THE ACCURACY VERSUS INTERPRETABILITY TRADE-OFF IN THE APPLICATION OF MACHINE MODELS FOR PREDICTING FRAUDULENT CREDIT CARD TRANSACTIONS

AUTHOR: OLABISI DABI
SUPERVISOR: JARUTAS ANDRITSCH

Faculty of Business, Law and Digital Technologies
Solent University Southampton

A dissertation submitted in partial fulfillment of the requirement for the degree of Master of Science in Applied Artificial Intelligence and Data Science

September 2022

DEDICATION

This project is dedicated to God almighty who blessed me with the wisdom to complete this course. To my family, my loving husband whose support and unwavering encouragement saw me through to the completion of the project.

ACKNOWLEDGEMENT

I would want to express my gratitude to everyone who helped make the completion of my study a success, including my loving husband, our adorable children, and the rest of my family.

I am greatly indebted to my supervisor, Dr. Jarutas Andritsch, who helped me by providing valuable corrections and suggestions and whose continued guidance and support not only helped me learn more but also got the project accomplished.

My gratitude also goes to my lecturers Dr. Shakeel Ahmad, Dr. Drishty Sobnath, Dr. Olufemi Isiaq, and Prins Butt who were of help in ensuring that the right knowledge was transferred to me.

LIST OF ABBREVIATION

LR	Logistic Regression
RF	Random Forest
DT	Decision Tree
NB	Naïve Bayes
MLPC	Multilayer Perceptron Classifier
NAMS	Neural Additive Models
CCFD	Credit Card Fraud Detection
ML	Machine Learning
AUC	Area under the ROC curve
ROC	Receive operating characteristic curve
CC	Credit card
SMOTE	Synthetic Minority Oversampling Technique
SOTA	State of the art
PIN	personal identification number

THE ACCURACY VERSUS INTERPRETABILITY TRADE-OFF IN THE APPLICATION OF MACHINE MODELS FOR PREDICTING FRAUDULENT CREDIT CARD TRANSACTIONS

Olabisi Dabi

MSC

Applied Artificial Intelligence and Data Science

2022

Abstract

Fraudsters adapt and get over increasingly complex obstacles put up by public or private organisations. Financial institutions are among those that need to act swiftly to avoid losses while keeping legitimate customers happy. Effective banking in the face of a growing volume of activities requires data analytics to back up traditional risk control measures, and a deeper insight into the mechanisms at work in fraudulent behaviour. In addition, the procedure of gathering evidence is important because fraud is a criminal and huge offense. Constraints imposed by legal, operational, and strategic constraints necessitate adjusting approaches to combating fraud. The initial part of this research is dedicated to figuring out how to evaluate a fraud detection model's effectiveness in terms of real-world issues encountered by banks at each level of the fraud prevention and management process. Second, it examines several different machine learning strategies that consider these peculiarities and workarounds the gap between fraudulent and nonfraudulent transactions, the dearth of fully trusted labels, the concept-drift phenomenon, and the inevitable compromise between detection accuracy and interpretability.

This state-of-the-art review throws light on a technological conflict between intrinsically interpretable models that have been improved for accuracy and black box machine learning models that have been augmented by post-hoc interpretation.

The paper concludes with a discussion of how genuine and promising hybrid sampling and machine learning approaches might give financial institutions and regulators real, short-term solutions without enveloping stakeholders with financial and moral interests in this technological race.

Table of Contents

LIST OF TABLES

LIST OF FORMULA

LIST OF FIGURES

CHAPTER ONE

1.0 Introduction

1.1 Research Motivation

The Credit card fraud detection process involves a wide range of techniques and procedures in developing an effective, interpretable, and accurate detection system. Credit card transactions are large and are used for different purposes by users at different geographical locations and currencies. This shows that credit card frauds are widely diverse. In recent times, there is news on the need for an increase in security on users' information and financial fraud as it relates to online and offline financial transaction systems. Fraudulent transactions are influenced by the fraudster's operation which has made fraud very difficult and challenging to detect. This act has negatively impacted users in many different negative ways including financial and emotional trauma once their cards are attacked by the fraudsters, and there is a need for an effective, accurate, and interpretability detective system to help salvage this situation. As a result of this, I am motivated to improve the CCFD system (irrespective of the geographical location) with the use of some accurate and explainable ML methods, while ensuring that the best class imbalance method is used for accurate and evenly distributed. One of the bank's goals is to detect credit card transactions and be able to quickly confirm or predict the risk of that transaction. With this, I will be improving customer service.

1.2 Research Problem

Credit card is widely used and has greatly facilitated transactions both for the users and Marchants. Credit card fraud is referred to any fraudulent card payment either using a debit or credit card. The improvement and maintenance of bank database security have been a major problem, as fraudsters are waiting for the slightest leak to gain access to the customer's information to carry out their activity. Credit card fraud has resulted in significant financial loss and is one of the most serious threats to business and commercial establishments today. Detecting credit card fraud is a challenging task when using a normal process. Also, traditional methods such as the cost analysis model, expert rules, etc. might have some shortcomings like high maintenance cost, low detection accuracy, and long detection time. As a result, there is a need for the development of CCFD models with accuracy and the ability to be able to interpret the behavior of each model. Several effective and efficient systems, models, processes, and preventive measures will aid in the prevention of Credit card fraud and the reduction of financial risks.

The imbalanced dataset is another challenge in the CCFD system. I am motivated will be using the resampling method, coupled with the hybrid methods which are SMOTEENN and SMOTETomek to get the most efficient dataset before running models on it. This will help data scientists solve the challenge of having to work with big data for an effective ML model.

1.3 Research aims and objectives

The objective of the research is to assess the performance of the fraud detection model using different ML algorithms to obtain higher accuracy. The accuracy is then compared with the interpretability of the machine models. This will also focus on the problems, methods, and measurements raised by fraud management when developing CCFD models that address the trade-off between accuracy and interpretability of detection. it provides a state-of-the-art (SOTA) review of the different ML methodologies-based approaches to process these data. we aim to study how interpretable ML would enhance the reliability and eventually contribute to the adoption and deployment of such a system. It examines how the approaches can influence short-term responses to banks and policymakers without compromising standards. Finally, it will focus on the use of the hybrid method in dealing with the problem of an imbalanced dataset for comparison with other methods while evaluating the performance of the prediction. This has been further outlined below.

- ✓ Identify through researching various literature, the possible ML algorithms that have or can be used in the detection of card fraud.
- ✓ Use the sampling method (SMOTE) which are Over-sampling, Under-sampling, and the Hybrid method (SMOTEENN and SMOTETomek) in dealing with the problem of imbalanced dataset currently faced by the fraud detection team.
- ✓ Measure the effectiveness of the classification accuracy obtained using the different data mining techniques.
- ✓ Review and compare the accuracy and interpretability trade-off among the different data mining techniques selected.
- ✓ Conduct the necessary research on the challenges of the current systems and possible future issues with CCFD.
- ✓ Using the model at 3 above, we will also identify and reduce the number of False Positives transactions (i.e., actual transactions that are wrongly identified as fraudulent transactions) that are associated with existing systems of CCFD and ensure that genuine transactions are not rejected.

1.4 Research Contribution

This thesis suggests a process based on ML techniques with a specific focus on the interaction between models, accuracy, and interpretability to meet the criteria laid out. AML algorithm specifically performs an initial classification of objects between the ones that are thought to be legitimate and the ones that potentially constitute a fraud attempt as the first component of the solution. Later, a module for AI interpretability that aims to explain the classification choice made by the ML model elaborates on listings that are thought to be fraud attempts. The item is now prepared for the final human review, which will determine whether the listing is accurate based on its attributes, classification result, and explanations.

To summarise the focus of this research, a ML model will perform a binary classification role will first be built, and then various cutting-edge ML explanatory techniques implemented and tested. One of these, a novel technique, achieves state-of-the-art performances by utilising the genetic algorithm's optimization skills to produce adversarial ML explanations. Furthermore, it has been empirically

demonstrated that the use of explanations can increase the accuracy and effectiveness of human validations and the acceptance of ML predictions.

1.6 Research Question

This study will be assessing the below questions

- ✓ What are the ML algorithms that have proven effective in the detection of credit card fraud
- ✓ Which of the sampling method (over-sampling, under-sampling, and hybrid) will best solve the problem of the highly unbalanced dataset
- ✓ Are the accuracy and interpretability trade-off of the different data mining techniques measurable
- ✓ What are the limitations and foreseeable future concerns of the current CCFD system
- ✓ Using several statistical methods, what is the most effective ML model in the prediction of credit card fraud

1.7 Hypothesis Tested

The study tested the following research hypothesis.

- ✓ Null Hypothesis (H0) - The transaction is not a fraud.
- ✓ Alternative Hypothesis (H1) - The transaction is a fraud.

1.8 Structure of the thesis

The study was divided into five chapters, except for the preliminary pages, which contain the title, declaration, dedication, abstract, acknowledgments, table of contents, list of figures, list of tables, abbreviations, and acronyms, as well as references and letter of transmittal in the back pages.

Chapter one contains the background and introduction to credit card fraud, its types, impact, and significance of the study. It involves the aims and objectives, research hypothesis, and questions.

Chapter two contains the literature review, previous research works, investigating references, and queried data to confirm this research can be achieved by using different classification models and ML python libraries.

Chapter three covers the pre-processing and how it was created for this study. It covers the choice of categorised features and guidance on tracking those features. It also involves addressing the problem of data imbalance using hybrid and other resampling methods.

Chapter four involves calculating the accuracy of various ML algorithms based on the classification model, comparing the model for the optimal accuracy, and then comparing it with the interpretability trade-off of the various ML method.

Chapter five will cover the summary of my findings, discussion, recommendations, and proposed areas for future study.

1.9 Work Schedule – Gantt Chart

Below is the work map to be followed in the completion of this research work.

Tasks	July				August - September				
	Wk1	Wk2	Wk4	Wk4	Wk5	Wk6	Wk7	Wk8	Wk9
Background studies									
Introduction									
Aims, Objectives and thesis									
Literature Review and related work									
Research question									
Data Collection									
Data Pre-Processing									
Model Training									
Methods / Value proposition									
Model evaluation									
Improving the performance									
Accuracy and interpretability									
The proposed system									
Results and Discussion									
Conclusions and Future Work									

Table 1.1 Work Schedule – Gantt Chart

10.0 Python libraries for data science and ML

During this study, python Jupyter note was utilised, and several python packages were imported which are Numpy, Pandas, seaborn, sklearn, imblearn, etc.

11.0 Proposed System

We decided to go an extra mile to visualize some of our EDAs, using Steamlit. This part is not included in our aims and objectives, that this will help present the EDAs for better visualization. The necessary libraries were imported which includes Streamlit and pickle etc., to carry out the task. Below is a visualization presented. Further images will be added to the Appendices.

Figure 1.1 Streamlit Visualisation

CHAPTER TWO

2.0 Background and Literature Review

2.1 Background

According to Gosset et al. (1999), defining fraud is difficult because the distinction between fraudulent and legitimate behavior is not always obvious. The weakest link is usually a customer or a store, but fraud is adaptive and moves to where it is most easily successful. Fraud is a wrongful or criminal deception intended to result in financial or personal gain, and it is as old as human existence. on the other hand, Alexopoulos et al. (2007) define fraud as "the deliberate and premeditated act perpetrated to achieve gain on false ground." The consequences of fraud are not limited to monetary losses, but can also result in violations of human rights, physical and psychological harm, and premature death. Fraud can be committed anywhere and in any private and government sector.

Historically, Credit card (CC) fraud has been a big problem, despite the increase in the level of security built around card transactions, criminals are still having their way around it.

The biggest occurrence of credit card fraud recorded in the UK happened in the mid-2000s when a group of European criminals used stolen and cloned details of £32,000 CC to defraud victims of more than 17 million. The fund illegally acquired was transferred to various accounts in different countries. An occurrence of CC fraud was also recorded in the USA in 2013 when a group of criminals stole a total value of $200 million. This was achieved through the creation of several false identities.

In recent years, the online payment method is being used because of the speedy increase in cashless electronic transactions. Among others, a credit card is one type of electronic payment method. A credit card is a thin rectangular piece of plastic or metal issued to a customer by a bank or financial services company to enable payment to a merchant of goods and services.

The card Issuer (Usually the Financial Institution) opens an account and assigns a line of credit to the user. An increase in credit card fraud is being experienced by financial institutions, despite the initiation of many new technologies, Despite the advantages that come with the electronic payment system. Scammers take advantage of the loopholes and always steal data using Skimming and phishing technology. They design a website to match a legitimate site, requesting personal details from the victim, which will be used to carry out fraudulent activity. Mails (Bait) directing victims to their bogus websites may also be sent. The email seems to be from a legit organization, requesting the victim to provide their personal information to solve an issue.

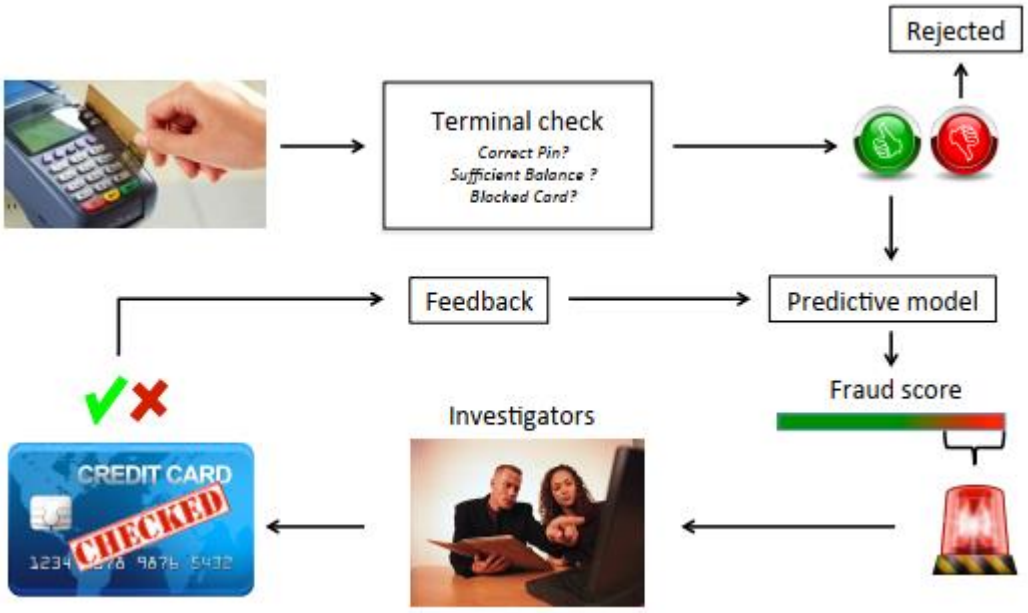


Figure 2.1 Credit Card detection process

Source: *https://www.researchgate.net/figure/The-Credit-Card-Fraud-Detection-Process_fig1_325658124*

2.1.1 Classification of credit card fraud

Credit card fraud has been divided into the below.

Offline fraud: It involves using a stolen physical card to carry out a fraudulent act.

Online fraud: it is perpetrated via the internet, phone, shopping, web, or in absence of a cardholder.

Type of Credit Card Fraud

Below are the common classifications of Credit Card fraud.

- ✓ **Card-not-present (CNP) fraud**: Once an account number and card's expiry date are known, CNP fraud can easily be committed. This can be accomplished via phone, mail, or the internet. It usually occurs when someone uses your card without being physically in possession of it. Merchants frequently use the card verification code to commit CNP fraud. CNP fraud is slightly more difficult, but if a fraudster has your account number, they most likely have your PIN as well. There are currently only 999 possible combinations for the four-digit verification code. Many fraudsters are attempting to determine the correct number. Overall, remote purchase fraud was reduced to £452.6 million in 2020, a 4% decrease from 2019. Online fraud against UK retailers is expected to total £262.3 million in 2020, a 9% increase over the previous year. Mail or telephone order (MOTO) fraud against UK retailers totaled £63.7 million, a 28% decrease from the previous year. Though there is a decrease of 4%, an increase in value worth 12% was recorded. There is need a to ensure that further decreases in fraud value are recorded despite the increase in value and issuance of the credit card.

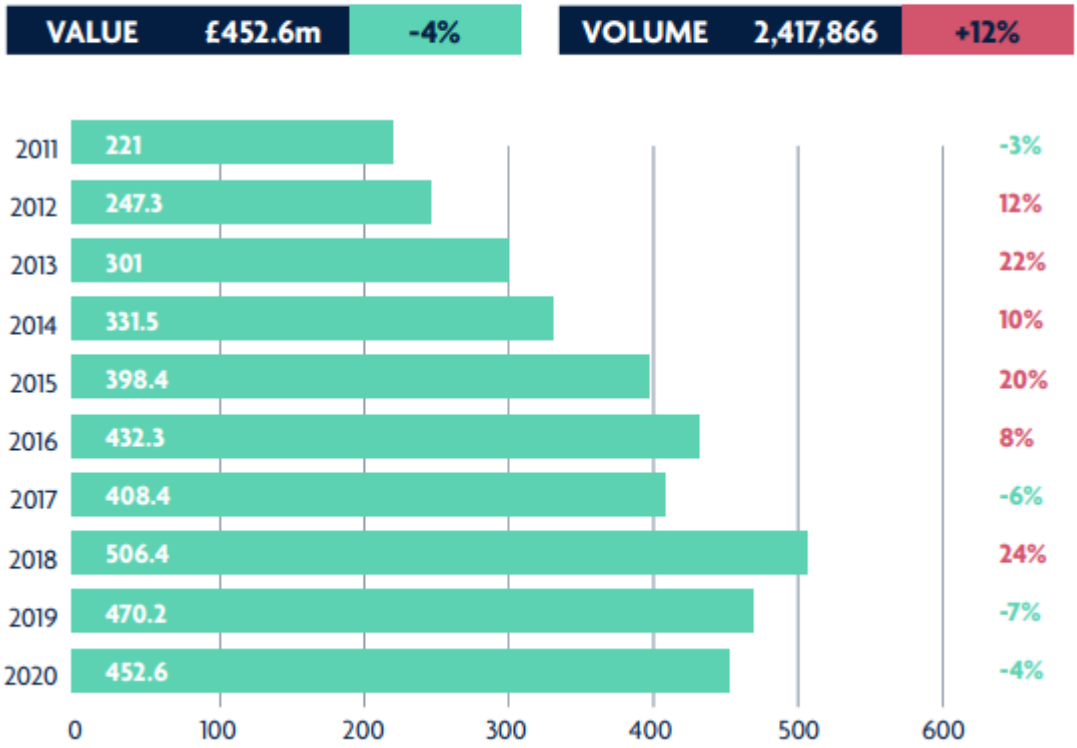


Figure 2.2 Remote purchase (CNP) fraud losses on UK-issued cards 2011 - 2020 (£m).

***Source**: fraud the fact 2021*

- ✓ **Counterfeit card fraud**: Skimming is the most common method of committing counterfeit card fraud. This counterfeit magnetic swipe card contains all your card information, including the card number, account number, and PIN. This counterfeit magnetized strip is then utilised to produce a fully working fake credit card. it is a replica of the original card, and fraudsters can simply insert it into a machine to pay for purchases or withdraw funds. Someone who has access to a user's credit card details can also perpetrate this type of fraud. They can use this information to make "fake plastic." In 2014, India ranked second in South Africa for counterfeit credit card fraud. Counterfeit card losses totaled £8.7 million in 2020, a 32% decrease from 2019 and a 95% decrease from the peak reported in 2008 (£169.8 million). To steal the information needed to make a fake card, criminals often affix hidden or masked devices to the card-reader slots of ATMs and unattended payment terminals (UPTs) such as self-service ticket dispensers at train stations, movie theatres, and parking garages. Counterfeit cards are typically used in countries that have yet to implement Chip and PIN technology.

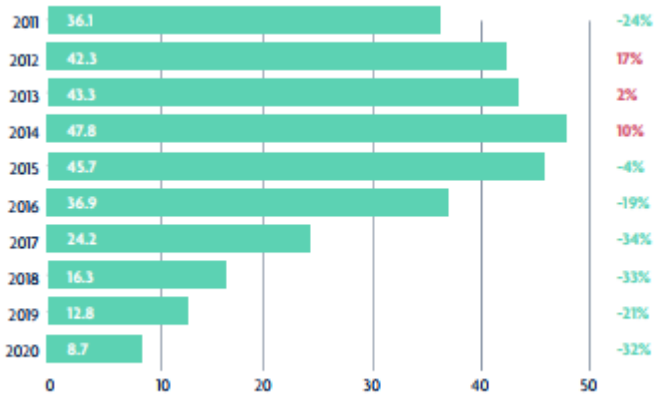


Figure 2.3 Counterfeit card fraud losses on UK-issued cards 2011 – 2020 (£m).

Source: fraud the fact 2021

- ✓ **Lost and stolen card fraud:** Under this type of fraud, the credit card will be taken from the user's possession, either through theft or loss. The fraudsters will then use a credit card to make payments for goods purchased. This type of fraud is difficult to commit using machines because they require a PIN. However, a stolen or misplaced card can be easily used to make online purchases. Losses from lost and stolen fraud decreased by 17% in 2020, falling to £78.9 million from £94.8 million in 2019. The number of incidents has also dropped dramatically, falling by 30% over the same time. Cards are typically stolen through low-tech means such as distraction thefts and ATM entrapment devices, which are then used to commit fraud. According to FRAUD - THE FACTS 2021, According to the data collection of data by Fraud – The facts 2021, 2020 observed a yearly decline in non-contact losses for the first time. The drop in contactless card fraud was due in part to fewer opportunities for fraudsters to perpetrate these types of scams because of the limitations enforced during the pandemic.

