the issuer guaranteed that controls were designed to prevent the theft of credit cards or use of stolen cards, and further examining the fraud immediately. There were no exceptional preventive techniques for distinguishing credit card fraud about thirty years back except carrying the card safely. A few strategies have as of now been proposed since 1990. In paper [3], analyzed different AI models like logistic regression, random forest, Naïve Bayes and found logistic regression with the highest recall. Well, the highest accuracy does not mean the best model always. We also need to view the weighted score of exactness and recall which finishes up the fittest model. Another research [7] has demonstrated the examination of K-NN, Naïve Bayes and Logistic Regression Classifier and presumed that logistic regression does not perform well for highly skewed data as it truly tends to over fit the data. Numerous machine learning models like Bayes Minimum Risk , Decision Tree, Support Vector Machine (SVM), Logistic Regression, have been focused in various occasions and some of them are cost-effective and some are truly expensive computationally. Fanning and Cogger (1998) built a fraud detection model utilizing a neural network. The input vector comprised of qualitative variables and financial ratios. After performance analysis, the outcome showed that their model is more viable at fraud detection compared to statistical methods, just as logistic regression. Bhattacharyya (2011) evaluated two exceptional data mining methods, Support Vector Machines, and Random Forests, collectively with logistic regression [4]. SVMs work in the high dimensional feature space without incorporating any extra computational complexity. SVMs seems alluring due to the integrity of a linear classifier and the potential to work in a feature rich space where the detection of a fraudulent transaction is hard to accomplish. Random forests are efficient since each tree is created separately from the others. Sahin (2013) stated a cost-sensitive method for fraud detection applying a decision tree and showed that this algorithm outperforms the current methods [5].

Malini performed the KNN algorithm together with outlier detection methods to optimize the most suitable solution for fraud detection [6]. These strategies are manifested to reduce the false alarm rates and increase the rate of fraud detection. In their paper, Zareapoor and Shamsolmoali [7] presented fraud detection strategies significantly Naïve Bayes, KNN, SVM, and Bagging Ensemble Classifier. They discussed the different concerns they faced during their research. For instance, there is a lack of real-world data since the banks and other financial organizations do not make their data public because of privacy concerns. Furthermore, the data is quite unstable as the percentage of fraudulent transactions is just 2%, whereas that of the

legitimate transaction is 98% in most cases. Tom Sweers [8] in his paper, describes Auto Encoders as an efficient neural network that can encode the data and decode it as well. The Auto encoders are trained to non-anomaly points in this approach. Here, any value over the threshold or upper-bound value could be computed as an anomaly. Jain (2016) introduced a cost-sensitive method based on Bayes Minimum Risk. The paper is comprised of real-life data set provided by a large European organization and the exactness of a calculation is around 50%. The importance of this paper was to obtain a proper algorithm and to decrease the rate of expense. Delamaire (2009) introduced the various kinds of credit card fraud and explained the suitable methods to fight them; such as clustering techniques, neural networks, genetic algorithms, and decision trees. Not only credit card fraud detection, machine learning algorithms are largely being used in predicting customer strategies in online shops also. In [28] the authors stated that machine learning algorithms like SVM, Decision Tree, Random Forests can be used to predict customer tendency of purchasing products from online stores. Also machine learning algorithms have got wide use in medical and other fields now-a-days. Similarly, in [29], the authors have shown the use of similar machine learning algorithms in predicting the prognosis of cervical cancer in women.

Chapter 3

Methodology

With a view to developing a system that can comfortably detect the credit card fraud alerts with machine learning algorithms we must go along with dataset approach. In order to consolidating with machine learning algorithms we have to train the machine. On condition that, training a machine with examples is the finest way to go around with. However, no other approach is as suitable as the dataset approach.

Training premium quality Machine Learning systems requires a significant amount of data based on previous transactions. In the contradiction, if we do not have plenty of data regarding previous counterfeit and regular transactions, it would be difficult to conduct a machine learning algorithm on it. Moreover, the standards of the machine learning algorithm formulation procedure rely on the standards of the process and inputs. However, there are an odd number of cases where training or processing datasets restrain both the same quantity of data and samples in two categories. Nevertheless, the quality of the dataset should always be maintained. Again, there would be some problems we might have to face, as we know that models might be subject to bias depending on the character and standard of previous data. From that narration, we should know that if the model makers or upholders could not gather and categorize the data methodically and decently there could be a chance for mixing up the details of fraudulent deals and the normal ones, which would generate crucial bias in our model's outcomes. On the other hand, if we own numeral data which might be skillfully maintained as well as unbiased, and if our reasoning matches the machine learning model precisely, the possibility exists that our fraud detection will work properly for all customers. As a result, we will surely benefit from using the dataset technique. Fig.(3.1) represents the workflow we followed to determine the best machine learning model for efficient credit card fraudulent detection.

3.1 Implemented Models

We have used six notable machine learning algorithms to train and test the machine with the acquired dataset. These six algorithms are very promising and regular in implementing machine learning techniques. A brief description of the working principles of these algorithms is given below:

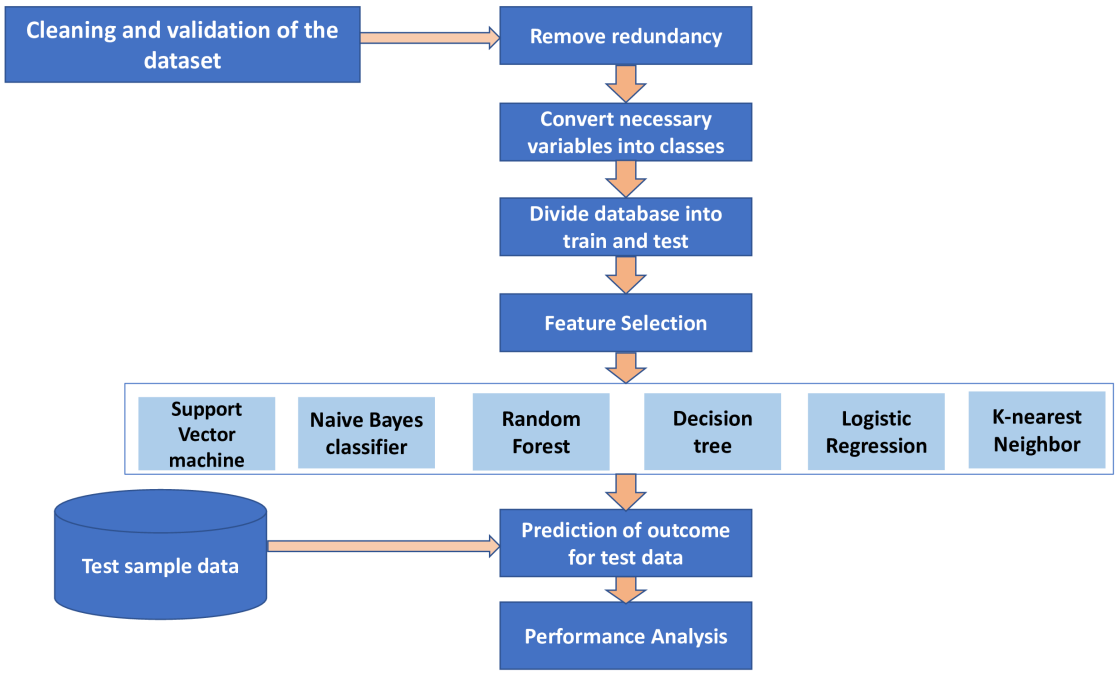


Figure 3.1: Workflow Diagram

3.1.1 Random Forest

It is a decision tree-based algorithm which incorporates building a few trees and improves the output combining with the generalization capacity of the model. This strategy of combining trees is called an ensemble strategy. Ensemble is a combination of individual trees to create a solid learner. It can be utilized to deal with regression and classification issues. In regression issues, the dependent variable is continuous. In classification issues, the subordinate variable is categorical.

Working Procedure of Random Forest

Random forest is a executed learning algorithm. The "forest" it builds, is kind of decision trees, which is trained with the "bagging" technique. In bagging strategy a combination of learning models increases in general result. One advantage of random forest is that it can be utilized as both classification and regression issues, which outline the bigger portion of current machine learning systems. Let's see at random forest in classification, since classification is some of the time considered the building piece of machine learning. Fig.(3.2) depicts how a random forest works in two trees.

Random forest got almost the same hyper parameters like a decision tree. But there's no need to connect a decision tree with a bagging classifier as the classifier- class of random forest can be utilized effectively. With random forest, it can deal with regression tasks by utilizing the algorithm's regressor. Random forest incorporates extra haphazardness to the model, while growing the trees. Instead of trying to find the foremost vital feature while splitting a node, it looks for the finest feature among a random subset of features, coming about in a wide difference that generally comes about in a better higher model. Therefore, only a random subset of the features is taken into consideration by the algorithm for splitting a node in random forest. You

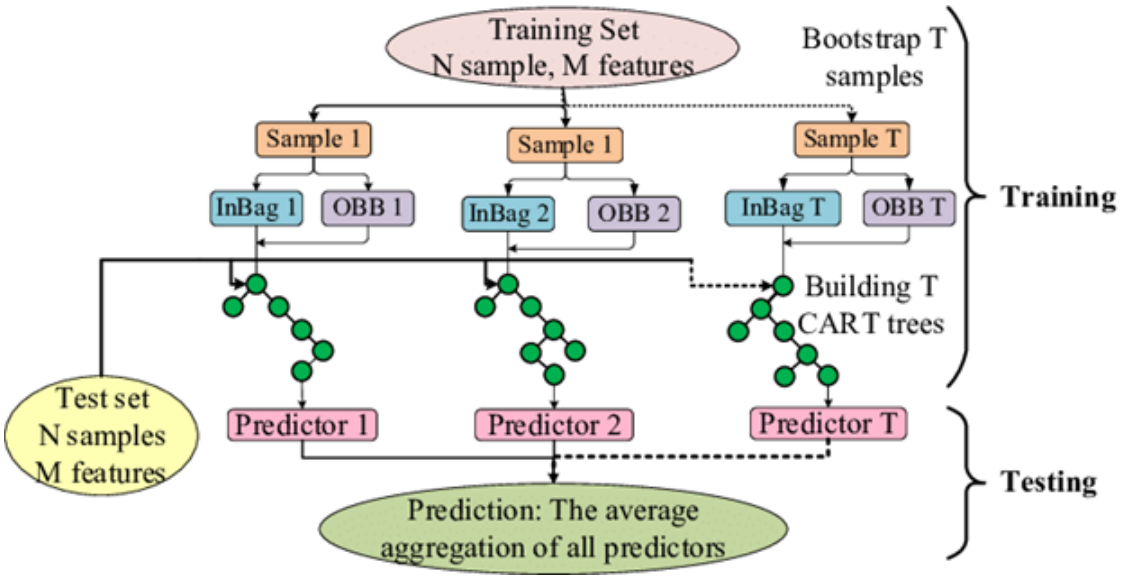


Figure 3.2: Random Forest Propagation

will be able to even make trees more arbitrary by also utilizing random thresholds for each feature instead of looking for the most excellent conceivable thresholds.

3.1.2 Decision Tree

It is a table with tree shaped which connects lines to accessible nodes. Each node is either a branch node taken after with more nodes or as it were one leaf node allotted by classification. With this approach of separating and resolving, decision tree simply separates the complex issue into basic ones and solves back the sub issues through rehashed utilizing, information mining strategy to assist in preparing different sorts of classifying information by constructing decision tree. The method of decision tree model is to develop a decision tree with high accuracy and small scale. There are numerous values of Decision tree method. Firstly, the high adaptability which could be a non-parameter method without any idea for the data distribution. Moreover, it can be clarified, which is additionally the reason of its distinctive utilization. Still, it has drawbacks that, the requirements transaction should be checked separately. 2mm

Terminologies of Decision Tree

- Root Node- This gets divided into two or more homogeneous sets and shows sample.
- Splitting- A node getting divided into two or more sub-nodes.
- Decision Node- When a sub-node divides into more subnodes.
- Leaf/ Terminal Node- Nodes that cannot be divided is called Leaf node.
- Pruning- Removing sub-nodes of a decision node is called pruning.
- Branch / Sub-Tree- A sub section of entire tree is called branch or sub-tree.

- Parent and Child Node- A node, which is divided into sub-nodes is called parent node of sub-nodes and sub-nodes are the child of parent node.
- Entropy- It is the measuring unit of homogeneity of a sample.

$$Entropy = \sum_{i=1}^{k}(P(value_i) \times log_2(P(value_i))) \tag{3.1}$$

- Gain: Gain is the difference of starting entropy (E1) and terminal node entropy (E2).

$$Gain = E1–E2 \tag{3.2}$$

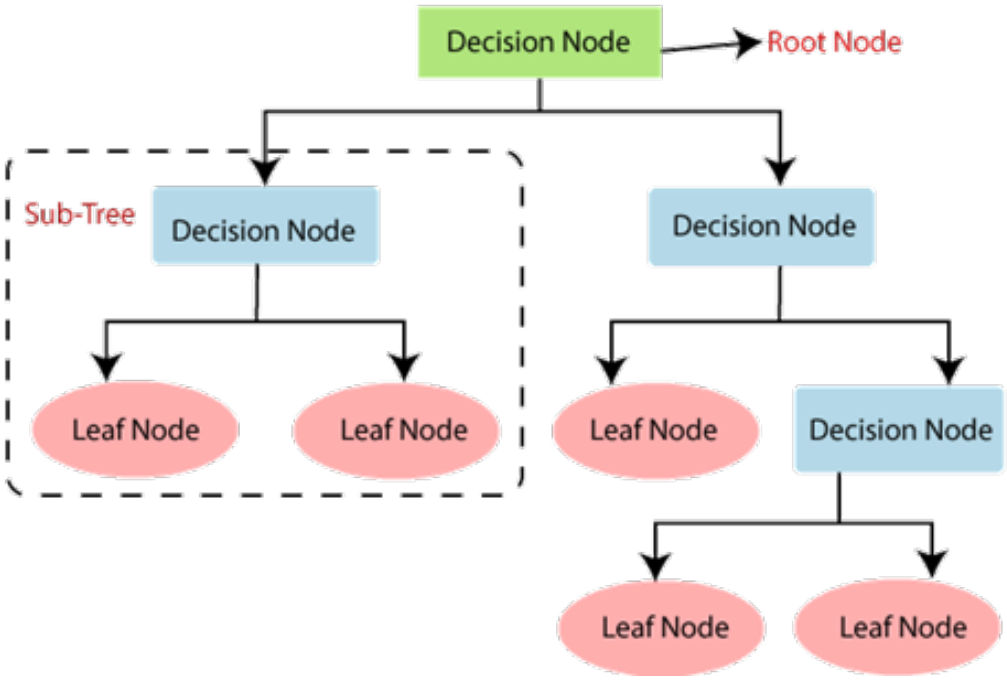


Figure 3.3: Decision Tree Algorithm

3.1.3 Logistic Regression

Logistic Regression is an administered strategy that returns the likelihood of the subordinate variable from the autonomous variable of a dataset. Logistic regression produces logistic curves which plot the values somewhere in the range of a zero, and one. This model works with subordinate variable that is categorical and examines the connection between numerous autonomous variables. Logistic regression provides more reliable results for fairly large samples. The assumptions made by Binary Logistic Regression are given below:

- The dependent variable should have mutually exclusive categories.
- There should be no outliers in the data.
- There should be no high correlations among the predictors

Working Principle

A binary logistic model includes a subordinate variable with two expected values, similar to yes/no, pass/fail, or true/false; these are allocated by a pointer variable with the two values named, "0" and "1." In this model, the log-odd is a linear combination of at least one autonomous variable. The autonomous variable, predictor, can be either binary or constant. The comparing likelihood of the value marked, "1" can shift somewhere in the range of "0" and "1." The logistic regression model is defined as follows:

$$log\frac{p}{1-p} = B_0 + B_1X_1 + B_2X_2 + ... + B_kX_k \tag{3.3}$$

The autonomous variables are denoted as X_1, X_2,...,X_k and p denotes the probability that the subordinate variable has a value of 1. B_0 is a constant and the coefficients of the autonomous variables are B_1,....., B_k.

