**A**

**PROJECT REPORT**

**ON**

**COMPARATIVE ANALYTIC MODELLING FOR**

**CREDIT CARD FRAUD DETECTION**

**WRITTEN BY**

**OCHEI DANIEL OGHENETEGA**

**MATRIC NUMBER: 180202009**

**SUBMITTED TO**

**THE DEPARTMENT OF PHYSICAL AND COMPUTER SCIENCES**

**COLLEGE OF NATURAL AND APPLIED SCIENCES.**

**McPHERSON UNIVERSITY, SERIKI SOTAYO, OGUN STATE.**

**IN PARTIAL FULFILMENT OF THE AWARD OF BACHELOR OF COMPUTER SCIENCES (B.Sc. COMPUTER SCIENCE)**

# CERTIFICATION

This is to certify that I, OCHEI DANIEL OGHENETEGA with Matric. No: 180202009 wrote the project report on CREDIT CARD FRAUD DETECTION SYSTEM in the Department of Physical and Computer Sciences, College of Natural and Applied Sciences, McPherson University, Seriki Sotayo, Ogun State under my Supervisors.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. K. E. Akinola Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. K. Olumurewa Date

# DEDICATION

This project report is dedicated to the Almighty God who bestowed upon me HIS grace, protection and strength to will myself forward, and my Mother who has supported me financially and spiritually in this journey. I am immensely blessed to have such a supportive guardian as yourself, who has continued to encourage and push me past my limits and cheer me on even through rough patches.

# ACKNOWLEDGEMENT

I would also like to thank my Project Supervisors, Dr. K. E. Akinola and Dr. Kayode Olumurewa for their input, wisdom and guidance throughout the course of this project.

I also express my deepest gratitude to all lecturers for all their continued efforts in imparting knowledge into their dear students who have come this far. From Mr. Joachim Braimah, to our Course Adviser, Mr. Aina to Dr. Femi Ayo and, Dr. Kayode Olumurewa, I thank all of you for bringing us this far in our university education.

# ABSTRACT

Due to the rapid growth of e-commerce, the use of credit cards for online purchases has increased and unexpectedly caused an eruption in credit card fraud. The use of any form of payment card for purchase has generated more cases of fraud associated with it. A lot of money, estimated in billions is lost every year to fraudulent credit card transactions. Fraud detection systems come into a synopsis when the fraudsters break down every prevention initiative put in place and start fraudulent transactions. Fraud detection based on analysing existing purchase data of a cardholder is a promising way in minimizing fraud. For better secure services from financial institutions, a reliable fraud detection mode is vital to support safe credit card usage. The design of efficient fraud detection algorithms is key for reducing these losses and more algorithms rely on advanced machine learning techniques. The detection of credit card fraud features statistical tests and data made on user data based on those behavioural and historical data.

This research work also aims to explore three methodologies and their implementation in the accuracy of detecting credit card fraud. Financial institutes have an increased incentive for the development of fast, effective and dynamic fraud detection systems and thereby focus more on implementing technology safeguards and cybersecurity to decrease the impact of credit card fraud activities.

The methodology used in this research topic includes a supervised learning algorithm called Logistic Regression, and unsupervised Learning Algorithms such as Isolation Forest and Local Outlier Detection, involving the K-Nearest Neighbours Algorithm. One extensively discussed proposed system uses logistic regression to build the classifier to prevent frauds in credit card transactions. This is used on a dataset obtained from Kaggle, which contains features Time, Amount, Class and transformed features as a result of a Principal Component Analysis dimentionality reduction. Comparisons among these algorithms in respect to the dataset’s distributed dataframe are then made to determine their ccapability to achieve a better result and can be adopted by credit card merchants. The model shows an overall evaluation of results in terms of accuracy, sensitivity and error rate.

# TABLE OF CONTENTS

[CERTIFICATION ii](#_Toc105099601)

[DEDICATION iii](#_Toc105099602)

[ACKNOWLEDGEMENT iv](#_Toc105099603)

[ABSTRACT v](#_Toc105099604)

[TABLE OF CONTENTS vi](#_Toc105099605)

[CHAPTER ONE 1](#_Toc105099606)

[1.1 BRIEF INTRODUCTION OF THE TOPIC 1](#_Toc105099607)

[1.2 STATEMENT OF PROBLEM 2](#_Toc105099608)

[1.3 SIGNIFICANCE OF THE STUDY 4](#_Toc105099609)

[1.4 GENERAL OBJECTIVES 4](#_Toc105099610)

[SPECIFIC OBJECTIVES 5](#_Toc105099611)

[1.5 SCOPE OF STUDY 5](#_Toc105099612)

[1.6 ORGANIZATION OF WORK 6](#_Toc105099613)

[CHAPTER TWO 7](#_Toc105099614)

[2.1 REVIEW OF RELEVANT LITERATURE 7](#_Toc105099615)

[2.2 RELATED WORKS 7](#_Toc105099616)

[2.3 SUMMARY OF RELATED WORKS 18](#_Toc105099617)

[CHAPTER THREE: MATERIALS AND METHODS/DESIGN METHODOLODY 23](#_Toc105099618)

[3.1 INTRODUCTION 23](#_Toc105099619)

[3.2 DATASET DESCRIPTION 23](#_Toc105099620)

[3.3 DESIGN APPROACH 24](#_Toc105099621)

[3.3.1 AMOUNT AND QUALITY OF DATA 25](#_Toc105099622)

[3.3.2 SUPERVISED AND UNSUPERVISED LEARNING TECHNIQUES 26](#_Toc105099623)

[3.3.3 SOFTWARE AND DEPENDENCIES USED 28](#_Toc105099624)

[CHAPTER 4: IMPLEMENTATION OF THE ANALYTICAL MODEL 51](#_Toc105099625)

[4.1 FRAUD DETECTION USING LOGISTIC REGRESSION MODEL 51](#_Toc105099626)

[4.1.1 LOGISTIC REGRESSION IN A DISTRIBUTED DATAFRAME 59](#_Toc105099627)

[4.2. OUTLIER DETECTION FOR UNSUPERVISED LEARNING ALGORITHM 63](#_Toc105099628)

[4.2.1 ISOLATION FOREST ALGORITHM 63](#_Toc105099629)

[4.2.2 LOCAL OUTLIER FACTOR 64](#_Toc105099630)

[4.3 ANOMALY DETECTION USING LOCAL OUTLIER FACTOR AND ISOLATION FOREST ALGORITHM 64](#_Toc105099631)

[CHAPTER FIVE: CONCLUSION AND RECOMMENDATION 73](#_Toc105099632)

[5.1 CONCLUSION 73](#_Toc105099633)

[5.2 CONTRIBUTION TO KNOWLEDGE 73](#_Toc105099634)

[5.3 RECOMMENDATION 74](#_Toc105099635)

[REFERENCES 75](#_Toc105099636)

# CHAPTER ONE

## 1.1 BRIEF INTRODUCTION OF THE TOPIC

A credit card or universally known as a payment card is a small plastic card issued to various users as a system of payment (Raj & Portia, 2011). It is branded as one of the methods of carrying out transactions and have become commonplace for individual finance over the past few years. In our daily lives, credit cards are used for purchasing goods and services with the help of virtual card for online transaction or physical cards for offline transaction. The credit card has a plethora of advantages, one being its easy access to credit, purchase and offering a guaranteed method of payment and providing consumers with a way to further implement a cashless policy in transactions. Fraudulent transactions can be carried out by an attacker by stealing the card information from the cardholder. These information may include the credit card number, the validity, the Card Verification Value (CVV) which is vital for completing online transactions and the name of the card holder. After the information gathering, the attackers can then use these cards for ridiculous purchases, putting both the cardholder and the institution at risk.

The good thing is, some major payment processes mine data from their card holders and their spending habits. The company builds a picture not only of where you spend the money but how much and how frequently. Some more advanced methods can track the IP addresses of where the transactions originated from. So if a charge tied to an IP address previously used for fraud is observed, the card is flagged and immediately reported.

## 1.2 STATEMENT OF PROBLEM

As the world continuously advances into a more digitalized format of even the most basic events, various cybercrimes involving but not limited to credit card fraud are on the increase. Credit Card fraud can be defined as when an authorized person uses a credit card for personal use without the approval or knowledge of the card owner and the card issuer does not have a clue of what the card is being used for (Ayorinde, 2021). The Federal Bureau of Investigation also defined credit card fraud as, “the unauthorized use of a credit or debit, or similar payment tool to obtain money or property.” Everyone involved in the card based payment process, ranging from cardholders, online merchants, card issuers and acquirers can potentially fall victim to scammers. This happens to be one of the biggest threats to financial institutions and business presences today. A 2019 report of the Bureau of Consumer Financial Protection on the Consumer Credit Card Market stated that, “fraud remains a constant and costly reality of the credit card market”. According to the Nielson Report, in 2020, over $28.6 billion was lost to fraud worldwide and an estimated $408 billion in worldwide losses is projected over the next ten years. Unauthorized card operations hit an amount of 16.7 million victims in 2017. Additionally, as reported by the Federal Trade Commission (FTC), the number of credit card fraud claims in 2017 was 40% higher than the previous year’s number. This goes to show that fraud can staggeringly affect an organization’s reputation, altruism and client relations.

These types of frauds have a certain trend to them.

* Stolen/Misplaced card: This method is the most prevalent. It has to do with stealing someone's credit card and using it as their own. Banks usually inform customers to notify them through the emergency lines anytime their card is stolen or misplaced. The attacker can use the stolen card information to purchase goods online, and the bank might not notify the owner until the end of the month.
* Data Breach: Since people carry out some of their transactions through the internet, their data is vulnerable to hackers. The hacker might adopt several ways to get the victim's data. One of the recommended ways to remedy this situation is to avoid saving important information on any device, or better still, to frequently clear data before getting into the wrong hands.
* Synthetic Fraud: A synthetic fraud is an act whereby a fraudster applies for a payment card on behalf of someone. The attacker gathers essential information of their victim like Social Security Number (SSN), date of birth, address, etc., and applies for a payment card on behalf of the victim. This is also known as the false application method.
* Mail Interception: Fraudsters can also intercept mails intended to go to the user's address. This is usually done after applying for a new card and then the fraudster can manipulate things to get the card before it gets to the owner. The money would have been gone before the card eventually gets to the owner.
* Skimming: This is done in a number of ways. Attackers can purchase credit card information from the dark web and then use a form of untraceable cryptocurrency from a block chain database. A tool called the embossing machine can then be used to rewrite a fake credit card with the information purchased from the web. The machine then proceeds to punch in the new numbers on the card and a skimmer/skimming machine plugged into a computer can then fully write that information to the card using the magnetic strip. Skimming is also particularly popular in cloning, where an original card/victim’s card is read into the skimmer and can be copied or cloned onto another counterfeit card with different information.

The use of these various methods of card fraud can and has been done on a much wider scale, affecting millions of users, institutions and businesses alike. Therefore this stresses an immediate response to the ever growing attack mechanisms employed by these scammers

## 1.3 SIGNIFICANCE OF THE STUDY

The benefit of the research on this topic is to provide basis on aiding financial institutions by improving existing predictive machine learning algorithms that can determine fraudulent acts on payment cards with a very high accuracy which will ease them in preventing fraudsters from carrying out transactions that were not approved or authorized by the legitimate owner of various accounts. Although the range of the problem is problematic and extensive, there has been relatively some of academic exploration has been done on fraud costs, the root causes, how and why it occurs, and productive ways to recognize, discourage and avert it.

One of the current solutions that help banks and financial institutions move forward is the machine learning approach in relation to the various aspects of data mining. Organizations, firms and businesses must incur significant resources as they strive to protect themselves from fraud and reputational consequences.

## 1.4 GENERAL OBJECTIVES

The main aim of fraud detection systems is to detect fraud more accurately before it is committed. To break it down, the key focus here is to identify suspicious events and report them to an analyst while letting normal transactions be automatically processed. Financial institutions are now increasingly turning to a data science and machine learning approach in this implementation and as such, it can bring significant improvements to the process.

### SPECIFIC OBJECTIVES

* To achieve a higher accuracy of fraud detection. Various data points are considered with patterns (even in the slightest changes in and details of behaviour patterns) are considered to have a higher precision and return more relevant results compared to rule-based solutions.
* To examine the possibility of little manual work needed for additional verification. The application of machine learning filters out a high percentage of normal patterns, leaving only a minute percentage to be verified by experts. Even for a small bank, transaction checks cannot be done manually for every consumer. Enhanced accuracy leads reduce the burden on analysts.
* To better reduce the frequency of false declines, where the system wrongly flags or identifies a legitimate transaction as suspicious and cancels it.
* To explore the various machine learning techniques to provide an accurate model producing accurate results
* To enhance the ability to identify new patterns and adapt to changes, identifying new suspicious patterns and creating new rules to prevent types of scams.

## 1.5 SCOPE OF STUDY

In this report, the adaptation and implementation of the various methodologies framed under the possible detection and prevention of credit card fraud is studied and observed. This will require continuous and nominal data, which allows us to run tests involving logistical regression, decision tree which involves classification algorithm and tree representation. These supervised methods all fall under data mining, classifying the structure of the root, tree and leaf nodes considered in the operation.

Extensive experiments are conducted to train and test the proposed classifier using a standard database. This study surrounds the features and usage of machine learning and data mining using models such as logistic regression, Isolation Forest as well as observing other methodologies of supervised outlier detection methods. These operations are brought about using a dataset from Kaggle, an open source database which is encoded in a '.csv' file format. Models will be heavily implemented and analysed on this selected database and a detailed comparative analysis can be drawn and made on these analytic algorithms for their accuracy.

## 1.6 ORGANIZATION OF WORK

The first chapter of this report focused on the introduction, scope, purpose and general background of this research. It highlights the need for a fully functional and reliable credit card fraud detection system.

The second chapter takes a look at the relevant literature from authors who have greatly contributed knowledge in various aspects of this research, as well as some related study carried out by researchers with appropriate information and statistical data presented.