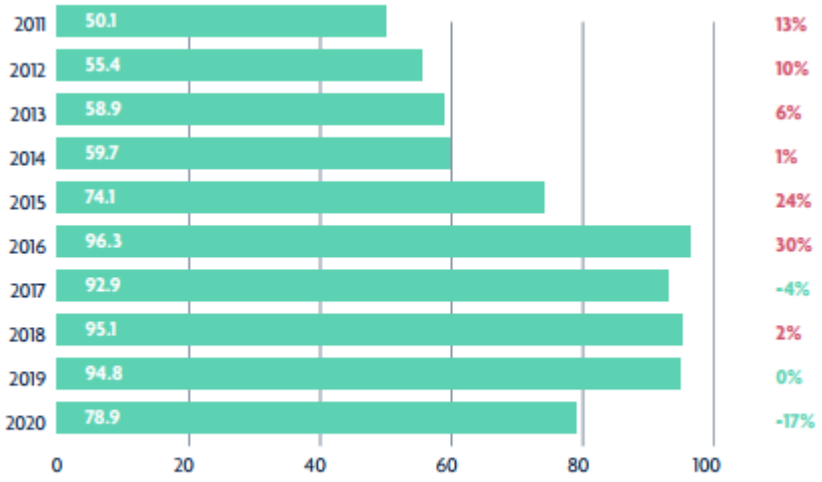


Figure 2.4 Lost and stolen card fraud losses on UK-issued cards 2011 – 2020 (£m).

Source: fraud the fact 2021

- ✓ **Card ID theft**: Card ID theft fraud occurs when a fraudster uses the name and information of another person to apply for credit or a new credit card. Typically, they will take supporting documents, which will subsequently be utilised to confirm their bogus application. To prevent this type of fraud, banks have implemented a variety of safeguarding plans and actions. The most important is that only appropriate and original documentation is required. They will also frequently call the employers to confirm their identity. Unfortunately, fraudsters frequently falsify documents and provide false contact information for places of employment. Utility bills and bank statements, for example, are used to open bogus accounts. Application and Account takeover fraud is classified under card ID theft fraud. Application fraud takes place when dishonest individuals use stolen or falsified documents to create an account in the name of another individual. Thieves try to get their hands on important personal documents like bank statements and utility bills to get into what they want. Alternatively, they could use forged documents. Account takeover is the fraudulent use of another person's credit or debit card account by a criminal, who begins by gathering information about the intended victim and then contacts the card issuer pretending to be the genuine cardholder.

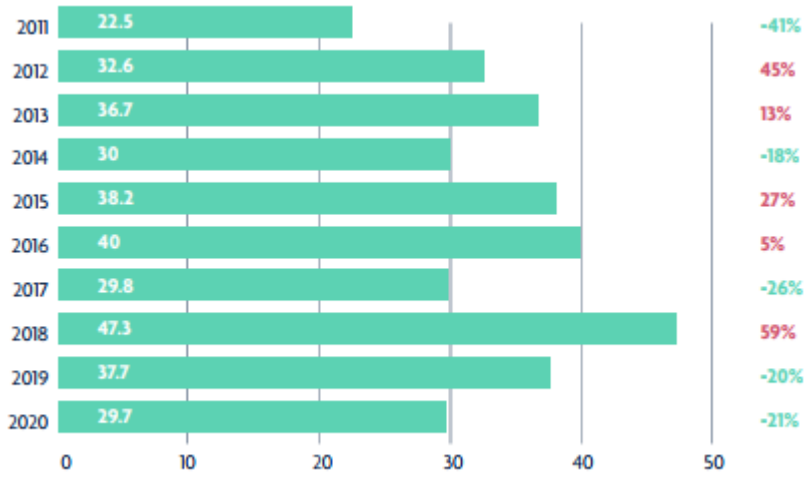


Figure 2.5 ID theft on UK-issued cards 2011 – 2020 (£m).

Source: fraud the fact 2021

- ✓ **Card not received fraud:** This occurs when a card is stolen while in transit after it has been sent out by the card issuer but before it reaches the genuine cardholder. To commit this type of fraud, criminals typically target properties with communal letterboxes, such as flats, student halls of residence, and external mailboxes. People who have their mail redirected when changing addresses are also susceptible to this type of fraud.

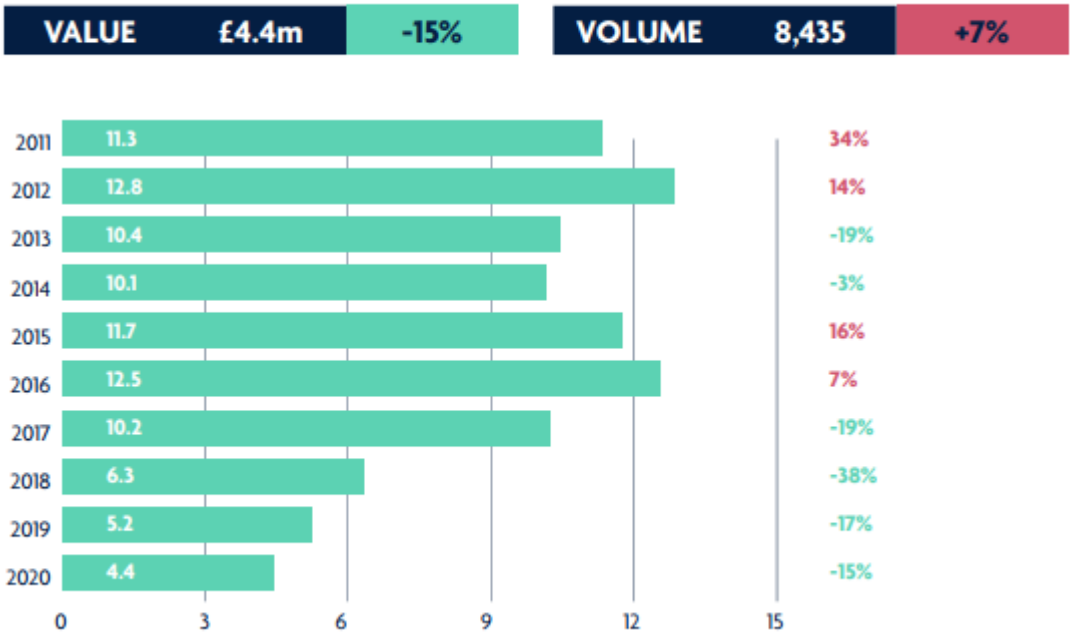


Figure 2.6 Card not received fraud losses on UK-issued cards 2011-2020 (£m).

Source: fraud the fact 2021

2.1.2 Significant of the study

This project's main contribution is to use ML algorithms to do performance analysis on a credit card transaction data set to identify fraudulent transactions using various criteria. It involves comparing the accuracy of the different machine techniques and comparing the interpretability of those models. To address the class imbalance issue using the hybrid and other sampling techniques. Several ML algorithms, including LR, DT, RF, Naive Bayes, and Neural Network will be applied to the data set to identify transactions that are fraudulent or legitimate, interpretable, and comparative result analysis will be presented. Financial organisations can use the findings of this study to improve the accuracy with which their ML systems detect fraudulent behaviour. This will make it simpler for banks to prevent fraudulent transactions from going through that haven't been authorised by the account holders. Despite the magnitude of the fraud, there has been the surprisingly little scholarly study of its costs, root causes, mechanisms, effects, and strategies for detection, deterrence, and prevention. The importance of knowing how to spot and avoid becoming a victim of fraud is growing as more criminals go undetected. The method that can help financial institutions move forward is the accuracy of the ML approach and how well they are interpretable for a better application.

2.1.3 Challenges of CCFD

Fraud detection systems are limited to the difficulties and challenges listed below. To achieve peak performance, an effective fraud detection technique should be able to address these challenges.

Imbalanced data: The data used to detect credit card fraud is unbalanced. This means that only a very small proportion of all credit card transactions are fraudulent. This makes detecting fraudulent transactions difficult and imprecise.

Misclassification importance: The impact of various types of misclassification errors varies greatly depending on the nature of the fraud detection task at hand. A false positive for fraud is less damaging than a false negative for a legitimate transaction. Since more inquiries into the first case will undoubtedly reveal the incorrect categorisation.

Overlapping data: Many transactions may give the impression that they are fraudulent while they are valid (this is known as a false positive), and vice versa, a fraudulent transaction may give the impression that it is legitimate (false negative). Because of this, one of the most important challenges for systems that identify fraud is to achieve a low percentage of both false positives and false negatives.

Interpretability: In recent years, there has been a lot of focus placed on the idea of interpretability when it comes to machine learning algorithms. Even though ML has developed into an effective tool for solving problems, mapping complicated and nonlinear functions can be challenging to understand. In the field of fraud, the capacity to identify result drivers is essential to the process of persuading domain specialists to trust the detections made by such systems. Models that can be intrinsically interpreted, as well as post hoc procedures that are either specific or agnostic, are the three components that make up this constraint's solution.

Fraud detection cost: The system needs to consider not only the cost of fraudulent behaviour that has been identified, but also the cost of avoiding it, and then compare those two costs to the gain that has been obtained. For example, preventing a few pounds worth of fraudulent transactions does not result in any additional revenue.

Accuracy under constraints: In the field of machine learning (ML), one of the most significant challenges is the trade-off between precision and interpretability as both concepts are seen as "contradictory." Recent approaches like deep neural networks, for example, have a particularly difficult time breaking through the interpretability barrier. In numerous application domains, there is a tension between accuracy and interpretability, and although the results of these tests are still up for discussion, the performance of the models that are the most representative is compared.

Challenges with drifting of frauds: The fraud patterns shift whenever the criminals who commit the fraud alter their behaviour in response to newly developed products and control measures. The emergence of this word, which is referred to as concept drift, takes place when the fundamental distribution of the target notion is dependent on hidden circumstances, which necessitates re-training the model.

Dealing with the unlabeled dataset: In addition to the constraints that unsupervised models have when applied to the unbalanced dataset, they are less effective when applied to information that is only partially labeled. One characteristic of fraud that has been carried out successfully throughout history is that it has evaded detection. This indicates that "nonfraudulent" observations and "fraudulent" observations are frequently co-occurring in real-world fraud datasets. It is recommended to employ semi-supervised or unsupervised anomaly detection algorithms when working with data that only has a limited level of trust placed in its labels.

2.1.4 Impact of Fraud

After a thorough review of the impact of CC fraud worldwide, the CCFD system is necessary and worth investigating. According to Merchant Savvy (2020), Payment fraud losses have more than tripled since 2011 and are expected to exceed $40 Billion by 2027, from 2011 to 2020, it rose from $9.84 billion to $32.39 billion. According to the UK finance report (fraud – The fact 2021), A total of £983 million in card fraud was stopped by banks and card companies in 2020. A total of 2,835,622 card fraud cases with a value of £574.2m compared to 2,745,539 cases reported in 2019 with a value of £620.6. Although, there is a reduction. However, the value involved is on the high side, hence the need for an effective control system. According to Federal Trade Commission (2020), 393,207 cases were recorded in 2020 compared to 2019 with 271,927 cases in the US under identity theft reported. $28.5 billion was lost worldwide in 2020 due to credit card fraud.

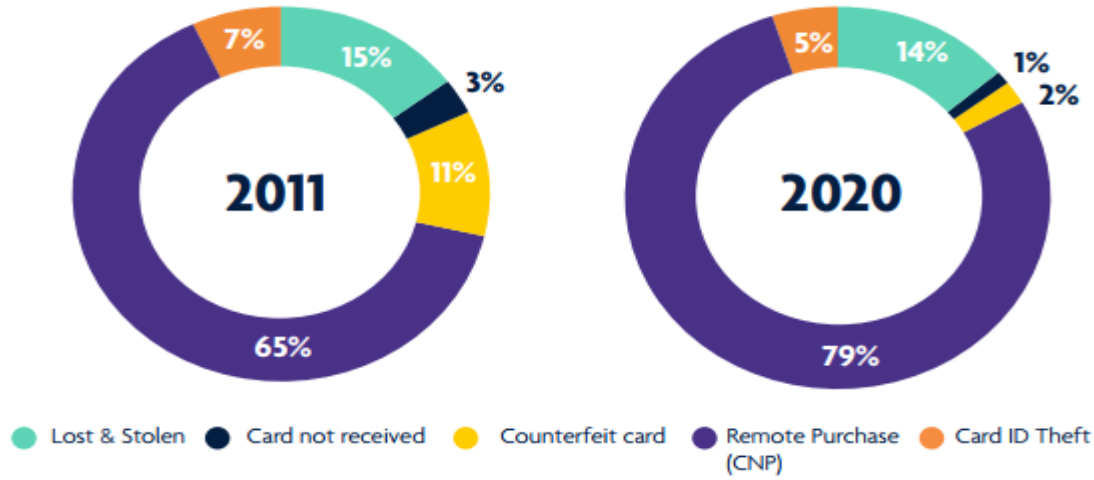


Figure 2.7 Card fraud losses split by type (as a percentage of total loss).

Source: fraud the fact 2021

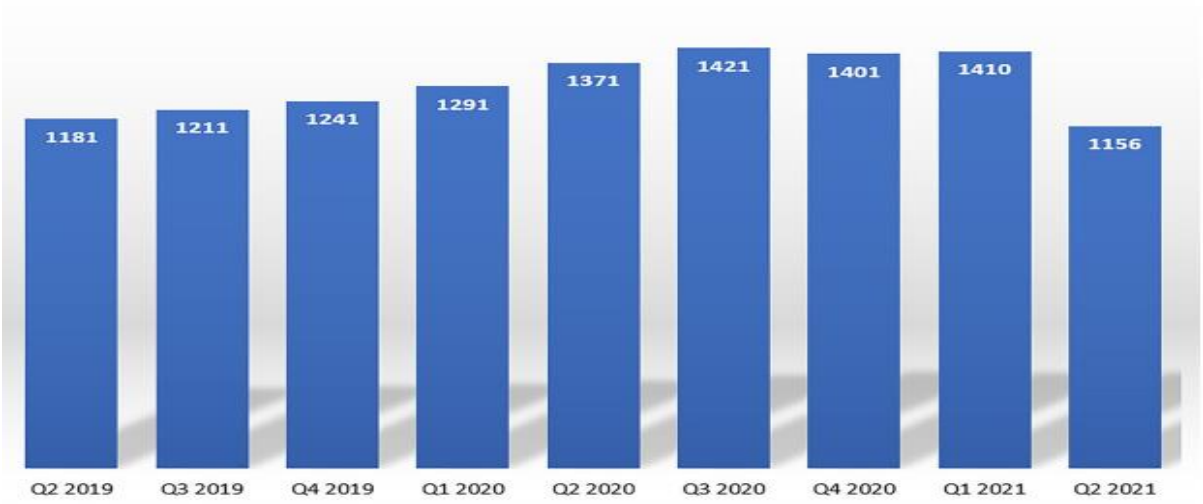


Figure 2.8 Amount of Visa cards issued worldwide.

Source: *https://www.researchgate.net/figure/Amount-of-Visa-credit-issued-worldwide-50_fig2_360408387*

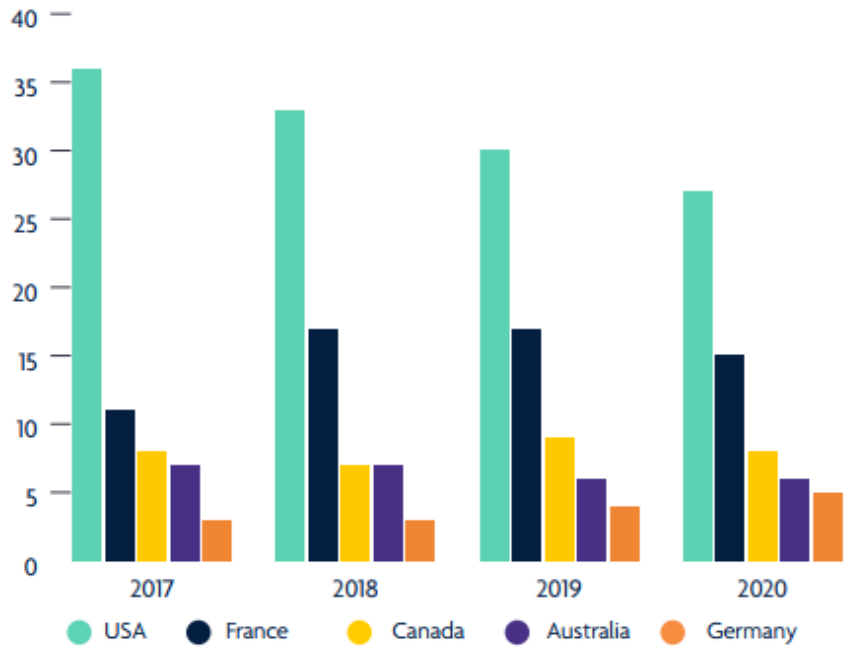


Figure 2.9 Top five countries for fraud on foreign-issued cards occurring in the UK 2017-2020.

Source: fraud the fact 2021

The chart displays losses as a fraction of total fraud at UK-acquired merchants using foreign-issued cards, with the USA topping the list for the four years. This may be due to the USA population compared to other countries.

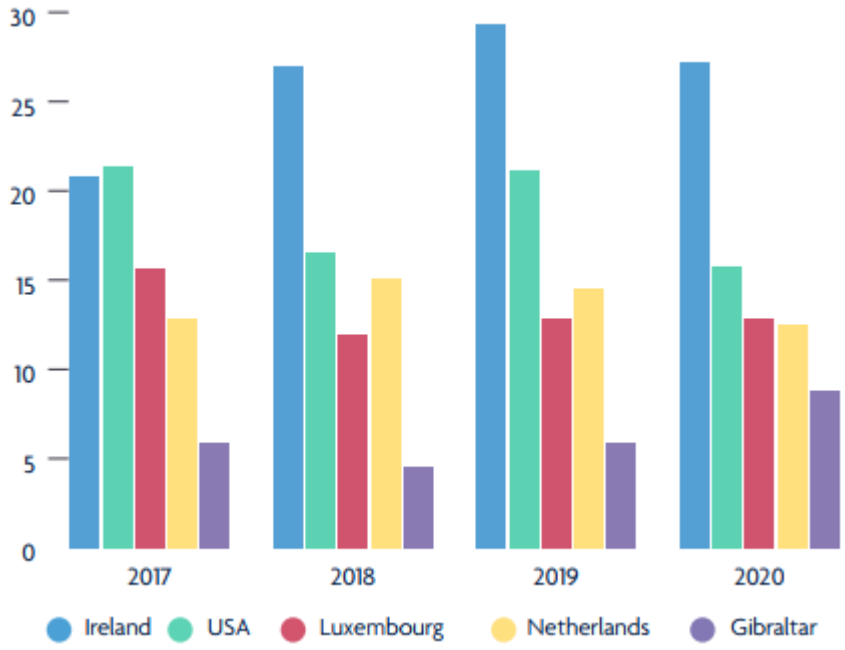


Figure 2.10 Top five countries where fraud on UK-issued cards occurs 2017-2020.

Source: fraud the fact 2021

Percentage Losses on UK-issued cards or card details used fraudulently overseas with Ireland and USA topping the list.

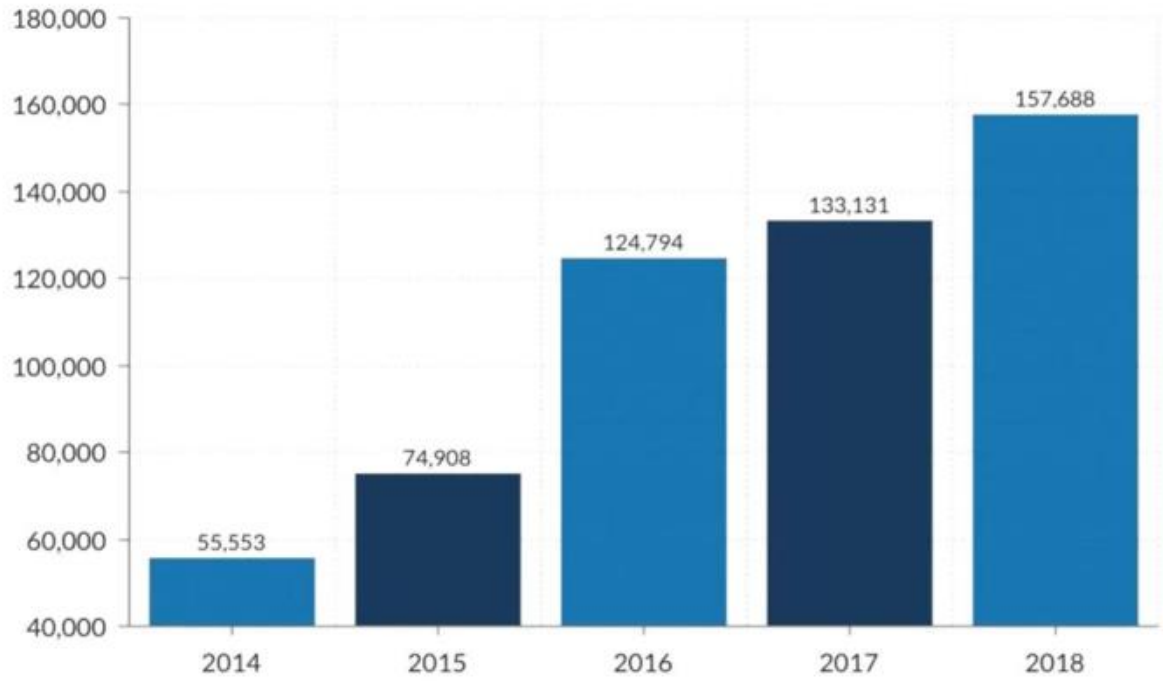


Figure 2.11 Credit Card Fraud Reports in the United States

Source: *https://www.researchgate.net/figure/Amount-of-Visa-credit-issued-worldwide-50_fig2_360408387*

This shows the values and the year for which credit card fraud was reported in the USA. There is an increase in fraud, and it can be connected to the increase in the issuing of credit cards.