3.1.4 K-Nearest Neighbour

The nearest neighbor analysis notion has been utilized in a variety of anomaly detection techniques. The k-nearest neighbor algorithm stands out among other classifier algorithms utilized in credit card fraud detection. It is a supervised learning approach in which the majority of the K-nearest neighbor category is used to characterize the outcome of a new occurrence query. Three major factors affect the performance of the KNN algorithm:

- The distance metric is used to find the nearest neighbors.
- The distance rule is used to classify objects based on their K-nearest neighbors.
- The number of neighbors used to divide the new sample into groups.

Among other fraud detection approaches of administered statistical pattern recognition, the K-nearest neighbor rule consistently achieves high performance. KNN is non-parametric, which implies that it makes no suppositions. Instead of anticipating a standard design, the model is entirely formed up of the data provided to it.

Working Principle

Technically, we classify any new transaction by examining k of its -classified, nearest neighbor points and determining the majority votes from nearest neighbors with similar attributes to those used to map the points. The transaction then indicates fraud if the nearest neighbor is fraudulent. The Euclidean distance formula can be used to determine the distance between two data instances from Fig.(3.4).

The k in KNN refers to the number of nearest neighbors taken into account while assigning a label to the current point. A higher k-value is more resilient to exceptions because it analyzes more points while making a prediction.

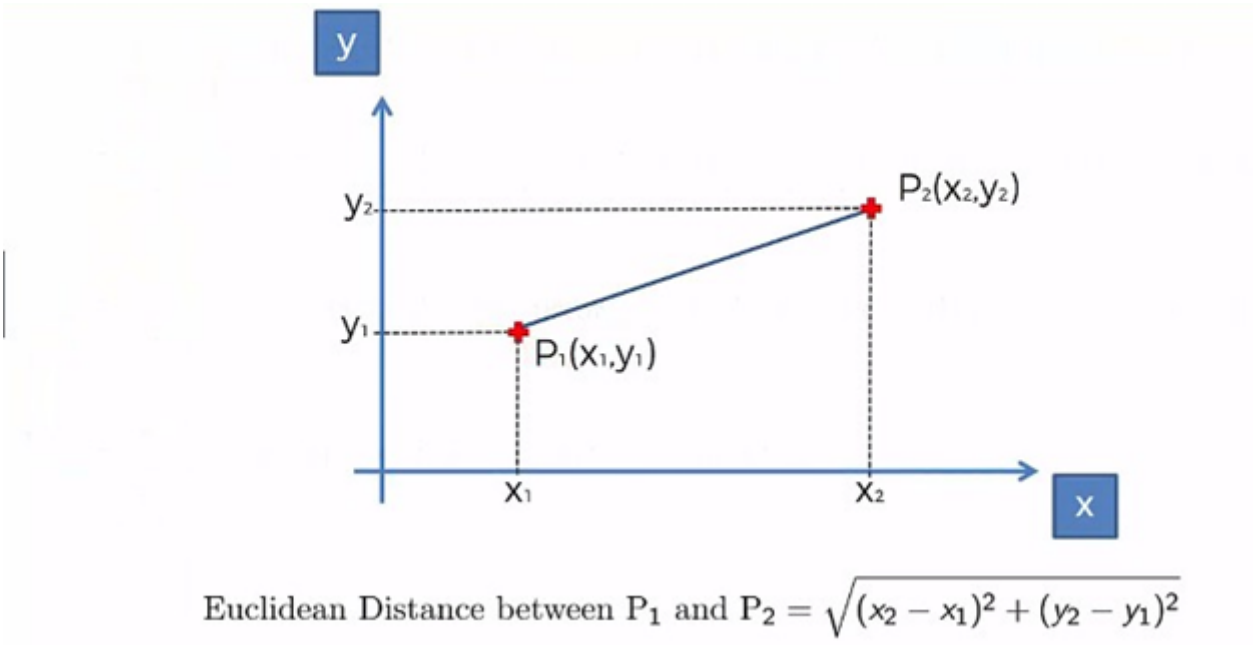


Figure 3.4: Euclidean Distance in KNN

3.1.5 Naive Bayes

Bayesian network classifiers are exceptionally imperative within the region of machine learning and it is under the set of directed categorization models. Naïve Bayes classifier is additionally a prominent Bayesian Network which is based on Bayes theorem of conditional likelihood and thus, may be a classifier based on likelihood which considers Naïve i.e., solid freedom suspicion. The Naïve Bayes machine learning classifier tries to discover a course which is known as result course based on probabilities, additionally conditional probabilities of how numerous times it happened from the preparing information.

This kind of learning is very proficient, quick and tall in exactness for real-world scenarios, and is known as directed learning. Moreover, usually exceedingly profitable since it calculates the parameters by utilizing exceptionally little preparing information which is utilized for classification and is based upon word autonomy. Naïve Bayes is basic to actualize and get it and employments solid presumptions. It gives exact results additionally Naïve Bayes works viably in numerous ranges related to machine learning.

In Naïve Bayes, we compute the probabilities of particular categories given a content is performed by the utilize of joint probabilities of categories and words. Within the probability-based classification strategy, the probabilities of the target classes are calculated together with likelihood of the test information. The naïve Bayes classification approach will move forward the precision of credit card extortion discovery. The proposed technique is done in python and results will be decided in terms of precision, accuracy, recall and F- measure.

Implementing Naive Bayes

The Naïve Bayes machine learning classifier predicts a class known as end result lesson which is based on probabilities, additionally conditional probabilities of its occurrence from the preparing information. This sort of research is exceptionally valuable, quick and good in precision for real-life frameworks.

The first step for Naïve Bayes classification calculation is the Bayes hypothesis for conditional probability, where 'x' is given information point and 'C' may be a course:

$$P(C/x) = P(x/C)/P(x) \tag{3.4}$$

And other steps are done by making the assumption for a data point x = x1 to xj, and the event probability of each of its attribute inside given class is independent. So the probability of x can be in this way: Posterior probability of target class c P(c—X) is calculated from P(c), P(X—c) and P(X).Apparently naïve Bayes classifiers have worked a part in numerous circumstances, most imperatively document classification and spam sifting. They require training data to get the essential parameters. It can be superfast compared to other strategies. Here each distribution can be evaluated as dimensional free distribution.

3.1.6 Support Vector Machine

The Support Vector Machine was first presented by Vapnik (1995). SVM is a supervised training algorithm that can detect subtle patterns in large datasets. This statistical method of learning can be used to evaluate data for classification as well as regression. SVM modeling includes two stages, initially for the training of data set and plotting a model, and then the model is utilized to predict information from a test data set.

Working Principle

An SVM model is essentially an illustration of various categories in a multidimensional hyperplane. The hyperplane will be constructed iteratively to minimize the error. The principal objective of SVM is to classify datasets in order to determine the maximum marginal hyperplane.

Support vectors- Data points that are adjacent to the hyperplane are referred to as support vectors. These data points influence the location and orientation of the hyperplane and maximize the classifier range.

Hyperplane- It is a decision plane that is divided into separate classes by a set of points. The magnitudes of the hyperplane depend on the features present in the datasets. We need to choose the hyperplane with the highest margin to avoid the chance of a misclassification.

Margin- Margin is defined as the distance between two lines on the nearest data points of distinct classes. It is computed as the perpendicular distance between the line and the support vectors.

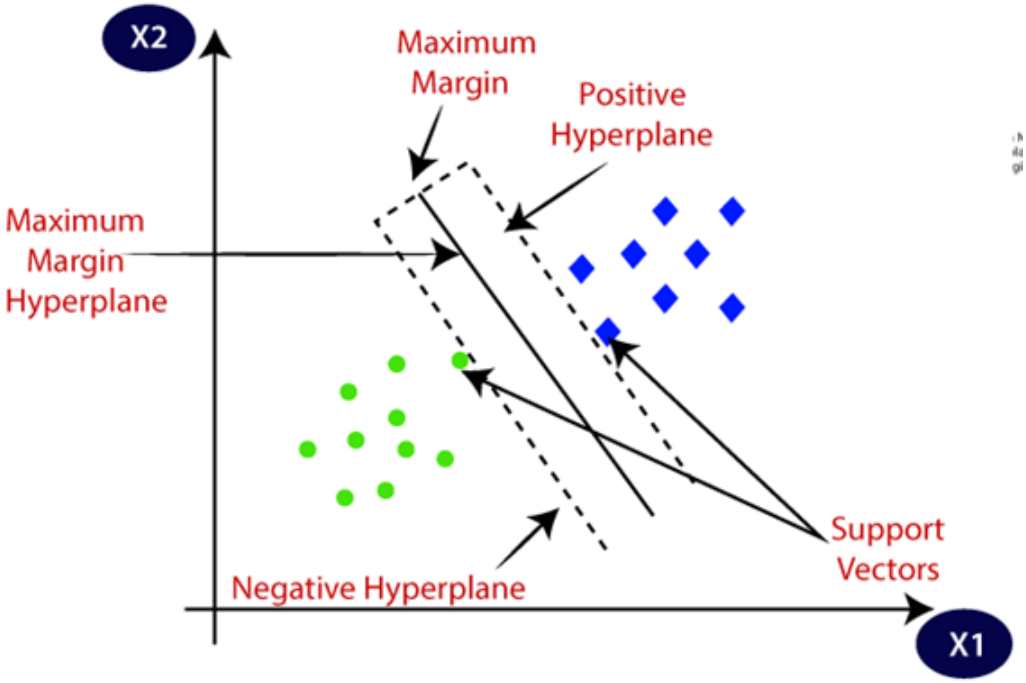


Figure 3.5: Support Vector Machine

Implementation of SVM in Python

In practice, the SVM algorithm is carried out with the help of a kernel. A low- dimensional input space is converted into a higher-dimensional space using the kernel. It creates SVM more effective, adaptable, and precise. The categories of kernels used by SVM are presented below: Linear Kernel:

It can be calculated by taking the dot product of any two observations. The linear kernel formula is as follows:

$$K(x, x_i) = \sum(x \times x_i) \tag{3.5}$$

Polynomial Kernel: It can figure out if the input space is curved or nonlinear. Following is the formula for polynomial kernel:

$$K(x, x_i) = 1 + \sum(x \times x_i \times d) \tag{3.6}$$

Here, the degree of polynomial is d, in this case Radial Basis Function (RBF) Kernel: It represents an input space in an infinite-dimensional space. The formula below explains it mathematically:

$$K(x, x_i) = \sum(x \times x_i^2) \tag{3.7}$$

Chapter 4

Dataset & Experimentation

4.1 Dataset

We know that most machine learning techniques can handle mixed datasets. As our dataset has dealt with both qualitative and quantitative data thus our dataset is also a mixed method which we are following to build a system that can do the work accurately. Our simulated dataset contains both lawful and deceitful transactions that took place in between 1st January 2019- 31st December 2020, which can counterbalance credit cards of one thousand customers doing their transaction with a pool of eight hundred merchants. The whole dataset contained 1,048,575 transactions of which 6,006 were actual fraudulent, 22 columns with relevant information of the transactions as well as the customers [35].

For our simulation, we shaped and pruned the dataset by removing the irrelevant features and at the same time converted some important features into more narrowed down ones. As such, we ended up with 16 essential and highly relevant features. Fig.(4.1) & Fig.(4.2) depicts the graphical representation of the features.

4.2 Data Manipulation

We use machine learning to solve well-structured tasks like as classification and clustering. We need to keep in mind that a machine learning algorithm learns from structured training data during establishment. Even it is also learning endlessly from physical world data during formation so that the algorithm can improve its model with both dataset and real-life experiences.