The third chapter focuses on the methodology; data collection and sampling, data pre-processing, data cleaning, categorical variables, feature scaling and resampling, correlation and selection, logistic regression, Outlier Detection Methods such as Isolation Forest and Local Outlier Factor in their functions and implementation of the system.

Chapter four of this project research delves into the findings and results of the application of the above methodologies. This is capped off with a concluding final chapter, regarding future research on this topic, contributions to the research and limitations encountered doing this research.

# 

# CHAPTER TWO

## 2.1 REVIEW OF RELEVANT LITERATURE

Credit card fraud detection is a very active area of research and learning in data science and many works have been done over the years in relation to this topic and its constituents. Numerous literatures pertaining to anomaly or fraud detection in this domain have been published already and are available for public usage. In this chapter, we will view and synthesize some of these research articles to identify and point out work that has already been done. This section discusses machine learning and data science for a variety of methods. These methods can be broadly classified into supervised methods such as Logistic Regression, Decision Tree, Random Forest, XG Boost and unsupervised methods such as the K-means Clustering and Auto encoder in Keras. Various researchers like Makki, (2019), Mohari, Dowerah, Das, Koucher, F., & Bora, (2021), Shirgave et. al (2019 Akinola, Adekunle, Adebayo & Okolie, S. O. (2019), Maniraj (2019), Sadineni (2020) amongst many others have greatly contributed knowledge to this topic in discussion, some of which identified supervised and unsupervised methods in data science implementations and also proposing other ways or methods in solving this recurring issue in transactions for a working credit card fraud detection system model.

## 2.2 RELATED WORKS

Data science, machine learning together with deep learning techniques can be implemented and can be used to tackle fallacious activities performed on credit card. Mohari, Dowerah, Das, Koucher, & Bora, (2021) explored this by identifying ten of them and compared them in their research. They compared Logistic Regression, Random Forest, Ada Boost, Artificial Neural Network (ANN), and Hidden Markov Model (HMM). KNN Classifier, Decision Tree, Isolation Forest and Local Outlier Factor. Of all ten methods, their results show that Local Outlier Factor fraud accuracy is greater than the rest of the algorithms. Vengatesan, Kumar, Yuvraj, KUMAR & Sabnis, (2020) compared the accuracy of logistical regression and KNN algorithms. They proposed a working model of a fraud detection system based on classification algorithm, which takes dataset as input, applied pre-processing techniques which finds errors where noise or inconsistent values from dataset and their prompt elimination . The performance analysis was measured using logistical regression and KNN algorithm, based on customer transaction.

Darwish (2020) tried to achieve two main objectives in the study; enhancing the accuracy of the classifications output by credit card fraud detection systems and lowering the response times of these systems. To achieve the first goal, the author proposed a hybrid model that fuses two classifiers to generate a new (or enhanced) one. The first classifier used is the K-means classifier, which deals with overlapping data because such data cause poor accuracy. The second classifier is the artificial bee colony algorithm (ABC), which is used to enhance the performance of the system. The first classifier forms the first level, and the second classifier forms the second level of the classification process proposed in the same model. C# programming language generated the database used in this work, where the number of instances was 100,000. In addition, 12 features were selected to include in the training phase. The selected features were based on a rule engine.

Kim, Kim, and Kim, (2019) tried to evaluate the detection problem by extracting the general pattern of the dataset to represent the fraud. In other words, the enhancement of the clustering methods relies only on the clusters used; this technique is called general enhancement. The authors proposed an approach that enables the application of local enhancement as well as general enhancement for fraud detection in financial transactions. They proposed the “Hierarchical Clusters-based Deep Neural Networks (HC-DNN)” method that uses the anomalous features of hierarchical clusters that are pretrained based on an auto encoder as the initial weights for neural networks. In detail, the data are grouped based on abnormal features that refer to fraud. These features are then used as the starter parameters for the input layers of neural networks.

Behera and Panigrahi, (2015) analysed their model by calculating the suspicion score according to the extent of deviation from the normal patterns. Using this, the transaction can then be efficiently classified as legitimate or suspicious (fraudulent). When a transaction is found to be suspicious, a neural network based learning mechanism is deployed to determine whether it was actually a fraudulent activity or an occasional behavioural deviation by a genuine user. They also performed extensive experimentation with stochastic models and showed that the combined use of clustering techniques along with learning helps in detecting fraudulent activities effectively while minimizing the generation of false alarms. A case study in credit card fraud detection conducted by Carneiro, Dias, Da Cunha and Mialaret (2015) in review of cluster analysis and artificial neural networks seemed to further buttress their application for their usage in qualitative data normalization as well as using an MLP trained automatically normalized data, presenting promising results. They also highlighted performance comparisons between ANNs trained by using raw data and ANNs trained by using clustered data and the use of different clustering algorithms and configurations in order to reduce information loss.

Dal Pozzolo, Boracchi, Caelen, Alippi and Bontempi, G. (2015) addressed credit card fraud detection. In this study, the authors relied on the fact that "the features of the financial transactions in institutions change over time". This shows that the problem of credit card fraud detection should be considered in real time. Therefore, they converted this problem into real working transactions. In terms of artificial intelligence, the class should not be provided to the classifier immediately during the training stage. The key idea of the proposed approach is to follow a strict strategy that has three main steps; analysing the real conditions under which the real transactions are performed, employing these conditions to train the classifier using two main data sets and testing the classifier after the training stage is completed and supporting it by using the interactions and feedback of the users to improve the accuracy of the classifier. Kasa, Dahbura, Ravoori, and Adams, (2019) highlighted a hybrid approach of identifying groups of customers based on engineered behavior profiles and then building classifiers specific to those groups with Random forest and XGBoost classifiers being trained on an entire sample and compared against classifiers trained at the transaction level across each cluster. Their approach were centered around three approaches; profiling accounts, clustering the accounts and building the classifiers. Their customer groups included;

* Global Spenders: Customers who spend farther away from home.
* Diverse Spenders: Customers who spend across a larger number of industries
* Low Spenders: Customers whose transactions are a lower dollar amount
* Frequent Spenders: Customers who have a higher frequency of transactions within a given time period.
* Big Spenders: Customers whose transactions are higher dollar amounts.
* Loyal Spenders: Customers with transactions concentrated over a smaller number of industries.
* General: No specifically discernible behaviors.
* Local Spenders: Customers who spend closer to home.
* Big and Infrequent Spenders: Customers whose transactions are higher dollar amounts and have a low frequency of transactions within a given time period.
* Glocal Spenders: Customers who regularly spend both close to and far away from home.

Clustering was also able to determine distinct behavioural patterns across account holders for each cluster. These cluster behaviours are also viewed across samples. Knowing the customer groups and behavioural tendencies of clusters that perform well and those that perform poorly is valuable in trying to improve fraud detection.

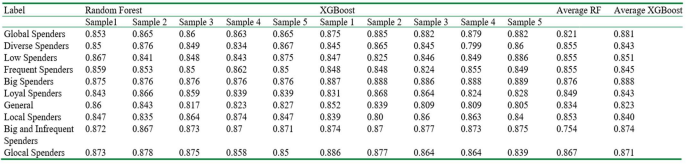


Figure 2.1 AUC values for customer groups

Shirgave et al. (2019) reviewed credit card fraud detection by comparing techniques using instruments like accuracy, precision and specificity. They proposed a fraud detection system which uses a supervised Random Forest Algorithm. The precision of detecting fraud in credit card is increased using this proposed system. More et al. (2021) also used a Random Forest fraud detection algorithm. This increased the accuracy of detecting fraud in credit card transactions and helped to solve fraudulent activities in the real world. The authors used a dataset containing 100,000 transactions made by cardholders, showing that 0.262% of all transactions made are fraud. There was the issue of an imbalanced dataset. However, the unbalanced dataset was processed, showing that 80% pf the dataset was used for training the model while 20% was used for testing. Despite being an unbalanced dataset, the model works well for credit card detection.

In illustrating the modelling of a dataset using machine learning with credit card fraud detection, Maniraj et al. (2019) tried to detect transactions that are 100% fraudulent as they minimize the incorrect fraud classification. The goal was analysing pre-processing datasets and deploying several anomaly detection algorithms like the Isolation Forest algorithm on the transferred credit card transaction data. The results of this study showed that the algorithm reaches over 99.6% accuracy, but the particularity is about 28% when a tenth of the data was used.

Tran, Tran, Huong, Heuchenne, HienTran and Le, (2018) focused their study on a proposed real time data driven approach for detecting credit card fraud. Their approaches brought about a high-level of detection accuracy and a low false alarm rate. Lamiek et al. (2018) used Outlier Detection Method for ATM Fraud. This involved an extracted feature to reflect the nominal behaviour of the accounts in each group of known legitimate transactions. A preprocess was applied to obtain only a withdrawal transaction from an Automated Teller Machine (ATM) and the transaction amount greater than zero is applied. Carcillo et al. proposed an extension consisting of the definition of a number of outlier scores and their integration into the supervised approach, generated using different supervised techniques.

Prakash, Murthy, Ashok, Prithvi and Kira, (2018) implemented their work using RStudio for R programming language which allows program development for data science field estimation. This is based on information regarding analysis done for the proposed system. Using the datasets involving European cardholders, they were able to make comparative results and determined that Decision Tree algorithm, in comparison to Logistic Regression and Support Vector Machine was indeed more promising, having a higher accuracy than other algorithms and methods used.

Jain, Tiwari, Dubey and Jain, (2019) made an extensive review on the basis of quantitative measurements such as accuracy, detection rate and false alarm rate. They achieved this by calculating the True Positive, the False Positive and False Negative generated by the system or an algorithm and use these in quantitative measurements to compare the performance of different systems. It was then found that the neural network and Naïve Bayesian network gives the highest accuracy, the SVM and Decision Tree offers medium level of accuracy while fuzzy logic system and logistic regression gives low accuracy. They also noted that ANNs are expensive to train and can easily be over trained.

Varmedja, Karanovic, Sladojevic, Arsenovic and Anderla, (2019) used a dataset from Kaggle containing transactions made in September 2013 by European cardholders over the span of two days. It captured 31 numerical features and they used classical algorithms to show that oversampling data can greatly improve fraud detection rate. The study also proved that the usage of classical algorithms in its implementation is just as successful as deep learning, and it is also stressed that feature selection and balancing of dataset is shown to be extremely important in achieving significant results. Patil et al. (2018) research study also proposed a robust framework to process large volumes of data to build a strong analytical model, evaluated with the help of confusion matrix (a field of statistical classification, also known as error matrix).

DECISION TREE

Rocha & de Sousa Junior, (2010) identified bank frauds by using CRISP-DM and Decision Tree techniques. They evaluate some transactions using decision trees and CRISP-DM to help identify and prevent bank fraud. Like many researchers who came after them, they identify decision trees as an essential concept in artificial intelligence. After the information regarding bank transactions, the analysis identified different fraudulent activities from internet bank transactions. Save et al. (2017) proposed a system which detects fraud in credit card transaction processing using a decision tree with combination of Luhn's algorithm and Hunt's algorithm. These algorithms can be used for validation of billing and shipping addresses. In this case, transactions can be detected as exceptions to the cluster – a process known as outlier detection.

HYBRID CLASSIFIERS

Warghade, Desai and Patil, (2020) analysed various machine learning techniques by using multiple metrics for judging multiple classifiers. Their research has been able to improve fraud detection rather than 10 misclassifying a genuine transaction as fraud. In their model, they opted for synthetic techniques like SMOTE for the conventional oversampling method. And to yield a better result, synthetic sampling methods like SMOTE with advanced boosting methods like Local Outlier factor, Isolation Forest, and SVM can be applied. As a result of the parallel processing model, LOF and Isolation Forest is fast and robust to the outlier. Isolation Forest gave a 99.74% accuracy score; Support Vector Machine provided a percentage of 45.84% accuracy score. LOF gave a great 99.66% accuracy score, making the prediction correct, misclassifying the genuine transaction as fraud.

Thennakoon, Bhagyani, Premadasa, Mihiranga and Kuruwitaarachchi, (2019) addressed real time credit card fraud detection by using predictive analytics, proposing a system by detecting four different patterns of fraudulent transactions by also addressing related problems identified by past researchers. They also accessed sampling methods that efficiently address the distribution of data and a major impact of using resampling techniques to obtain a higher performance.

In addressing the problem of detecting credit card fraud through transactions, Jiang, Song, Liu, Zheng and Luan (2018) dealt with the problem of online shopping fraud and the concept of drift. They proposed a strategy consisting of four stages; based on both the previous transaction data and the information of the cardholders, they used the clustering method to divide the cardholders into different groups for the purpose comparing their behaviours. Also, they proposed a sliding window strategy to group the transactions in each group to extract the behavioural patterns for each cardholder, they trained a set of classifications for each group to measure behavioural patterns; and they used a group of classifiers by training them on cardholder behaviours and output the highest behaviour pattern. A feedback mechanism was used to solve the concept of drift problem. Four dataset simulators were generated to manually create the data sets.

Marchal and Szyller, (2019) proposed a clustering-based method. In this study, the fraud detection problem in ecommerce is exploited by highly skilled hackers. The methods proposed to address such problems suffer from low accuracy and effectiveness. In addition, the methods used for detecting fraud may make some mistakes in identifying fraudulent transactions. The reason behind such shortcomings is that the proposed approaches focus on order analysis rather than anything else. Motivated by these facts, the authors proposed a method that focuses on the hackers themselves. The key idea is to extract some recognized features, such as the address of delivery, customer name, and methods of payment, and then, based on these features, the similarity among the attackers is calculated. Based on these similarities, the attackers are grouped in some clusters for detection. A main feature of their proposed method is that two current methods, agglomerative clustering and sampling, are selectively used in a reasonable amount of time for recursively grouping orders into small clusters. The dataset used for the training process was inspired by the Zalando website. This website periodically receives approximately 29 million orders (some of them are normal and others are fraudulent).