2.2 Literature review

Since credit card fraud has increased over the past years, and after Monitoring users' behaviour and eliminating questionable credit card transactions, numerous research studies have aimed to reduce credit card fraud on a global scale. Credit card fraud research was boosted by the introduction of the "one-time credit card number" (Saxena and Ponnapalli, 2012). One method proposed in the study for generating a unique credit card number even while the user is disconnected from the internet was implemented. A credit card number is created by the system using a predetermined secret. Each customer's private key can be used to complete a transaction while logged in, making it impossible to undo an online purchase. The authors are still leveraging the existing credit card numbers scheme and traditional infrastructure to handle internet transactions.

Credit card fraud research was boosted by the introduction of the "one-time credit card number" (Saxena and Ponnapalli, 2012). One method proposed in the study for generating a unique credit card number even while the user is disconnected from the internet was implemented. A credit card number is created by the system using a predetermined secret. Each customer's private key can be used to complete a

transaction while logged in, making it impossible to undo an online purchase. The authors are still leveraging the existing credit card numbers scheme and traditional infrastructure to handle internet transactions. The purpose of this model was to lower the likelihood of fraudulent activity involving credit cards. With this method, a unique credit card number is generated on the user's device and sent to the card issuer and the retailer. The card issuer employs validation criteria, such as a one-time password or a string of letters, to verify the user's identity before allowing the transaction to proceed. A one-time credit card number is produced on the user's device, and the server accepts this number and then compares it to the number the user sends to the merchant using the shared key between the user and the server. If the two sets of numbers agree, the user is verified. A model employing a robust dynamic security code scheme was developed by Essebag et al. (2018). The system allows you to modify the security code on your credit, debit, or prepaid card. While these dynamic security code values are only applicable to online transactions, they can be generated by the system's dynamic security code generator and used with the existing payment infrastructures. The technique can also be utilised in situations unrelated to payments, including checking balances.

Credit card numbers with a one-time usage were generated using a hash function by Yingjiu and Zhang (2004). To produce a new one-time password, the current one is hashed with a secret shared between the cardholder and the issuer. Each credit card in this system contains a tiny computer chip used for storing information and computing hashes.

McDonald (2010) took a new approach to the problem of credit card security, by developing a system for cardless, secure online purchases utilising a credit/debit card. The system is focused on the purchaser, and it confirms users using a Personal Digital Identity Token (PDIT). The PDIT is a biometric identification tool for the cardholder that connects to a set of established identity credentials. The system keeps track of at least one supplier of online credit/debit card services and offers a way to conduct secure online transactions. It hides the card details while making online purchases, anonymizes the transaction, and makes it unseen from hackers and fraudsters.

In a report published by Johari and Gupta 2011, the development of the online validation process, which are one-time password systems, biometrics, mobile devices e.t.c. The authors suggest a completely new paradigm for credit card transactions involving both in-person and online transactions.

Trivedi 2020 proposed a mechanism for detecting credit card fraud that relies on ML algorithms. Using a dataset of credit card fraud that was slightly skewed, the authors evaluated the effectiveness of various ML methodologies, including LR, RF, Artificial Neural Networks (ANN), DT, Naive Bayes, SVM, and Gradient Boosting Classifier Strategies. They demonstrated that RF outperforms other ML classifiers with a 95 percent accuracy rate. However, RF is regarded as a laborious model.

Gupta, Shalini, and Johari (2002) conducted an in-depth study and analysis of the existing online authentication technologies, which include biometrics, one-time

password (OTP) systems, mobile devices, and the Public Switched Telephone Network for cardholder authentication. Single-use one-time passwords are used in the one-time-password (OTP) system to authenticate users. When a customer submits their credit card details for verification, the payment gateway issues a one-time password to enter on their computer, smartphone, or other mobile device. However, if this is an online purchase, the retailer might prompt the cardholder to enter a one-time password on his device or website. If the one-time password is entered successfully, the transaction will be approved, and the cardholder's identity will be confirmed.

The current methods of online authentication, including biometrics, one-time password (OTP) systems, mobile devices, and Public Switched Telephone networks for cardholder authentication, were examined and analysed by Shalini, Johari, and Gupta 2011. Single-use one-time passwords are used in the one-time-password (OTP) system to validate users. After the customer's credit card details have been sent to the payment gateway, the customer will get a one-time password by text message or email. The retailer will require the cardholder to enter the transaction-specific one-time password on his device or website. If the entered one-time password is correct, the cardholder's identity is verified, and the transaction is processed. This method places a significant burden on the cardholder, as they will be required to input a one-time password for every online transaction.

Virtual credit card numbers are often issued by some credit card companies. An example is Capital One's Bank which offers this service to all its clients. This is done through a third party called ENO, an intelligent assistant. It assists Capital One clients with various requests. The client must be using a computer and Eno installed with either google chrome or Mozilla Firefox to generate a virtual credit card number. Customers of Capital One cannot access this service with any other internet browser. Eno does not have access to its customers' payment histories, its sole responsibility is to give virtual credit card number services to Capital One customers. Therefore, they cannot detect fraudulent transactions based solely on the customer's payment history. Eno provides each business with their credit card number for use in payments.

Another organisation that offers virtual credit card numbers is City Bank, which is a credit card issuer. However, only a select group of consumers who own "Only Select Citi cards" are eligible for this service. The process is easy for those cards that do meet the requirements. Before using the internet interface to obtain a virtual credit card number, the consumer must first register their Citi credit card with the service. The created virtual card numbers are valid for up to a year. Until a new number is requested, the consumer may use the virtual number if they like.

2.3 Related Work of CCFD Techniques

The six main techniques for detecting credit card fraud are identified in this section.

2.3.1 Logistic Regression

LR is a method that uses one or more factors to estimate the likelihood of a binary classification response. It uses statistical models such as regression analysis,

discriminant analysis, and other analyses. (Altam, Macro and Varetto 2019), (Sahin and Duman 2011).

There are more benefits to using LR in credit card fraud cases. It can forecast some outcomes of the existence or nonexistence of the distinctive values when replied set of variables. For each of the model's independent variables, odds ratios can be assessed using the coefficients of LR. it applies to a wider variety of research scenarios when compared to characteristic analysis.

Jordan (2002) presented a comparison of LR and Naive Bayes. The asymptotic error of the discriminative LR algorithm is lower, according to the authors' mathematical analysis of each algorithm they provided. As a result, the generative Naive Bayes classifier may also reach its asymptotic error more quickly. There have been some instances where LR underperformed Naive Bayes, but this is mostly seen in small datasets.

A similar study, conducted with ML methods, may be found in Sadineni (2020). Artificial Neural Network (ANN), Support Vector Machine (SVM), LR, DTs, and RF were only a few of the ML methods looked at in this research. Like previous researchers, we evaluate the efficacy of the concepts by measuring things like precision, accuracy, and false alarm rate. Using the Kaggle data set, Joshi and Aruna examined exactly 150,000 transactions (2020). The study found that the database has numerous fields. The dataset, which contained both relevant and irrelevant variables, was evaluated based on the principal component to extract the relevant data, such as transaction amount and time of the transaction, etc. The results were: LR 95.55%, DT 98.47%, Radom Forest 99.21% accuracy, Artificial Neural Network (ANN) 99.92% accuracy, and Support Vector Machine (SVM) 95.16% accuracy.

Shen et al. (2007) investigated some classification techniques (Neural Networks, DT, and LR) in the detection of fraud. The authors demonstrate that the projected classifier of LR and neural network techniques is more effective at resolving the issue at hand than the DT.

Sahin and Duman (2011) used highly skewed data to apply classification models based on LR (L.R.) and Artificial Neural Networks (ANN) and to problems of detecting credit card fraud. This research shows that artificial neural network classifiers are superior to logistic regression classifiers for this analysis task. L.R classifiers tend to suit the training data as they grow due to insufficient work sampling.

The dependent variable in CCFD could have a value of 0 (non-Fraudulent transaction) or 1. (fraudulent transaction). LR, in contrast to conventional linear regression, makes no assumptions about the distribution of the dependent variable or the error terms, nor does it assume a linear relationship between the independent variables and the dependent variable. It is defined as the below.

$$\log(\frac{p}{1-p}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

Where X1, X2,…, Xk are the independent variables and p is the probability that the dependent variable has a value of 1. $\beta 1$, βk are coefficients of the independent variables, and $\beta 0$ is a constant.

2.3.2 Naïve Bayes

The naïve Bayes algorithm was introduced by John and Langley in 1995. The probabilistic classifiers in Naive Bayes enable this model to predict data from various classes. The choice is carried out based on conditional probability. This model employs a collection of algorithms rather than a single algorithm, but they all follow the same principle. According to this model, each variable contributes to the outcome similarly and distinctively Awoyemi (2017). This model has an additional benefit over others in that it only needs a small amount of training data. The decision of the highest probability is based on the Bayes theorem. Known values and probabilities are used to estimate Bayesian probability.

A comparison of supervised data mining methods for preventing fraud was presented by Sherly (2012). Several techniques, including, Neural Networks, DTs, and Naive Bayes classifiers, were evaluated by the author. According to the study, neural networks work best with larger databases and take a while to train. Bayesian classifiers are ideal for different data sizes and are much more accurate and quicker to train, but they take longer to use when applying to new instances.

According to Awoyemi (2017), there are two challenges associated with identifying fraudulent activity on credit cards. The first problem is that the characteristics of both valid and fraudulent transactions are in a state of constant flux, and the databases that are used to track credit card fraud are severely unreliable. They further examined the data performance on highly skewed credit card fraud data using LR, k-nearest Neighbor, and naive Bayes. The study looked at a representative sample of 284,807 transactions from cardholders located all over Europe. The study is done in Python, and the researchers utilised three different approaches for both the raw data and the data that has been preprocessed. The performance of the various methods was evaluated using several different metrics, including accuracy, precision, specificity, sensitivity, flat classification, and Matthew's correlation coefficient rate. The results provide the best accuracy for naive Bayes, k-nearest neighbour, and LR classifiers, with respective values of 97.92%, 97.69%, and 54.86%. The comparison of the three methods made it evident that the k-nearest Neighbor strategy surpasses the naive Bayes method as well as the LR method.

A meta-classification approach was used by Pun and Lawryshyn (2012) to enhance the detection of credit card fraud. The method uses DT, K-Nearest Neighbor, and Naive Bayesian algorithms to build three base classifiers. The performance of the whole system is improved by 28% when the naive Bayesian algorithm is used as the meta-level method to integrate the predictions produced by the basic classifier. The

following formula represents the Nave Bayes supervised ML algorithm.

$$p(A|B) = \frac{p(A|B).p(A)}{p(B)}$$

The Bayes theorem gives a mechanism for determining the subsequent probability P (A|B), often known as the likelihood of result (A), provided that certain conditions are met (B). The Bayes Theorem can identify the next probability even in the absence of any previous understanding of clear conditions by employing a probability ratio known as P (B|A) = P (B) to tie it to the prior probability of the result. The naive assumption that each factor influences the outcome in its unique way is the foundation of the Nave Bayes Theorem.

Real-world data experiments have repeatedly demonstrated that the Nave Bayesian classifiers perform on par with more advanced induction algorithms. According to Clark and Niblett (1989), Bayesian classifiers in the medical domain are just as accurate as rule14 induction techniques like the CN2 and ID3 algorithms. John and Langley (1995) demonstrate that the Naive Bayesian classifier outperforms the DT algorithm by using kernel density estimation rather than a Gaussian distribution. However, this approach is known as "Naive" because it erroneously (Naively) believes that the class's attributes are independent of one another. Following classification, Bayes' rule is applied to determine the likelihood that the correct class will be identified given the specific attributes of the credit card transaction.

2.3.3 Decision Tree

Quinlan (1986) created the DT approach, which can handle consecutive data. The DT is a table of different tree appearances composed of the root, internal, and leave nodes.

The DT was combined with Hunt's algorithm and Luhn's algorithm in a study by Save et al. (2017) to identify fraudulent transactions. The shipping address and billing address of the non-fraudulent user were verified by the paper. It is assumed that these addresses must match for the transaction to be considered valid. If not, the transaction is considered suspicious because a fraudulent one is more likely to differ from the address of the legitimate user. The process of "Outlier detection" was described in the paper, which concluded that the card validation was accurate and had few false alarms.

Complex problems are broken down into simpler ones by the DT, which then builds a DT based on the learned information obtained through the data mining technique. The DT model is based on the construction of a tree with extreme accuracy and small scale.

The trained system generates a set of conditions at each level, and the DT centers its decision on those conditions. The DT is built using data mining methods that divide a dataset of records using the depth-first greedy or the breadth-first approach Kalyanakrishnan and Gaikwad (2014). There are lines connecting each of the nodes

and leaves. Each node may be a branch node with additional nodes following it, or the DT classification method may only assign one leaf node.

CRISP-DM and DT algorithms were employed to detect bank fraud by Rocha and Sousa (2010). Some financial dealings are evaluated using DTs and CRISP-DM to aid in the detection and prevention of bank fraud. They, like many other scholars who came after them, saw DTs as a foundational concept in AI. By analysing the data from bank transactions, the investigation uncovered further instances of online banking fraud Rocha and Bruno (2010).

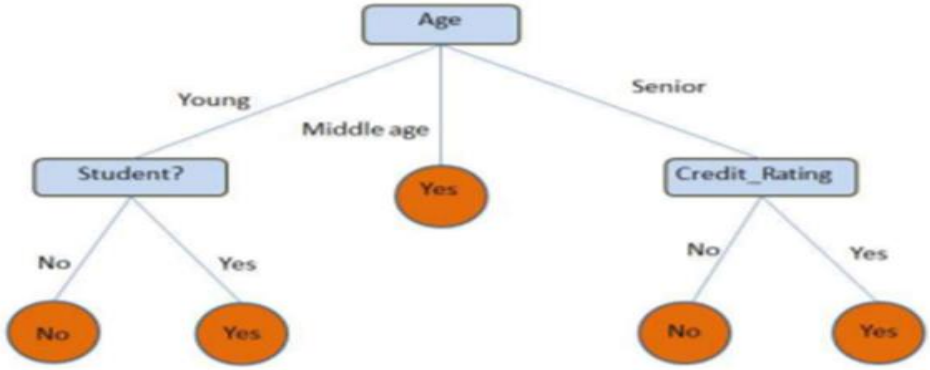


Figure 2.12 A simple DT

Source: Semantic Scholar 2020

2.3.4 Random Forest

A classifier for aggregate data is the RF model. It employs several trees by combining numerous DT classifiers. The basic objective of using multiple trees is to train them enough such that each one contributes to the construction of a model. After the tree had been constructed, the results would then be integrated. This model is dependent on a particular dataset that employs several DTs.

Lakshmi (2018) examined the effectiveness of various ML algorithms, including LR, DT, and RF, for detecting credit card fraud. They made use of a well-known Kaggle dataset of credit card transaction, which contains 284,808 credit card transactions from a set of data from European banks. The R programming language is used to apply the three techniques to the dataset. Based on sensitivity, specificity, accuracy, and error rate, the methods' performance is assessed for a variety of variables. They investigated 5, 10, and 21 variables separately, which is a different number. The average outcome reveals that the accuracy for the RF, DT, and LR classifier are respectively 95.5, 94.3, and 90.0.

The RF fraud detection system was utilised by More, Rashmi & Awati (2021). This method has steadily made it easier to spot fraud in credit card transactions and can help solve fraud in the real world, according to Rashmi and Awati (2021). The data set used in their study was made up of 100,000 cardholder transactions. Based on what was found, 0.262% of all transactions are fake. Even though the dataset was

very unbalanced, it was still processed. This shows that 80% of it was used to train the model and 20% was used to test it. Recall (sensitivity), Accuracy, and precision were the metrics that were utilised in the performance evaluation. The accuracy level was 0.9793, which demonstrated that the proposed technique had increased accuracy for a significant portion of the training data. Additionally, 20,000 separate transactions were discovered. 19,830 of them were members of class 0, while the remaining 170 were members of class 1. According to the findings of the study, the model is successful at detecting fraudulent credit cards even when applied to an imbalanced dataset. According to the findings of the research conducted by Dornadula, Vaishnavi, and Geetha (2019), a comparative examination of the three classifiers (DT, Naive Bayes, and RF) revealed that the RF approach performed much better than both the DT and Naive Bayes Techniques.

Shirgave, Suresh, and Awati (2019) talked about CCFD which is based on ML. They use metrics like specificity, accuracy, and precision to compare the different ML fraud detection techniques. They also recommended an FDS that uses the supervised RF method. The system they suggested makes it easier to spot credit card fraud. The problem of idea drift in fraud recognition is well addressed by the proposed method. This is accomplished through the ranking of the alert as well as the utilisation of the learning-to-rank method.

2.3.5 Neural Networks

Artificial Neural Networks (ANN) are computer models that attempt to emulate the biological neural networks in our bodies and can easily adapt to new information.

According to Bulus (2013), improvements in the system for payment approval have assisted in the fight against card payment fraud. In the early 1980s, online authorization of credit card payments was made possible. At the point of sale, transaction data is sent to the card issuer, who then decides right away whether to approve or reject the payment. In the 90s, the application of "neural network" computer systems, involved sophisticated statistical modeling techniques, to analyze transactions and detection of fraudulent transactions. Today, it is used in almost every transaction in the United States.

The impact of each input's contribution to the outcome prediction depends on how much weight that input has. Appropriate weights for the connections must be established to create a neural network that is a reliable predictor. Backpropagation is the most popular technique for determining the ideal connection weights. Rumelhart, Hinton, and Williams (1986) introduced this technique, and it was thanks to their work that artificial neural network research became well-known in the field of ML. Backpropagation makes use of the mathematical technique known as gradient descent, which incrementally modifies a function's parameters to reduce the output network's squared error function. If the function has several of them, the gradient descent method might not find the best minima.

Another comparison study on the use of Bayesian and Neural Networks to detect credit card fraud was conducted by Maes (2002). The outcome demonstrates that

while the Bayesian Networks require less training time to produce better fraud detection results, Artificial Neural Networks detect fraud more quickly.

Ogwueleka (2011) suggested a method for creating a CCFD system that combines neural network technology with traditional data mining techniques to better address the credit card fraud problem. Real-time transaction entry is used in conjunction with unsupervised neural network design. The study offered a method that reduced the classification of non-fraudulent transactions as fraudulent and guaranteed an accurate and dependable outcome. The study established a firm foundation for the employment of intelligent detection approaches in an operational fraud detection system and furthered the validity and effectiveness of ANNs as a research tool.

Using three phases—the first neural network, the fuzzy C-means clustering, user authentication, and verification of the card details technique, Panigrahi and Behera (2015) created a system that identifies credit card fraud using neural networks and fuzzy clustering. To determine whether a credit card transaction is legitimate, suspicious, or fraudulent, it must pass these three tests. The neural network learning mechanism is used after the transaction is identified as suspicious. The computation time also increases, but this system can reduce the production of false alarms and result in a more accurate CCFD system.

Below is the sigmoid function used in calculating each network layer.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Below is the squared error function.

$$E = \frac{1}{2}(y - f(x))^2$$

Where y is the instance's class label and f(x) is the network prediction obtained from the output unit.

2.4 Handling imbalance dataset

The simplest approach entails randomly eliminating the majority class of members (Batista 2004). Other approaches try to better divide the classes to speed model learning. This is an example of choosing which observations to keep using near miss and condensed nearest neighbor (CNN) Hart (1968). However, techniques like Tomek links (Tomek, 1976) and edited nearest neighbors (ENN) Wilson, 1972) attempt to pick out unwanted observations. The most recent techniques combined both latter techniques. One-sided selection (OSS), for instance, combines CNN and Tomek links Kubat and Matwin, (1997). The ENN method enhances OSS using the neighborhood

cleaning rule method (NCL) (Laurikkala, 2001). These techniques, however, are debatable and performed worse than oversampling techniques.

To balance the dataset and match the effectiveness of the minority class, under-sampling entails taking fewer instances from the majority class (Drummond and Holte, 2003). Under-sampling is predicated on the notion that the majority class contains duplicate data that can be eliminated. Under-sampling has a major drawback in that it removes too many examples from the majority class when the imbalance is skewed. The lack of data could result in the algorithm performing worse. It is important to note that under-sampling expedites the learning process, which makes it an intriguing option when a dataset is unbalanced.

Oversampling involves repeating random examples of the minority class to rebalance the dataset (Drummond and Holte, 2003). Repeating certain samples doesn't add information, which is the fundamental drawback of oversampling, and it can overfit the learning algorithms. Additionally, because oversampling artificially expands the train set, learning time is prolonged.

A strategy for detecting credit card fraud using ML was proposed by Varmedja (2019). The dataset about credit card fraud used by the authors was gotten from Kaggle European credit card users' recent transactions are included in this dataset. The researcher used the Synthetic Minority Over-sampling Technique (SMOTE) over-sampling technique to address the dataset's class imbalance issue. To evaluate the effectiveness of the suggested approach, the following ML techniques were used: RF, NB, and multilayer perceptron (MLP). The experimental findings showed that the RF algorithm operated best, with a 99.96% accuracy rate in detecting fraud. The accuracy ratings for the NB and MLP approaches were 99.23% and 99.93%, respectively.

2.5 Conclusion

To achieve my goals and answer my research questions, my data was changed to a workable dataset with labels. These will enable us to reflect on and analyse the accuracy and interpretability of the models. We also had a larger dataset to work with, compared to the initial dataset used during the pilot study. This will improve productivity and efficiency.