The criteria is to train the dataset with numeral examples that we had in the dataset. In case of every algorithm we used, the train and test score depicts the fitness and rigorousness of the methods used. Our whole process of data processing, modification and experimentation contains the following steps:

Data Noise Reduction: The process of reducing irrelevant and unnecessary data is called data noise reduction. The dataset we collected from Github initially had 22 columns or features that described the name, location, timestamp, transaction numbers, credit card information, latitude, longitudes of the customers and transactions. But in order to train the machine learning models, we did not need all this information as such the unnecessary features were considered as data noise which

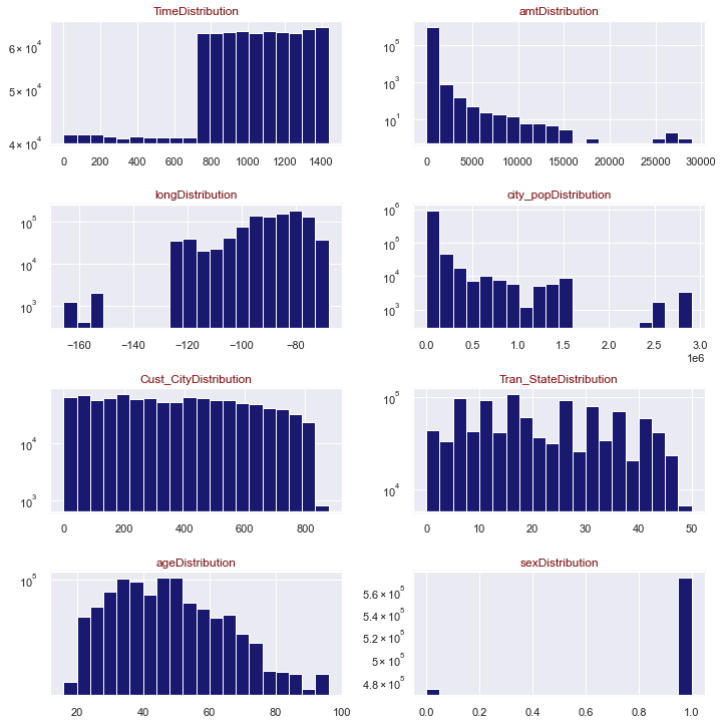


Figure 4.1: Modified Dataset Description I

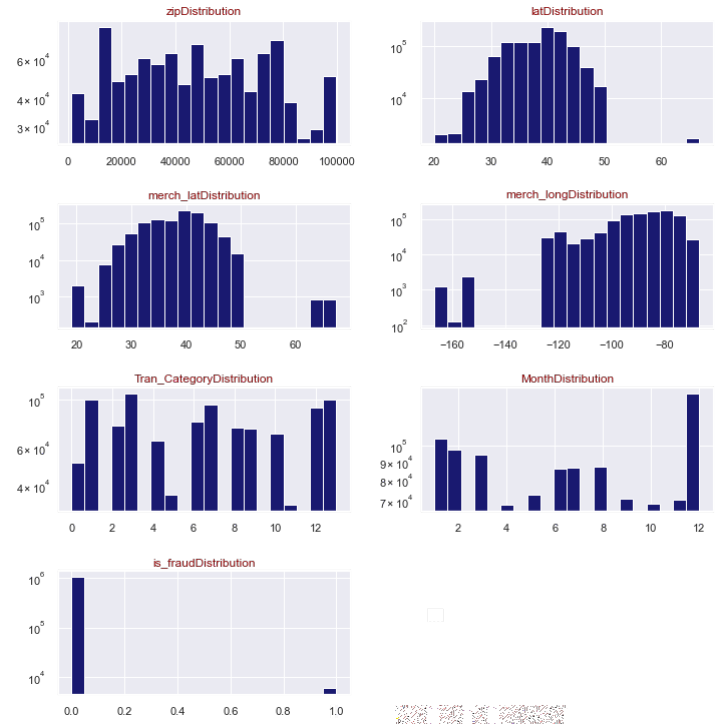


Figure 4.2: Modified Dataset Description II

we had to prune.

Data division into label and feature: In order to instruct the model algorithms, we divided the columns into two categories i.e., label and feature. Features are the input columns of the dataset while label is the output column that we are trying to eventually predict. In our dataset, the column 'is_fraud' is the label while all the other columns are features.

Dividing dataset for model training: After we pruned and shaped the dataset, we divided the whole dataset i.e., 1,048,575 transaction records into two groups viz- the train set and the test set. Our training set contained 70% i.e., 734,002 records. The machine used this data to train different machine learning algorithms in order to prepare itself for further fraud detection test.

Dividing dataset for model testing: The testing set contained the other 30% i.e., 314,573 transaction records for the machine learning algorithms to test and check how good they can detect frauds. The test score of varies from algorithm to algorithms since some are more efficient than the other.

Data conversion and mapping: After successful training and testing, we decided to determine the correlation of different features with the fraud transactions. And to do so, we had to calculate some vital features from existing irrelevant features like 'age of the customers' from 'Date of Birth' and 'month of the transactions' from 'timestamp'. This method us called data conversion while assigning numerical values to salient and relevant non-numerical features is called data mapping.

Correlation generation: After we converted and mapped all the relevant data, the dataset was once again ready for correlation generation. We generated the correlation of all the important features against the transaction types.

4.3 Dataset Processing

Initially the dataset contained 23 columns out of which 22 denoted relevant & salient features of the transactions while the last column was the class column that represented whether the transactions were fraud or legitimate. This column is called label or target variable.

First, we extracted some very important but missing features from the existent ones such as, 'age' from 'DOB', 'month' from 'trans_date_trans_time' as well as mapped some existing features to numbers for the machine readability. We dropped the irrelevant columns (features) such as 'transaction id', 'cc num', 'name of the cardholder' and such. We used 14 features of the dataset. Secondly, we divided the dataset into X & Y representing 'Features' & 'Label' respectively. The whole dataset was then divided into training (70%) and testing (30%) sets maintaining an equal distribution of the classes between the datasets.

4.4 Features of the dataset

Extracting the most salient & relevant features of a transaction is a very important task in credit card fraud detection. The dataset used to train the models may con-

tain many unnecessary features like 'Customer Name', 'Card Number', 'Transaction Number', 'Date of Birth' and more that does not have any adherence with the fraudulence. Since the owner of the card or the card itself has nothing to do with the fraud, any features indicating the owner's personal credentials should be ignored. Rather the features relating to the transactions, customers and merchants should be used. Table (4.1) contains detailed information of the features we converted and extracted for model training and validation purposes.

Table 4.1: Salient Features with descriptions

Sl.	Features	Descriptions	Status
1.	amt	Amount of Transaction	Given
2.	zip	Zip Code of Customer	Given
3.	lat	Latitude of Customer	Given
4.	long	Longitude of Customer	Given
5.	merch_lat	Latitude of Merchant	Given
6.	merch_long	Longitude of Merchant	Given
7.	city_pop	Population of the Transaction City	Given
8.	Time	Time when the transaction took place	Converted
9.	Tran_Category	Type of the purchased product	Converted
10.	Cust_City	City of Customer	Converted
11.	Tran_State	State where transaction occurred	Converted
12.	Month	Month of Transaction	Extracted
13.	age	Age of Customer	Extracted
14.	sex	Sex of Customer	Converted

4.5 Simulator

We conducted our experiment and machine learning algorithms in a Python based Integrated Development Environment (IDE)- Jupyter Notebook. Our used programming language was python 3. With all required libraries like Numpy, Pandas, Matplotlib, Seaborn and more we finished our experimentation on the dataset and presented our results in Chapter 5.

4.6 Data Analysis Metrics

To evaluate the data we got from the simulations, we calculated multiple evaluation metrics like precision, recall, F1-Score, accuracy, specificity etc. All these metrics give us the appropriate comparison of our simulated data.

4.6.1 Confusion Matrix

Confusion Matrix or error matrix is a graphical representation or layout of the performance of an algorithm. It is mostly used in machine learning fields as well as in problems relating to statistical classification. In a confusion matrix that is

generated as the result of machine learning algorithm, there are these following factors that play great roles:

- True Positive (TP): When the model predicts the value to be positive and that prediction turns out to be true, it is considered as true positive.
- True Negative (TN): When the model predicts the value to be negative and that prediction turns out to be true, it is considered as true negative.
- False Positive (FP): When the model predicts the value to be positive, but that prediction turns out to be false, it is considered as false positive.
- False Negative (FN): When the model predicts the value to be negative and that prediction turns out to be false, it is considered as false negative.

4.6.2 Precision, Recall & Accuracy

In case of a single algorithm, we determined the training score and the testing score along with the metrics- precision, recall and accuracy of the individual models. Precision, recall and accuracy are the key metrics of any pattern recognition technique that helps to detect a specific pattern in a given set of data.

- Precision: In pattern recognition and machine learning classification, precision is the fraction of relevant instances among the retrieved instances. It is also called positive predictive value. Mathematically precision can be denoted as:

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

- Recall: Much like precision, recall is also a vital factor in pattern recognition, classification and information retrieval. It is the fraction of relevant retrieved instances. Mathematically it can be denoted as:

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

- Accuracy: Accuracy is the best metric of comparing the achieved results from a model simulation. It is the ratio of the total number of correct predictions and the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{4.3}$$

- Specificity: It refers to the ratio of true negative to the summation of true negative and false positive. The model is as follows:

$$Specificity = \frac{TN}{TN + FP} \tag{4.4}$$

- Receiver Operating Characteristic (ROC) Curve: This is a curve that shows the performance of a classification model at all possible thresholds. It uses two parameters viz. True Positive Rate (sensitivity) and False Positive Rate (specificity).

Chapter 5

Experimental Result & Result Analysis

We have used our dataset of 10,48,575 transactions to evaluate the models and found significant results. But this dataset is highly imbalanced because there are 6006 fraud transactions and 10,42,569 non-fraud transactions. So, to solve this issue, we used a technique of using equal or almost equal amount of random non-fraud transactions. Since there are 6006 fraud transactions, we took 6006, 10000, 15000, 20000, 25000, 30000, 35000, 40000, 45000 and 50000 non-fraud transactions and compared the metrics of all the models to find the best solution.

5.1 Full Dataset

Our dataset has dealt with both qualitative and quantitative data thus our dataset is also a mixed method which we are following to build a system that can do the work accurately. The full dataset consists 1,048,575 transactions of which 6,006 were actual fraudulent & 1,042,569.

Table 5.1: Old Dataset Description

Category	Values
Fraud Transactions	6006
Non-Fraud Transactions	1042569
Train set	734003
Test set	314572
Total Transactions	1048575

We divided the dataset into train and test sets for our model simulations. As supervised machine learning models first train themselves with a portion of the dataset, we used 70% of our dataset i.e., 734,003 transactions as our train set and the remaining 314,572 transactions as test set.

We have applied accuracy, specificity and ROC metrics to evaluate the performance of the models with the full dataset. In Fig.(5.1) & Fig.(5.2) we can observe that the findings are over-fitting and excessively accurate, implying that an imbalanced dataset cannot provide accurate results. Table (5.2) contains the performance metrics value of the old dataset model simulations.

Table 5.2: Performance Metrics of Old Dataset

Models	Accuracy	Precision	Recall	Specificity
Decision Tree	99.71	89.48	56.16	99.96
KNN	99.54	79.68	39.18	99.94
Random Forest	99.54	100	20.26	100
SVM	99.42	18.75	0.17	99.99
LR	99.36	0.44	0.06	99.93
Naive Bayes	99.15	33.69	49.17	99.44

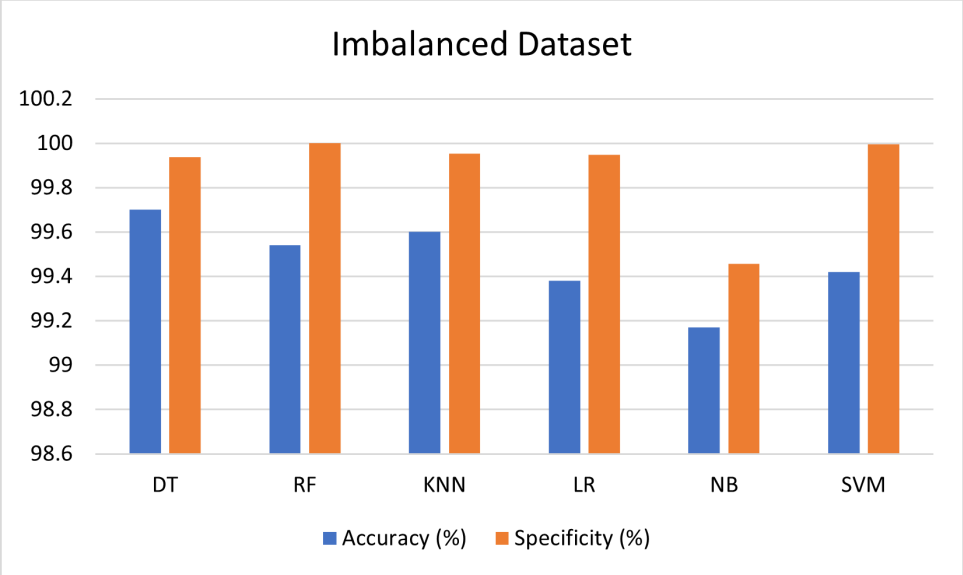


Figure 5.1: Accuracy & Specificity of models with full dataset

5.2 Balanced Dataset

In order to balance our dataset, we have divided our dataset into 10 new divisions. In all these 10 sets, we have taken 6006 fraud transactions but in case of the non-fraud transactions, we have taken 6006, 10000, 15000, 20000, 25000, 30000, 35000, 45000 & 50000 non-frauds and created balanced sets named as Balanced-A (6006 frauds, 6006 non-frauds), Balanced-B (6006 frauds, 10000 non-frauds), Balanced-C (6006 frauds, 15000 non-frauds), Balanced-D (6006 frauds, 20000 non-frauds), Balanced-E (6006 frauds, 25000 non-frauds), Balanced-F (6006 frauds, 30000 non-frauds), Balanced-G (6006 frauds, 35000 non-frauds), Balanced-H (6006 frauds, 40000 non-frauds), Balanced-I (6006 frauds, 45000 non-frauds) & Balanced-J (6006 frauds, 50000 non-frauds).

We have used precision, recall, accuracy and specificity metrics to evaluate the models and their performances. Since our dataset was highly imbalanced, we tried applying a different approach of taking equal and close to equal amounts of non-fraud transactions. We have also mapped the accuracy and specificity model wise so that we can sort the best performing models easily.

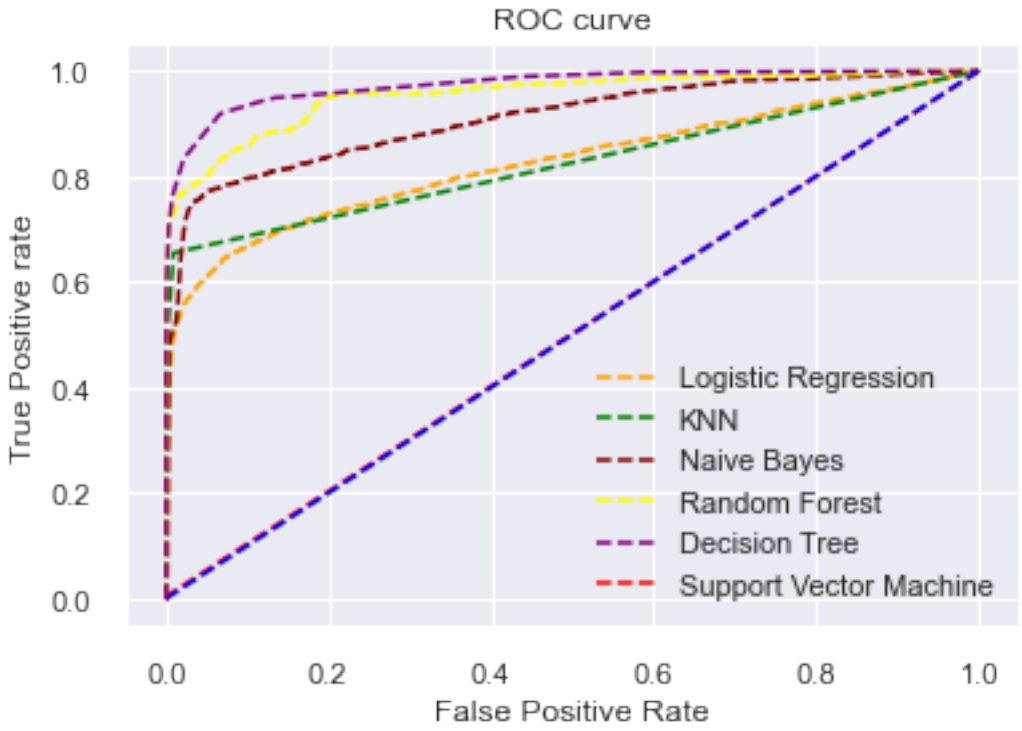


Figure 5.2: ROC curve of model simulations with full dataset

5.2.1 Balanced-A: Non-Fraud 6006, Fraud 6006

In Balanced-A set, we have taken 6006 fraud and 6006 non-fraud transactions which resulted in 12012 total transactions, 8409 transactions in train set and 3603 transactions in test set. Fig.(5.3) & Fig.(5.4) depict the values of accuracy, specificity and ROC of Balanced-A respectively.