Wang and Han (2019) used a support vector machine (SVM) to improve the accuracy of the classifier in the process of detecting fraud in credit card transactions. The key idea behind using an SVM is to split the features that represent transactions, where these features are used for the clustering process. In other words, the data are cleaned initially. Then, the features of transactions are extracted. Third, the features are measured to calculate the similarity among them. To isolate the features as much as possible, the SVM is used. Fourth, the K-means clustering algorithm is used to cluster the data based on the isolated features. The classifier is then trained on the clusters. The classifier that deals with fraudulent transactions is used to detect fraud. The database used for training contains 5310 records in total. Among them, 490 records are fraudulent data and 4820 are non-fraudulent data, and 1174 characteristic variables are included.

HIDDEN MARKOV MODELS

Rahmawati, Sarno, Fatichah and Sunaryono, (2017) proposed a method for detecting fraud on credit applications. They used the Hidden Markov Model (HMM) and activity information recorded in the event log to identify fraudulent activities. This automated system bases the probability and possibility of fraud on the event log by identifying symptoms of fraudulent activities. The automation and analysis was based on 90 cases and the results show that the Hidden Markov Model method can be used to detect fraud with an accuracy of 94%. This model was able to report 10 of the 90 cases as fraudulent and the rest 80 as genuine transactions. Since Hidden Markov Model is viewed as the statistical tools for engineer and scientists to solve various problems, Bhusari and Patil (2016) used this to obtain a high fraud coverage combined with a low false alarm rate by analysing the pattern of spending profile of the card holder and getting the percentage calculation of each spending profile (low, medium and high) based on price distribution. It was observed that medium spending profile has maximum percentage of 50, followed by low profile 40% and then 10% of high spending profile, as shown below;

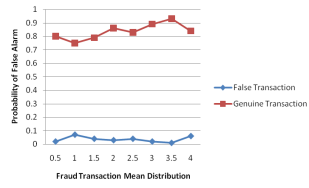


Figure - Probability of False Alarm compared with Fraud Transaction Mean Distribution.

Makki, (2019) explored fraudulent on Imbalanced classification algorithms for machine learning, such as C5.0 algorithms and Support Vector Machine (SVM). The authors also delved into Naïve Bayes conditional probability rule for classification, logistic regression and cost sensitive models (CS). Ultimately, they concluded that out of all the above mentioned algorithms, the C5.0 algorithm, SVM and ANN are the most eligible ones used in evaluating performance, simply because they produced the most balanced and reasonable outcomes for measures of Accuracy, Sensitivity and the fact that these three algorithms are not constrained by mathematical or statistical equation.

Sadineni (2020) worked on related study on machine learning application, where the analysis considered various techniques to identify frauds carried out by attackers. Accuracy, precision and false alarm rates were used in the performance analysis of the techniques. 150,000 transactions stored in the Kaggle repository were analyzed and the dataset, containing relevant and irrelevant attributes was analyzed based on the principal component to extract the relevant details like transaction amount, time if transaction, etc. Results showed that Random Forest achieved an accuracy of 99.21%, Decision Tree was 98.47%, Logistic Regression was 95.55%, SVM was 95.16% and ANN was 99.92%. These results, unlike others, showed that ANNs was more accurate than other techniques.

## 2.3 SUMMARY OF RELATED WORKS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author | Title | Method Used | Strength | Limitations |
| Vengatesan (2020) | Credit Card Fraud Detection using Data Analytics Techniques | Proposed a working model of the system, involving preprocessing techniques, logistical regression and KNN algorithm for production analytics | The KNN algorithm is produced best result such as statistical measure | Outlines the infinitesimal number of trades fraudulent in nature |
| Carcillo et al. (2019) | Combining Unsupervised and Supervised Learning in Credit Card Fraud Detection | Taking the outlier scores completed on the dataset, involving a hybrid approach | The implementation and assessment of different levels of granularity for the definition of an outlier score | The use of global outlier scores indicated a strong deterioration in accuracy and inconsistencies in the behaviour of precision metrics used |
| Makki (2019) | An experimental Study with Imbalanced Classification Approaches for Credit Card Fraud Detection | Implementing Class Imbalance solutions like classification algorithms and a selection of performance measures | Their research was able to show that SVM and ANN are the best methods. | While these approaches improve sensitivity, it led to an increase in the number of false alarm rates |
| Akinola et al. (2019) | Multifactor Authentication Model for Integrating Iris Recognition into an Automated Teller Machine | A virtual server arrangement with the use of a built Java application, Irisoft | Iris recognition produced a high-level of accuracy and reliability. | A 1.6% chance of an authentic user may be denied access. |
| Jain et al. (2019) | A Comparative Analysis of various Credit Card Fraud Detection Techniques | Comparing the performance of different systems by using measures generated from the system in quantitative environments | Neural Networks and Naïve Bayes networks give the highest accuracy in comparison to others | ANNs are expensive to train and can easily be overtrained |
| Prakash et al. (2018) | ATM Card Fraud Detection System using Machine Learning | Use of R programming language with RStudio and a GUI for confusion matrix decision tree algorithm analysis | Their results showed that decision tree had a higher accuracy than other algorithms | Discovered that the standard data mining algorithms did not fit well with classification problems |
| Tran et al. (2018) | Real Time Data-Driven approaches for Credit Card Fraud Detection | They used data-driven approaches without anomalies in the training set | Their proposed approaches to a high-level of accuracy and a low false alarm rates | Improvement on the detection ability of the proposed system |
| Niu et al. (2019) | A Comparison Study of Credit Card Fraud Detection: Supervised versus Unsupervised | They evaluated five supervised and four unsupervised learning models to leverage transactions to determine abnormal transactions. | All models performed well, with XG Boost achieving the best performance | The label availability and data imbalance restrict the supervised learning performances |
| Varmedja (2019) | Credit Card Fraud – Machine Learning Methods | The use of SMOTE technique was used for oversampling | They proved that the usage of classical algorithms is as successful as deep learning | Stresses the need for feature selection for metrics such as accuracy and precision |
| Patil et al. (2018) | Predictive modelling for credit card fraud using data analytics | Proposed a robust framework to build a strong analytical model with the help of confusion matrix | They were able to obtain a higher performance with decision tree in terms of accuracy | They had the issue of overfitting of tree in memory as data increases |
| Lamiek et al. (2018) | ATM Fraud detection using outlier detection | They used extracted feature to reflect behaviour of the amount and the usage of preprocesses | The use of Isolation Forest and Local Outlier Factor in addition to accurately identify fraud | Neural Networks and Random Forest models could not detect any frauds in a local-only account group |
| Rahmawati (2017) | Fraud detection on event log of Bank Financial credit business process using Hidden Markov Model Algorithm | They based the possibility of fraud on the event log by identifying symptoms of fraudulent activities | The method yielded a 94% accuracy. | The state probability of fraud has to be greater than the value of state probability of no fraud |

# CHAPTER THREE

# MATERIALS AND METHODS/DESIGN METHODOLODY

## 3.1 INTRODUCTION

It is imperative that companies overseeing the use of credit cards are able to recognize fraudulent credit card transactions to ensure credibility and make sure their customers are not charged for items not purchased by them, thus a method that is simple and fast in detecting frauds. However, in trying to generate and implement such, they can be overshadowed by a few challenges including the huge size of data used, organization of an Imbalanced dataset, and trying to adapt techniques in solving these problems as attackers themselves change styles to bluff detection systems. This section of the report aims to explore a given dataset, enumerate and analyze the development stage, as well as view the various methodologies used in formulating a working model to predict accurate results.

## 3.2 DATASET DESCRIPTION

There aren't a lot of active credit card frauds data available publicly, as this information will contain and include sensitive or confidential information. However, the dataset used for this research work is obtained from Kaggle. This dataset contains transactions made by credit cards in September 2013 by European cardholders. These transactions, 284,807 of them had 492 fraudulent activities and the remaining 284,315 that occurred in two days. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. It contains numerical input variables consists of features V1 through V28, the result of a PCA dimensionality reduction that was used in order to protect sensitive information. Principal Component Analysis (PCA) enables the execution of an exploratory data analysis to reveal the inner structure of data and explain its variations. The only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. The feature, 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

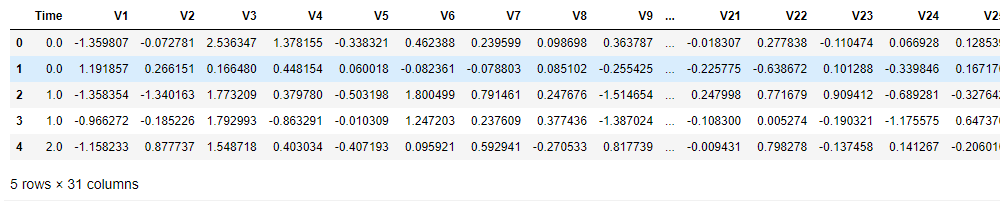


Figure 3.1

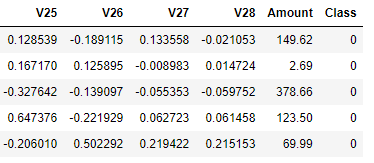


Figure 3.1 and 3.2 showing the first five rows of the overall dataset

## 3.3 DESIGN APPROACH

The main approach involving development in this study includes the implementation of machine learning in detecting fraudulent transactions automatically. This based operation is useful for identifying hidden correlations in data, and less time is needed for verification methods. The implementation of a machine learning based methodology differs a lot from a conventional fraud detection, which takes an enormous amount of time and finds only obvious fraudulent activities. Besides that, multiple verification methods are needed in a conventional detection, which is inconvenient for the user.

The implementation of Machine Learning in credit card fraud detection system involves a process of data investigation using data science and the development of a model that will provide the best results in revealing and preventing fraudulent transactions. This is achieved by putting together the meaningful features of card users' transactions (as seen in the dataset description). This information is then run through a trained model which analyzes patterns to be able to classify whether a transaction is fraudulent or legitimate.

Machine Learning steps implementation also involves Data Mining, which implies classifying and grouping data to search millions of transactions to find patterns and detect fraud. Pattern recognition involves detecting classes, clusters and patterns of suspicious behavior. Machine Learning represents the selection of a model or a defined set of models that fits a business problem.

### 3.3.1 AMOUNT AND QUALITY OF DATA

Training high-quality Machine Learning models requires significant internal historical data. That means if you do not have enough previous fraudulent and normal transactions, it would be hard to run a Machine Learning model on it because the quality of its training process depends on the quality of the inputs. Because it is rarely the case that a training set contains an equal amount of data samples in two classes, dimensionality reduction or data augmentation techniques are used for that. PCA is one of the most popular techniques for Anomaly Detection. It searches for correlations among features, which in the case of credit card transactions, is time, location, and amount of money spent and determines which combination of values contributes to the variability in the outcomes.

### 3.3.2 SUPERVISED AND UNSUPERVISED LEARNING TECHNIQUES

Unsupervised learning is a type of algorithm that learns patterns from untagged data, using these algorithms to analyse and cluster. Unsupervised learning provides an exploratory path to view data, allowing businesses to identify patterns in large volumes of data more quickly when compared to manual observation. The implementation of this method is particularly common in computer vision, computer anomaly, anomaly detection, and recommendation engines. Unsupervised Machine Learning methods use unlabelled data to find patterns and dependencies in the credit card fraud detection dataset, making it possible to group data samples by similarities without manual labelling.

While unsupervised learning has a lot of benefits in applying its methodology, some challenges can occur when it allows machine learning models to execute without any human intervention. Some of these include:

* Longer training times
* A higher risk of inaccurate results
* Computational complexity due to a high volume of training data

Supervised Learning is a subcategory of machine learning and artificial intelligence, defined by its use of labelled datasets to train algorithms to classify data or predict outcomes accurately. It uses a training set to teach models to yield the desired output. Supervised learning can be separated into two types of problems in the processes of data mining: Classification and Regression: Classification uses an algorithm to accurately assign test data into certain categories. It recognizes specific entities within the dataset and attempts to draw conclusions on how those entities should be labelled or defined. Classification algorithms include linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbour, and random forest. Regression is used to understand the relationship between dependent and independent variables. Popular regression algorithms include logistical regression, linear regression and polynomial regression.

Supervised learning models may offer invaluable business advantages such as improved automation, but there are a few challenges when building sustainable supervised learning models. Some of these challenges include:

* Datasets can have a higher likelihood of human error, resulting in algorithms learning incorrectly.
* Training supervised learning models can be very time intensive
* Unlike unsupervised learning models, supervised learning cannot cluster or classify data on its own.
* Supervised learning models can require certain levels of expertise to structure accurately.

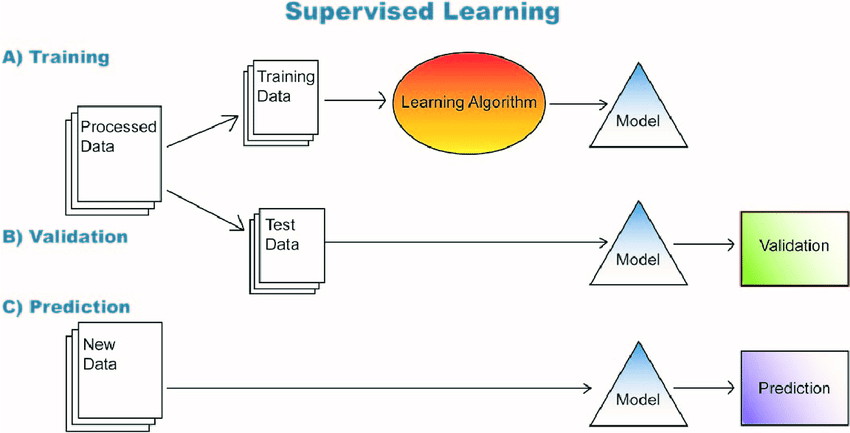


Figure 3.3.1 Supervised Learning

### 

### 3.3.3 SOFTWARE AND DEPENDENCIES USED

Throughout this study the use of machine learning, the software in use is Jupyter Notebook, a web based interactive computing platform provided by Project Jupyter. The notebook combines live code, equations, narration text, visualization and offers a streamlined, document-centric experience. It uses the IPython variable in shell. IPython is an interactive shell that is built with Python. It provides a more useful shell environment to execute python code in REPL (Read Eval Print Loop). It provides more interactive features like syntax highlighting, code completion, etc. Jupyter Notebooks may be just one of many web based interactive software, but it is certainly one of the better favored software for project development.