CHAPTER THREE

3.1 Methodology

This chapter covers the method used in distinguishing between fraudulent and non-fraudulent transactions. The steps taken in this investigation are shown in the below diagram. These goals are to explore different machine models and compare the accuracy and interpretability trade of those models. Research methodology is essential as it aids in achieving the research purpose. The validity and reliability of the research might be impacted by the choice of the best methods and approaches. As a result, any method or strategy chosen must be supported considering the goals and objectives of this study. The first part of this paper discusses how to conduct the performance evaluation needed to develop a fraud detection model by adapting answers to the practical questions raised by the banking industry at each level of the fraud management process. It also considers a wide range of ML strategies that tackle these problems, such as the discrepancy between illegal and legal actions, the scarcity of completely trustworthy labels, the concept drift phenomena, and the unavoidable compromise between detection accuracy and interpretability. This state-of-the-art review throws some insight into a technological competition between intrinsically interpretable models that are enhanced for accuracy and black box ML models that are improved by post-hoc interpretation. Before discussing the methodology used in this investigation, we analysed the dataset at length.

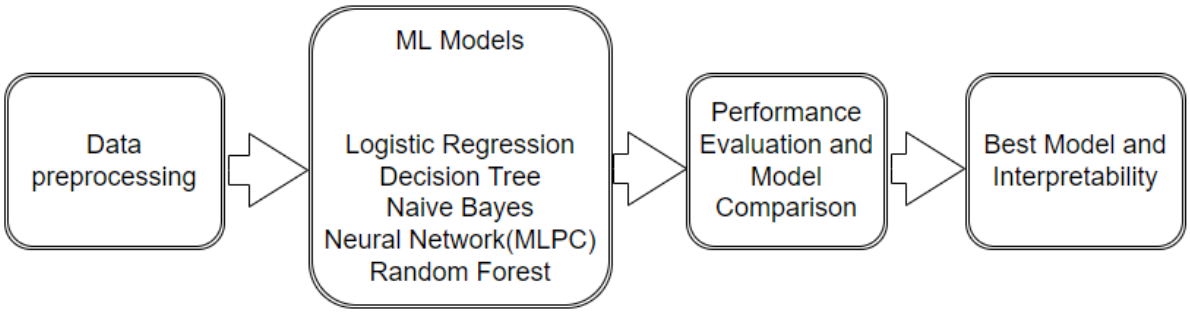


Figure 3.1 Classification Methodology

3.2 Dataset

The dataset used in this study was obtained by Emily Smith via Kaggle. A fictional credit card transaction is modeled after the data in this set, which includes both legitimate and fraudulent transactions. It considers purchases made with a total of 693 different businesses utilising the credit cards of 320 different clients. The total number of transactions in the dataset is 975036, and 5412 of those transactions were determined to be fraudulent while 969624 are non-fraudulent. It involves 22 variables with numerical and categorical features. Fraudulent transactions are represented as '1' while non-fraudulent transactions as '0'. The dataset was obtained from the below link. It involves transactions from February 2021 to March 2022.

https://www.kaggle.com/datasets/emilysmithh/credit-card-fraud-detection?resource=download&select=Data.csv

```
0    0.994449
1    0.005551
Name: fraud, dtype: float64
```

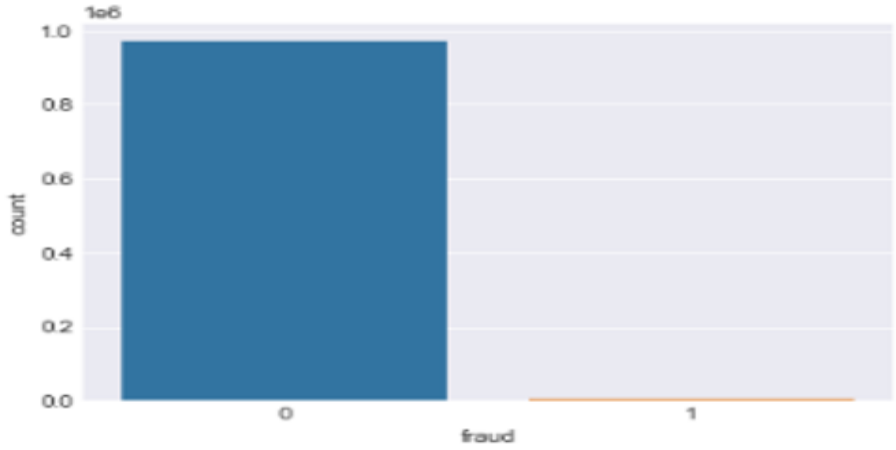


Figure 3.2 Imbalance on the target variable

	trans_date_trans_time	credit_card_number	merchant	category	amount	first_name	last_name	gender	street	city	state
0	02/01/2021 00:02	4750000000000	fraud_Kling-Grant	grocery_net	19.460000	Carrie	Washington	F	6114 Adams Harbor Suite 096	Kingsford Heights	IN
1	02/01/2021 00:03	4330000000000000	fraud_Huel-Langworth	misc_net	13.010000	Scott	Martin	M	7483 Navarro Flats	Freedom	WY
2	02/01/2021 00:05	4720000000000000000	fraud_Streich, Hansen and Veum	gas_transport	50.020000	Robert	Drake	M	463 Willie Estates	Burbank	OK
3	02/01/2021 00:06	180000000000000	fraud_Johns Inc	entertainment	6.110000	Jared	Camacho	M	4257 Perez Mall	Canton	OH
4	02/01/2021 00:08	4540000000000000000	fraud_Spinka Inc	grocery_net	32.140000	Nathan	Mendoza	M	767 Adam Mill Apt. 115	Espanola	NM

zip_code	latitude	longitude	city_population	job	day_of_birth	trans_number	unix_time	merchant_lat	merchant_long	fraud
70531	30.251000	-92.500200	1261	Broadcast presenter	07/01/1972	9b91366a6e0d4423eba54063e68e6bd4	1362182265	31.209622	-92.753662	0
36507	30.863500	-87.764400	19090	Science writer	5/30/1929	6562620b1a8fb8f80d2c725401e2584e	1362182331	30.433248	-88.487129	0
56672	47.087400	-93.919600	2097	Insurance risk surveyor	10/27/1987	041ae844a3776c559dcc28b08f2a65e4	1362182337	47.070617	-93.355866	0
35811	34.778900	-86.543800	190178	Television production assistant	04/01/1973	df878abf52d433fdf1699f6bb9e3c759	1362182373	35.545742	-86.735160	0
52576	41.200100	-92.135400	568	Commercial horticulturist	7/24/1969	894e23c22aff3c694a317251dc1b0641	1362182379	42.085221	-92.755333	0

Figure 3.3 Original dataset

Below is a graph showing the descriptive statistics for the amount characteristic of the contrast between fraudulent and non-fraudulent transactions. For a normal distribution, the output claims that the minimum and maximum values of the amount feature are 1,00 and 28948.9, but for a fraudulent distribution, they are 1,18 and 1371.81. The output shows that the average of the legitimate distribution for the amount feature is $67.63, whereas the average of the fraudulent distribution is $530.57. When looking at the standard deviation, the same analysis holds.

	Row Type	Overall Amount Distribution	Non-Fraudulent Amount Distribution	Fraudulent Amount Distribution
0	count	975036.000000	969624.000000	5412.000000
1	mean	70.213255	67.632401	532.603975
2	std	160.831716	154.783802	391.002744
3	min	1.000000	1.000000	1.180000
4	50%	47.420000	47.200000	409.770000
5	95%	195.970000	189.780000	1084.595500
6	99.9%	1503.557250	1505.843770	1288.749420
7	max	28948.900000	28948.900000	1371.810000

Figure 3.4 Descriptive statistics of the Feature (Amount)

3.3 Data Pre-processing

Preprocessing data is essential before the application of ML models. The predictive output can be influenced by the different models and the various methods used. The goal of data preparation is to make the data more manageable and standardised before analysis, to detect and eliminate missing values, and to increase the number of unique records in the dataset. The dataset is both categorical and numerical. The categorical dataset must be encoded before using them for modeling. By applying feature scaling, the independent variables are within the same range. An application of the box-cox transformation was made to lessen feature skewness. On the unbalanced initial dataset, resampling techniques including under-sampling, oversampling, SMOTETomek, and SMOTEENN were used to prevent overfitting and bias in our training model. To carry out the purpose of this project, we have used the Python data processing package pandas and the ML library sci-kit learn. The step for this session is shown below.

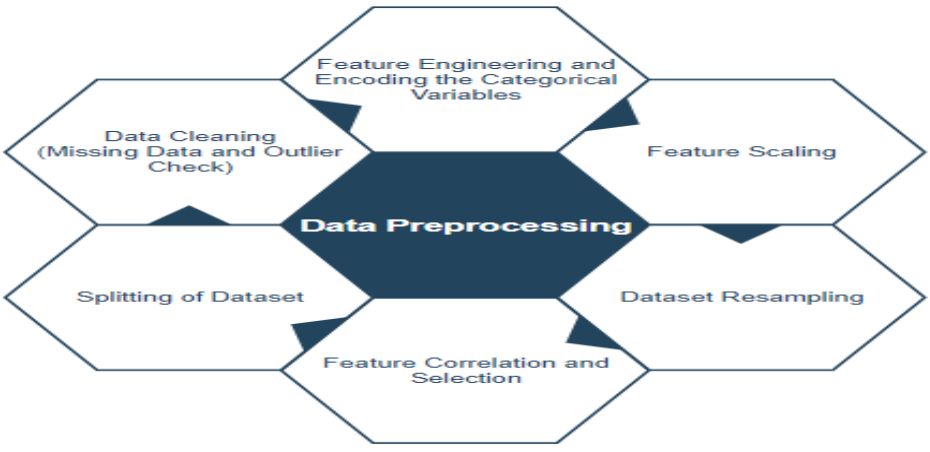


Figure 3.5 Dataset preprocessing steps

3.3.1 Data Cleaning

The data cleaning process was conducted on the credit card dataset imported into python. The data cleaning process included handling null and missing values and handling outliers.

The dataset has 22 variables and a total of 975036 transactions. There were no null values in this dataset. More importantly, our dataset has no missing data.

```
df.isnull().sum().max()
```

```
0
```

Figure 3.6 Null value

```
Index(['trans_date_trans_time', 'credit_card_number', 'merchant', 'category',
       'amount', 'first_name', 'last_name', 'gender', 'street', 'city',
       'state', 'zip_code', 'latitude', 'longitude', 'city_population', 'job',
       'day_of_birth', 'trans_number', 'unix_time', 'merchant_lat',
       'merchant_long', 'fraud'],
      dtype='object')
```

There are 22 columns in the dataset

```
df.shape
```

```
(975036, 22)
```

Figure 3.7 Shape of the dataset

	credit_card_number	amount	zip_code	latitude	longitude	city_population	unix_time	merchant_lat	merchant_long	fraud	year	hour	age
count	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000
mean	417197755852789416.000000	70.213255	48815.857405	38.534111	-90.231395	89042.679016	1345505091.924963	38.534049	-90.231559	0.005551	2021.105341	12.805828	48.769642
std	1308903056712911616.000000	160.831716	26893.982150	5.076803	13.754579	302594.109465	9320993.106374	5.111292	13.756833	0.074295	0.306992	6.817871	17.362207
min	60416207185.000000	1.000000	1257.000000	20.027100	-166.672300	23.000000	1328054544.000000	19.027785	-166.671242	0.000000	2021.000000	0.000000	17.000000
25%	180000000000000.000000	9.640000	26237.000000	34.620500	-96.798000	743.000000	1337961209.250000	34.730153	-96.900836	0.000000	2021.000000	7.000000	35.000000
50%	3520000000000000.000000	47.420000	48174.000000	39.354300	-87.476900	2456.000000	1345380789.000000	39.364304	-87.446250	0.000000	2021.000000	14.000000	47.000000
75%	4640000000000000.000000	83.010000	72042.000000	41.940400	-80.158000	20328.000000	1354320131.750000	41.958414	-80.239186	0.000000	2021.000000	19.000000	60.000000
max	4980000000000000000.000000	28948.900000	99783.000000	66.693300	-67.950300	2906700.000000	1362182379.000000	67.510267	-66.950902	1.000000	2022.000000	23.000000	98.000000

Figure 3.8 Descriptive Statistics

Outliers

Outliers can be referred to as observations that are distant from the other data. In figure 3.8 below, most transactions have amounts between 0 and roughly 2500, but there are a few outliers with exceptionally large amounts. If the outliers are very few, excluding them from our analysis could be sensible. Additionally, we should be aware that these outliers should not represent fraudulent transactions. In most cases, fraudulent transactions involving large sums of money and their removal from the data can lead to inaccurate model forecasting.

Therefore, we can effectively create a model that predicts transactions as fraud realistically and unaffected by outliers. It might not be all that helpful to train our model on these extreme outliers.

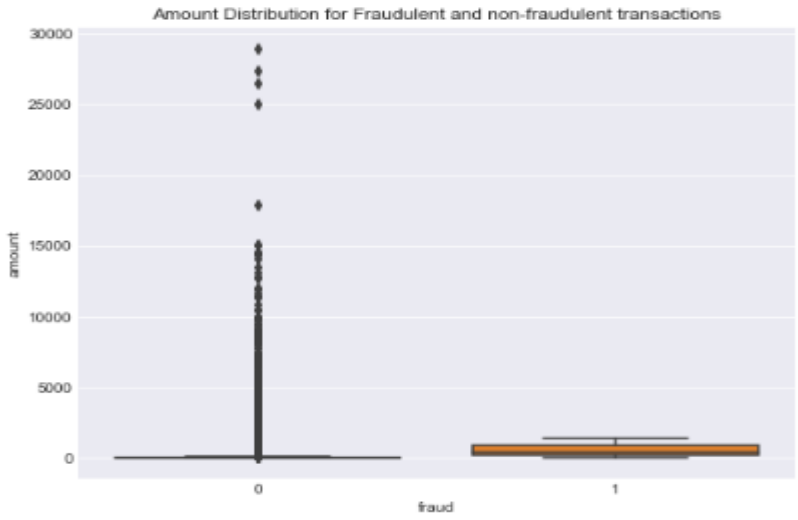


Figure 3.9 Distribution of Amount concerning *the target variable.*

3.3.2 Feature Engineering

Under this heading, we will be constructing data visualization for a comprehensive view of the overall distribution of the data. Feature engineering is also needed from the raw data to be used by the models to scale, integrate and select the features for better performance. One of the aims of the project is to improve the accuracy and precision and interpret the trade-off between ML by processing feature engineering on the dataset. We will be visualising various features.

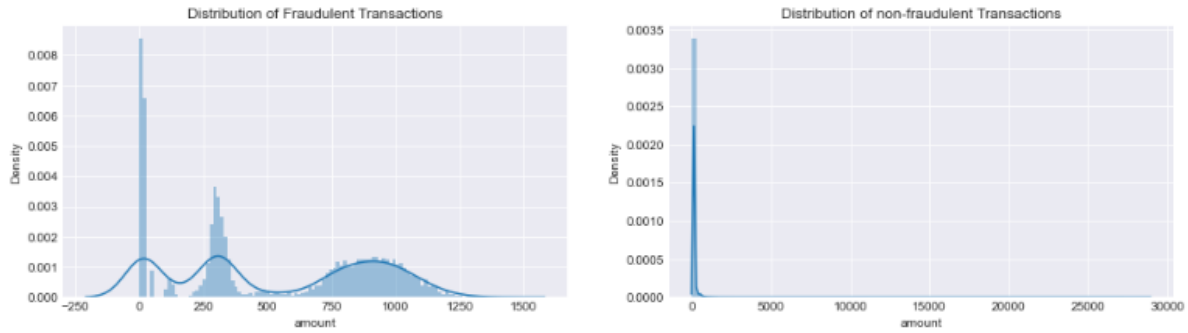


Figure 3.10 Comparison of the amount concerning *the target variable.*

From the above figure, the amount involved in fraudulent transactions is lesser than in non-fraudulent transactions. More diagrams will be added to the appendix.

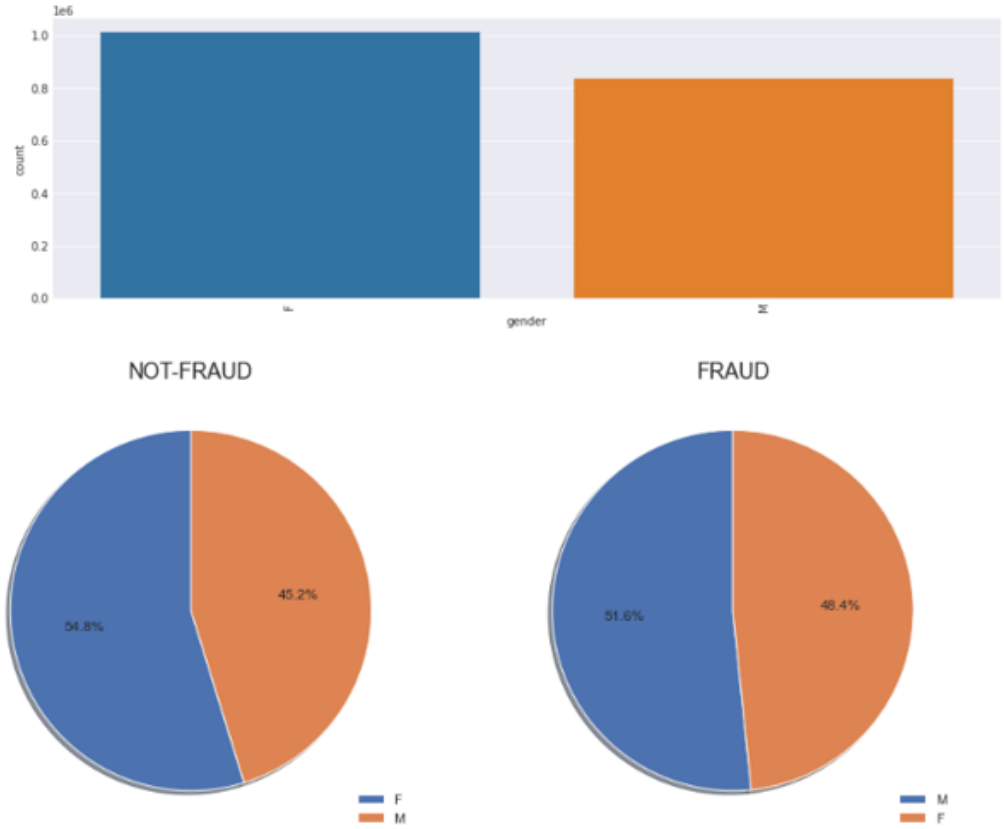


Figure 3.11 Gender distribution concerning *the target variable*.

As can be seen in the preceding visualisation, women represent a significant amount of the total number of transaction frequencies. Although women made up a somewhat larger share of the fraudsters (51.6%) than men (48.4%), fraud committed by women accounted for only 0.49% of all transactions, whereas fraud committed by men accounted for approximately 0.63%.

It can be inferred that Females transact more than males, and will be prone to error, the females can be educated and trained to be more careful since they are prone to fraud due to their transaction frequency. The males are a bit more drawn away to be involved in fraud, although the fraud rate is almost the same as the females.

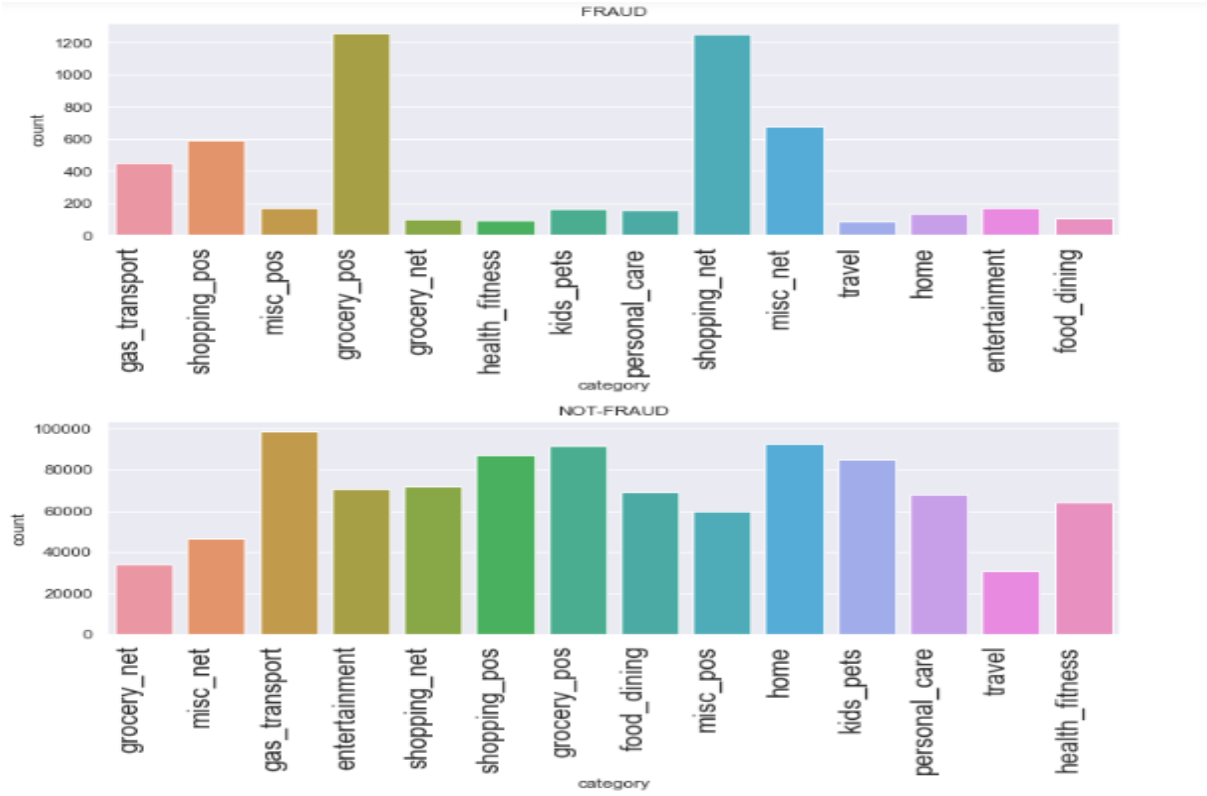


Figure 3.12 EDA for the category feature

From the above figure, it can be observed that about three categories have more than 1% of fraudulent transactions involved. These are - shopping_net, misc_net, and grocery_pos. The shopping_net category has the highest number of fraudulent transactions. The fraudsters may identify a weak link in the category and are taking advantage of it. These three categories need to be reviewed to reduce the fraud rate.