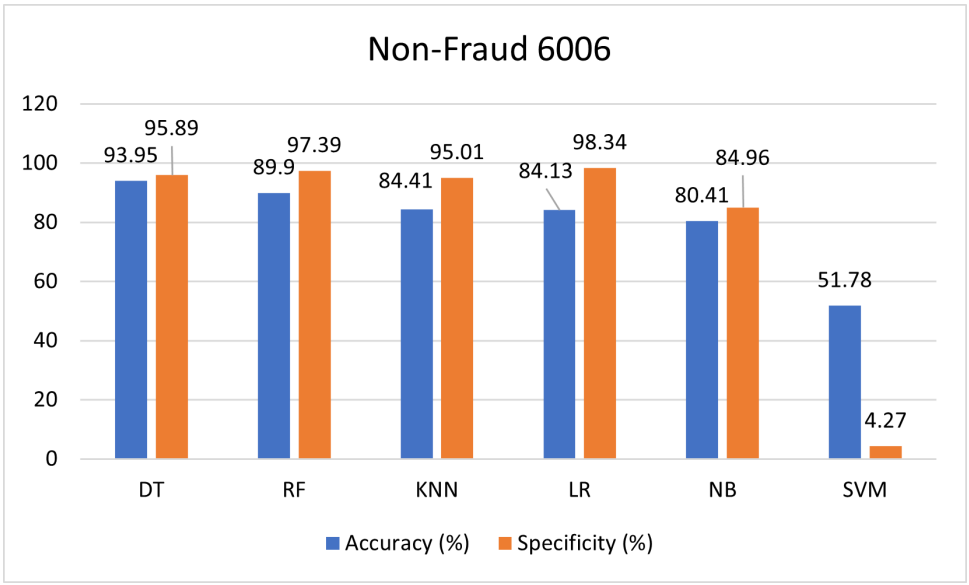


Figure 5.3: Accuracy & Specificity of models in Balanced-A set

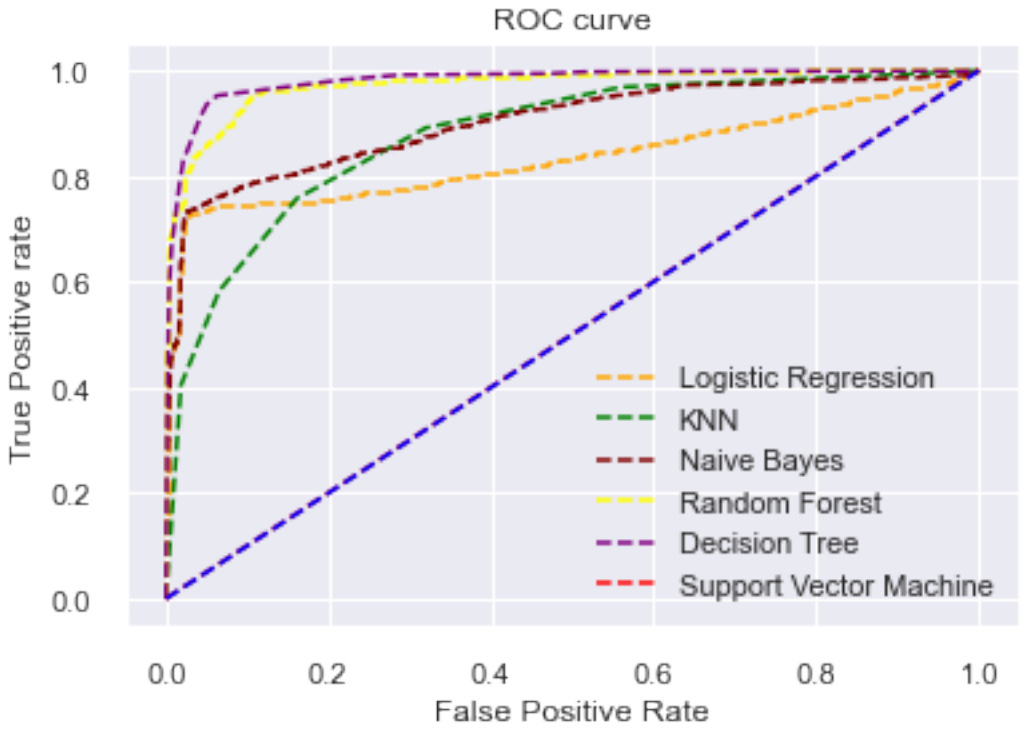


Figure 5.4: ROC curve of model simulations with Balanced-A set

5.2.2 Balanced-B: Non-Fraud 10000, Fraud 6006

In Balanced-B set, we have taken 6006 fraud and 10000 non-fraud transactions which resulted in 16006 total transactions, 11204 transactions in train set and 4802 transactions in test set. Fig.(5.5) & Fig.(5.6) depict the values of accuracy, specificity and ROC of Balanced-B respectively.

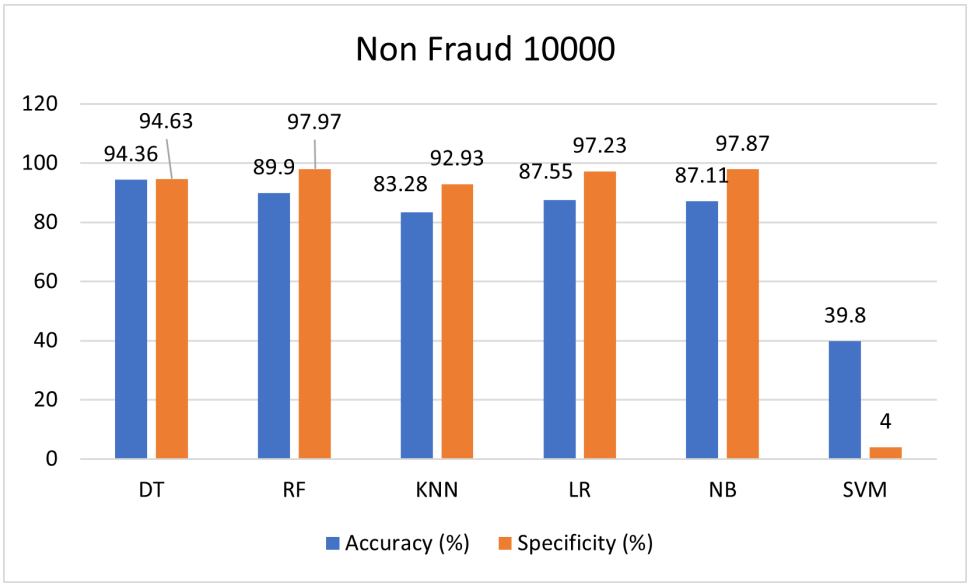


Figure 5.5: Accuracy & Specificity of models in Balanced-B set

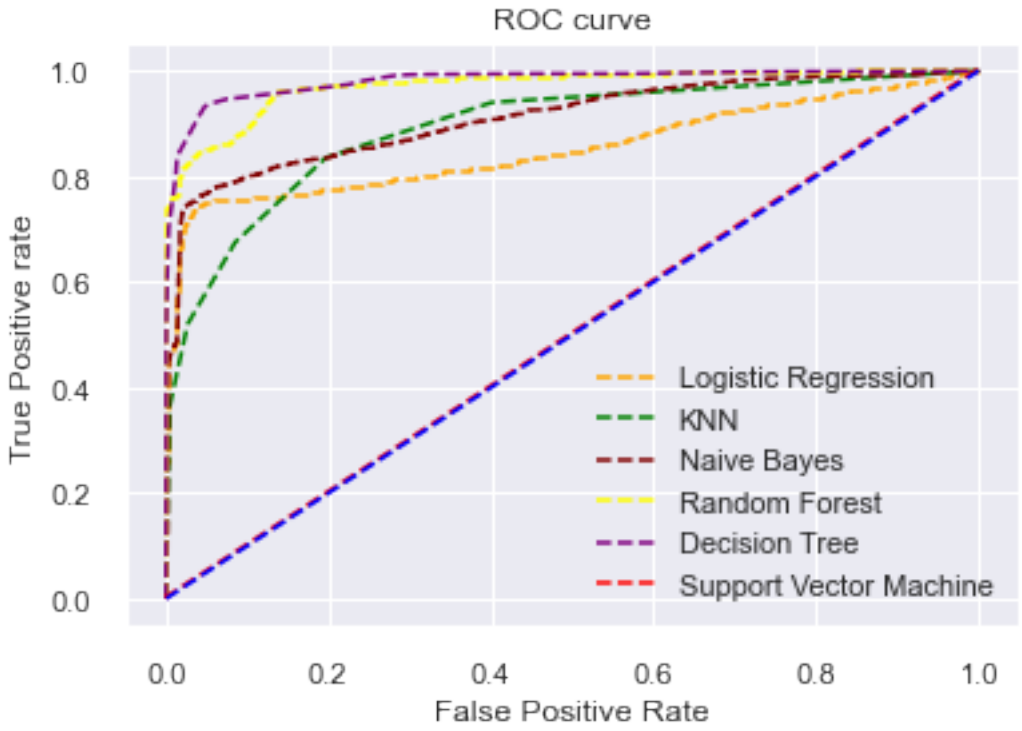


Figure 5.6: ROC curve of model simulations with Balanced-B set

5.2.3 Balanced-C: Non-Fraud 15000, Fraud 6006

In Balanced-C set, we have taken 6006 fraud and 15000 non-fraud transactions which resulted in 21006 total transactions, 14704 transactions in train set and 6302 transactions in test set. Fig.(5.7) & Fig.(5.8) depict the values of accuracy, specificity and ROC of Balanced-C respectively.

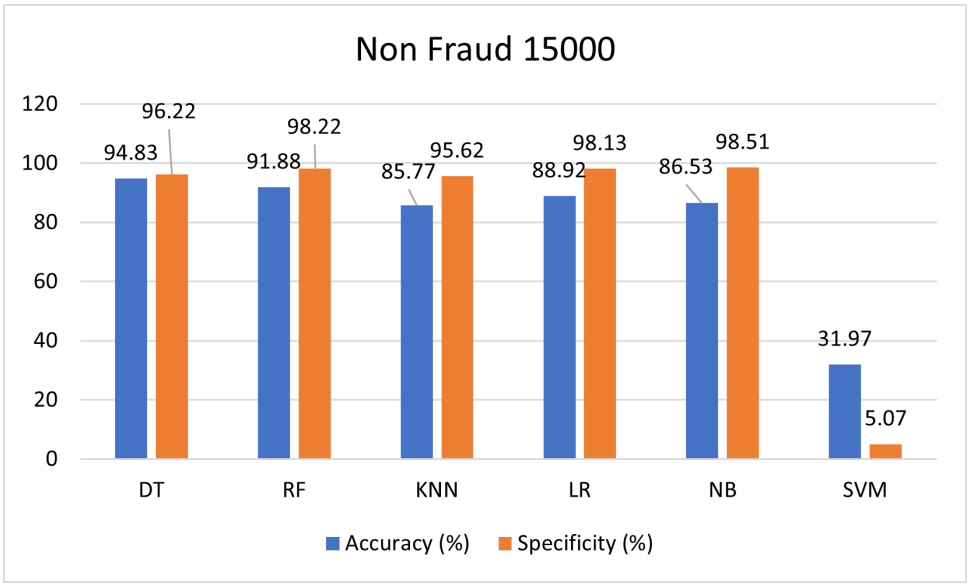


Figure 5.7: Accuracy & Specificity of models in Balanced-C set

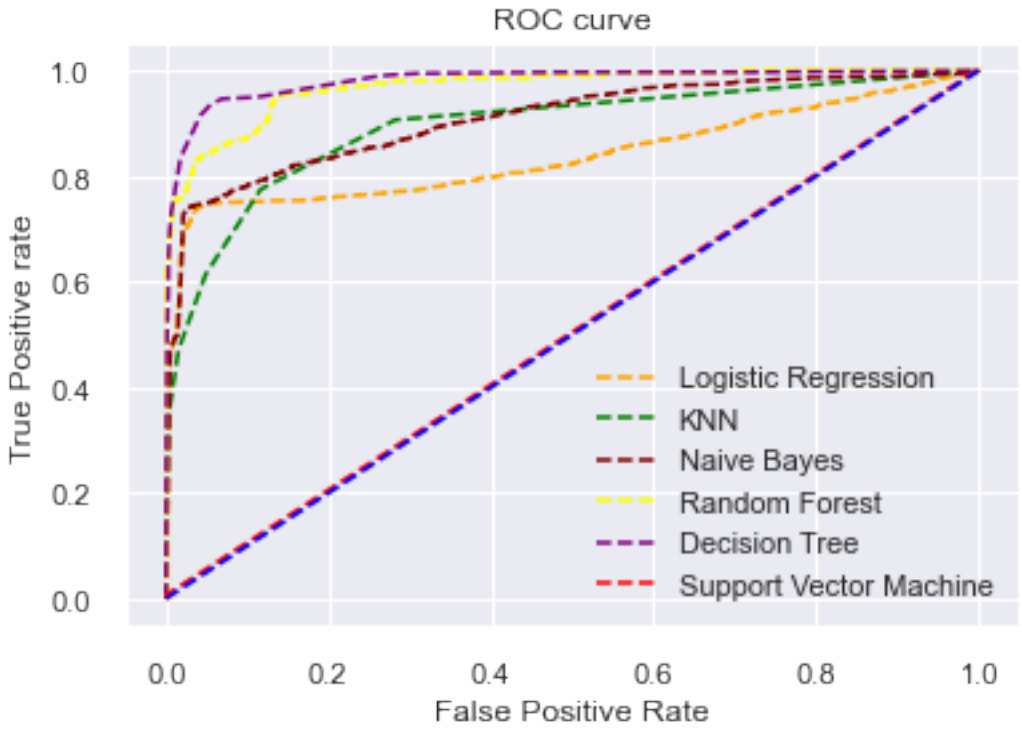


Figure 5.8: ROC curve of model simulations with Balanced-C set

5.2.4 Balanced-D: Non-Fraud 20000, Fraud 6006

In Balanced-D set, we have taken 6006 fraud and 20000 non-fraud transactions which resulted in 26006 total transactions, 18204 transactions in train set and 7802 transactions in test set. Fig.(5.9) & Fig.(5.10) depict the values of accuracy, specificity and ROC of Balanced-D respectively.