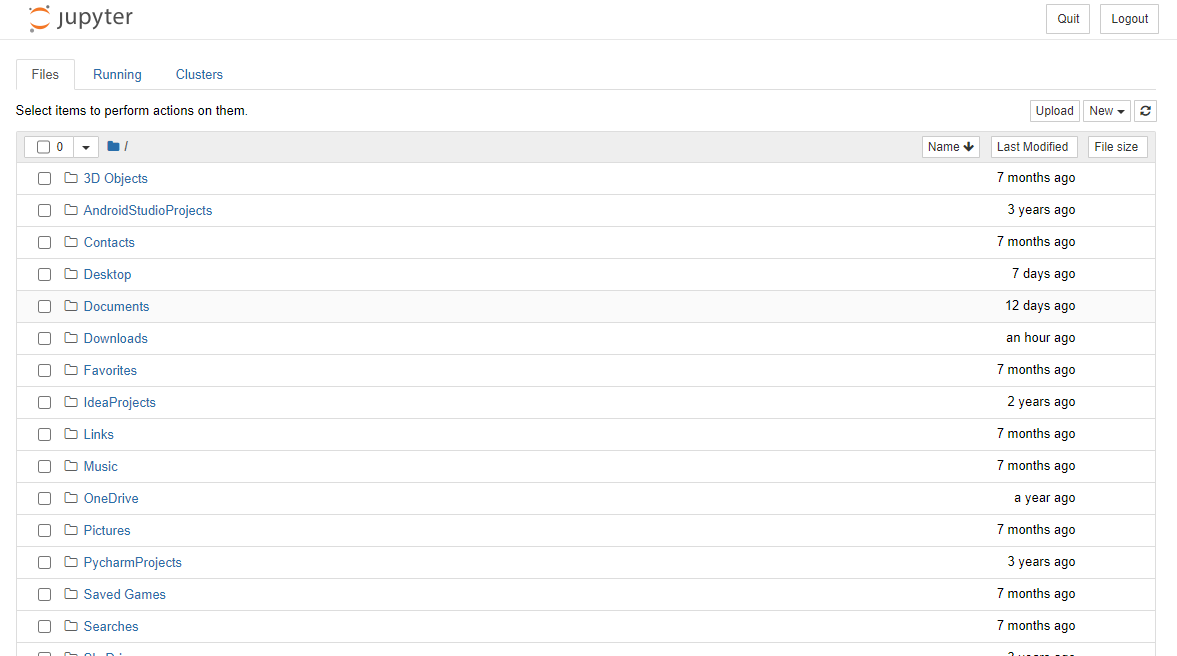


Figure 3.3.2 The Jupyter Web Interface Landing Page after being launched by the Command Prompt

Below are some of the dependencies used during the course of this project study:

* NumPy: NumPy is a Python library used for working arrays. It provides a high performance multidimensional array and tools to manipulate those arrays. It also has functions for working in the domain of linear algebra, Fourier transform and matrices.
* Pandas: The Pandas module is an open source python library that provides high performance data structures and data analysis tools. It is used to process data from csv files for analysis and processing. Pandas is also capable of offering an in-memory 2d table object called Data Frame.
* Sklearn: This is the most robust library for machine learning in Python. It provides a vast selection of efficient tools for machine learning and statistical modeling. This includes classification, regression, clustering and dimensionality reduction. Note that importing sklearn functions have to be specified and issued at the beginning of the project data.
* Scipy: Scipy builds on NumPy, providing a large number of functions that operate on NumPy arrays. It is useful for different scientific, engineering and mathematical applications. It allows the user to manipulate and visualize data using a wide range of high level commands.
* Matplotlib: This is a cross-platform library for making 2d plots from data in arrays, providing data visualization and graphical plotting library, and it's numerical extension NumPy. Jupyter Notebook is able to display plots if code in input cells and works seamlessly with matplotlib library.
* Pylab: This is a module that provides a namespace by importing functions from the modules NumPy and Matplotlib. It gets installed alongside matplotlib as a module.
* Seaborn: Seaborn aids in better understanding of the data by making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structure.

For their usage and implementation, they are imported as such:

Figure 3.3.3 Basic Dependencies needed for this project work

With the intricate use of attributes of Principal Component Analysis (PCA) in conventional data transformation, methods have been involved almost extraordinarily amidst the continuos usage of evolutionary algorithms to automate the optimization of parameters and attributes of existing Machine Learning methods. It is of this basis that hybrid machine learning algorithms are formulated on an architecture that is slightly different from the conventional work flow.

The hybrid learning model seamlessly combines different algorithms and procedures from similar or different expansions of knowledge or areas of application with the objective of complementing or enhanced by each other.

It is common knowledge that no single machine learning method can fully be applicable to solve a problem. As displayed in previous chapters, accuracy and precision scores obtained using a singular model in comparison to others were subject to certain drawbacks such as the lack of adequate fraudulent data for an effective training set and the technique's performance varying depending on the environment initiated in. The aim of a hybrid approach is to merge two or more technique’s together to give better, accurate and sustainable result (Harwani et al., 2020). Another intricate focus is to reduce the number of false positives, false declines in which a legitimate transaction is flagged as fraudulent. This is a very unfavourable situation which can greatly affect an institution.

This hybrid framework proposes a boosting-like algorithm in applications with limited training data and existing knowledge bases in the form of analytical models and indicators (Figure 3.3.4). Through the use of further implemented algorithms such as Isolation Forest and Outlier Detection, with Logistic Regression as a viable addition for a more accurate model. This can be possible with the aforementioned Scipy dependency, which will bring up notable libraries available for import.

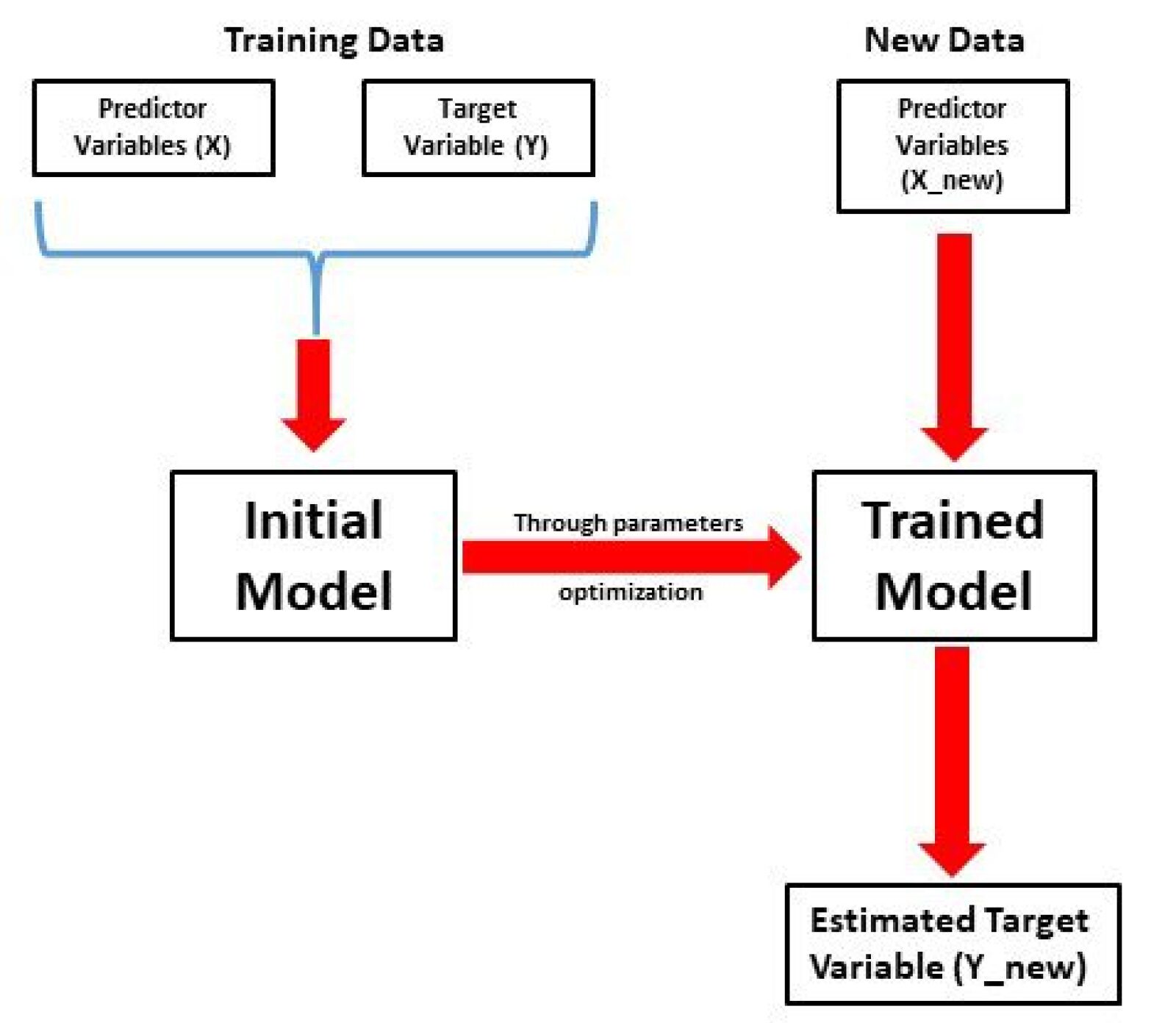


Figure 3.3.4 Outline of a functional hybrid model process

Evaluation Metrics

To effectively assess the statistical compounds surrounding this research, ...of certain evaluation metrics have to be considered. Evaluating machine learning models or algorithms is essential for any project because they are used to measure the quality of the statistical or machine learning model. These metrics are particularly necessary in credit card fraud detection mainly because of the frequency is false declines and false positives. Institutions take extreme care in preventing flagging a legitimate transaction as fraudulent as it greatly affects customer relationships and their reputation. A crucial part of building an efficient predictive model lies in the evaluation of the model. The most frequent metric used is ‘Accuracy’. High Accuracy does not necessarily mean that the model is performing better in every situation or environment. The model involved may not always be considered accurate as it may sometimes become misleading in certain situations like imbalanced class datasets. Several of these metrics and attributes have been ...and many others have been implemented in this research, as well as other fields. Metrics like Accuracy, Precision and Recall are among the quintessential classification metric easily suited for binary as well as a multiclass classification problem. This feature is attributed to F1 score, which maintains a balance between the precision and recall for a classifier.

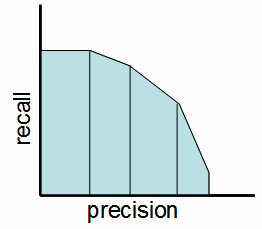


Figure 3.3.5. Precision - Recall Trade-off

Because the bulk of analysis is focused on credit card fraud detection, the evaluation of the performance of the Model can be based on just a few metrics, such as:

Confusion Matrix

A confusion matrix is an N x N matrix that is used for the evaluation of the performance of a certain classification model, where N is the number of target classes. The matrix gives a comparison involving the actual target values and those predicted by the machine learning model. It is a technique for generally summarizing the performance of a classification model or algorithm.It provides more astute, intricate and insightful details on the performance of the classification model.

The usage of confusion matrix is \*prominent\* in displaying the values of the number of predictions between the true and predicted values of each class.

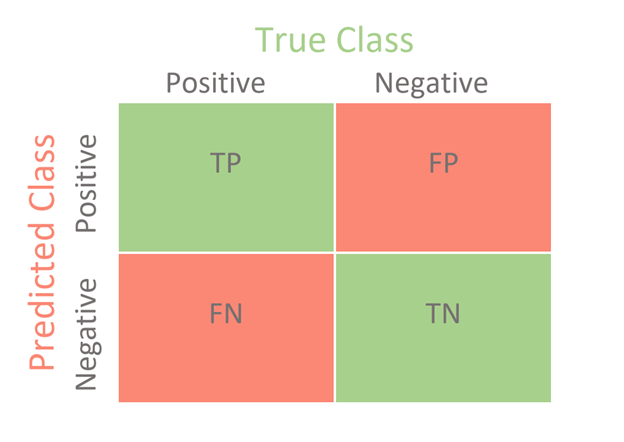


Figure 3.3.6. Class Distribution of the working model

Understanding True Positive, True Negative, False Positive and False Negative in a Confusion Matrix

True Positive (TP): Here, the predicted value matches the actual value. This means that the actual value was positive and the model predicted a positive value

Therefore, it can be said that the Observation is Positive, and the model classified it as Positive.

True Negative (TN): Here, the predicted value matches the actual value. This means that the actual value was negative and the model predicted a negative value.

Therefore, it can be said that the Observation is Negative, but the model classified it as Positive.

False Positive (FP): This is also known as the Type 1 error. In this scenario, the predicted value was falsely predicted because the actual value was negative but the model predicted a positive value.

Therefore, it can be said that the Observation is Negative, but the model classified it as Positive

False Negative (FN): This is also known as the Type 2 error. In this scenario, the predicted value was falsely predicted because the actual value was positive but the model predicted a negative value.

Therefore, it can be said that the Observation is Positive, but the model classified it as Negative.

The Confusion Matrix in according to our problem:

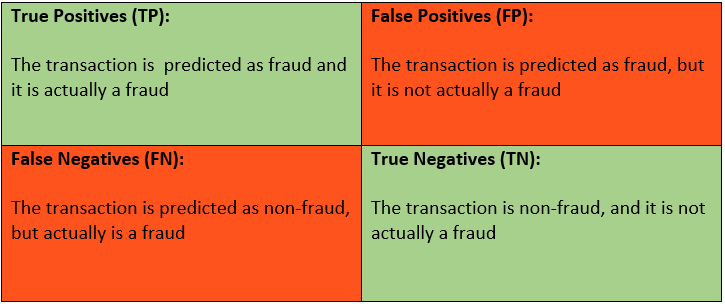


Figure 3.3.7 Outline of Class Distribution effect

Sklearn has two great functions: confusion\_matrix() and classification\_report(). Sklearn confusion\_matrix() returns the values of the Confusion matrix. The output given is slightly different. It shows that it takes and accesses the rows as Actual values and the columns as Predicted values. The rest of the concept remains the same. Sklearn classification\_report() outputs precision, recall and f1-score for each target class. In addition to this, it also has some extra values: micro avg, macro avg, and weighted avg.