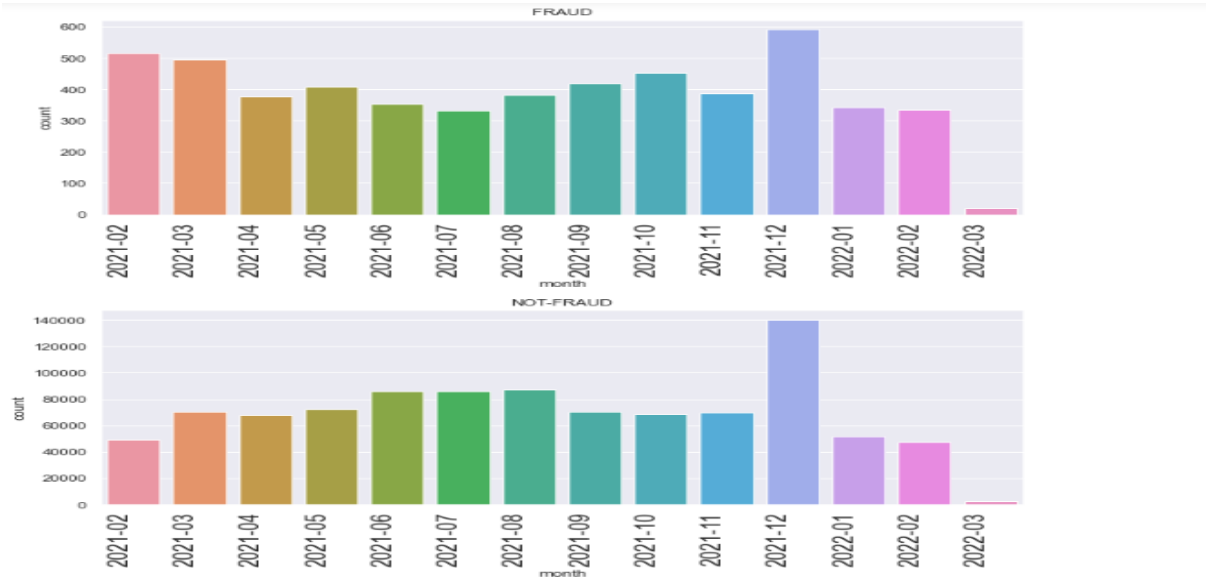


Figure 3.13 EDA for the Month feature

According to the 'trans_year_month' data, most transactions happened in the February, March, July, and December months of 2021 December is the month with the highest fraudulent activity and this can be linked to the festive season, as people tend to spend more during this season.

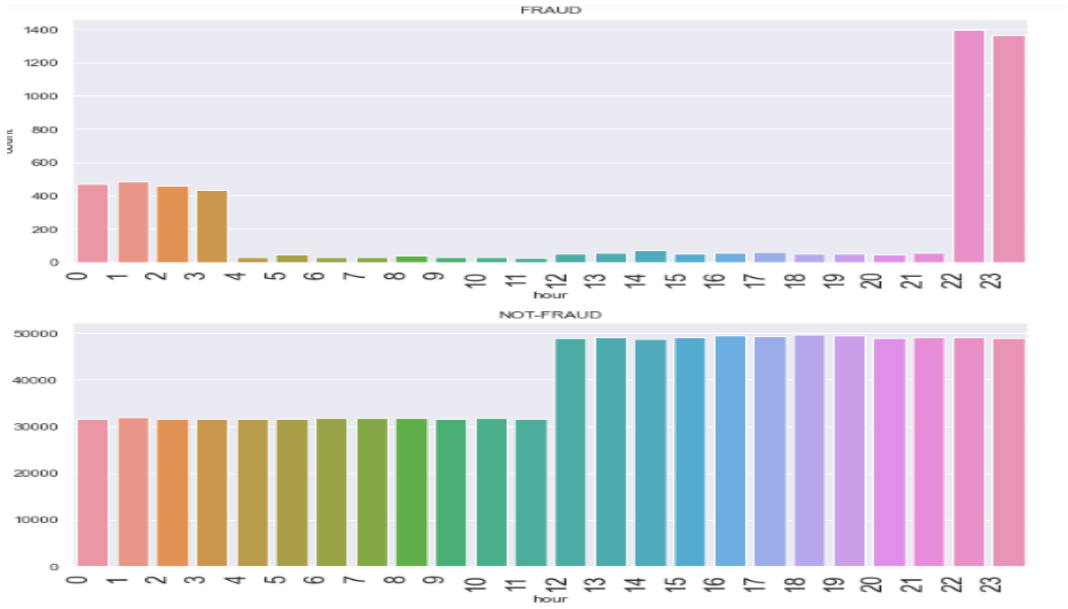


Figure 3.14 EDA for the time feature

The above plot shows that fraudulent transactions occurred in the late hours of the day and early hours. it means fraudsters operate mostly at midnight when people are sleeping.

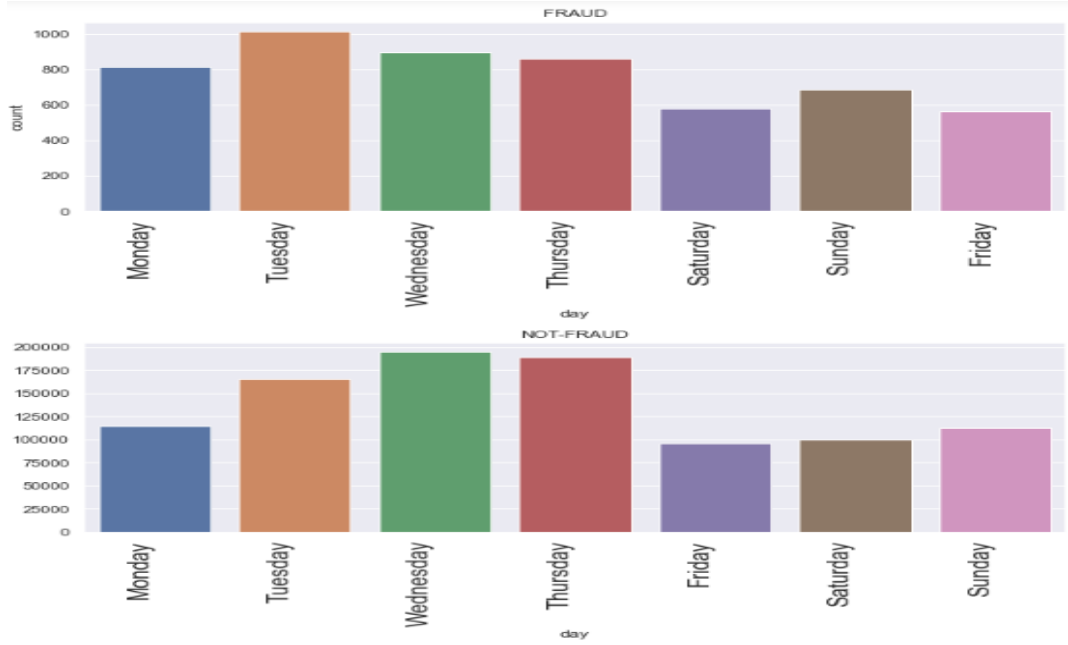


Figure 3.15 EDA for the day feature

Most of the transactions occurred on Tuesday, Wednesday, and Thursday as seen above.

```
Age Age-Range
20-29   Youth
30-39   Young Adult
40-59   Middle Aged Adult
60-100  Old Adult
```

	age_range	fraud	Transaction count	age_count	Transaction percentage
0	Youth	0	124789	125550	99.393867
1	Youth	1	761	125550	0.606133
2	young_Adult	0	228215	229146	99.593709
3	young_Adult	1	931	229146	0.406291
4	middle_aged_adult	0	376308	378296	99.474486
5	middle_aged_adult	1	1988	378296	0.525514
6	old_adult	0	230454	232142	99.272859
7	old_adult	1	1688	232142	0.727141

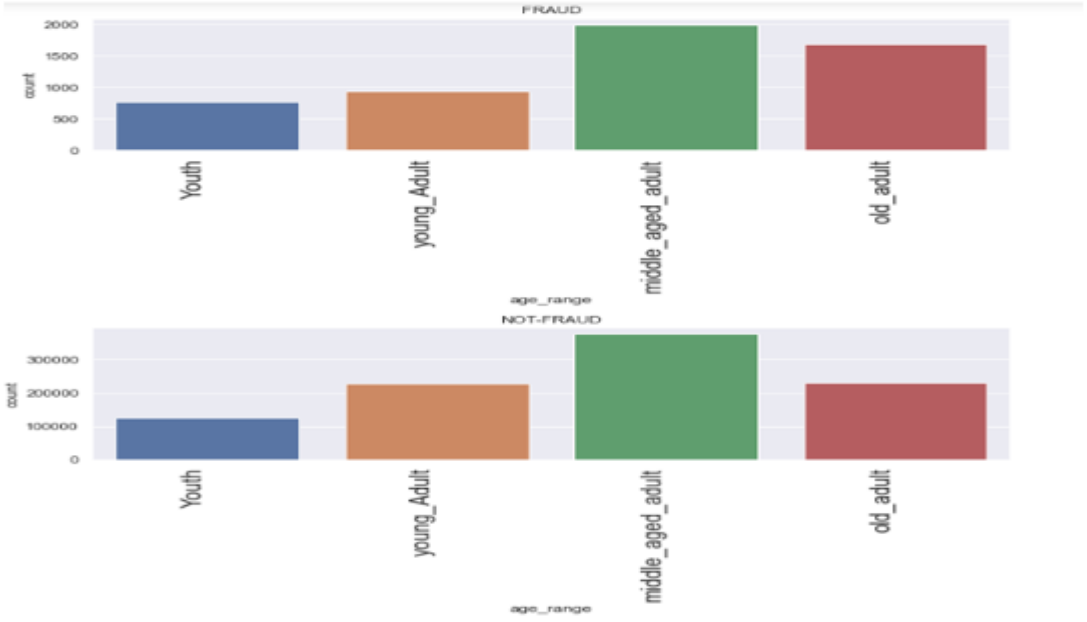


Figure 3.16 EDA for the age feature

The plots above show that people between the ages of 40-59 performed the highest number of transactions in the dataset. Furthermore, the 30-39 age group has also processed a substantial number of transactions. In terms of the total number of transactions done by a certain age group, the age group between the ages of 40-59 is the most affected, with approximately 1% of transactions performed by these people being fraudulent, this can be linked to the transaction frequency recorded around this age group. These people are much more susceptible to fraud; thus, their transactions should be monitored with greater attention, and they should be informed about the fraud that is occurring to curb fraudulent transactions.

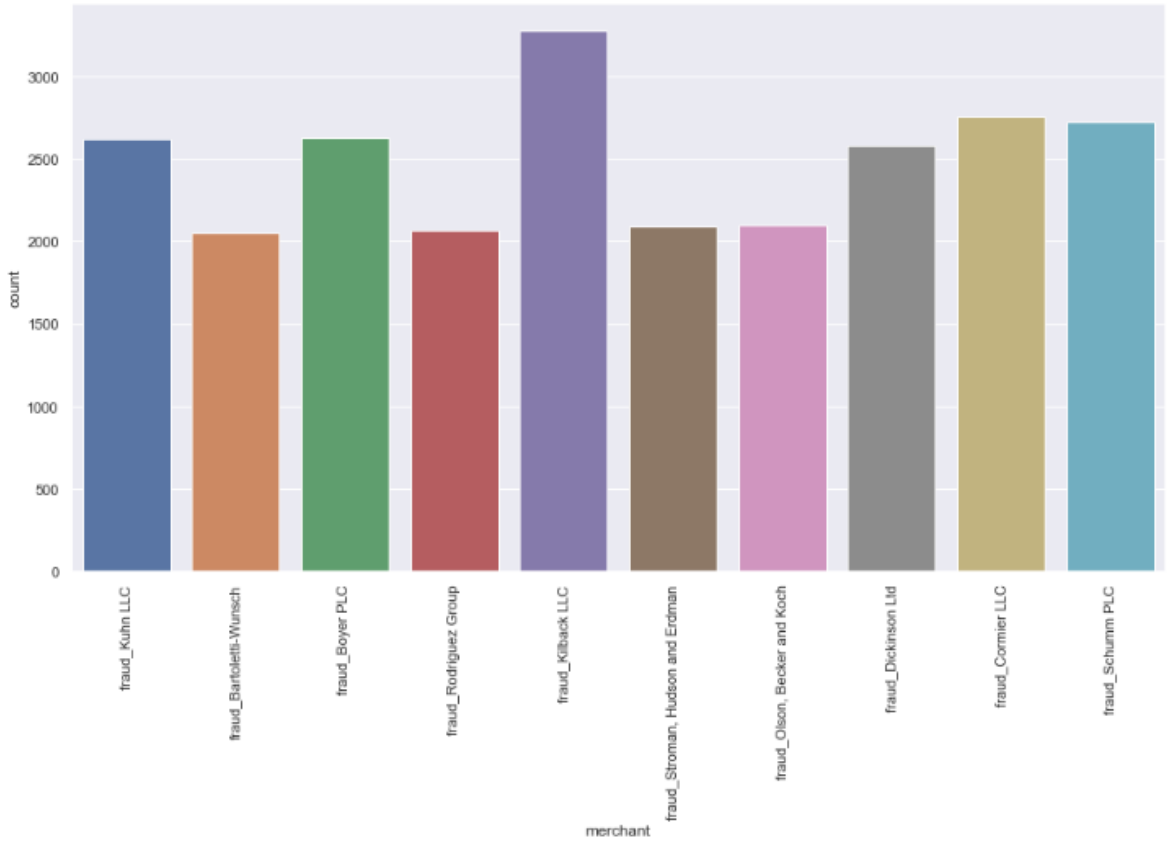


Figure 3.17 Top *merchants with high transaction volumes*

	merchant	fraud	Transaction count	merchant_count	Transaction percentage
481	fraud_Herman, Treutel and Dickens	1	28	950	2.947368
660	fraud_Kozey-Boehm	1	34	1396	2.435530
394	fraud_Goyette Inc	1	33	1495	2.207358
678	fraud_Kuhic LLC	1	32	1464	2.185792
144	fraud_Boyer-Reichert	1	31	1441	2.151284
345	fraud_Fisher-Schowalter	1	31	1465	2.116041
1203	fraud_Terry-Huel	1	31	1493	2.076356
1179	fraud_Streich, Dietrich and Barton	1	30	1464	2.049180
925	fraud_Padberg-Welch	1	37	1817	2.036324
390	fraud_Gottlieb, Considine and Schultz	1	30	1477	2.031144
553	fraud_Jast Ltd	1	30	1477	2.031144
380	fraud_Gleason-Macejkovic	1	31	1527	2.030124
1230	fraud_Towne, Greenholt and Koepp	1	30	1480	2.027027
596	fraud_Kerluke-Abshire	1	28	1383	2.024584
834	fraud_Miller-Harris	1	19	947	2.006336
525	fraud_Hudson-Ratke	1	37	1865	1.983914
1085	fraud_Schmeler, Bashirian and Price	1	29	1464	1.980874
1259	fraud_Vandervort-Funk	1	36	1830	1.967213
868	fraud_Mosciski, Ziemann and Farrell	1	28	1467	1.908657
1025	fraud_Rohan, White and Aufderhar	1	19	997	1.905717

Figure 3.18 Top *20* merchants *with high fraudulent transaction volumes*

It has been noted that there are more than 600 distinct values; this will make plotting and analysis difficult. As a result, we will only examine the top retailers who have the most fraudulent transactions. It can be seen from the plots and data frames provided above that 144 merchants have more than 1% of their transactions classified as fraudulent. It is possible to check for fraudulent activity at the above merchants. Additionally, they can be informed about current fraudulent activities and taught how to stay clear of them.

```
df.job.nunique()
```

```
491
```

```
# Select top 10 job with fraudulent transaction
df["job"].value_counts().head(10)
```

```
Film/video editor             7426
Exhibition designer           6887
Surveyor, land/geomatics      6539
Naval architect               6523
Materials engineer            6284
Designer, ceramics/pottery    6181
Systems developer             5816
IT trainer                    5782
Financial adviser             5740
Environmental consultant      5703
Name: job, dtype: int64
```

```
#20 jobs with high transaction frequencies
high_trans_freq_jobs = df.job.value_counts().head(20).index.tolist()
print(high_trans_freq_jobs)
```

```
['Film/video editor', 'Exhibition designer', 'Surveyor, land/geomatics', 'Naval architect', 'Materials engineer', 'Designer, ce
ramics/pottery', 'Systems developer', 'IT trainer', 'Financial adviser', 'Environmental consultant', 'Chartered public finance
accountant', 'Copywriter, advertising', 'Chief Executive Officer', 'Scientist, audiological', 'Comptroller', 'Sub', 'Paramedi
c', 'Agricultural consultant', 'Podiatrist', 'Magazine features editor']
```

Figure 3.19 Top 10 jobs *with high fraudulent transaction volumes*

It has been noted that there are more than 400 distinct values; this will make plotting and analysis difficult. As a result, we will only examine the top retailers who have the most fraudulent transactions. The jobs listed above have high transaction frequencies that have been observed. The transactions in the data frame with the job feature of 'Film/video editor', 'Exhibition designer', 'Surveyor, land/geomatics', 'Naval architect', 'Materials engineer', 'Designer, ‘ceramics/pottery’, 'Systems developer', 'IT trainer', 'Financial adviser' etc. have been seen to have completed fraudulent transactions. People in employment with a high number of fraudulent transactions can be warned about credit card transaction fraud so that they can use their credit cards more carefully. The fact that almost all transactions are fraudulent may indicate that there is a problem with the data point. That is, the person representing a certain profession may be at fault because it is extremely uncommon that all transactions performed by a person representing specific employment are fraudulent. As a result, conducting a background check on the credit card holder may be beneficial in this scenario. More features will be discussed in the appendix session.

3.3.3 Feature encoding

In the literature, several options for encoding categorical features are provided. Label encoding and one-hot encoding are the two most prevalent options Kotsiantis (2007).

Label encoding is an encoding approach that transforms a category feature with n categories into a numerical feature with n distinct number values. However, after being converted into numbers, the categories become sorted using label encoding. This might present problems for classifiers that try to calculate a distance between samples, such as LR, neural networks, or SVMs classifiers. Tree-based classifiers, on the other hand, are unaffected. However, because their method of classification involves dividing the dataset into categories, tree-based classifiers are unaffected by the order of the categories owing to label encoding.

One hot encoding technique is converting a categorical feature with n categories into a binary feature with n - 1 that describe the values of the categorical feature. This approach is suitable for distance-based classifiers because it does not rank the categories; rather, all of them are equally distant from one another, unlike label encoding, where some categories would be mistakenly thought to be closer than others.

Class embeddings are another excellent method that preserves the proper spacing between categorical feature categories. Additionally, it uses a lot less memory than one-hot encoding. It entails mapping a feature's various categories to a k-dimensional space (with k n-categories) and then modifying the mapping using a gradient descent procedure. Using embeddings, Guo and Berkhahn (2016) encoded categorical features in a supervised manner. The pre-one-hot encoded feature set is likened to having an additional layer of neurons thanks to the embedding maps. Gradient descent is used to adjust them to the classification task.

Gradient descent is used to adjust them to the classification task. Word2Vec embeddings were recently employed by [Russac et al., 2018] to replace one-hot encoding for a distance-based classifier (LR) in a CCFD challenge. This work was done in partnership with the chair of data science at the University of Passau.

Frequency encoding can be helpful in the event of an unbalanced classification problem like detecting credit card fraud, as demonstrated by Carcillo et al (2018) and Pozzolo (2015). The ratio of the Minority Class to the Majority Class is used to substitute the category's value in frequency encoding (the percentage of fraud in this category for our application).

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

Figure 3.20 Sample of converted categorical features using One-Hot Encoder

3.3.4 Feature Selection

The characteristics of typical credit card transactions are numerous. They give too many categories or are too scarce, some attributes like the cardholder's city may not be significant for the classifiers or may cause overfitting problems. Additionally, some features could have redundant information (for instance, the ZIP code includes the county value) or be strongly associated, such as the relationship between the cardholder's age and the average transaction value.

In this situation, feature selection might be helpful to reduce information duplication, clean up the feature set, and maintain only the data that is pertinent to the ML task. In addition, feature selection would result in a reduced number of features and less connected features with duplicate data, which would improve comprehension of the classifier's choice. Additionally, feature selection may improve classifier performance (less overfitting) and speed (smaller feature sets and datasets) Bhattacharyya et al (2011).

Noghani and Moattar, (2015) suggested an iterative method for choosing features in a program that detects credit card fraud. The features that boost a RF classifier's performance the most are gradually added.

Leopold Fossi and Gabriele Gianini performed feature selection for CCFD using the Shapley value Shapley, (1953) Fossi and Gianini, (2019). The Shapley value measures a player's contribution to the team. A good player may not contribute much because another individual already brings his abilities to the team, whereas a less-than-stellar player may satisfy the team's needs and gain more from it.