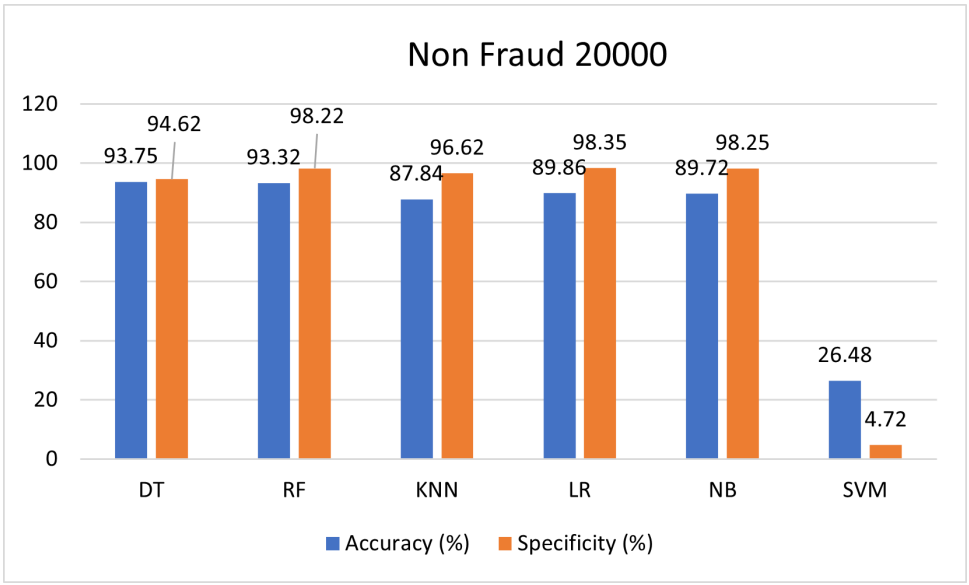


Figure 5.9: Accuracy & Specificity of models in Balanced-D set

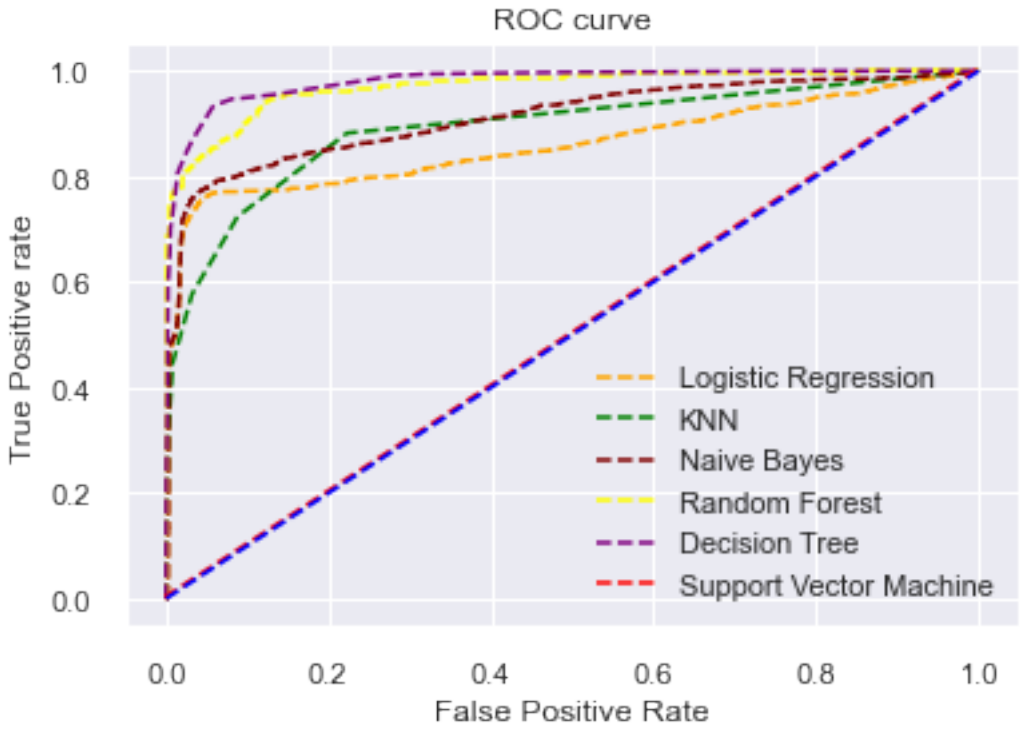


Figure 5.10: ROC curve of model simulations with Balanced-D set

5.2.5 Balanced-E: Non-Fraud 25000, Fraud 6006

In Balanced-E set, we have taken 6006 fraud and 25000 non-fraud transactions which resulted in 31006 total transactions, 21704 transactions in train set and 9302 transactions in test set. Fig.(5.11) & Fig.(5.12) depict the values of accuracy, specificity and ROC of Balanced-E respectively.

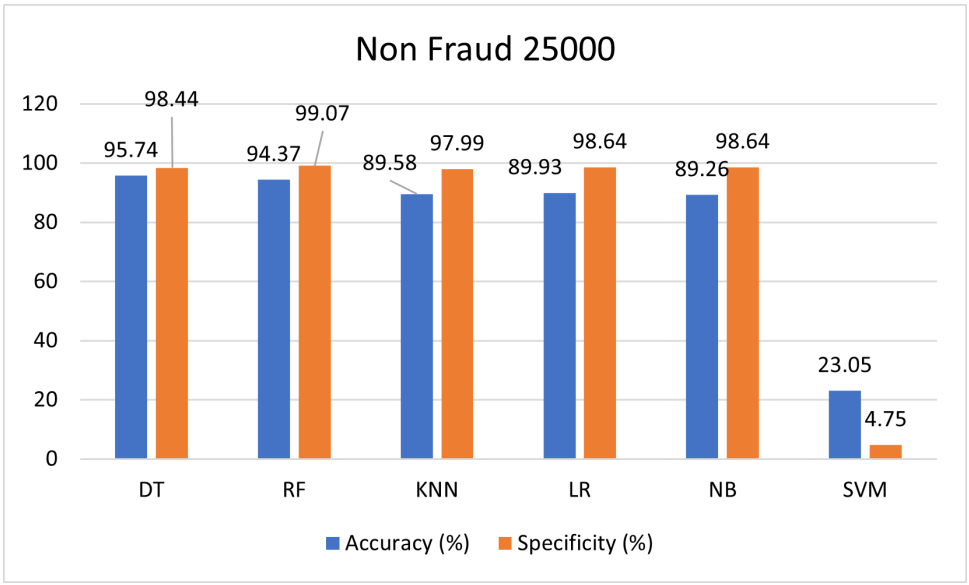


Figure 5.11: Accuracy & Specificity of models in Balanced-E set

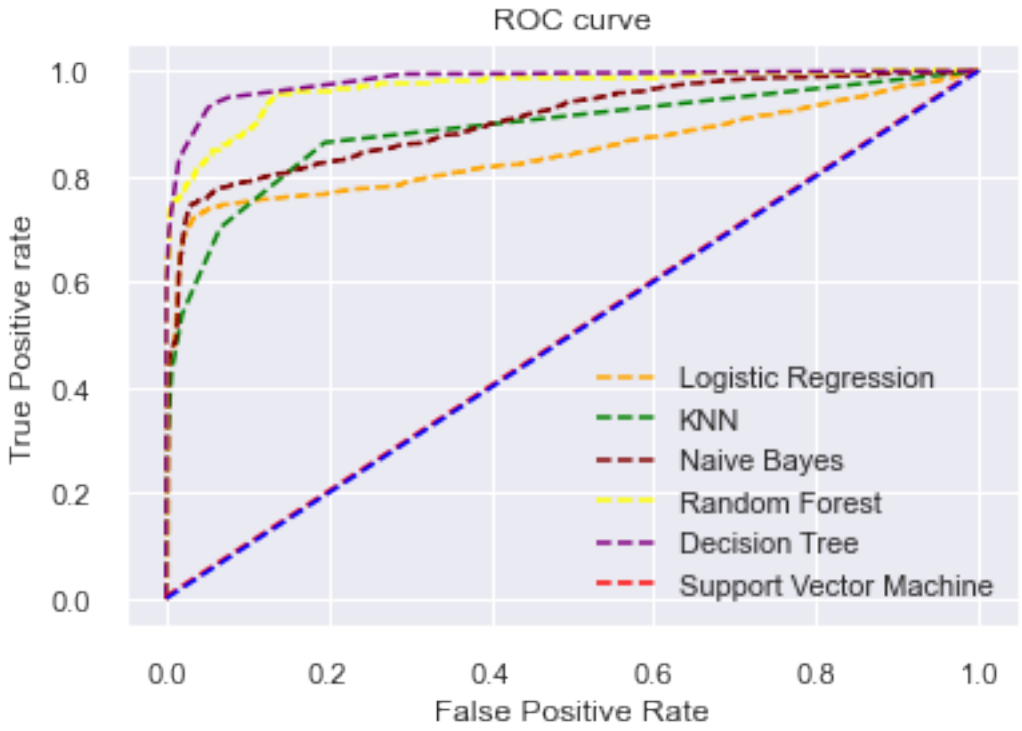


Figure 5.12: ROC curve of model simulations with Balanced-E set

5.2.6 Balanced-F: Non-Fraud 30000, Fraud 6006

In Balanced-F set, we have taken 6006 fraud and 10000 non-fraud transactions which resulted in 36006 total transactions, 25204 transactions in train set and 10802 transactions in test set. Fig.(5.13) & Fig.(5.14) depict the values of accuracy, specificity and ROC of Balanced-F respectively.

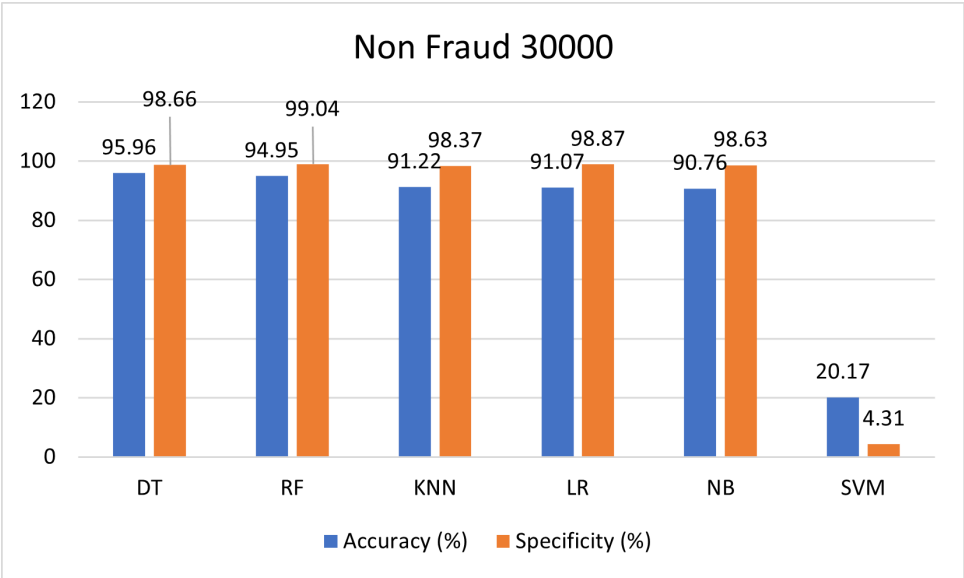


Figure 5.13: Accuracy & Specificity of models in Balanced-F set

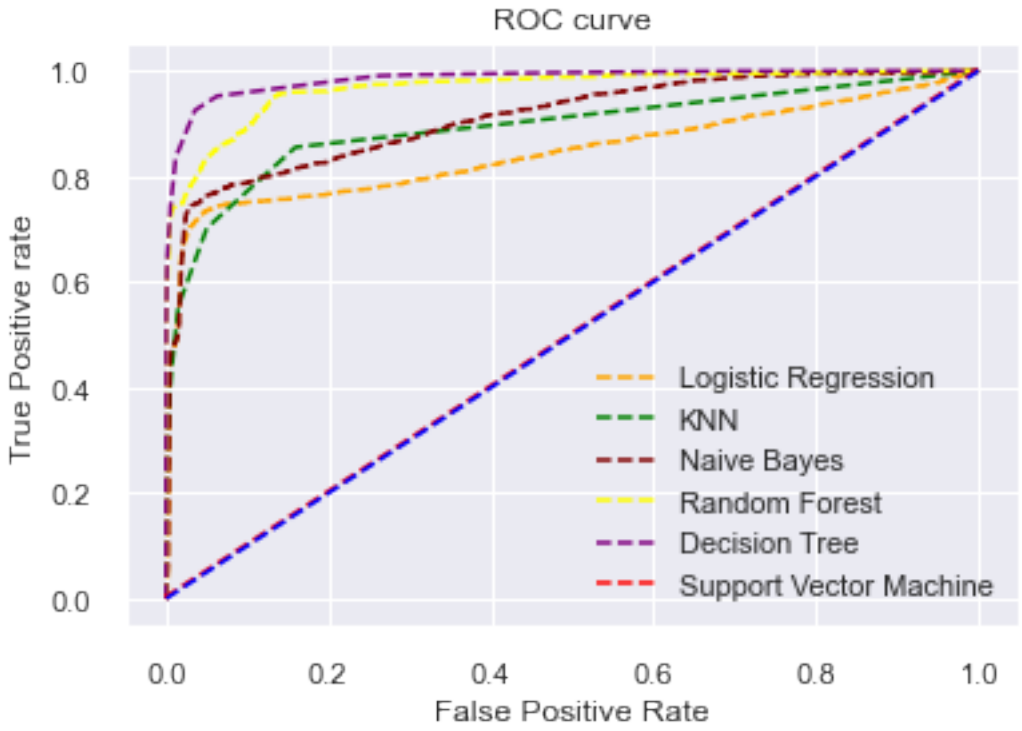


Figure 5.14: ROC curve of model simulations with Balanced-F set

5.2.7 Balanced-G: Non-Fraud 35000, Fraud 6006

In Balanced-G set, we have taken 6006 fraud and 35000 non-fraud transactions which resulted in 41006 total transactions, 28704 transactions in train set and 12302 transactions in test set. Fig.(5.15) & Fig.(5.16) depict the values of accuracy, specificity and ROC of Balanced-G respectively.