# confusion matrix in sklearn

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import classification\_report

# actual values

actual = [1,0,0,1,0,0,1,0,0,1]

# predicted values

predicted = [1,0,0,1,0,0,0,1,0,0]

# confusion matrix

matrix = confusion\_matrix(actual,predicted, labels=[1,0])

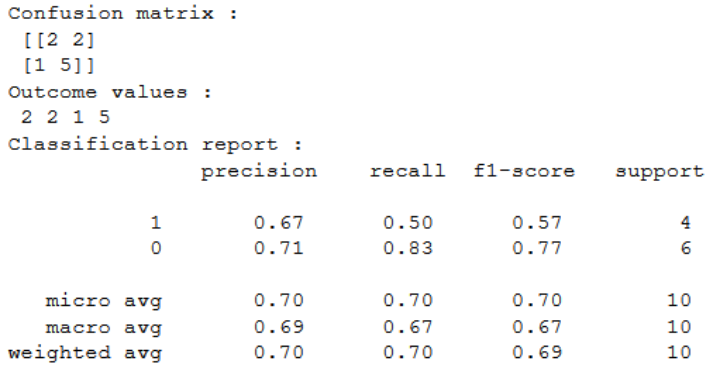
print('Confusion matrix : \n',matrix)

# outcome values order in sklearn

tp, fn, fp, tn = confusion\_matrix(actual,predicted,labels=[1,0]).reshape(-1)

print('Outcome values : \n', tp, fn, fp, tn)

# classification report for precision, recall f1-score and accuracy

matrix = classification\_report(actual,predicted,labels=[1,0])

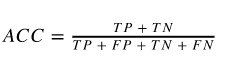
print('Classification report : \n',matrix)

Figure 3.3.8 Confusion Matrix with the Scikit-learn library in Python

Accuracy

Although Accuracy is not recommended for imbalanced data, because the great number of correct predictions of the negative class will make the accuracy high, even if we have a lot of wrong predictions for the positive class.

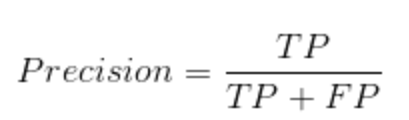
Equation (i);



Precision vs. Recall

Precision gives a definite description of how many of the correctly predicted cases actually turned out to be positive.

To calculate Precision:

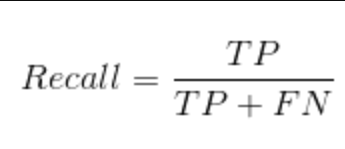


Equation (ii);

This would determine whether the model is reliable or not. Low precision means the more false positives are predicted by the model

Recall describes how many of the actual positive cases that was able to be predicted correctly with the model.

To calculate Recall:



Equation (iii);

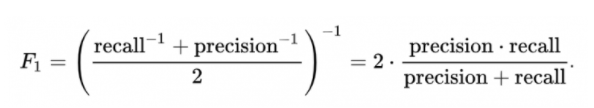
Equation 3 depicting evaluation of Recall

Recall focuses on outlining the proportion of actual positive cases that are correctly identified. It can also be regarded as the ratio of True Positives to all the positives in the dataset. Low recall means the more false negatives the model predicts

Despite their seemingly clashing attributes, Precision and Recall are useful for imbalanced datasets, because they don't involve the true negatives. They are only concerned with the correct prediction of the positive class.

F1 Score

The F1 Score is used when both the scores of precision and recall are needed for the evaluation of the model



It is the harmonic mean of precision and recall values for a classification problem.The F1 Score maintains a balance between the precision and recall for the classifier. If the precision is low, the F1 is low and if the recall is low again the F1 score is low.

AUC-ROC curve (Area Under Curve — Receiver Operating Characteristic Curve)

AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes. AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values. The ROC is a trade-off between the True Positive Rate (TPR) and False Positive Rate (FTR) for a predictive model using different probability thresholds.The True Positive Rate (TPR) is plot against False Positive Rate for the probabilities of the classifier predictions. The area under the curve is then calculated. The False Positive Rate is the probability of a false alarm (Figure 10).

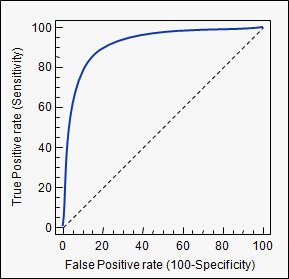


Figure 3.3.9 True Positive and False Positive relation

TPR is also known as Recall, therefore: True Positives/True Positives + False Negatives

FPR is known as the negation of specificity = 1 - Specificity = 1- True Negatives/(True Negatives + False Positives)

Therefore FPR = False Positives/(True Negatives + False Positives)

Here, the ROC curves can be used to decide on a Threshold value. The choice of threshold value will also depend on how the classifier is intended to be used.

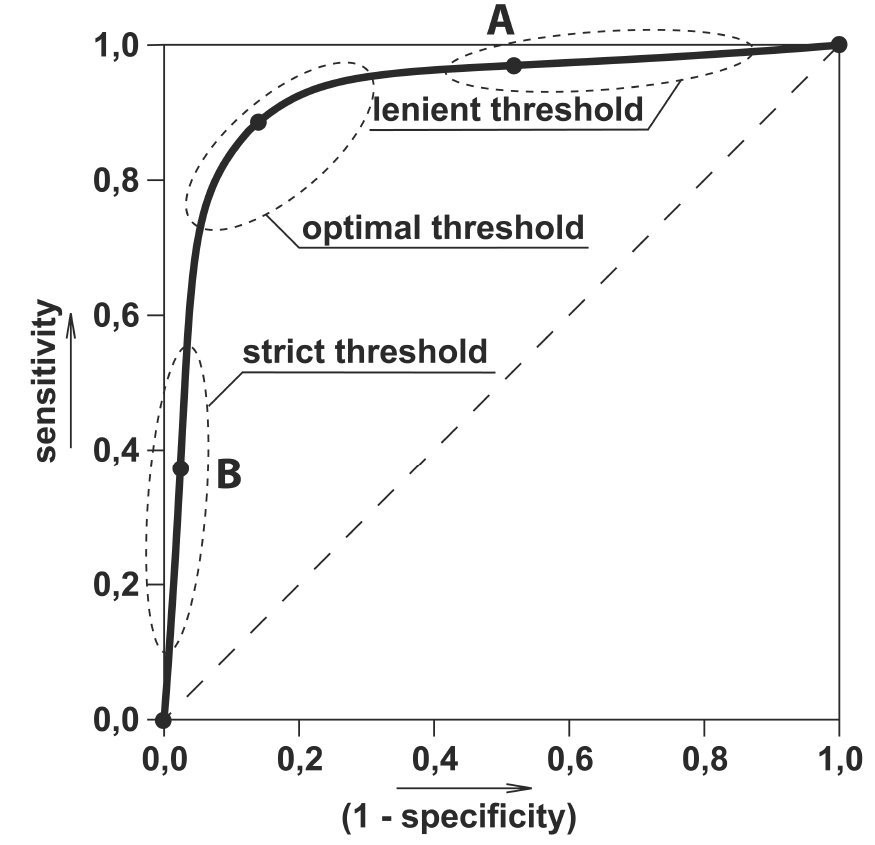


Figure 3.4. Threshold Specificity

AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1).

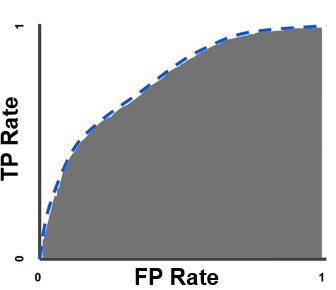


Figure 3.4.1 ROC Curve of True Positive and False Positive Rate

Hybrid Algorithms

Hybrid methods combine two or more machine learning computing methods for higher performance with optimum results. Generally, in machine learning, the combining of models is done by using two approaches, “Ensemble Models” and “Hybrid Models”. Ensemble Models use multiple machine learning algorithms to bring out improved predictive results, in comparison to using a single algorithm. Despite it's perceived nature There are different approaches in Ensemble models to perform a particular task.

Exploring Logistic Regression

Logistic Regression is commonly used to estimate the probabilities than on instance belong to a particular class. In this case, the model developed uses logistic regression to build the classifier to prevent frauds in credit card transactions, basically known as a binary classifier.

Logistic Regression constitutes of several hyperpameters; C, Solver, Penalty and Max\_iter.

C: This is a control parameter that has full control of the penalty strength. The higher the value of C, the less the model is standardized.

Solver: It is of great significance to try different solvers as each solver's performance or convergence is notably different from others.

Penalty: Here, it is possible to specify regularization Techniques.

Max\_iter: This is the maximum number of iterations taken for the solver to converge.

An example of Logistic Regression performance on an undersampled data:

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import GridSearchCV

logmodel= LogisticRegression()

param\_grid = {

‘penalty’ : [‘l1’, ‘l2’, ‘none’],

‘C’ : np.logspace(-4, 4, 20),

‘solver’ : [‘lbfgs’,’newton-cg’,’liblinear’],

‘max\_iter’ : [100, 1000,2500]

}

logcls\_us = GridSearchCV(logmodel, param\_grid = param\_grid, cv = 3, verbose=3, n\_jobs=-1)

bestlogcls\_us = logcls\_us.fit(X\_train\_us, y\_train\_us)

bestlogcls\_us.best\_estimator\_

The best parameters were found as:

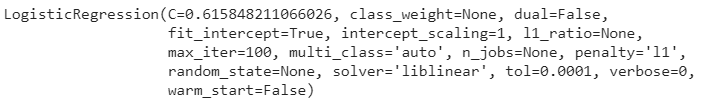


Figure 3.4.2 Logistics Analytics

from sklearn.metrics import confusion\_matrix, accuracy\_score, f1\_score, precision\_score, recall\_score, classification\_report, roc\_curve, auc

def model\_performance(y\_test, y\_pred):

print(f’Confusion Matrix:\n ‘,confusion\_matrix(y\_test, y\_pred))

print(f’\n Recall:’, recall\_score(y\_test, y\_pred))

print(f’\n Accuracy Score:’, accuracy\_score(y\_test, y\_pred))

print(f’\n Precision Score:’, precision\_score(y\_test, y\_pred))

print(f’\n F1 Score:’ ,f1\_score(y\_test, y\_pred))

compute\_roc(y\_test, y\_pred, plot=True)

return

predict\_us = bestlogcls\_us.predict(X\_test\_us)

model\_performance(y\_test\_us,predict\_us)

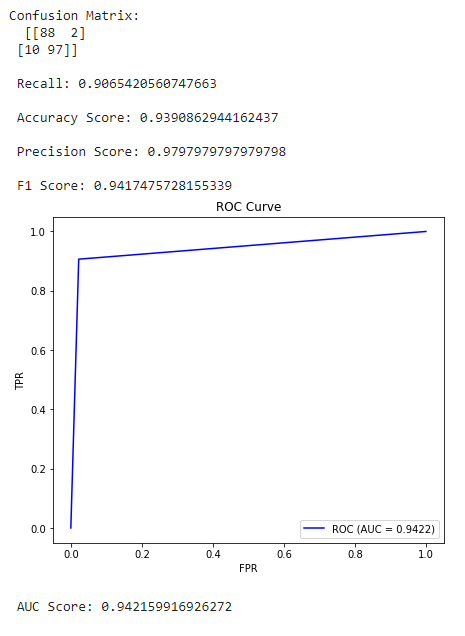


Figure 3.4.3 Analytics of Logistics Results; AUC yielding 94% Accuracy

For the model performance, a recall score of 90% was recorded. This classification states that 90% of the total Fraud transaction is correctly predicted by the classification model. The Accuracy score obtained was 93%.

Performing this same on an oversampled data;

param\_grid = {

‘penalty’ : [‘l1’, ‘l2’, ‘none’],

‘C’ : np.logspace(-4, 4, 20),

‘solver’ : [‘lbfgs’,’newton-cg’,’liblinear’],

‘max\_iter’ : [100, 1000]

}

logcls\_os = GridSearchCV(logmodel, param\_grid = param\_grid, cv = 3, verbose=3, n\_jobs=-1)

bestlogcls\_os = logcls\_os.fit(X\_train\_os, y\_train\_os)

bestlogcls\_os.estimator

The best parameters were found as:

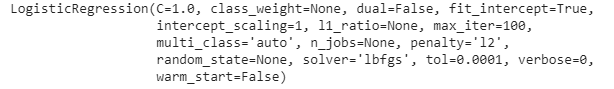


Figure 3.4.5 Analytics on Oversampled data

predict\_os = bestlogcls\_os.predict(X\_test\_os)

model\_performance(y\_test\_os,predict\_os)

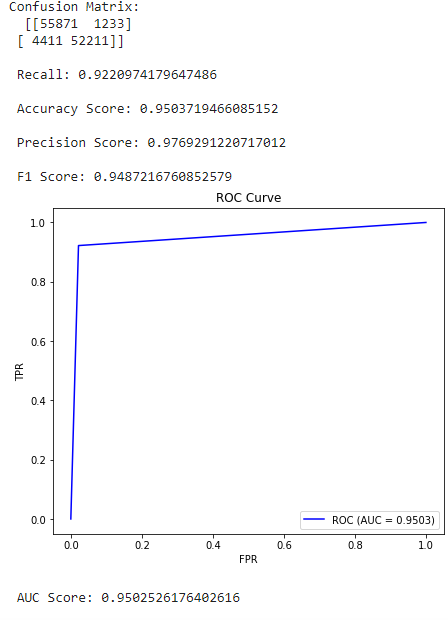


Figure 3.4.6 AUC Curve yields 95% on Oversampled data

For the model performance on an over-sampled dataset, there seems to be an improvement in the performance as in under-sampling data but there could also be the possibility of information loss which can have a negative effect on the model performance. A recall score of 92% was obtained, which states that 92% of the total fraud transaction is correctly predicted by the classification model. The Accuracy score obtained was 93%.