When the data is supplemented with many new informative features, feature selection could become necessary. At various times of the day, Fawcett and Provost (1997) proposed to characterise the evolution of pertinent CCFD features. To reduce the overall number of features, they then indicated that a feature selection process was necessary.

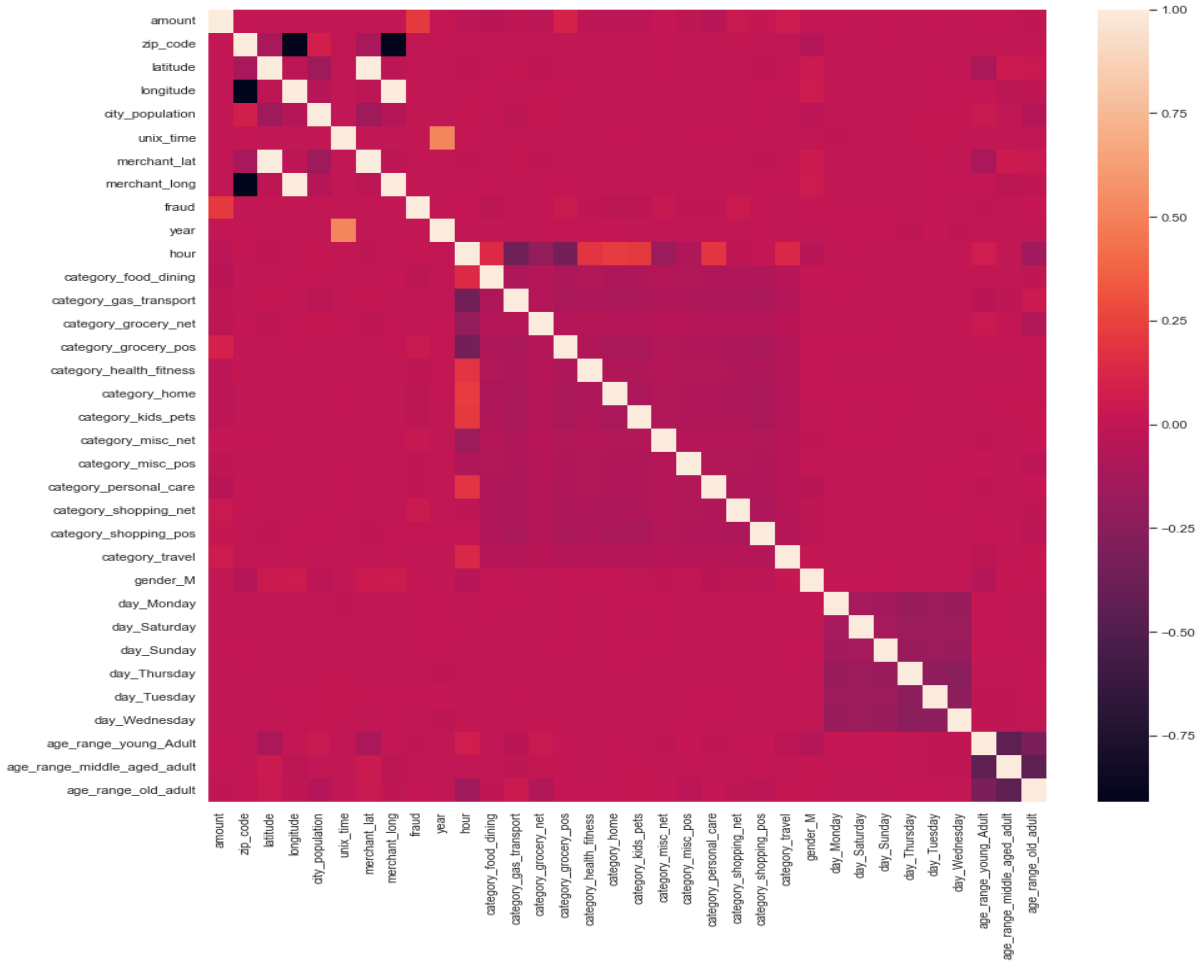


Figure 3.21 Correlations between the columns

There are a lot of variables, we need to get the variables with the highest correlation using a function that outputs the variables with the correlation between them and above a certain threshold.

```
#getting the correlation above 85%
corr_features = correlation(data,0.85)
corr_features
```

```
{'merchant_lat', 'merchant_long'}
```

Figure 3.22 Correlation above 85%

There are no features that correlate above 85%. Hence, it is safe to proceed without model building.

3.3.5 Feature Correlation

While there are many datasets to choose from, not all datasets can help you build a machine learning model that can make the necessary prediction. It's possible that utilising some of the features will result in more precise forecasts. Accordingly, enhancing ML models requires feature correlation. Traits with a high correlation are more prone to be linearly dependent, meaning that they have about the same amount of influence on the dependent variable. If there is a strong relationship between the two characteristics, we can rule out one of them. A heatmap showing

the similarity between the original and resampled data sets is shown in Figure 3.23 (both low and high sampling rates). Given the size of the dataset and the little insight provided by the heatmap, we opted to conduct feature selection to enhance in the prioritisation process.

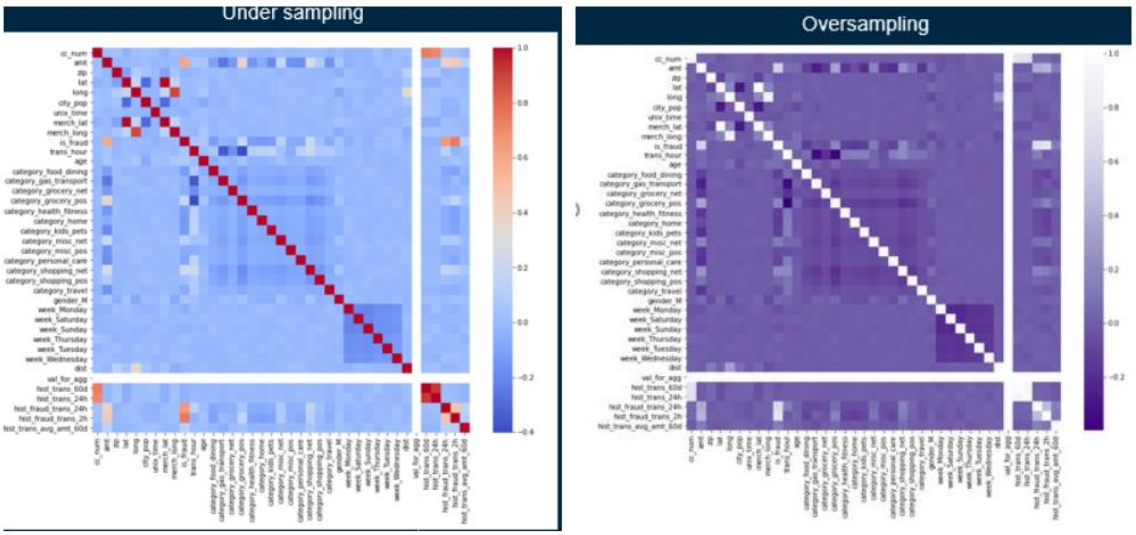


Figure 3.23 Heatmap for Undersampling and Oversampling

When unnecessary features offer no more meaningful information than the existing subset of variables, the collecting of critical features for the ML model's implementation is known to decrease the training period, make learning easier to understand, interpret, and reduce model over-fitting**.** As a result, one of the crucial processes in data preprocessing is feature selection. The dataset's comprehensive and extremely detailed information may have a significant impact on how well our model performs.

3.3.6 Feature scaling

Most features in a real-world dataset will have range, magnitude, and unit. When a feature's magnitude is greater than the others, a problem occurs since that feature will inevitably take precedence over other features. The influence of different quantitative units should therefore be eliminated by scaling raw data to meet classification algorithms. Zhang (2020). To rescale the characteristics between 0 and 1, the StandardScaler approach was employed in this study. The distribution that is produced has a standard deviation of 1. Because variance equals standard deviation squared, the variance is likewise equal to 1. Additionally, 1 squared is 1. The mean of the distribution is around 0, this is due to the StandardScaler.

$$z = \frac{x - \mu}{\sigma}$$

```
#scaling
scaler = StandardScaler()
X = scaler.fit_transform(X)
```

Figure 3.24 Scaling the values

3.4 Dealing with imbalanced data

The class imbalance in fraud datasets, with the "fraud" class, sometimes amounting to less than 1% of the entire dataset, is one of its unique characteristics. Additionally, the two classes overlap. Therefore, these two occurrences render standard supervised ML techniques useless. The aim is to give the minority class greater weight by changing the algorithm utilised or the data itself using one of three techniques: under-sampling negative observations, over-sampling positive observations, or hybrid methods that mix under- and over-sampling. Below is the method used in this study.

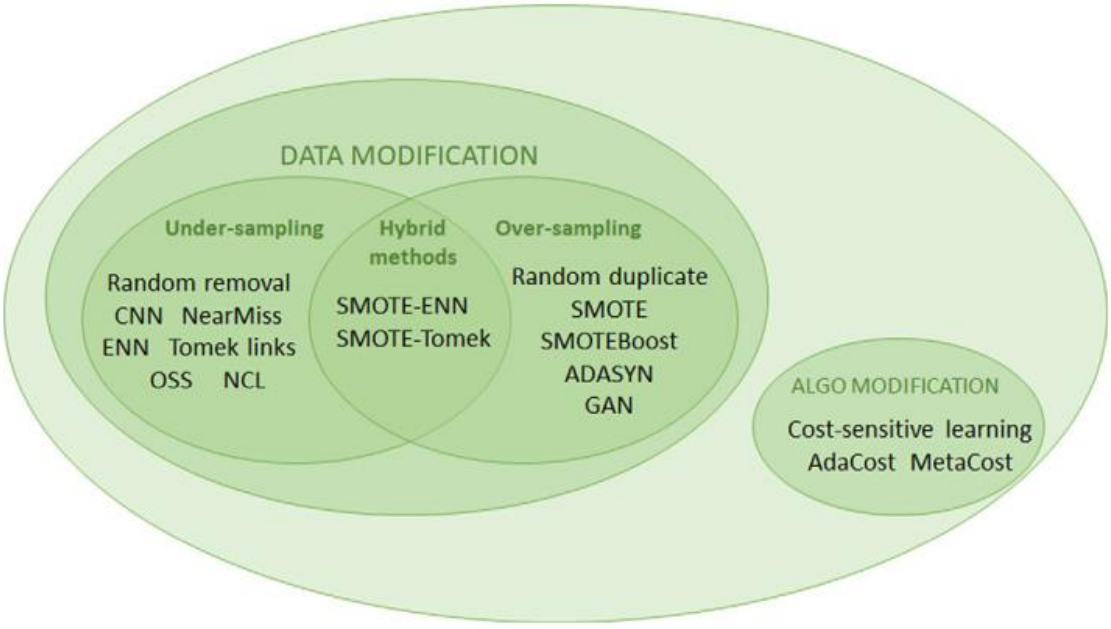


Figure 3.25 Summary of solutions to deal with imbalanced data

Source: *Published online by Cambridge University Press*

3.4.1 Under-Sampling

One of the most popular sampling methods involves randomly deleting members of the majority class to reduce the sample size of the majority class to an equal number of members of the minority class. The main issue with RUS is that it removes data at random, potentially resulting in the loss of vital information that may have been recorded. Pozzolo (2015)

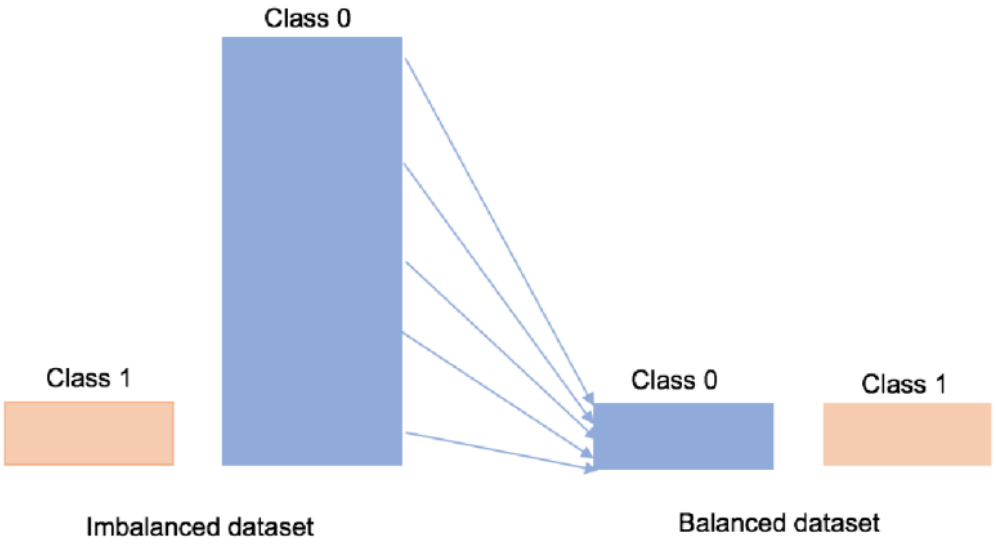


Figure 3.26 Under-sampling approach

Source: *KDnuggets* 2022

3.4.2 Over-Sampling

Minority class oversampling replaces minority class instances with new ones at random while ignoring the majority class. This tactic intentionally duplicates minority class examples, which doesn't offer any new information but raises the minority class examples' misclassification cost and, ultimately, may result in overfitting on the few artificially duplicated cases. Oversampling naturally lends itself to intentionally generating minority class instances that are similar but distinct.

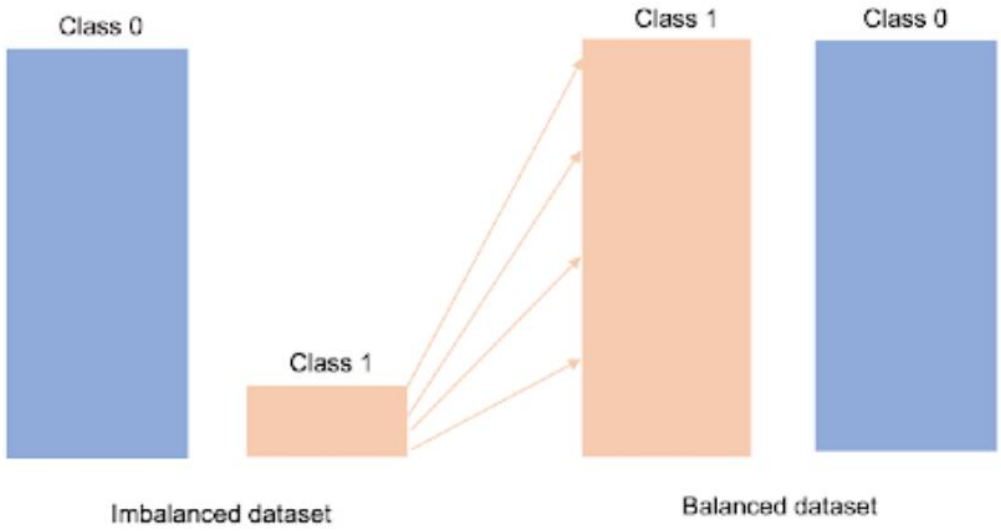


Figure 3.27 Over-*sampling approach*

Source: *KDnuggets* 2022

3.4.3 SMOTE

To successfully populate the feature space in regions where minority instances are present, Chawla (2002) suggests SMOTE, an oversampling strategy that creates additional artificial examples of the minority class by interpolating between close minority class examples. The authors compare the technique to naive random oversampling for a C4.5 DT. They successfully reduce the number of disjuncts in the generated tree by interpolating minority class examples with synthetic examples, resulting in greater generalisation as compared to the specialisation effect that emerges from randomly replicating minority class cases. The method has since prompted additional studies and extensions Nitesh (2013) and Hui Han (2005) and has also been used to balance classes in CCFD Pozzolo (2013).

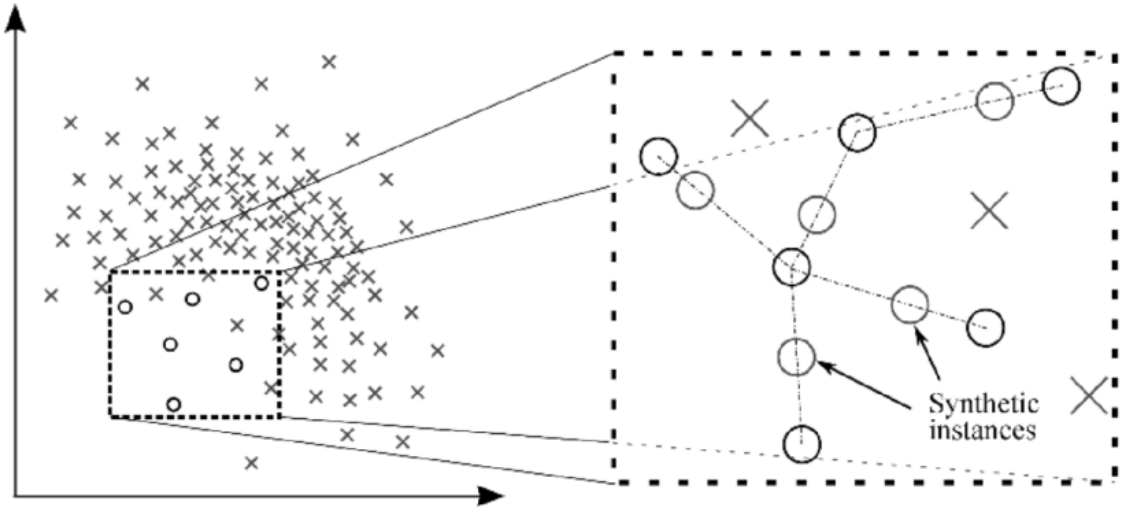


Figure 3.28 SMOTE *approach*

Source: *KDnuggets* 2022

3.4.4 SMOTETomek

This is a hybrid sampling method and is a technique for adjusting the distribution of class sizes that includes the ideas of both under-sampling and oversampling. It employs Tomek links for under-sampling and SMOTE for oversampling. Oversampling methods, such as SMOTE and ADSYN, introduce noise and outliers into the minority class Morais (2009), which has an impact on the newly oversampled dataset. The ML models that might be created with this new dataset could be negatively impacted. Therefore, it's important to come up with practical solutions for handling this noise. To do this, SMOTE Tomek employs Tomeklinks to lessen the noise that the oversampling SMOTE method introduces Wang (2019) and Boardman (2020).

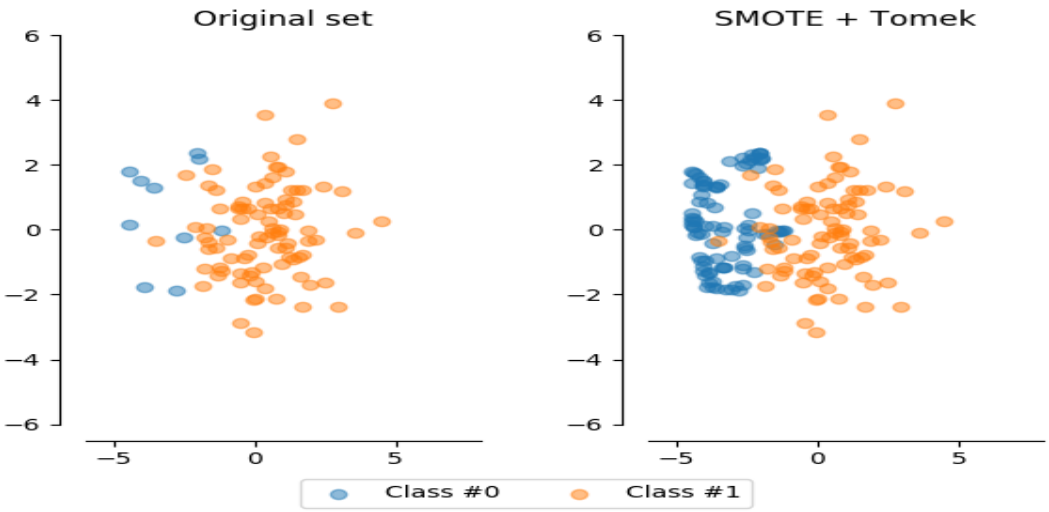


Figure 3.29 SMOTETomek *approach*

Source: *Lemaitre* (2016)

3.4.5 SMOTEENN

This technique, created by Batista et al. (2004), combines the SMOTE ability to create synthetic examples for the minority class and the ENN ability to delete observations from both classes if they are found to have a different class from their K-nearest neighbour in the majority class. This application of SMOTEENN, combines under-sampling with the use of ENN and oversampling with the usage of SMOTE, and can produce excellent results. To detect noise in data, the under--sampling technique ENN (Edited Nearest Neighbor) computes KNN for each minority class example. Based on these results, it determines whether there are imposters among those samples; if so, the example is eliminated.