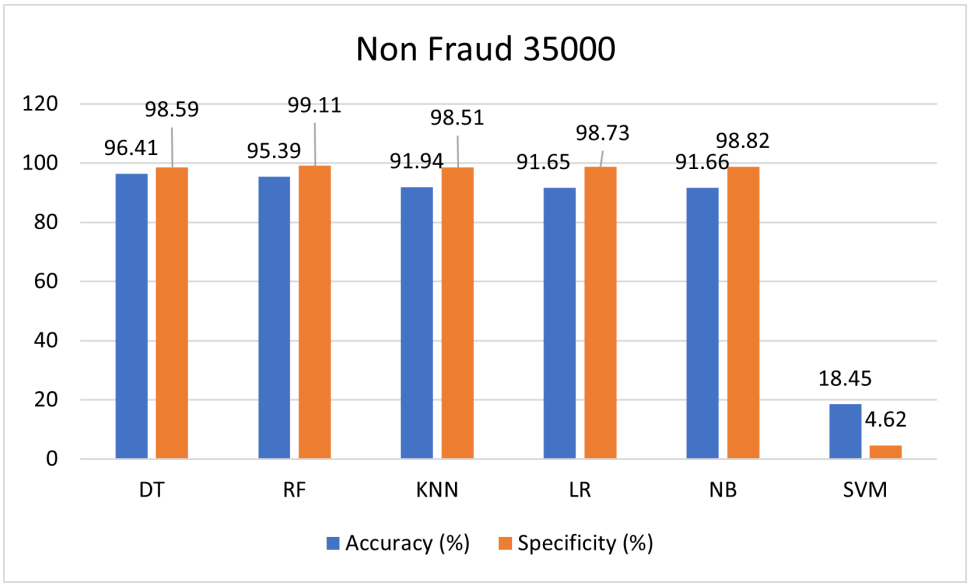


Figure 5.15: Accuracy & Specificity of models in Balanced-G set

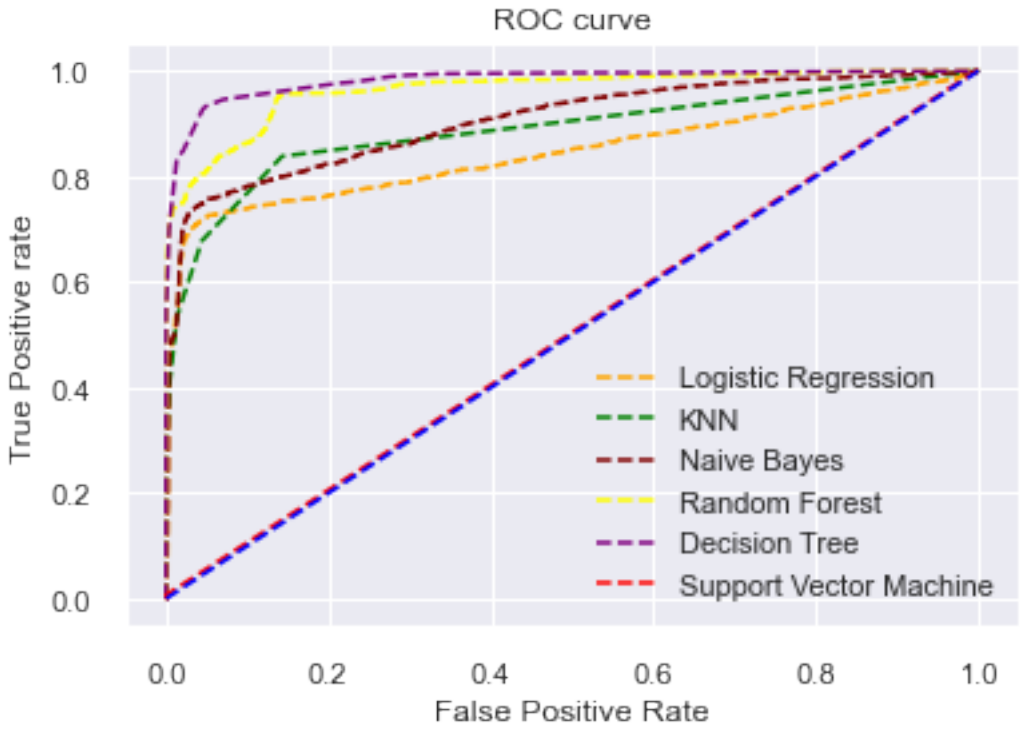


Figure 5.16: ROC curve of model simulations with Balanced-G set

5.2.8 Balanced-H: Non-Fraud 40000, Fraud 6006

In Balanced-H set, we have taken 6006 fraud and 40000 non-fraud transactions which resulted in 46006 total transactions, 32204 transactions in train set and 13802 transactions in test set. Fig.(5.17) & Fig.(5.18) depict the values of accuracy, specificity and ROC of Balanced-H respectively.

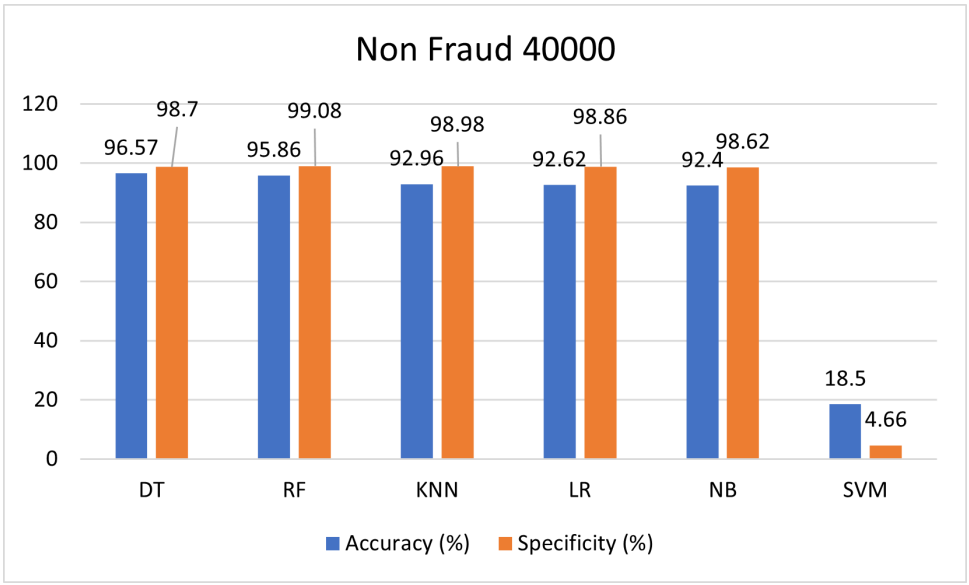


Figure 5.17: Accuracy & Specificity of models in Balanced-H set

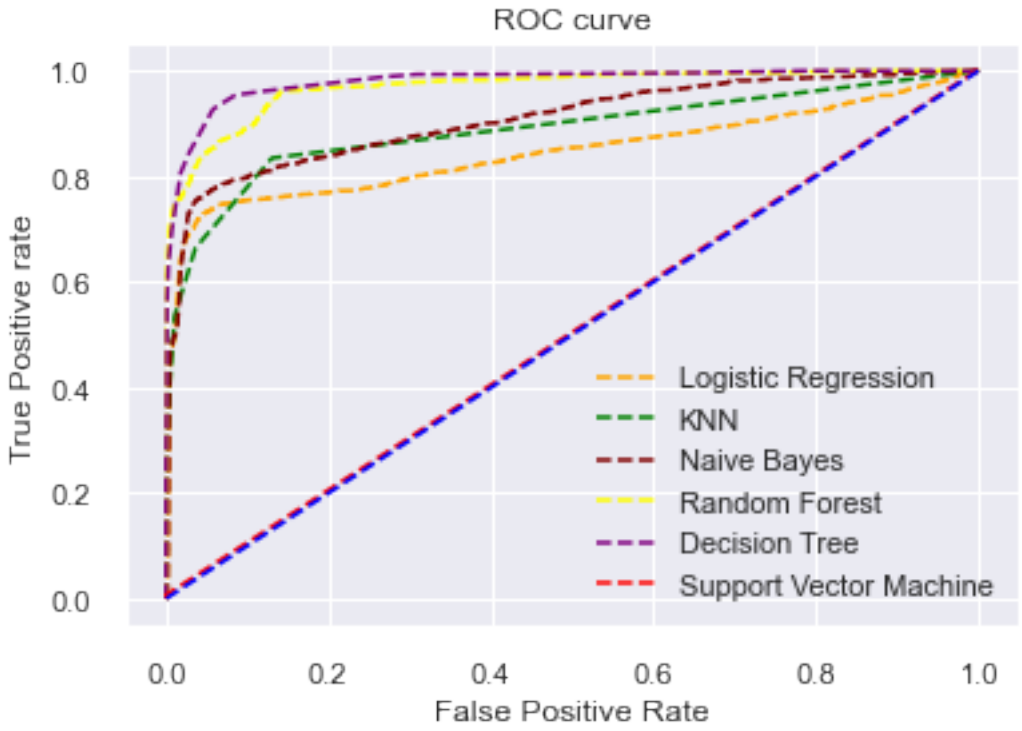


Figure 5.18: ROC curve of model simulations with Balanced-H set

5.2.9 Balanced-I: Non-Fraud 45000, Fraud 6006

In Balanced-I set, we have taken 6006 fraud and 45000 non-fraud transactions which resulted in 51006 total transactions, 35704 transactions in train set and 15302 transactions in test set. Fig.(5.19) & Fig.(5.20) depict the values of accuracy, specificity and ROC of Balanced-I respectively.

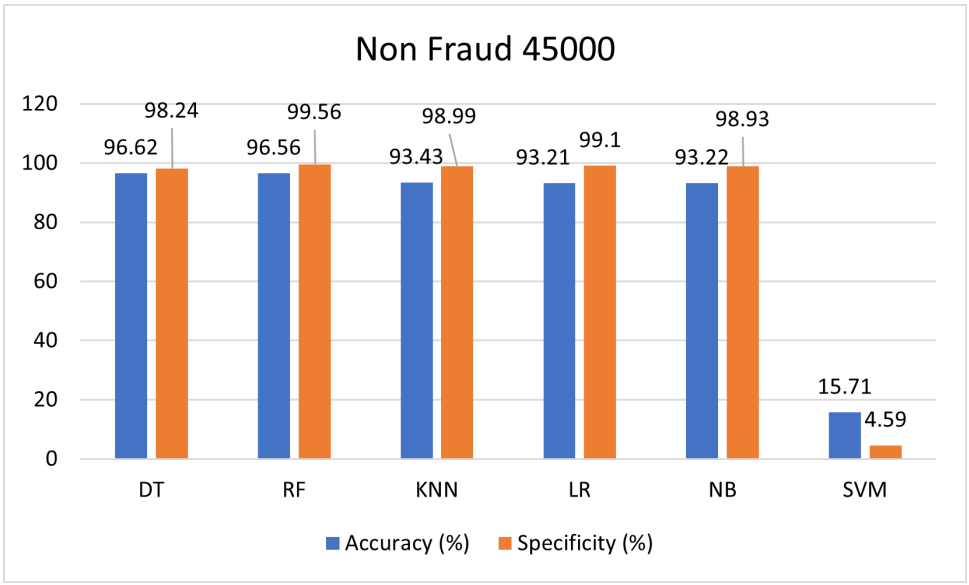


Figure 5.19: Accuracy & Specificity of models in Balanced-I set

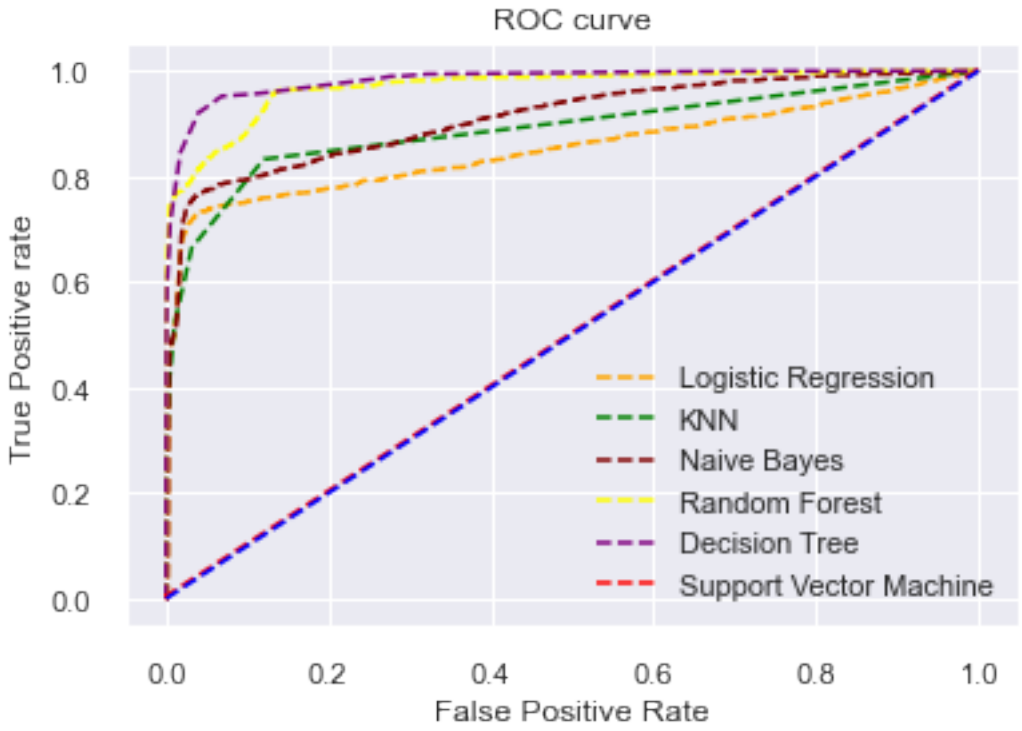


Figure 5.20: ROC curve of model simulations with Balanced-I set

5.2.10 Balanced-J: Non-Fraud 50000, Fraud 6006

In Balanced-J set, we have taken 6006 fraud and 50000 non-fraud transactions which resulted in 56006 total transactions, 39204 transactions in train set and 16802 transactions in test set. Fig.(5.21) & Fig.(5.22) depict the values of accuracy, specificity and ROC of Balanced-J respectively.