Exploring Isolation Forest and Outlier Detection

The idea behind Isolation forest lies in it's attempts to separate each point in the data. Here, an abberant point could be separated in a few steps while closer normal points could take significantly more steps to be isolated. Isolation forest is a tree-base model that is developed to detect anomalies and aberrant factors. It is based on the fact that anomalies are the data points which are few and different. These properties result in susceptible mechanism to anomalies which is known as Isolation. This method greatly differs from all other existing methods and is highly useful. In detecting anomalies, rather than the basic distance and density measures it introduces the use of isolation as an efficient, even more effectively. Isolation Forest has small memory requirement and low linear time complexity. It builds a good performing model with a small number of trees using small sub-samples of fixed size, regardless of the size of a data set.

M. Breunig, Hans-peter Kriegel, Raymond T. Ng and Jörg Sander introduced the Local Outlier Factor (LOF) Algorithm within the year 2000 and it's an algorithm to search out the anomalous informations by measuring the local deviation of a given data point with relation to its neighbours. Outliers supported the local density are detected using this algorithm. Locality is given by nearest neighbours and density is calculated by their distance. By comparing the local density of an object to the local densities of its neighbours, one can identify regions of comparable density, and points that have a substantially lower density than their neighbours. the info point is taken into account as an outlier if it's very small density as compared to its neighbours. Outlying patterns could also be divided into two types: global and local outliers. the object which is significantly having a large distance with regard to its k-th neighbour is termed Global outlier while as object whereas a neighborhood outlier includes a distance to its k-th neighbour that's large relatively to the average distance of its neighbours to their own k-th nearest neighbours.

# CHAPTER FOUR

# IMPLEMENTATION OF THE ANALYTICAL MODEL

## 4.1 FRAUD DETECTION USING LOGISTIC REGRESSION MODEL

Logistic regression is a statistical analysis method used to predict a data value based on prior observations of a dataset. The model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. Logistical regression is selected when the dependent variable is categorical. This means they have the features of outputs such as "True" and "False" or "Yes" and "No". This method is mainly used to solve binary classification problems, such as spam and identification.

In this work, fraud detection using a logistic regression model is proposed. This system uses logistic regression to build the classifier to prevent frauds in credit card transactions.

As shown in Figure 4.1.1, as with every program written throughout this study, dependencies have to be imported to ensure their implementation in the project work. As previously pointed out, this will include pandas for reading .csv files, NumPy for working arrays, and some sklearn functions. Model\_selection is a method for setting a blueprint to analyze data and using it to measure new data. This in conjunction with the train\_test\_split function which splits arrays or matrices into random train and test subsets, splitting the data into training and test data. The sklearn.linear\_model function is a logistic regression regression classifier, a classification algorithm rather than a regression algorithm, used to estimate discrete value like 0 or 1, yes/no, true/false. It is also called "Logit" or "MaxEnt Classifier".

The last dependency specified is a module that implements several loss, score and utility functions to measure classification performance. In this case, it deals with the accuracy classification score, which computes the subset accuracy. It returns the mean accuracy on the given test and data, and aids in checking the performance of the model.

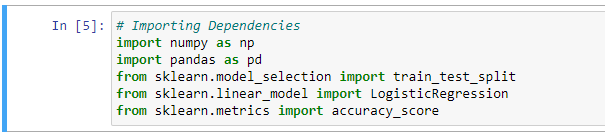


Figure 4.1.1. Dependencies while operating in Jupyter notebook

The downloaded format for the dataset from Kaggle is in a .csv file. This is then specified, along with the path in a variable. This variable is then used for the necessary information obtained and operated on.

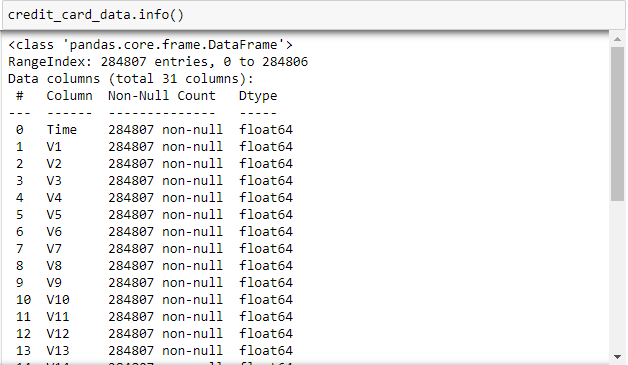


Figure 4.1.2. A general information overview of the dataset

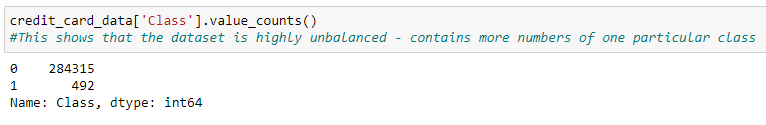


Figure 19a. A description of the Class Value counts of the respective cases presented

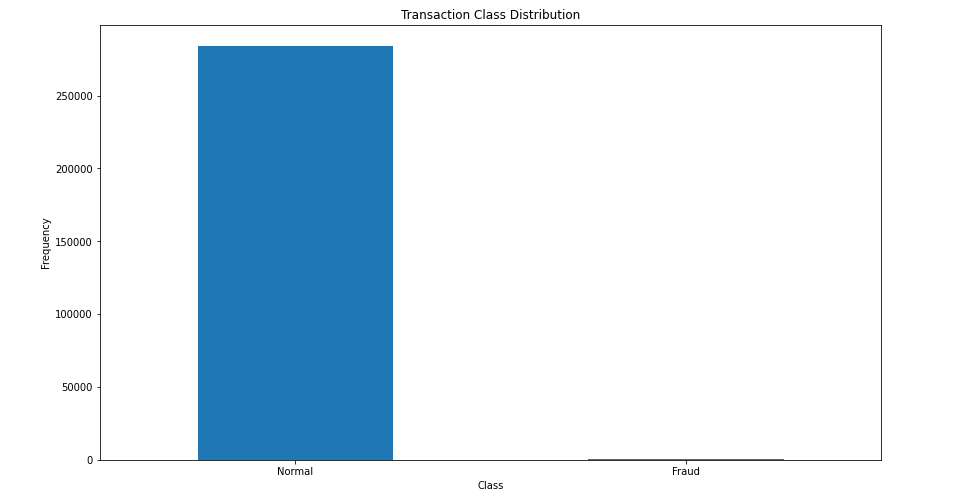


Figure 4.1.3. A Graphical Transaction Class Distribution of the two relevant cases

The output shown in Figure 4.1.3 shows that this dataset is highly unbalanced. An unbalanced dataset is one in which the target variable has more observations in one specific class than the others or contains more numbers of one particular class. A big challenge which will be addressed is that models trained on unbalanced datasets often have poor results when predicting a class or classifying unseen observations.

For the data to be accurately studied and progressed, it will need to be separated for analysis, with each class divided into variables:



Figure 4.1.4. The class data of the respective cases are stored into two distinct variables

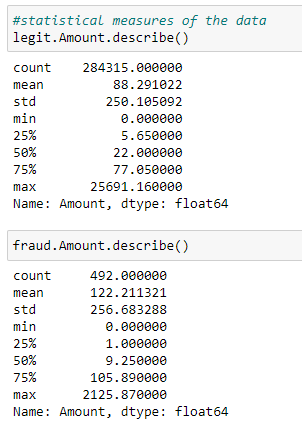


Figure 4.1.5. A statistical description of the divided variables

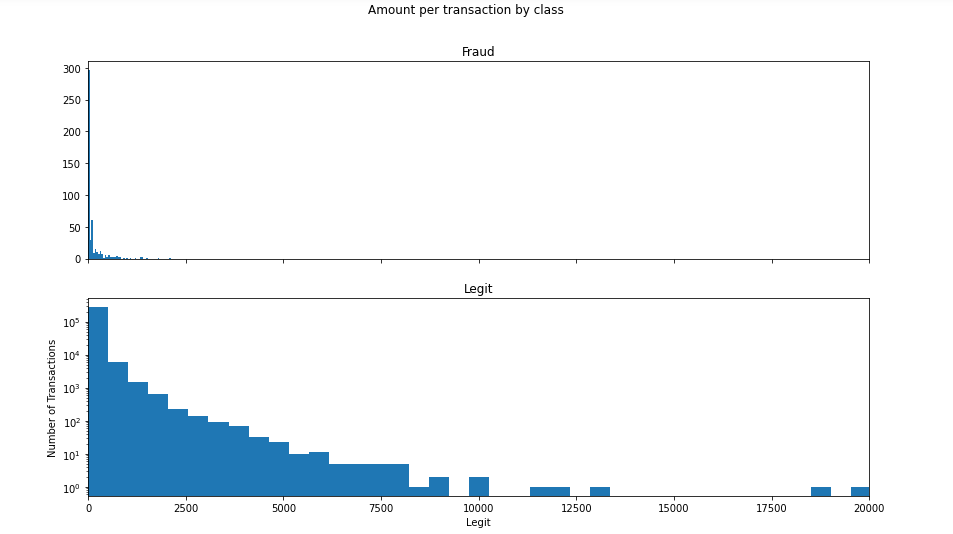


Figure 4.1.6. A graphical representation of amount per transaction of both cases

When solving the unbalanced dataset issue, under sampling is implemented by building a sample dataset containing similar distribution of normal transactions and fraudulent transactions. Under sampling refers to a group of techniques designed to balance the class distribution for a classification dataset having an uneven class distribution. This was done by taking a sample out from the class of legitimate transactions (since they are the dominant Class value) and downsizing the value to that of the fraudulent variable.

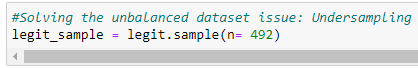


Figure 22. Successfully downsizing the number of legitimate transactions to equal that of fraudulent transactions.

The new variable sample as a data frame is then concatenated with the other previously defined data frame to form a new dataset.



Figure 4.1.7. The newly defined sample variable is concatenated with the fraudulent transactions with a row column into a new dataset variable

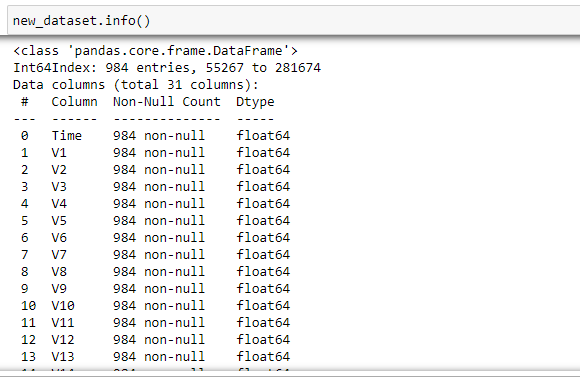


Figure 4.1.8. descriptive information of the new dataset used

As shown in Figure. 25, the number of entries to be used for the dataset are changed and selected at random, as read by the pandas module and shows the change in the even distribution of values.

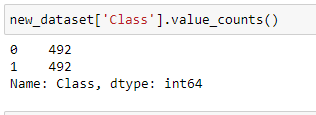
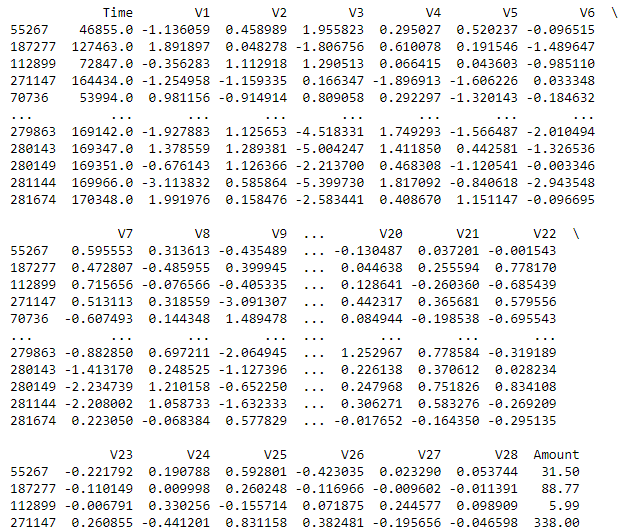


Figure 4.1.9. The value of the legit samples is downsized to equal value as the fraudulent transactions

This new defined dataset is then split into Features and Targets. The target variable of a dataset is the feature of a dataset about which you want to gain a deeper understanding. Features are important as they are building blocks of a dataset, upon which the model is designed and developed.



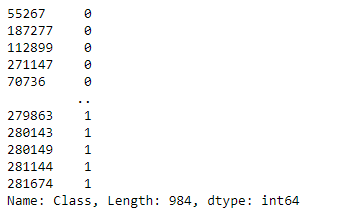


Figure 4.2.1 and 4.2.2 showing the output after column for variable X is dropped

There are two main features when training the Logistic Regression model; the train and test model. This involved creating four variables of equal training and their relative testing values using 20% of the data. This was selectively created in such a way that while 80% of the data goes to the training set, 20% will go to the testing data. The basis of these parameters is used to test the accuracy score in the model evaluation, observing the evenly distributed classes of data in both the training and testing data.

Putting the finishing touches to the model, the use of the function, "fit" to mould the data to Logistic Regression Model. The variable X\_train contains features of the training data and Y\_train contains binary labels.

These will fit our prepared data into the logistic regression model and predictions and conclusions can be derived from it.

In testing this model, only the values of X\_train are given to the model and it will predict the class of the value. It is observed that the Accuracy on Training data is 93%. But it is also noted that the accuracy score on test data is also of great significance. After running the model on the testing data was 95%. This is the best outcome possible as it shows all carefully planned steps were performed correctly. The worst case scenario is if the corresponding results were far apart from one another. If the accuracy score on Training data is very different from the accuracy score of test data then it means the model is over fit or under fit.