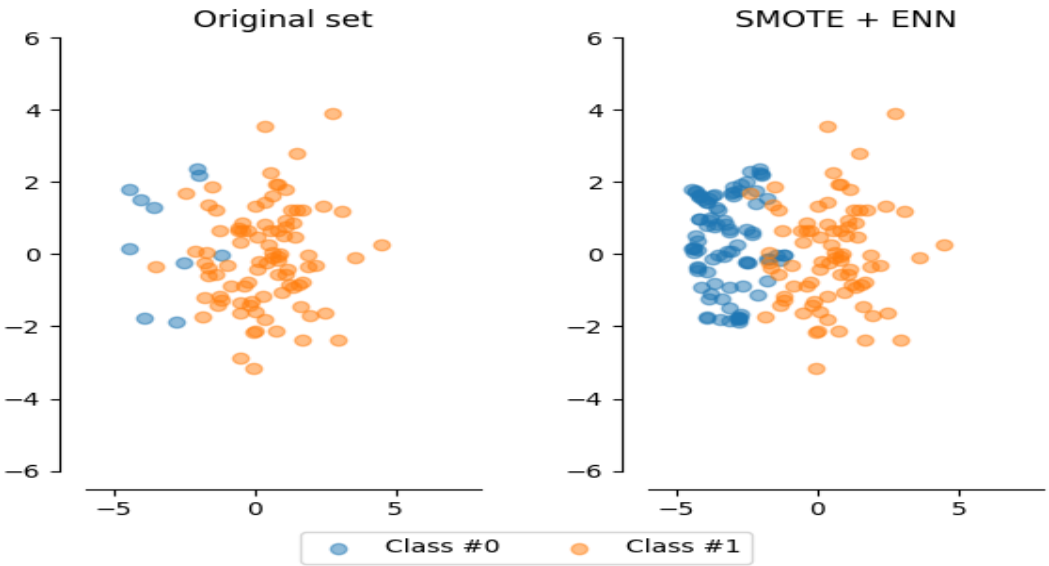


Figure 3.30 SMOTETomek *approach*

Source: *Lemaitre* (2016)

3.5 Machine learning Algorithms

In this study, we analysed the fraudulent transactions, using we used both supervised (classification) ML models. The ML models used in this inquiry are covered in the next subsection. We also discussed how to build a model and select the best hyperparameter settings.

3.5.1 Logistic Regression

In terms of classification and regression problems, this is the simplest solution. It has several applications, including email classification, tumor diagnosis, and spam filtering. The probability is determined by whether the outcome is binomial or multinomial. The sigmoid function is employed to characterise the data and the connection between the dependent and independent variables. The data can be used in the ongoing investigation to establish if a certain transaction is fraudulent. While highly effective, it has the potential to overfit high-dimensional datasets. It outperforms competing methods while making no assumptions about how classes are distributed in the feature space. A limitation of this method is that it makes the linearity assumption between dependent and independent variables. Classification and regression are two types of supervised learning, and their respective yield factors vary depending on the task at hand. The classification problem is concerned with putting several input variables into the precise category to which they belong when the algorithm's output falls into one of the various pre-selected categories. It employs a binary classification in which the conditional probability of one of the two interpretations of the response variable is set to compare a linear combination of two or more input variables adjusted by the logistic function (thus, the other name for this model: logical model). The goal of a binary classification model is to correctly predict one of two classes for a response variable (often 0 or 1). An explanation of the LR may be found in the Logistic Function, often referred to as the Sigmoid Function, which takes an arbitrary real input x and returns a probability value between 0 and 1.

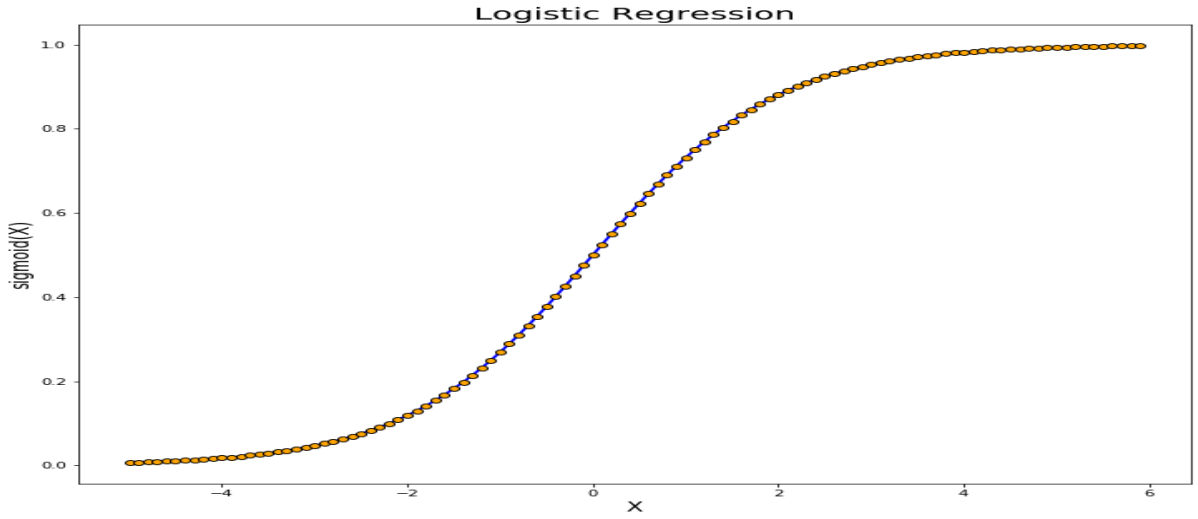


Figure 3.31 Logistic Regression

Source: Toward data science

3.5.2 Decision Tree

Machine learning systems use state-of-the-art algorithms for data analysis. In practical ML contexts, the DT model is the most used approach. DTs perform at an exceptional rate and level of intelligence, particularly when employed to gather and assess large amounts of data. To complete a transaction, the DT model uses the previously retrieved features. The process begins with a central inquiry, or "root question," and then "branches" out from there, employing specifics to develop "components" that culminate in "leaves," or conclusions. When continuous data is divided based on a specific parameter, regression and classification using DTs are called supervised learning.

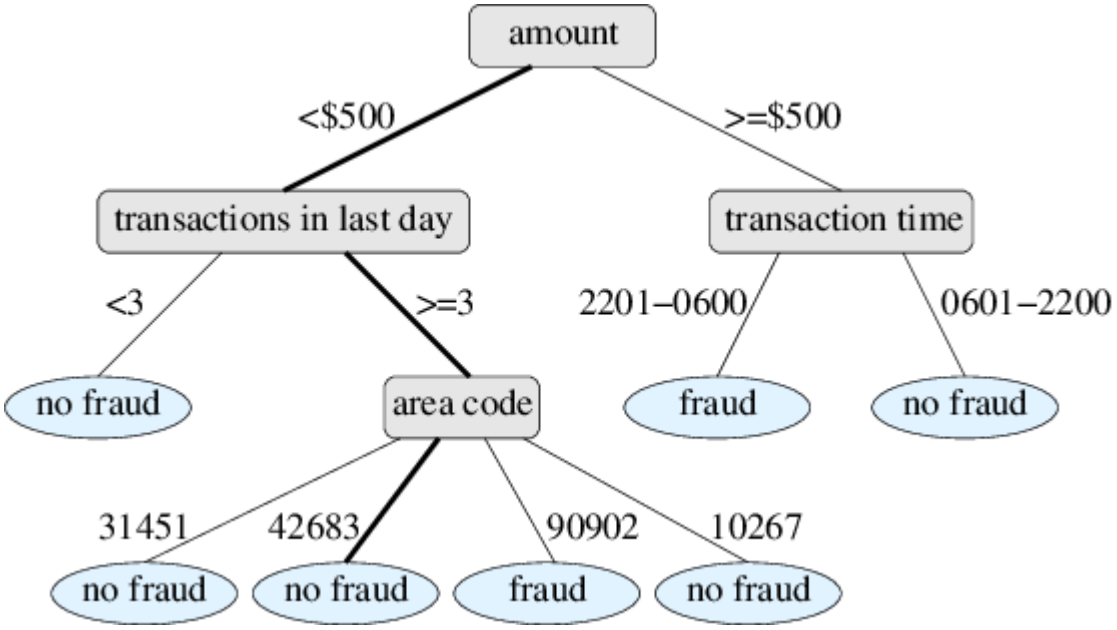


Figure 3.32 Fraud detection classification decision

Source: Semantic Scholar 2020

One of the DT's aims is the development of a training model for predicting the category of the response variable. Predictions are made to classify transactions using this strategy. It is a group of edges connecting several branches and nodes. A tree's edges show the outcome of an evaluation made by the interior nodes. The terminal nodes stand for the label of a certain class. The dataset will be iteratively divided into classes using the Depth-First Breadth technique until each element in the collection has been assigned to a class. This method is beneficial because it does not necessitate feature scaling, it is resistant to outliers, and it can automatically deal with missing information. It reduces training time and is especially effective at tackling classification and regression problems. An important drawback is that, as the size of the dataset grows, the single tree may get more complex and lead to overfitting. The input feature is labeled on a node deeper in the tree, rather than on a leaf node. Arcs extending from an input node represent the several possible values of the target variable, each of which is linked to a subsequent decision node with a different input feature. It uses a variety of methods to determine whether to divide a node into several child nodes. The DT divides the nodes into sub-nodes

depending on all available criteria, then chooses the one that produces the most similar-looking offshoots. Information gain, Measure of goodness, Variance reduction, and Gini impurity are some of the metrics that the DT employs on the prospective subset to achieve the quality of splitting into two or more nodes. The advantages of DTs include their ability to evaluate both categorical and numeric data, their ease of understanding and interpretation, the fact that they require almost nil data preparation, and the fact that they can be used to model extremely large datasets with minimal difficulty. One of the problems with the tree is that it is not robust, such that even small changes to the training data could have a big effect on the accuracy of the predictions it makes.

3.5.3 Naive Bayes

It is a type of probabilistic classifier model, that suggests that it may predict data from several classes at once. Multiple class predictions are made possible by probabilistic classifiers. The choice is selected based on conditional probability. This approach employs several algorithms, but they all share a similar basis, instead of using a single method. It is assumed in this model that each feature contributes to the output in an equal and unique way. This model offers several benefits when compared to other models because it only needs a limited quantity of training data Awoyemi (2017). With the help of training data, the Naive Bayes ML classifier attempts to predict a class known as the result class using probabilities and conditional probabilities of its occurrence. In real-world circumstances, this sort of learning is particularly effective, quick, and accurate, and it is also known as supervised learning Kiran (2018). Naive Bayes classification begins with the Bayes theorem for conditional probability, in which "x" is a given data point and "C" is a class.

$$P(C|x) = \frac{P(x|C)}{P(x)}$$

3.5.4 Multilayer perceptron Classifier

The backpropagation algorithm is used to train the multilayer perceptron (MLP) method. There are typically three layers in an MLP neural network: an input layer, an output layer, and a set of hidden layers. There are connections among each neuron in each layer of the architecture and every neuron in the layer below and above it. Another characteristic of this network is that while every neuron in the hidden and output layers has an activation function, the input layer does not. Although the setting of weights in the MLP is a random process, the network trains by calculating the discrepancy between the computed output and the actual output and then iteratively adjusting the weights to lower the residual.

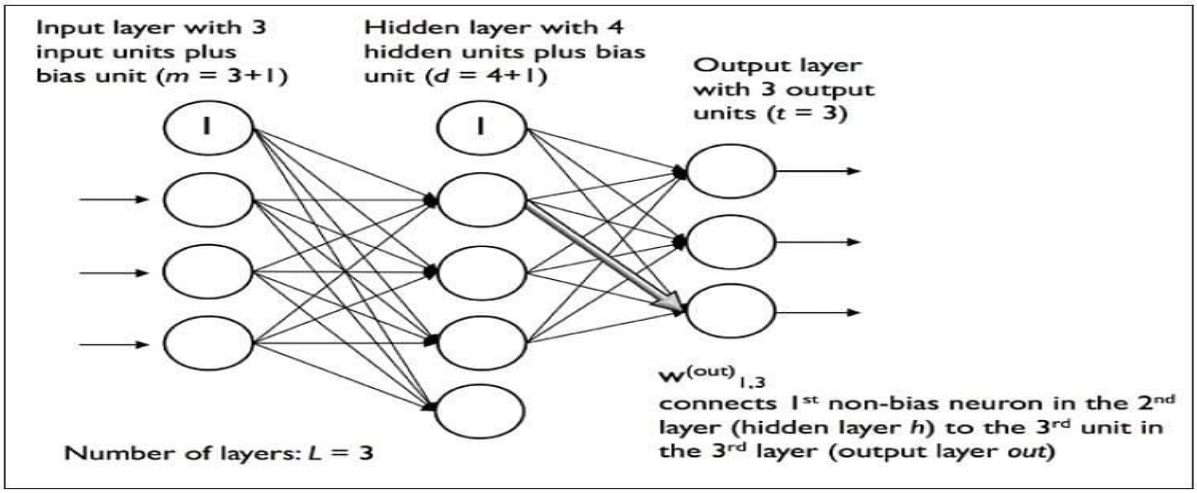


Figure 3.33 MLPC review

Source: simplilearn 2022

3.5.5 Random Forest

RF is a popular ML technique. It is a technique for dealing with classification and regression problems. The "forest" is made up of a staggering variety of DTs. A class is predicted by each distinct tree. The class that receives the most vote receives preference when making a forecast. Therefore, a bagging procedure is employed to generate a set of DTs that will ultimately coalesce into a forest. The speed and accuracy with which the model may be run without the need for feature selection is a major plus for this approach. Due to its being sensitive to data that contains a broad range of values and attributes that contain more than one value, this technique has the drawback of being able to detect fraudulent activity with relative ease. Bagging is a training method that commonly uses the Bootstrap approach for a high variance algorithm that is employed in ML Brownlee (2021). This algorithm builds the DT ensemble, sometimes known as the "forest". Algorithms that aggregate numerous models into a single package include bagging and RFs. In various kinds of predictive modeling issues, both algorithms work quite well. One of the best algorithms for spotting financial system fraud, it uses machine learning to identify suspicious activities. RFs can be used for both classification and regression problems, which is one of the numerous benefits. Finding the best attributes among all attributes for modeling, especially during the splitting of the node, is more important because the Random method always attaches randomness when it starts to create the tree Donges (2021). The RF hyperparameter increased the model's capacity for prediction or accelerated model execution. One of the problems with ML models is the overfitting problem. Even so, a RF classifier is helpful because it can generate many of forest trees and won't overfit the model.

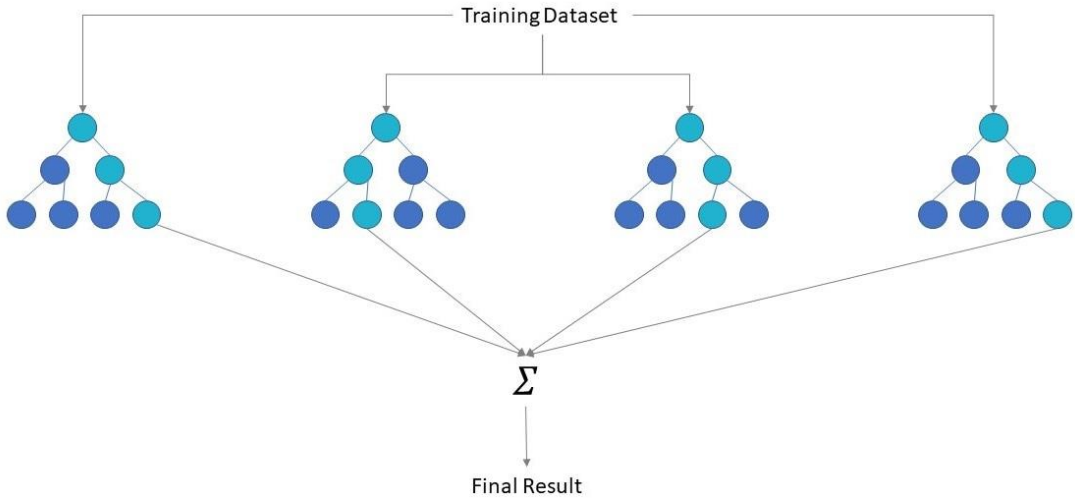


Figure 3.34 MLPC review

Source: IBM cloud education 2020

3.6 Data splitting

We divided the total dataset into a 70% training set and a 30% test set for each experiment. We utilised the test set to evaluate the performance of the model we developed after training it on the training set. We used a random seed when splitting the data to ensure a consistent data split every time the program was run. ScikitLearn Library's Stratified Shuffle Split cross-validator was employed (https://scikitlearn.org). By combining Stratified Fold and Shuffle Split, which return stratified randomised folds constructed by maintaining the percentage of samples for each class, this object is created. The StratifiedShuffleSplit is made for classification jobs in which maintaining class percentage after data splitting is necessary. The following can be written as the algorithm's pseudocode:

```
sk = StratifiedKFold(n_splits=10, random_state=0, shuffle=True)
```

Here, the test size is set to 0.3, and the number of re-shuffling and splitting iterations is set to n splits, int, default=1. The same data will be split due to the random state.

3.7 Hyperparameter Tuning

After completing the development of an ML model, we will be presented with design alternatives for selecting the model architecture. Typically, we are unsure of the ideal model architecture. Therefore, a variety of possibilities will be considered. In an ideal ML method, we would ask the computer to carry out this investigation and choose the best model architecture on its own. Using a set of parameters called hyperparameters, which dictate the model architecture, hyperparameter tuning aims to find the optimal model structure. The hyperparameter answers questions related to model design, like the minimum and maximum depth, the number of trees to be created in the RF, and layers of neuron number required in the neural network layer creation.

Our best model was a hyperparameter tuned in this study, and during grid search, 3-fold cross-validation was done to make sure our model was not overfitting. Throughout the entire process of adjusting the hyperparameters, we used a Python tool called GridsearchCV. Our minimum sample leaves are set between the range of 1 and 4, our maximum depth is set between the range of 10 and 50, and our n estimator is set to 200.

CHAPTER FOUR

4.0 Implementation

In this chapter, we used the ML models discussed in the previous chapter to demonstrate the results of our research. To evaluate the effectiveness of our model, various metrics, including the AUC score, were employed. We will offer metrics for each model based on its performance on our baseline, under-sampling, over-sampling, Smote, and hybrid datasets to determine which of our models is best effective in predicting credit card fraud.

4.1 Metrics

A research project must include an evaluation of the ML algorithms. This will demonstrate how each algorithm performed and allow you to determine whether it produces results that are satisfactory or not. In classification algorithms, accuracy is often used as a performance metric for models. However, it is not the only accurate method. In this study, we used F1-score, recall, precision, accuracy, confusion matrix, and receiver operating characteristic area under the curve (ROC AUC) score as evaluation tools (This is the main metric we have implemented to evaluate our model). This is because it provides the score and a plot showing how each model did, making it the most popular statistic among all metrics.

4.1.1 Accuracy

Accuracy is defined as the proportion of correct predictions to all input samples. It functions perfectly. only when each class contains an equal number of samples. Consider our training set, which has 2% of class B cases and 98% of class A examples. At that time, our model may surely achieve 98% accuracy by just predicting that each training sample will belong to class A. If a comparable model is used on a dataset where Class A accounts for 60% and Class B for 40%, the test accuracy drops to 60%. Although classification accuracy is quite good, it gives the appearance that we have achieved a very high degree of precision.

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

Formula 4.1 Accuracy ***source:*** *https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9*

4.1.2 Recall

The recall is what counts. When the results of the positive number that should have been obtained are divided by the overall sample size, the result should be considered a positive value.

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

Formula 4.2 Recall ***source:*** *https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9*

4.1.3 Precision

precision is determined by dividing the classifier's actual positive results by its predicted positive results.

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

Formula 4.3 Precision ***source***:https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

4.1.4 F1-Score

The accuracy of the test is evaluated using the F1 score. It is between recall and precision and is the mean. It enables a report on the classification's strength and degree of precision. High precision and low recall indicate that our accuracy is relatively high, but keep in mind that it may miss several possibilities that are hard to categorise. To simplify this, it means that the model performed best with a higher F1 score. It is calculated as seen below.

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

Formula 4.4 F1-Score ***source****:* https://towardsdatascience.com/whats-the-deal-with-accuracy-precision-recall-and-f1-f5d8b4db1021

4.1.5 Confusion Matrix

When it comes to the performance of the model in terms of producing matrices, the Confusion Matrix gives us a comprehensive breakdown of the information. It achieves good results, particularly when using a binary classification with samples that fall into two categories: yes or no, true or false.

		Actual Values: Positive (1)	Actual Values: Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Table 4.1 Confusion Metrics **Source**:https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

The four key terms are True Positives, True Negatives, False Positives, and False Negatives.

- True Positives: this occurs when the algorithm successfully predicted YES, and the actual outcome was YES.
- True Negatives: In this situation, the algorithm had predicted NO, but the actual result was NO.
- False Positives: In this case, the actual result was a NO while the algorithm had predicted a YES.
- False Negatives: In this case, the actual result was a YES, the algorithm predicted a NO.

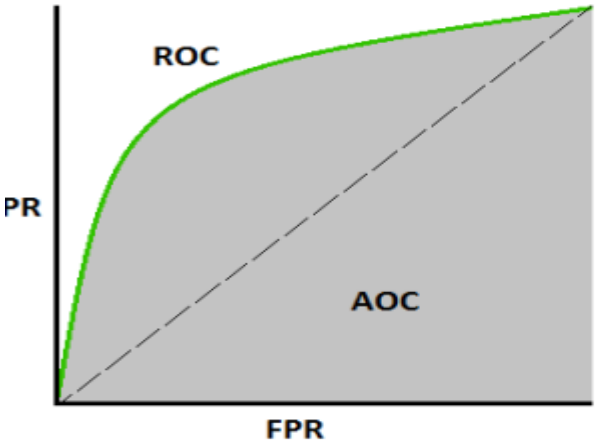
4.1.6 ROC AUC Score

The AUC (Area Under Curve) and ROC Receiver Operating Characteristics) is a common statistic for modeling evaluation. The degree of separability, often known as the area under the curve (AUC), is a measurement of how well a model can differentiate between classes. Concerns regarding classification ought to be evaluated according to the several criteria that are stated. A higher AUC score indicates that the model is more accurate when it predicts that 0 classes will be 0 and 1 class will be 1. The probability of the curve is ROC. The FPR (False Positive Rate) x-axis and TPR (True Positive Rate) y-axis are plotted on this ROC curve.