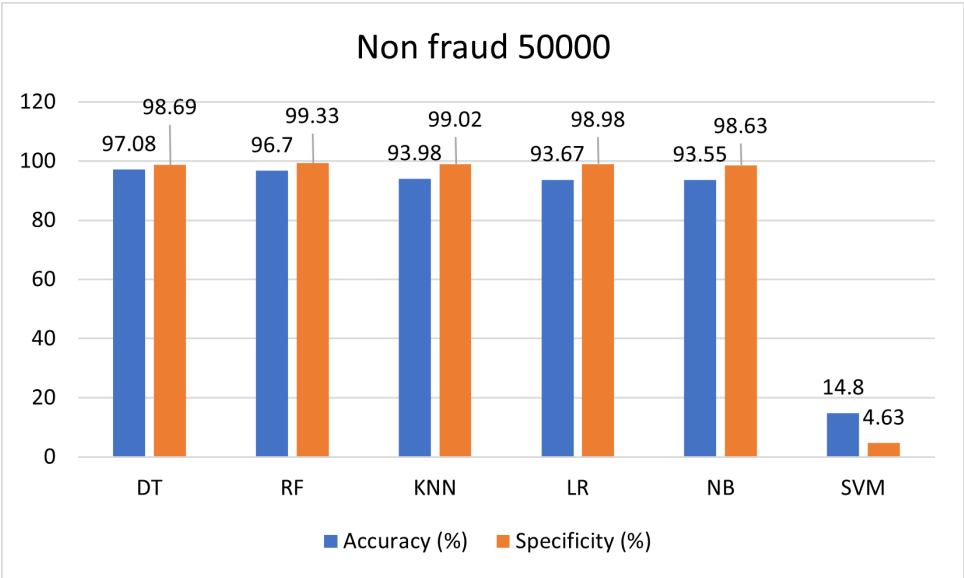


Figure 5.21: Accuracy & Specificity of models in Balanced-J set

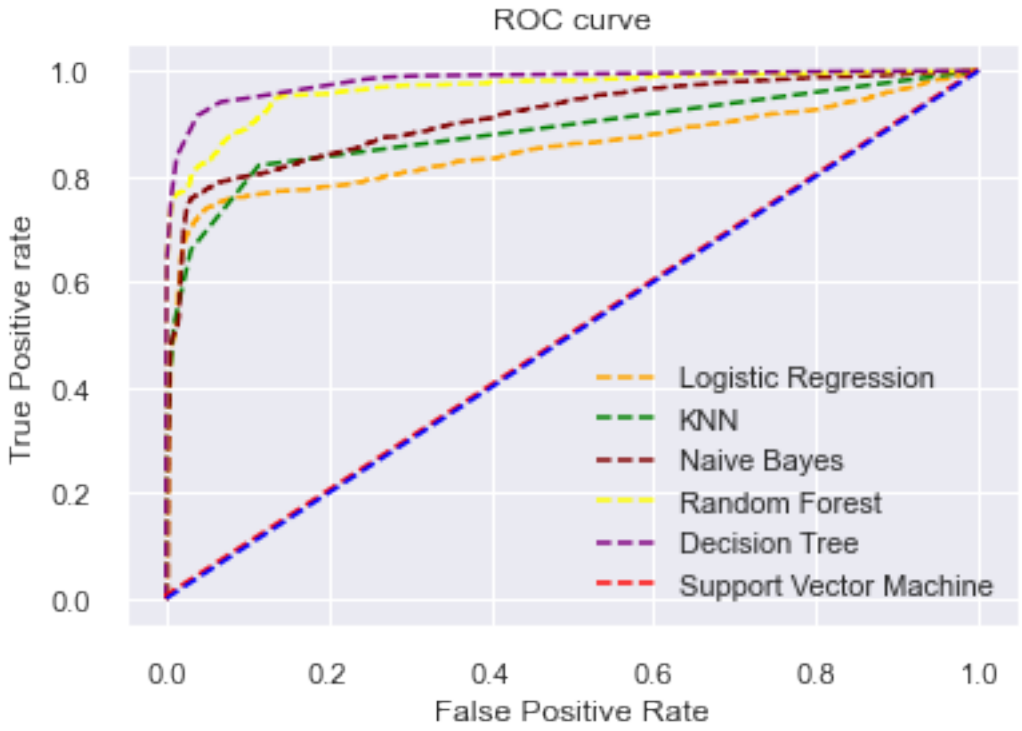


Figure 5.22: ROC curve of model simulations with Balanced-J set

5.3 Model wise evaluation

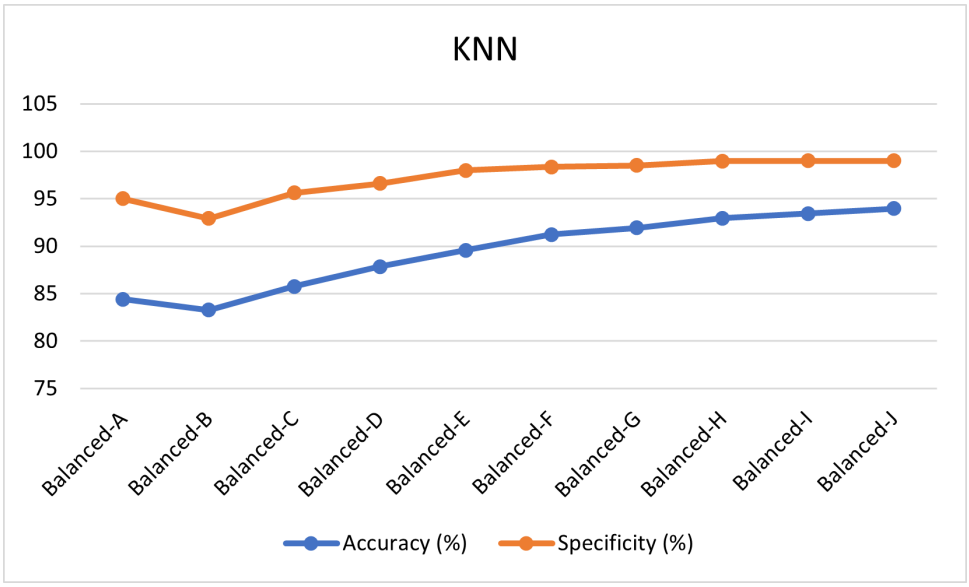


Figure 5.23: Accuracy & Specificity of KNN with variable non-fraud transactions

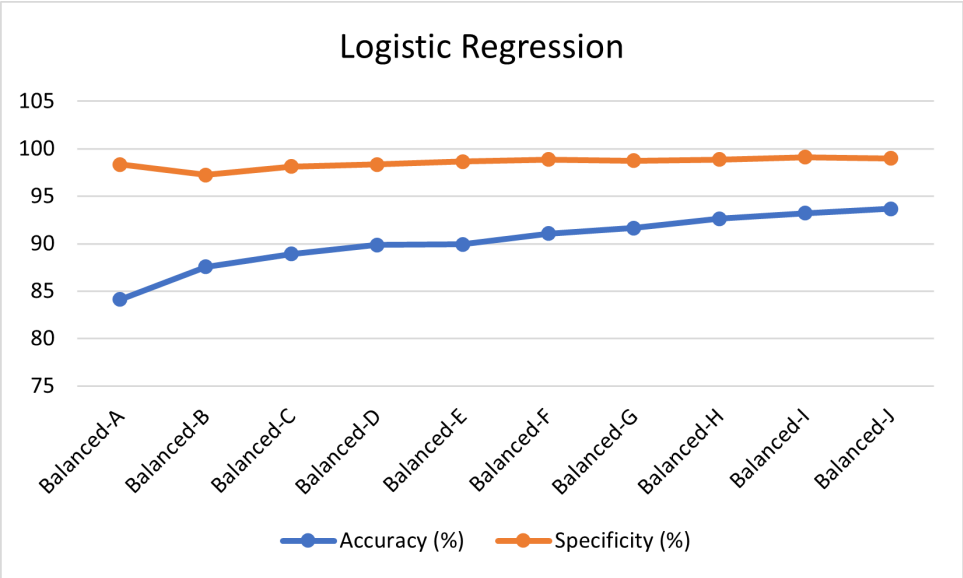


Figure 5.24: Accuracy & Specificity of Logistic Regression with variable non-fraud transactions

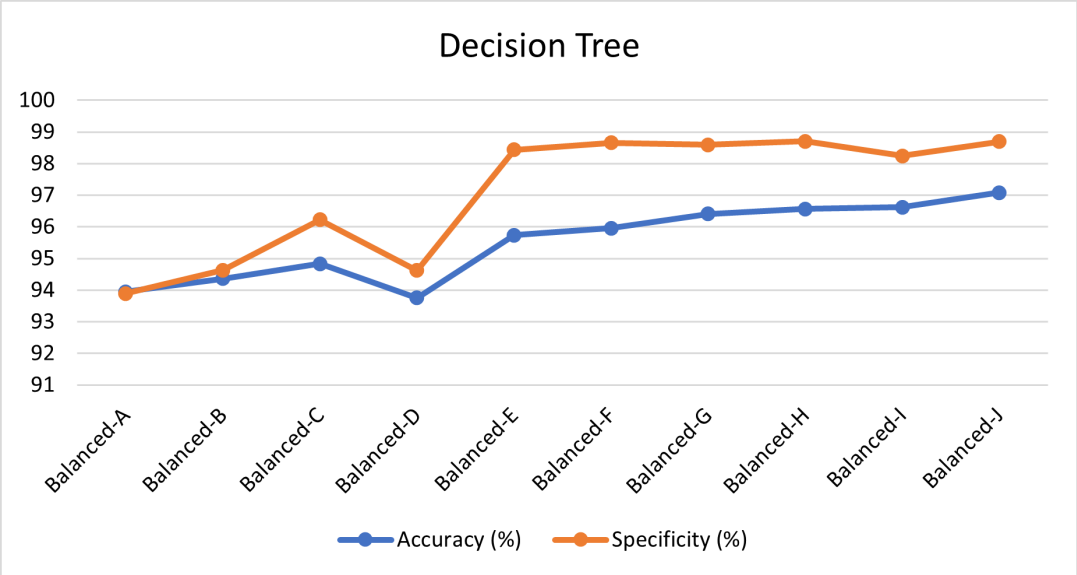


Figure 5.25: Accuracy & Specificity of Decision Tree with variable non-fraud transactions

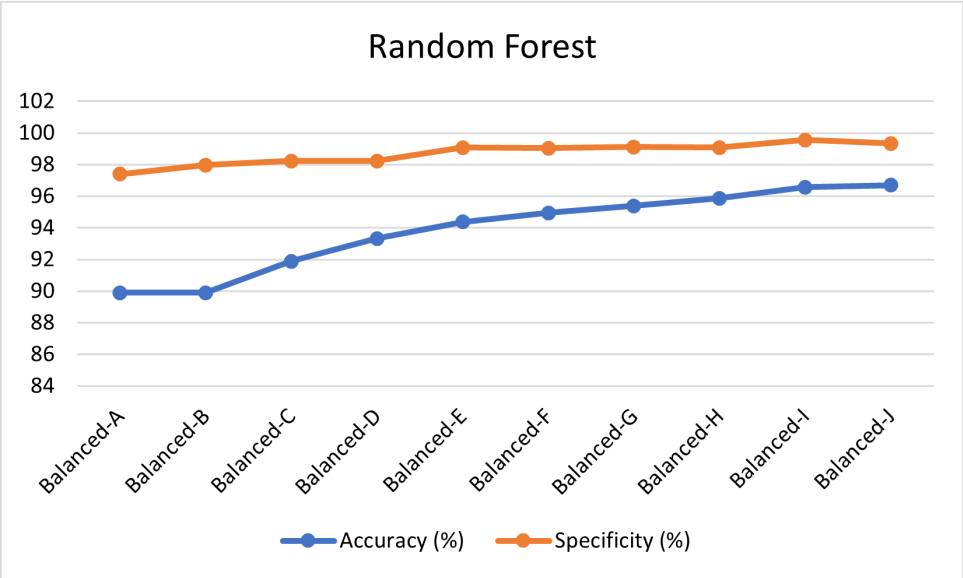


Figure 5.26: Accuracy & Specificity of Random Forest with variable non-fraud transactions

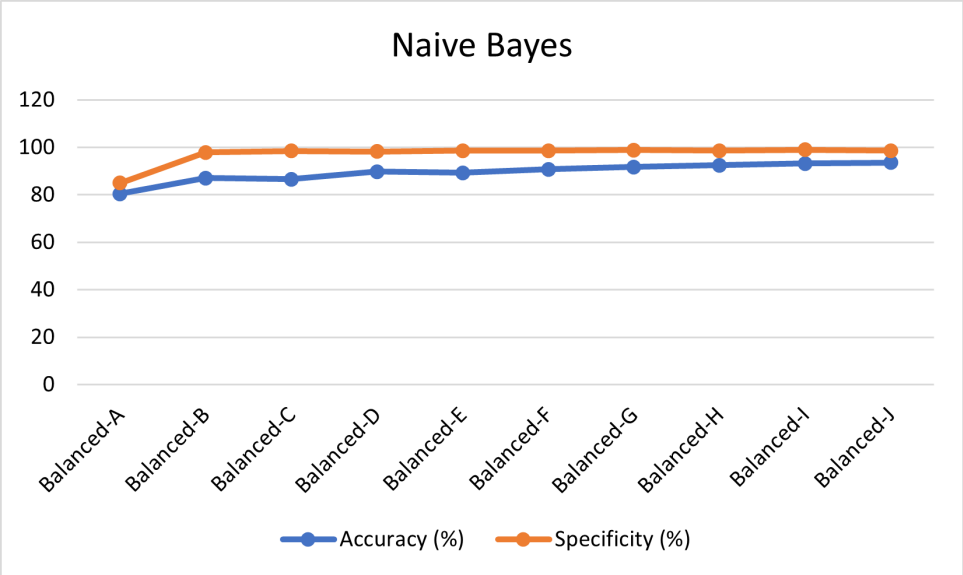


Figure 5.27: Accuracy & Specificity of Naive Bayes with variable non-fraud transactions