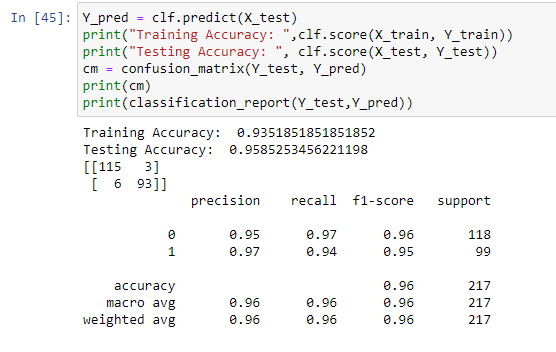


Figure 4.2.3. Final Training and Test results; 93% and 95% respectively.

### 4.1.1 LOGISTIC REGRESSION IN A DISTRIBUTED DATAFRAME

In the case of a distributed dataframe, using Logistic Regression; 

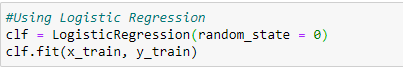


Figure 4.2.4 and 4.2.5 showing the use of Scikit-learning to improve upon the dependencies

Figure 4.2.6 shows that when evaluating the model, it is found that the test set has 99.9914% accuracy. This doesn’t mean that the model is perfect.

1. Position (0,0): True positives i.e. how many safe cases are there that our model predicted correctly. Here 71069 are the number of CORRECTLY PREDICTED safe cases.
2. Position (0,1): False Negatives, i.e. how many safe cases are there that our model predicted incorrectly. Here, the number of the misclassified cases is 4. Hence 4 legitimate cases were misclassified as fraudulent. This potentially causes little threat as it's better to stop some safe transactions with slightest chance of fraud.
3. Position (1,0): False Positives, fraudulent cases that the model predicted incorrectly. Here, the number of the misclassified fraud cases is 57. These cases were misclassified as safe. This can create great loss to the organization and poses a great threat because the fraudulent cases get a pass.
4. Position (1,1): True negatives, fraud cases that the model predicted correctly. Here, the number of correctly predicted fraud cases is 72.

Despite having an accuracy over 99.9%, the model predicted 57 fraud cases incorrectly. This is known as Accuracy Fallacy.

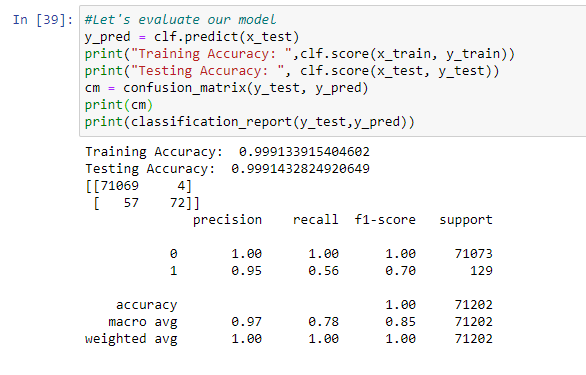


Figure 4.2.6. Training and Test accuracy model for a distributed dataframe yields 99% each.

For measuring model performance, precision, recall, f1 score and AUC-ROC curve is needed for this rectification. It is important to note that  the greater the F1 score the better it is.

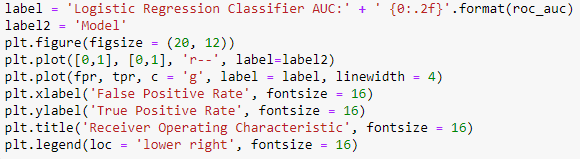


Figure 4.2.7. ROC Curve is needed for fallacy reduction

Calculating the True Positive Rate and the False Positive Rate;



Plotting the Curves of the thresholds;



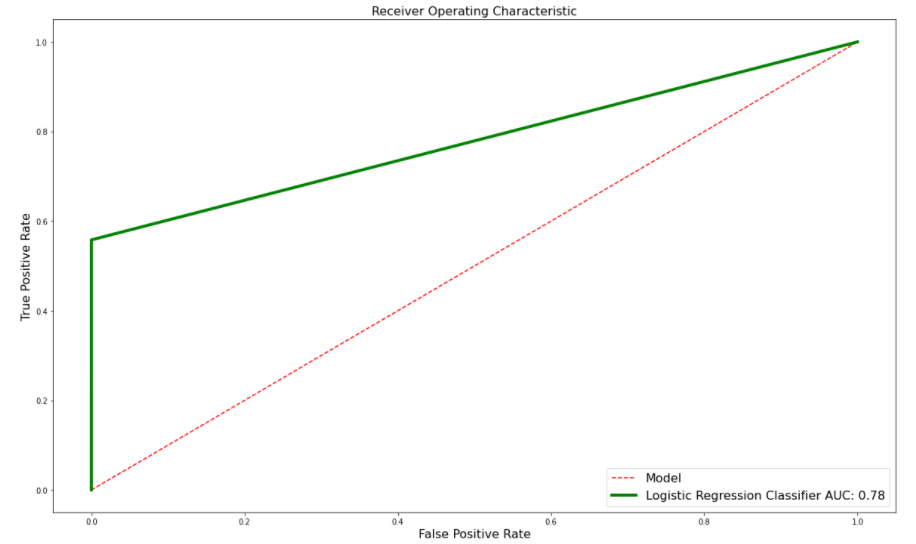
****

Figure 4.2.7. The result of a Logistic Regression Model without accuracy fallacy, yielding 0.78

The AUC parameter used for the accuracy of the Logistic Regression Model has an accuracy of 0.78, a relatively fair score in comparison.

## 4.2. OUTLIER DETECTION FOR UNSUPERVISED LEARNING ALGORITHM

### 4.2.1 ISOLATION FOREST ALGORITHM

One of the newest techniques to detect anomalies is called Isolation Forests. The algorithm is based on the fact that anomalies are data points that are few and different. As a result to these properties, anomalies are susceptible to a mechanism called isolation.

This method is highly useful and fundamentally different from all existing methods. It introduces the use of isolation as a more effective and efficient means to detect anomalies than the commonly used basic distance and density measures. Moreover, this method is an algorithm with a low linear time complexity and a small memory requirement. It builds a good performing model with a small number of trees using small sub-samples of fixed size, regardless of a size of a dataset.

Typical machine learning methods tend to work better when the patterns they try to learn are balanced, meaning the same amount of good and bad behaviours are present in the dataset.

How Isolation Forest Works

The algorithm isolates observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature. Isolating anomaly observations is easier because only a few conditions are needed to separate those cases from the normal observations. On the other hand, isolating normal observations require more conditions. Therefore, an anomaly score can be calculated as the number of conditions required to separate a given observation. When creating a decision tree, the outliers will be split initially (at the initial depth) because outlier is a completely different value so the root node will be selected in such a way that the outlier will be split. If the root node gets divided, all valid data points will be moved. In the extremely rare cases of outliers, less number of values will be moved or split. The score of the root node at one end will be much less than the assigned score, because the depth will increase. The leaf node is gotten quickly and the derived scores will be much less. It also works on anomaly score.

### 4.2.2 LOCAL OUTLIER FACTOR

The LOF algorithm is an unsupervised outlier detection method which computes the local density deviation of a given data point with respect to its neighbours.

The number of neighbours considered (parameter n\_neighbours) is typically chosen 1 greater than the minimum number of objects a cluster has to contain, so that other objects can be local outliers relative to this cluster, and 2 smaller than the maximum number of close objects that can potentially be local outliers. In practice, such information is generally not available and taking n\_neighbours = 20, in this particular layout appear to work well in general.

## 4.3 ANOMALY DETECTION USING LOCAL OUTLIER FACTOR AND ISOLATION FOREST ALGORITHM

For the purpose of starting a new to alienate the accuracy score measure, it is imperative to effectively begin on a new slate. This will involve the following dependencies; NumPy for working arrays, Pandas for analysis and processing, matplotlib for graphical plotting (as is the case for plotly for statistical, financial, geographic, scientific, and 3-dimensional use-cases.) The filter warnings is set to 'ignore' never display warnings which match.

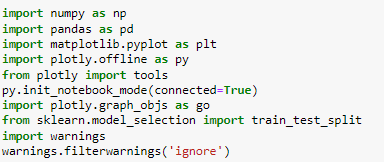


Figure 4.2.8. Importing the necessary dependencies for the model analysis

Shuffling all attributes in the dataset is highly important as it gives a fluid briefing at that level;

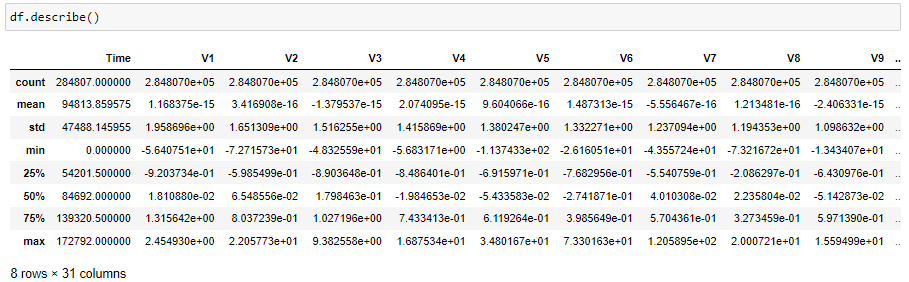


Figure 4.2.9. Attribute description of the dataset after shuffling

Viewing all columns in the dataset;

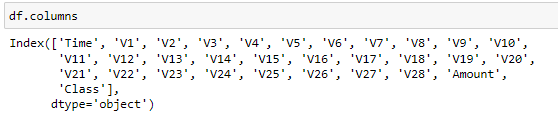


Figure 4.3.1. Columns in the dataset

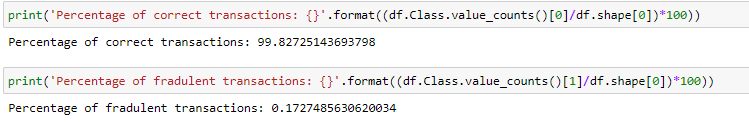


Figure 4.3.2. Percentages of both legitimate and fraudulent transactions

When plotting the heatmap for the graphical visualization of how much a feature affects the class, a sizable representation is required;

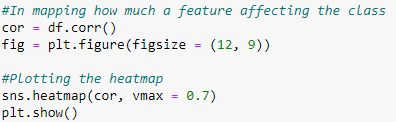


Figure 4.3.3. Graphical plotting of the Class Representation

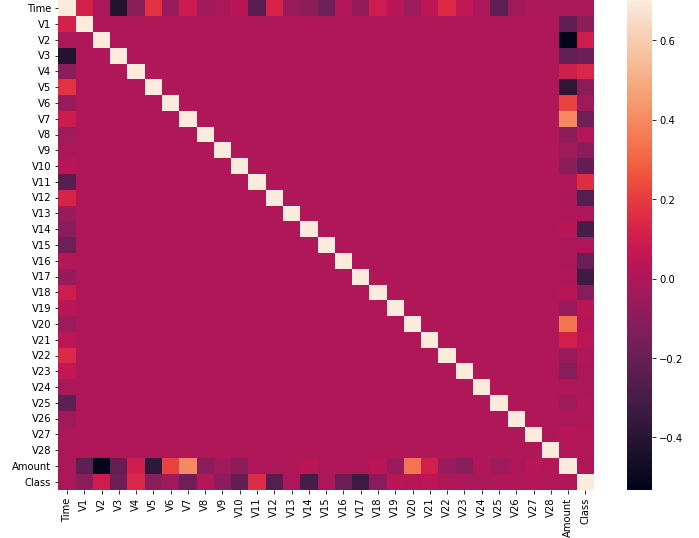


Figure 4.3.4. Heatmap of an attribute affecting the Class Distribution

For proper overall function of the model, there is the need to delete the least and greatest values. Using the above analysis, the features outlined below are selected. The Class Variable is included also because of the intention to create a new dataframe using the new features.

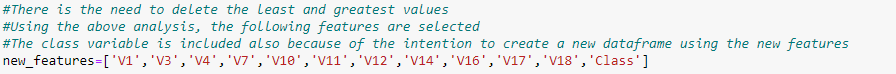
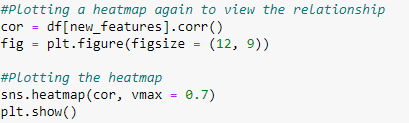


Figure 4.3.5. Newly extracted features from the dataset

Plotting the heatmap again to view the relationship;



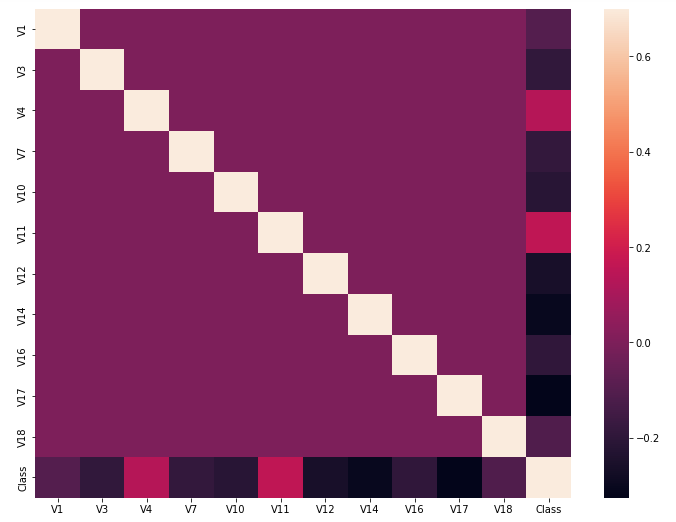
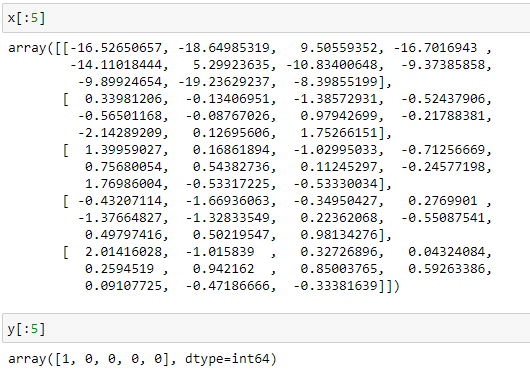


Figure 4.3.6. Effects of the removed values

It can be observed that the class rows and columns are darker and brighter, meaning all the variables in the new dataset have a significant affect.