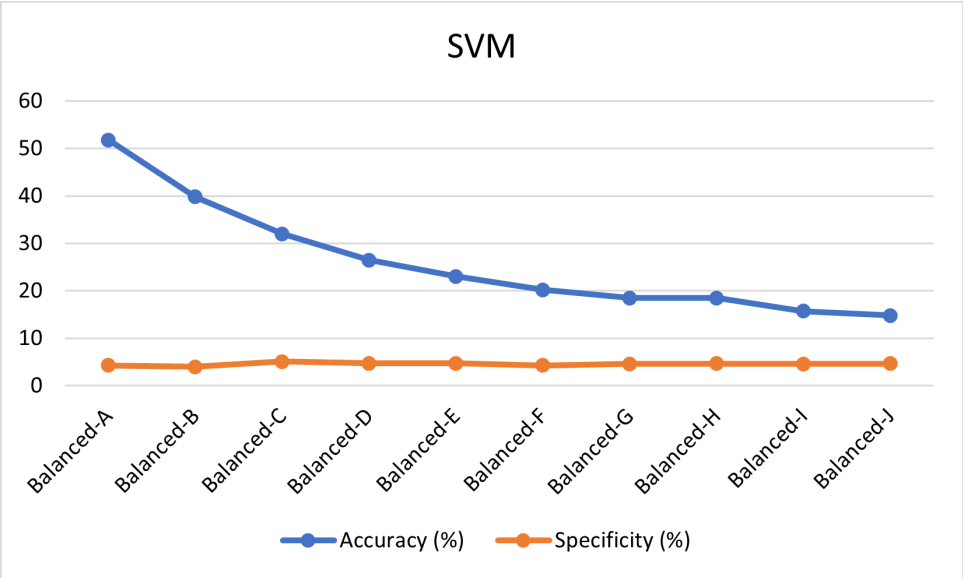


Figure 5.28: Accuracy & Specificity of Support Vector Machine with variable non-fraud transactions

Chapter 6

Result & Discussion

6.1 Comparison

From the results obtained, analysing the accuracy, specificity, and sensitivity, we have seen that all the models tend to give better performance with 50,000 non-fraud transactions in the training and testing set. We have considered the results of the simulation with 50000 non-frauds as our best result. Table (6.1) shows the performance metrics of our best result.

Table 6.1: Performance Metrics of best simulation

Algorithms	Accuracy(%)	Specificity(%)
Decision Tree	97.08	98.69
Random Forest	96.7	99.33
K Nearest Neighbour	93.98	99.02
Logistic regression	93.67	98.98
Naive Bayes	93.55	98.63
Support Vector Machine	14.8	4.63

From Table (6.1), it is evident that Decision Tree has the best performance metrics value. We have compared our results with some relevant papers cited in Chapter 2 and Table (6.2) displays the comparison.

Table 6.2: Tabular comparison with relevant papers

Reference	Best Algorithm	Accuracy(%)	Precision(%)	Specificity(%)
[32]	Random Forest	96.2	99.7	98.7
[31]	Random Forest	94.99	95.99	95.1
[33]	KNN	92.26	91.11	81.19
[34]	Random Forest	94.37	89.87	81.58
[30]	Random Forest	98.6	99.7	90.5
Our Model	Decision Tree	97.08	95.2	98.69

Fig. (6.1) shows the graphical representation of the comparison of our best model performance metrics and that of other relevant papers.

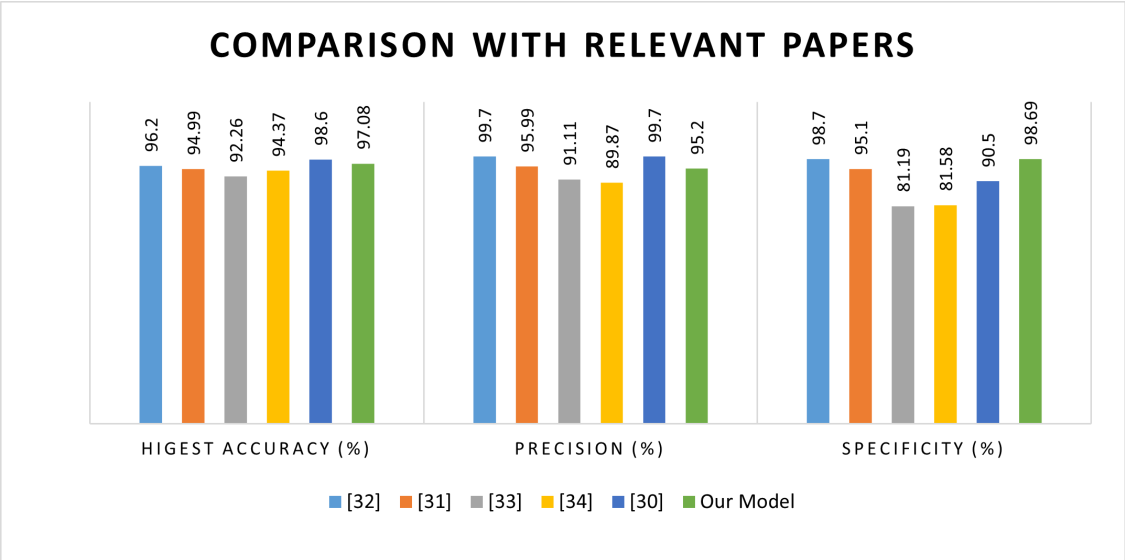


Figure 6.1: Comparison of our results with relevant papers

6.2 Discussion

The accuracy and specificity of the machine learning models are approximately 100 in Fig.(5.1), suggesting that the models are performing with 100% accuracy. However, this accuracy is far too high in general. Since our full dataset is highly imbalanced, the overall number of fraud transactions in the dataset is just 0.57% of the complete set. As a result, the models produce over-fitting results, as seen in Fig.(5.1) & Fig.(5.2).

But after we divided our dataset into ten balanced datasets (Balanced-A to Balanced-J), the accuracy & specificity of the simulated models showed better performance. From Fig.(5.3) to Fig.(5.21), we can see the accuracy and specificity of the balanced datasets. Among the six models we used for simulation and evaluation, Decision Tree in Balanced-I has given the best accuracy. Of all the balanced set, set Balanced-J has given the best performance according to the geometric progression in accuracy and specificity of all the sets. The accuracy of Decision Tree, Random Forest, KNN, Logistic Regression, Naive Bayes & Support Vector Machine in Balanced-J are 97.08%, 96.7%, 93.98%, 93.67%, 93.55% and 14.8% respectively. Just to compare, we also simulated the models with 51,000 non-fraud transactions. The accuracy of the models were 96.88%, 96.53%, 93.92%, 93.66%, 93.57% and 14.47% respectively. As such, we can clearly say that the accuracy of the models start to decrease for any balanced set where number of non-fraud is greater than 50000.

After getting the model performance metrics of the new balanced sets, we compared those results with some of the relevant papers we related to. Random Forest has the best accuracy of 98.6% and specificity of 90.5%, as shown in [30]. The best algorithms in [31] & [32] are also Random Forest, with accuracy 94.99% and 96.2%, respectively. In [34], Random Forest is likewise the best algorithm. S. Khatri, A. Arora & A. P. Agrawal revealed in [33] that K- Nearest Neighbour provided the highest results in detecting fraud transactions, with an accuracy of 92.26%. In comparison to these results, the best algorithm we have is Decision Tree with an accuracy of 97.08% and specificity of 98.69%. In terms of accuracy and specificity,

our Decision Tree takes the lead among all of these results. One of the most probable reasons is because we divided the old imbalance dataset into ten new balanced datasets with varying amounts of non-fraud transactions, which is a novel approach to such problems.

Chapter 7

Conclusion & Future Work

Real-time credit card fraudulence detection is possible in today's world with proper data pruning, noise reduction, feature extraction, and model training. With an accuracy of 97.08% and a specificity score of 98.69%, Decision Tree outperformed the other five supervised machine learning models. There are various existing techniques for dealing with dataset imbalance, such as oversampling and under-sampling, however we employed a new method. It takes time to identify the appropriate amount of non-frauds to train the model with our method, but the end result is satisfactory and efficient.

Bibliography

[1] Sahai, L., & Gursoy, K. (2019). "Real-time credit card fraud detection."

[2] Mester, L. J. (2020). "Modernizing Our Payments System," Evolutionary Computation, IEEE Transactions on, vol. 1, no. 1, pp. 53–66, 1997

[3] Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M., & Anderla, A. (2019, March). Credit card fraud detection-machine learning methods. In 2019 18th International Symposium INFOTEH-JAHORINA (INFOTEH) (pp. 1-5). IEEE.

[4] Awoyemi, J. O., Adetunmbi, A. O., & Oluwadare, S. A. (2017, October). Credit card fraud detection using machine learning techniques: A comparative analysis. In 2017 International Conference on Computing Networking and Informatics (ICCNI) (pp. 1-9). IEEE.

[5] Bhattacharyya, S., Jha, S., Tharakunnel, K., & Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision support systems, 50(3), 602-613.

[6] Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications, 40(15), 5916- 5923.

[7] Malini, N., & Pushpa, M. (2017, February). Analysis on credit card fraud identification techniques based on KNN and outlier detection. In 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB) (pp. 255-258). IEEE.

[8] Zareapoor, M., & Shamsolmoali, P. (2015). Application of credit card fraud detection: Based on bagging ensemble classifier. Procedia computer science, 48(2015), 679-685.

[9] Sweers, T., Heskes, T., & Krijthe, J. (2018). Autoencoding Credit Card Fraud. Bachelor Thesis.

[10] Jain, R., Gour, B., & Dubey, S. (2016). A hybrid approach for credit card fraud detection using rough set and decision tree technique. International Journal of Computer Applications, 139(10), 1-6.

[11] Delamaire, L., Abdou, H., & Pointon, J. (2009). Credit card fraud and detection techniques: a review. Banks and Bank systems, 4(2), 57-68.

[12] Bhanusri, A., Valli, K. R. S., Jyothi, P., Sai, G. V., & Subash, R. R. S. Credit card fraud detection using Machine learning algorithms.

[13] Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. Procedia Computer Science, 165, 631-641.

[14] Maniraj, S., Saini, A., Ahmed, S., & Sarkar, S. (2019). Credit card fraud detection using machine learning and data science. International Journal of Engineering Research and, 8(09).

[15] Lakshmi, S. V. S. S., & Kavilla, S. D. (2018). Machine learning for credit card fraud detection system. International Journal of Applied Engineering Research, 13(24 Pt. 1), 16819-16824.

[16] Saragih, M. G., Chin, J., Setyawasih, R., Nguyen, P. T., & Shankar, K. (2019). Machine learning methods for analysis fraud credit card transaction. International Journal of Engineering and Advanced Technology (IJEAT). ISSN, 2249-8958.

[17] Kim, E., Lee, J., Shin, H., Yang, H., Cho, S., Nam, S. K., & Kim, J. I. (2019). Champion-challenger analysis for credit card fraud detection: Hybrid ensemble and deep learning. Expert Systems with Applications, 128, 214-224

[18] Suryanarayana, S. V., Balaji, G. N., & Rao, G. V. (2018). Machine learning approaches for credit card fraud detection. Int. J. Eng. Technol, 7(2), 917-920

[19] Yee, O. S., Sagadevan, S., & Malim, N. H. A. H. (2018). Credit card fraud detection using machine learning as data mining technique. Journal of Telecommunication, Electronic and Computer Engineering (JTEC), 10(1-4), 23- 27.

[20] Bahnsen, A. C., Aouada, D., Stojanovic, A., & Ottersten, B. (2016). Feature engineering strategies for credit card fraud detection. Expert Systems with Applications, 51, 134-142

[21] Sadgali, I., Sael, N., & Benabbou, F. (2019). Performance of machine learning techniques in the detection of financial frauds. Procedia computer science, 148, 45- 54

[22] Pun, J. K. F. (2011). "Improving credit card fraud detection using a meta-learning strategy" (Doctoral dissertation).

[23] Pun, J., & Lawryshyn, Y. (2012). Improving credit card fraud detection using a meta-classification strategy. International Journal of Computer Applications, 56(10)

[24] Navamani, C., & Krishnan, S. (2018). Credit card nearest neighbor based outlier detection techniques. Int. J. Comput. Tech, 5(2), 56-60.

[25] KRISHNAN, S. Credit Card Nearest Neighbor Based Outlier Detection Techniques.

[26] Lucas, Y., & Jurgovsky, J. (2020). Credit card fraud detection using machine learning: A survey.

[27] Lucas, Y. (2019). Credit card fraud detection using machine learning with integration of contextual knowledge (Doctoral dissertation, Université de Lyon; Universität Passau (Deutscheland)).

[28] M. R. Kabir, F. B. Ashraf and R. Ajwad, "Analysis of Different Predicting Model for Online Shoppers' Purchase Intention from Empirical Data," 2019 22nd International Conference on Computer and Information Technology (ICCIT), 2019, pp. 1-6, doi: 10.1109/ICCIT48885.2019.9038521.

[29] Abdullah, F. B. Ashraf and N. S. Momo, "Comparative analysis on Prediction Models with various Data Preprocessings in the Prognosis of Cervical Cancer," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2019, pp. 1-6, doi: 10.1109/ICCCNT45670.2019.8944850.

[30] N. Khare, S. Y. Sait, "Credit Card Fraud Detection Using Machine Learning Models and Collating Machine Learning Models," International Journal of Pure and Applied Mathemetics, Vol. 118, No. 20, 2018, 825-838.

[31] N. K. Trivedi, S. Simaiya, U. K. Lilhore, "An Efficient Credit Card Fraud Detection Model Based On Machine Learning Methods," International Journal of Advanced Science & Technology, Vol. 29, No. 5, 2020, 3414-3424.

[32] S. K. Shirgave, C. J. Awati, S. S. Patil, "A Review On Credit Card Fraud Detection Using Machine Learning," International Journal of Scientific and Technology Research, Vol. 8, Issue 10, 2019

[33] S. Khatri, A. Arora, A. P. Agrawal, "Supervised machine Learning Algorithms for Credit Card Fraud Detection: A comparison," 10th International Conference on Cloud Computing, 2020

[34] S. Xuan, G. Liu, C. Jiang, " Random Forest for Credit Card Fraud Detection," International Conference on Communication Systems and Network Technologies, 2018.

[35] K. Shenoy, "Credit Card Transactions Fraud Detection," retrieved from https://www.kaggle.com/kartik2112/fraud-detection