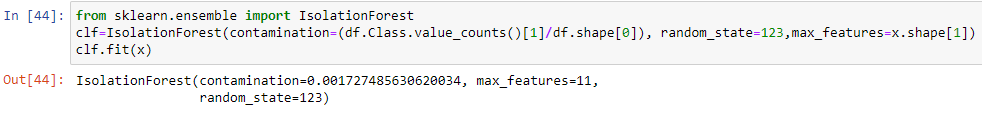
Now splitting the dataset into dependent variables (y) and independent variables(x);





Figur4.3.7. Array descriptions with respect to x.shape and y.shape

Now, importing and fitting the Isolation Forest Algorithm;



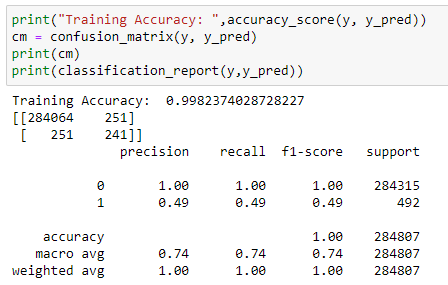
Predicting the Class; the tensor y\_pred which is the data predicted and calculated as output by the model;



Because the algorithm classifies one class as 1 and other as -1, it is important to note how many classes it predicted as fraudulent



Evaluating the model;



The Training Accuracy yielded 99%. To avoid the problem of accuracy fallacy;



label = 'Isolation Forest Classifier AUC:' + ' {0:.2f}'.format(roc\_auc)

label2 = 'Random Model'

plt.figure(figsize = (20, 12))

plt.plot([0,1], [0,1], 'r--', label=label2)

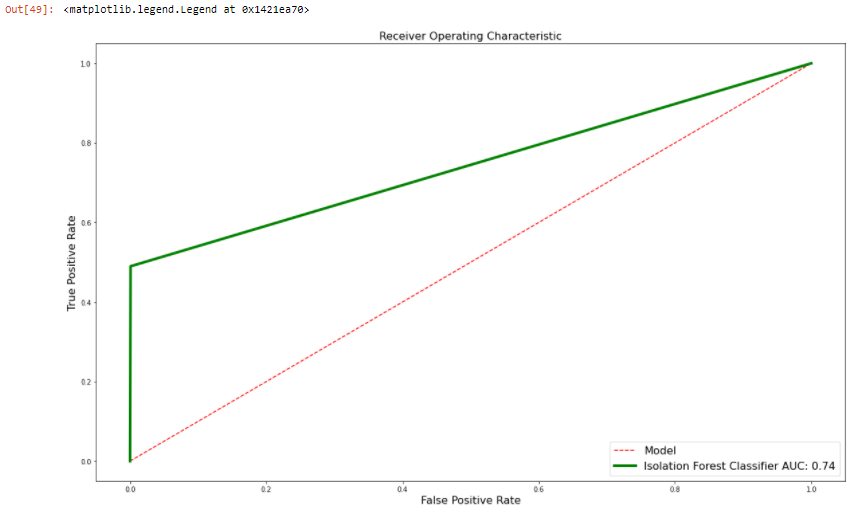
plt.plot(fpr, tpr, c = 'g', label = label, linewidth = 4)

plt.xlabel('False Positive Rate', fontsize = 16)

plt.ylabel('True Positive Rate', fontsize = 16)

plt.title('Receiver Operating Characteristic', fontsize = 16)

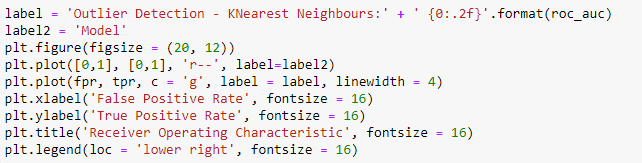
plt.legend(loc = 'lower right', fontsize = 16)

Figure. 4.3.8. Output of the Isolation Forest Algorithm yielding 74% accuracy

Since Local Outlier Factor and KNearest Neighbours are similar in terms of density-based scoring and captures and controls a certain degree of abnormality;



Plotting the layout of the model with respect to their characteristics;



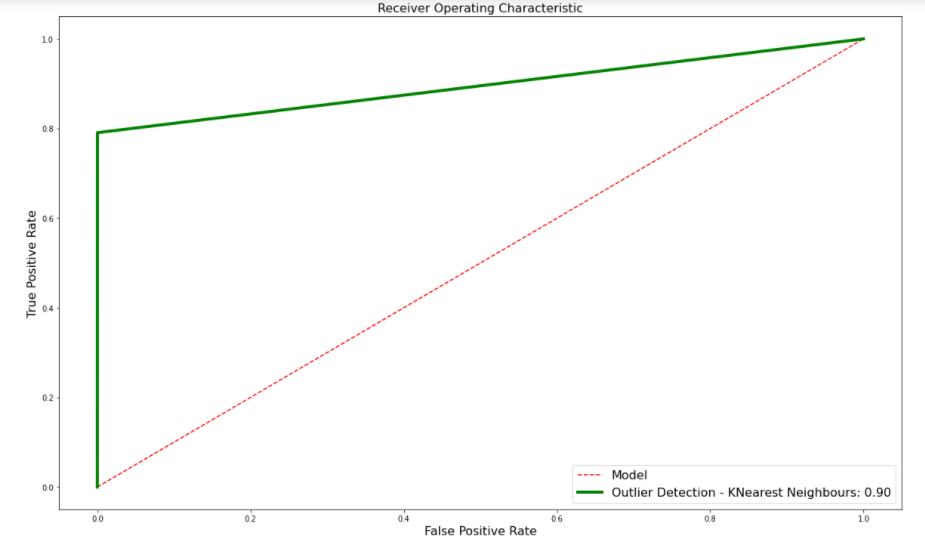


Figure 4.3.9. Outlier Detection Method’s AUC curve showing a 90% accuracy

# 

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATION

## 5.1 CONCLUSION

It can be gathered that using a selection of Random Models in the prediction of fraudulent activities has its challenges in their implementation. A notable frequency regarding the hindrance of the model reaching perfection is the inaccuracy of matrix parameters measured; a recurring error as regards the number of mismatched and miscalculated transactions in the False Positives region. Another inconsistency was with the inaccuracy of measurements, with most models for Class distribution with 99% accuracy. A model with such high accuracy does not necessarily mean perfection, known as Accuracy Fallacy. This phenomenon stems from the fact that although the overall score gives the illusion of nigh perfection, the matrix layout for True and False Positives, with True and False Negatives are not sufficient or accurate enough to warrant a pass for the model. As shown, the rectification methods used involved Area Under the Receiver Operating Characteristic Curve which illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is diversified.

## 5.2 CONTRIBUTION TO KNOWLEDGE

The evaluation and realization of a fully accurate model involved the use of several algorithms that were fitted to the model using evaluation metrics such as f1 score, confusion matrix, as well as accuracy, precision and recall. Accuracy score was used in the training and testing data for the Logistic Regression model which yielded 93% for Training data and 95% for Testing data, but when adjusting the features, both data parameters yielded 99%. This was quickly fixed by the usage of the ROC curve, which saw an AUC curve score of 78%. The Isolation Forest algorithm also had the problem of Accuracy Fallacy, which when rectified, yielded an accuracy of 74%, relatively because the number of False Negatives was reduced significantly. Also, Local outlier factor with KNN Classifier yielded an accuracy of 90%.

## 5.3 RECOMMENDATION

Although the lack of adequate fraudulent data for an effective training set proved to be a major thorn throughout the course of this research, they were still essential in preparing a model where the technique’s performance will vary depemding on the environment initiated in. That being said, more datasets for credit card fraud should be made for the purpose of research to find more effective ways to predict fraudulent activities. Institutions should adopt more advanced machine learning techniques to further improve their services in securing client’s data and protecting them by effectively predicting fraudulent activities in real time live events.

# 

# REFERENCES

Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud Detection in Credit Cards using Logistic Regression. *International Journal of Advanced Computer science and applications (IJACSA)*, *11*(12).

Akinola Kayode, E., Adekunle, Y. A., Adebayo, A. O., & Okolie, S. O. (2019). Multi-factor authentication model for integrating iris recognition into an automated teller machine. International Journal of Computer Applications, 181(45), 1-8.

Ayorinde, K. (2021). A Methodology for Detecting Credit Card Fraud (Doctoral dissertation, Minnesota State University, Mankato).

Behera, T. K., & Panigrahi, S. (2015, May). Credit card fraud detection: a hybrid approach using fuzzy clustering & neural network. In *2015 second international conference on advances in computing and communication engineering* (pp. 494-499). IEEE.

Bhusari, V., & Patil, S. (2016). Study of hidden markov model in credit card fraudulent detection. In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)* (pp. 1-4). IEEE.

Carcillo, F., Le Borgne, Y. A., Caelen, O., Kessaci, Y., Oblé, F., & Bontempi, G. (2021). Combining unsupervised and supervised learning in credit card fraud detection. Information sciences, 557, 317-331.

Carneiro, E. M., Dias, L. A. V., Da Cunha, A. M., & Mialaret, L. F. S. (2015, April). Cluster analysis and artificial neural networks: A case study in credit card fraud detection. In *2015 12th International Conference on Information Technology-New Generations* (pp. 122-126). IEEE.

Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., & Bontempi, G. (2015, July). Credit card fraud detection and concept-drift adaptation with delayed supervised information. In 2015 international joint conference on Neural networks (IJCNN) (pp. 1-8). IEEE.

Darwish, S. M. (2020). An intelligent credit card fraud detection approach based on semantic fusion of two classifiers. Soft Computing, 24(2), 1243-1253.

de Sousa Júnior, R. T., & Serrano, A. M. (2017, January). Predictive Analytics in Business Intelligence Systems via Gaussian Processes for Regression. In Knowledge Discovery, Knowledge Engineering and Knowledge Management: 7th International Joint Conference, IC3K 2015, Lisbon, Portugal, November 12-14, 2015, Revised Selected Papers (Vol. 631, p. 421). Springer.

Jain, Y., Tiwari, N., Dubey, S., & Jain, S. (2019). A comparative analysis of various credit card fraud detection techniques. Int J Recent Technol Eng, 7(5S2), 402-407.

Jiang, C., Song, J., Liu, G., Zheng, L., & Luan, W. (2018). Credit card fraud detection: A novel approach using aggregation strategy and feedback mechanism. IEEE Internet of Things Journal, 5(5), 3637-3647.

Kasa, N., Dahbura, A., Ravoori, C., & Adams, S. (2019, April). Improving credit card fraud detection by profiling and clustering accounts. In *2019 Systems and Information Engineering Design Symposium (SIEDS)* (pp. 1-6). IEEE.

Kim, J., Kim, H. J., & Kim, H. (2019). Fraud detection for job placement using hierarchical clusters-based deep neural networks. Applied Intelligence, 49(8), 2842-2861.

Makki, S. (2019). An efficient classification model for analyzing skewed data to detect frauds in the financial sector (Doctoral dissertation, Université de Lyon; Université libanaise).

Maniraj, S. P., 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).

Marchal, S., & Szyller, S. (2019, December). Detecting organized eCommerce fraud using scalable categorical clustering. In Proceedings of the 35th Annual Computer Security Applications Conference (215-228).

Mohari, A., Dowerah, J., Das, K., Koucher, F., & Bora, D. J. (2021). Credit Card Fraud Detection Techniques: A Review. *Soft Computing for Intelligent Systems*, 157-166.

Niu, X., Wang, L., & Yang, X. (2019). A comparison study of credit card fraud detection: Supervised versus unsupervised. arXiv preprint arXiv:1904.10604.

Patil, S., Nemade, V., & Soni, P. K. (2018). Predictive modelling for credit card fraud detection using data analytics. Procedia computer science, 132, 385-395.

Prakash, B., Murthy, G. V. M., Ashok, P., Prithvi, B. P., & Kira, S. S. H. (2018). ATM Card Fraud Detection System Using Machine Learning Techniques. Int. J. Res. Appl. Sci. Eng. Technol, 6(4), 5124-5129.

Rahmawati, D., Sarno, R., Fatichah, C., & Sunaryono, D. (2017, October). Fraud detection on event log of bank financial credit business process using Hidden Markov Model algorithm. In 2017 3rd International Conference on Science in Information Technology (ICSITech) (pp. 35-40). IEEE.

Raj, S. B. E., & Portia, A. A. (2011, March). Analysis on credit card fraud detection methods. In 2011 International Conference on Computer, Communication and Electrical Technology (ICCCET) (152-156). IEEE.

Sadineni, P. K. (2020, October). Detection of fraudulent transactions in credit card using machine learning algorithms. In 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) (pp. 659-660). IEEE.

Save, P., Tiwarekar, P., Jain, K. N., & Mahyavanshi, N. (2017). A novel idea for credit card fraud detection using decision tree. *International Journal of Computer Applications*, *161*(13).

Shirgave, S., Awati, C., More, R., & Patil, S. (2019). A Review On Credit Card Fraud Detection Using Machine Learning. Int. J. Sci. Technol. Res, 8, 1217-1220.

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019, January). Real-time credit card fraud detection using machine learning. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 488-493). IEEE.

Tran, P. H., Tran, K. P., Huong, T. T., Heuchenne, C., HienTran, P., & Le, T. M. H. (2018, February). Real time data-driven approaches for credit card fraud detection. In Proceedings of the 2018 international conference on e-business and applications (pp. 6-9).

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.

Vengatesan, K., Kumar, A., Yuvraj, S., KUMAR, V., & Sabnis, S. (2020). Credit card fraud detection using data analytic techniques. Advances in Mathematics: Scientific Journal, 9(3), 1185-1196.

Wang, C., & Han, D. (2019). Credit card fraud forecasting model based on clustering analysis and integrated support vector machine. Cluster Computing, 22(6), 13861-13866.

Warghade, S., Desai, S., & Patil, V. (2020). Credit Card Fraud Detection from Imbalanced Dataset Using Machine Learning Algorithm. International Journal of Computer Trends and Technology, 68, 22-28.