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

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

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



Formula 4.5 AUC-ROC Curve **source:** https://towardsdatascience.com/understanding-aucroc-curve-68b2303cc9c5

Results from several models used in this research are presented, each based on its own unique set of data (both under and over-sampled). To select the most effective

prediction model, the AUC score was utilised as a comparison statistic among other metrics to measure the performance of each model. The AUC measures how likely it is that a model will give a positive example a higher score than a negative example drawn at random. The better the model can predict both fraudulent and legitimate transactions, the higher the AUC score will be. Since AUC establishes a clear boundary (40) between the positive and negative classes, it can be used as a metric for judging a model's discriminatory power. A summary of the classification tool outputs is provided below.

4.2 Modeling

We will be starting with the original dataset before sampling.

4.2.1 Modeling original Dataset

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - Before Resampling	0.994065	0.994103	0.994103	0.991605	0.113636	0.009242
1	Decision Tree - Before resampling	0.998621	0.998287	0.998287	0.998206	0.908661	0.749838
2	Naive Bayesian - Before resampling	0.899867	0.900862	0.900862	0.943015	0.034955	0.670565
3	Neural Network (MLPC) - Before Resampling	0.519089	0.519724	0.519724	0.678682	0.008865	0.814815
4	Random Forest - Before resampling	0.999979	0.998178	0.998178	0.998090	0.896688	0.738791

Figure 4.1 Accuracy score on the original dataset

The accuracy, F1 score, and precision of each algorithm's performance were displayed in the above figure 4.1 and figure 3.9. As seen, all tree-based algorithms have the highest F1 Scores, which is approximately 0.99%, while MLPC has the lowest performance scores when using the original dataset for modeling, which is 0.67%. A model must have a high F1 Score to be regarded as the top performing model. We cannot trust these results because we are modeling from the original dataset without any pre-processing. There may be some bias in our results because of the high imbalance of the dataset and the possibility of some classifiers not performing well with it. The classification of our original dataset cannot be relied on, and as a result, we have used re-sampling techniques (under-sampling, over-sampling, smote, SMOTETomek, and SMOTEENN) to work on the skewness of the dataset and further improve its performance. The outcomes are displayed below.

4.2.2 Model result for under-sampling dataset

Since most of the records in the dataset belong to the majority class, the number of cases belonging to the majority class must be arbitrarily decreased to achieve an adequate sample size for the dataset. As a direct consequence of this, the dataset lacks important key data instances that are necessary for training.

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - Random Under Sampling	0.807596	0.817995	0.865573	0.922796	0.031539	0.781885
1	Decision Tree - Random Under Sampling	0.981376	0.961683	0.962904	0.976806	0.127603	0.974122
2	Naive Bayesian - Random Under Sampling	0.683531	0.663360	0.543665	0.698819	0.010011	0.829945
3	Neural Network (MLPC) - Random Under Sampling	0.983869	0.944235	0.958268	0.974271	0.113666	0.959335
4	Random Forest - Random Under Sampling	1.000000	0.965104	0.972750	0.982217	0.166456	0.975970

Figure 4.2 Accuracy score on the Under-Sampling dataset

Reviewing the result in table 4.2 and fig. 4.2, it can be shown that the tree algorithms still outperform them with an F1 Score of roughly 0.98% when modeling using the under-sampling data. RF and DTs perform similarly to a model using the original dataset. The maximum depth of a tree is determined by its learning rates, which were set at 0.1 and 5 respectively. The algorithms would select randomly from half of the training data with a subsampling of 0.5, preventing overfitting.

The tree algorithms have a high precision rate and recall, as shown by a comparison of the models' F1 scores with other metrics.

Naive Bayes is also a model for supervised classification and has performed poorly using the under-sampling method. Since the subject at hand is a classification problem and the dataset already includes preset classes to which items are allocated, the Naive Bayes model's F1 score of 0.68% indicates that it might not be the ideal algorithm to utilise.

4.2.3 Model result for over-sampling dataset

This method involves reproducing recent cases from the minority class, as well as occasionally simulating such examples. It increases the number of instances, which improves the accuracy of the model.

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - Random Over Sampling	0.802472	0.815738	0.860568	0.919903	0.030567	0.785582
1	Decision Tree - Random Over-sampling	0.981986	0.960899	0.948607	0.969044	0.095695	0.977819
2	Naive Bayesian - Random Over Sampling	0.690616	0.684262	0.602843	0.746780	0.011189	0.807763
3	Neural Network (MLPC) - Random Over Sampling	0.988479	0.943793	0.942258	0.965588	0.084707	0.959335
4	Random Forest - Random Over Sampling	1.000000	0.969697	0.968811	0.980037	0.148086	0.972274

Figure 4.3 Accuracy score on the Over Sampling dataset

We can see how the tree-based algorithms beat other methods, especially the RF, which has a precision of 0.14, and how the use of over-sampling has improved the performance of all the algorithms are performing compared to the under-sampling dataset in table 4.2. F1 score for the DT and RF is 0.98% and 0.96 respectively. We also observed that the Scores of other algorithms had improved, indicating that most algorithms perform better with oversampled datasets than with under-sampled ones. The RF algorithm is still the best among the tree algorithms when the AUC metric is compared against other metrics. We can rely on the accuracy prediction findings for credit card fraud because it has high precision and low recall.

4.2.4 Model result for SMOTE dataset

Below is the performance review on the SMOTE dataset.

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - SMOTE	0.803053	0.817167	0.865932	0.923003	0.031551	0.780037
1	Decision Tree - SMOTE	0.984735	0.953943	0.954381	0.972169	0.105752	0.968577
2	Naive Bayesian - SMOTE	0.692607	0.678297	0.588095	0.735152	0.010961	0.820702
3	Neural Network (MLPC) - SMOTE	0.988626	0.934403	0.944515	0.966804	0.086883	0.946396
4	Random Forest - SMOTE	1.000000	0.959525	0.968319	0.979761	0.145520	0.966728

Figure 4.4 Accuracy score on the SMOTE dataset

Using the SMOTE dataset, RF has shown the most effective performance result with an accuracy of 0.98, F1 of 0.97, and precision of 0.14. The tree algorithms

outperformed the other models. Second to this performance is the DT with an F1 of 0.97 and precision of 0.10. Reviewing the model with the least performance, as seen, it is Naïve Bayesian. Caution is required when using this method to avoid many noisy data points in feature space.

4.2.5 Model result for SMOTETomek dataset

Below is the performance review on the SMOTETomek dataset.

Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
LR - SMOTETomek	0.836641	0.826695	0.885932	0.934416	0.036612	0.772643
DT - SMOTETomek	0.988925	0.964478	0.955571	0.972797	0.106334	0.946396
Naive Bayesian - SMOTETomek	0.702815	0.710441	0.614340	0.755700	0.011339	0.794824
Neural Network (MLPC) - SMOTETomek	0.999077	0.965554	0.937561	0.963026	0.077532	0.940850
RF - SMOTETomek	1.000000	0.972013	0.966575	0.978796	0.137986	0.957486

Figure 4.5 Accuracy score on the SMOTETomek dataset

In Figure 4.4 above, the tree algorithms still outperformed other models with Random Tree having an accuracy of 0.96, F1 of 0.97, and a precision of 0.13. second to this is the DT with an accuracy of 0.95, F1 score of 0.97, and precision of 0.16. The Model with the least performance is Naïve Bayesian and this has been the same for previous sampling methods.

4.2.5 Model result for SMOTEENN dataset

The five models were also tested using SMOTEENN, and below is the output.

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - SMOTEENN	0.951841	0.938326	0.865625	0.922826	0.031900	0.791128
1	Decision Tree - SMOTEENN	1.000000	0.966960	0.957796	0.973942	0.104296	0.870610
2	Naive Bayesian - SMOTEENN	0.745042	0.737885	0.519379	0.678053	0.009571	0.835490
3	Neural Network (MLPC) - SMOTEENN	1.000000	0.993392	0.879799	0.930942	0.035717	0.794824
4	Random Forest - SMOTEENN	0.999056	0.977974	0.962863	0.976772	0.126214	0.826248

Figure 4.6 Accuracy score on the SMOTEENN dataset

The above output also showed that the tree-based models (RF and DT) performed well compared to other models. LR (with Accuracy of 0.92, precision of 0.03, and F1 score of 0.92) and MLPC (with Accuracy of 0.87, precision of 0.03, and F1 score of 0.93) performance are still fair when compared to Naïve Bayesian. Overall, the tree-based models are great.

4.3 Comparison of ROC-AUC Curve

4.3.1 Logistic Regression

In taking this research work further, ROC and AUC curve comparison was done using the different models covered in the study.

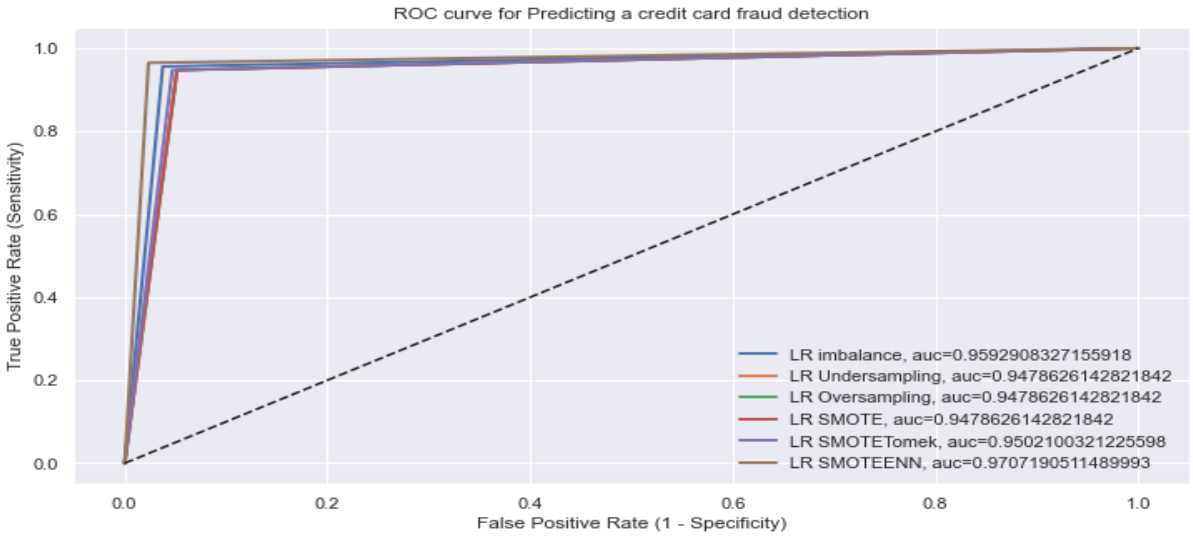


Figure 4.7 ROC curve for LR – sampling method

The above figure shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). The closer it gets to the 45degree diagonal line, the less accurate the test is, and the closer the AUC moves to the top left (or closer to 1), the better. Analysing the figure above, SMOTEENN happens to give the best ROC of 0.97 which is followed by SMOTETomek.

4.3.2 Decision Tree

The below speaks about the sampling methods using DT.

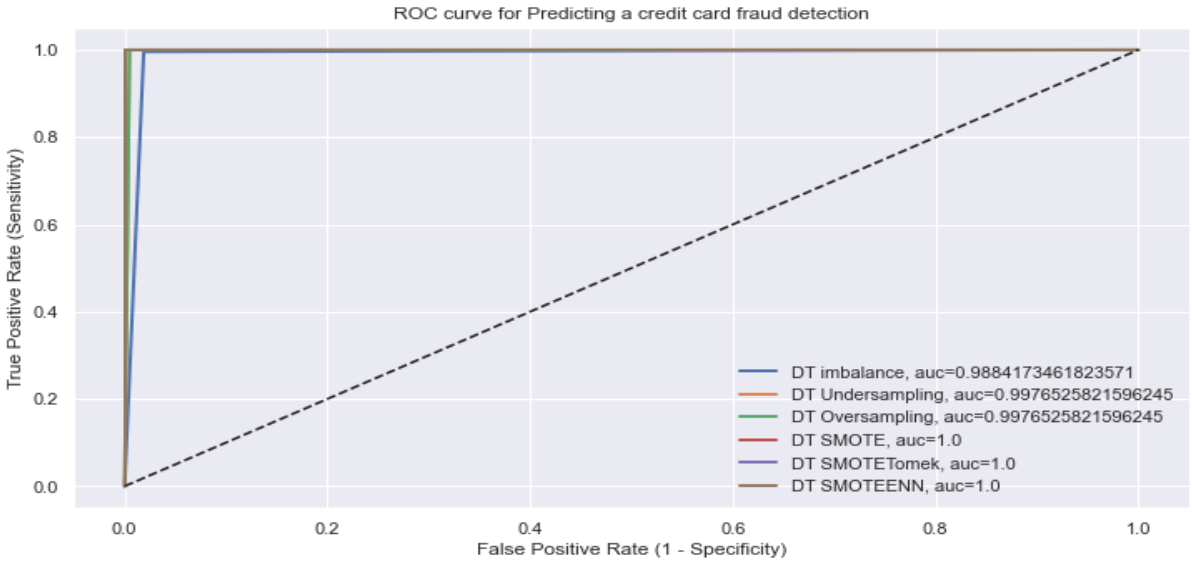


Figure 4.8 ROC curve for DT – sampling method

Analysing the sensitivity and specificity of the above figure, the smote, SMOTETomek and SMOTEENN are presenting an AUC of 1. This is showing how much of the plot is located under the curve. The other two sampling methods are just a little bit away from 1.

4.3.3 Naïve Bayes

ROC comparison was done for Naïve Bayes, using the various sampling methods.

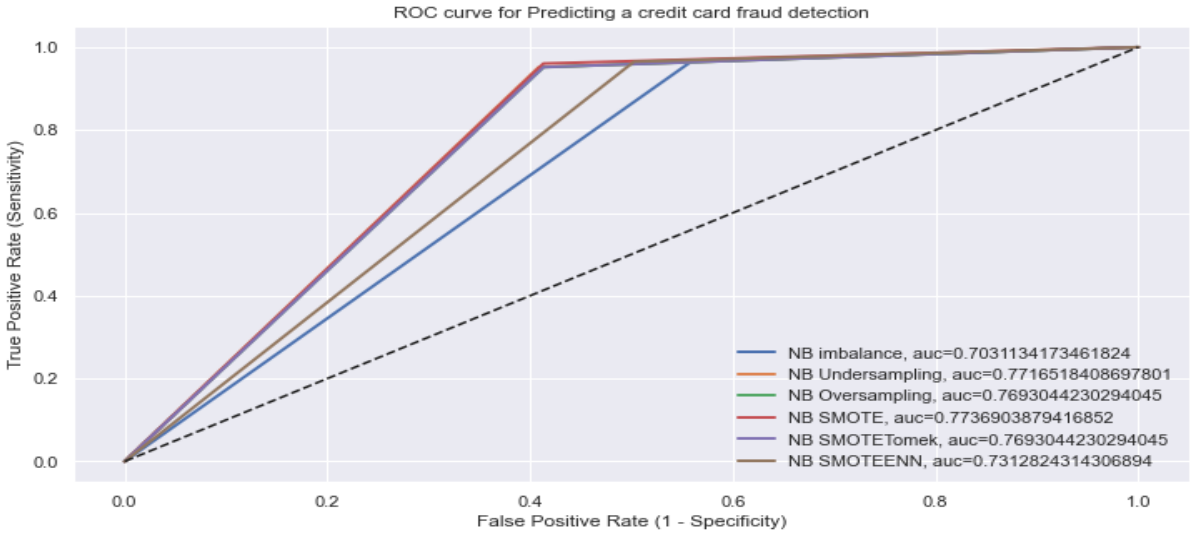


Figure 4.9 ROC curve for Naïve Bayes – sampling method

As seen, the best AUC happens to be SMOTE which is 0.77 with SMOTEENN being the least of 0.73. Generally, Naïve bayes has not been performing well in delivering the classification task in the study. It is not advisable to use this model in the detection of credit card fraud.

4.3.3 Multilayer perceptron (MLPC)

ROC comparison was done for MLPC, using the various sampling methods.

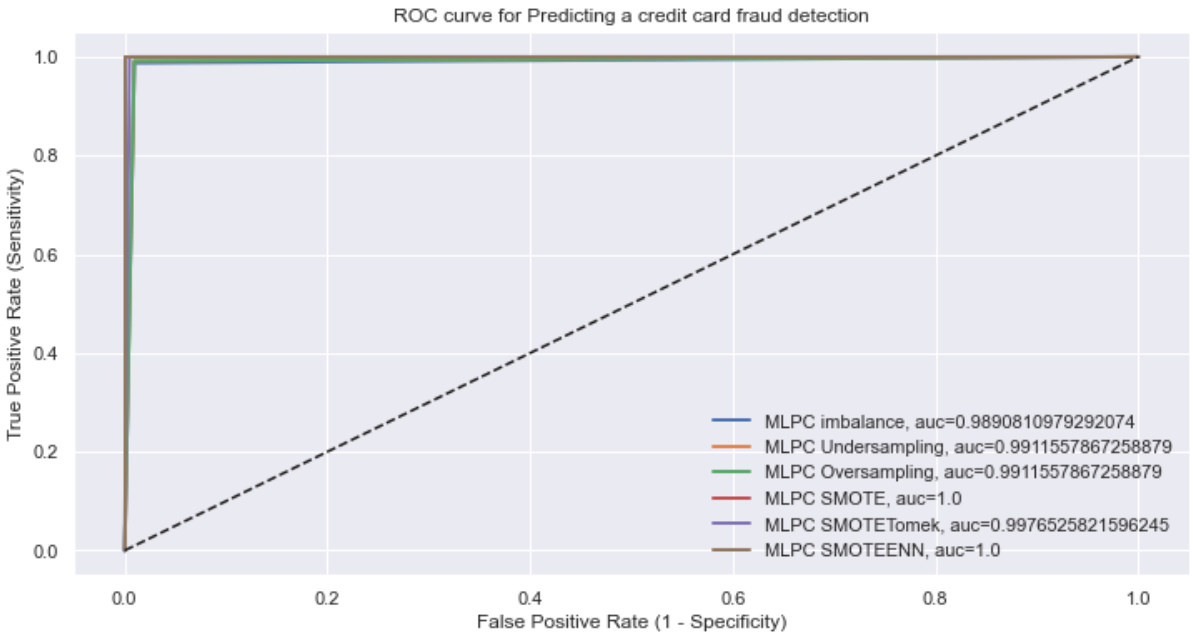


Figure 4.10 ROC curve for MLPC – sampling method

Analysing the sensitivity and specificity of the above figure, the smote and SMOTEENN are presenting an AUC of 1, with a near 1 for the SMOTETomek method. This is showing how much of the plot is located under the curve. The other two sampling methods are just a little bit away from 1.

4.3.3 Random Forest

The below speaks to the ROC comparison between the sampling methods using RF.

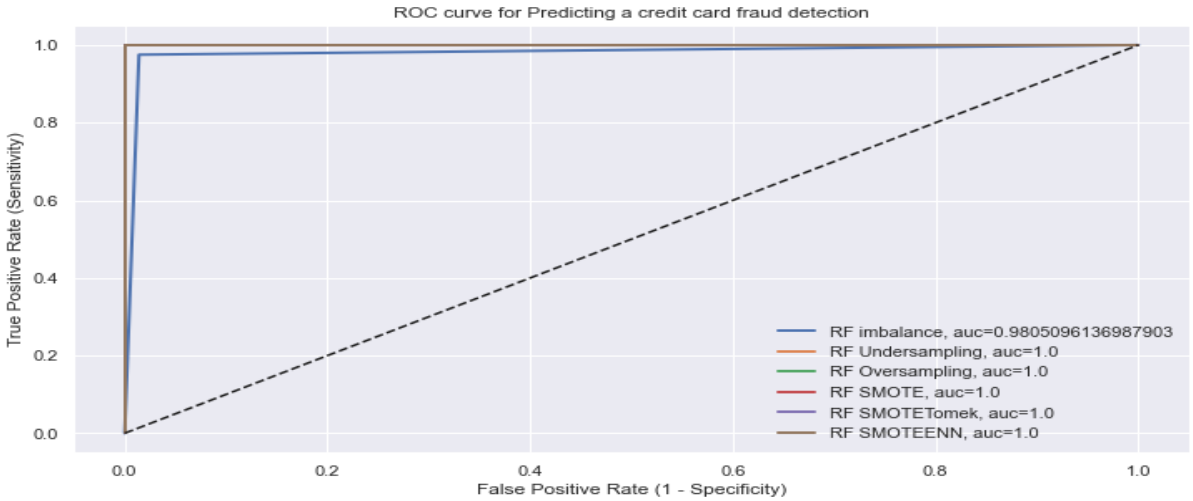


Figure 4.11 *ROC curve for RF – sampling method*

From this figure, RF has given the best AUC when compared to other models. As seen, it has presented an AUC of 1 for the five sampling methods. This has further confirmed that RF is the most effective model to achieve the goal of this research. This also applies to the F1 and precision comparison.

4.4 Hyperparameter tuning with the top models

To further evaluate the effectiveness of our best algorithms, we carried out a hyperparameter tuning on a subset of them. Given that hyperparameters are the core to ML classifiers, they help choose the best values for the algorithm's learning parameters.

	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Random Forest - SMOTEENN [Hyperparameter Tuned]	1.000000	0.993197	0.978339	0.985175	0.182626	0.835490
1	Decision Tree - SMOTEENN [Hyperparameter Tuned]	0.993191	0.945578	0.923756	0.955441	0.056720	0.815157
2	Logistic Regression - SMOTEENN [Hyperparameter...	0.969844	0.963719	0.917449	0.951969	0.050419	0.778189
3	MLP - SMOTEENN [Hyperparameter Tuned]	1.000000	0.990930	0.911839	0.948883	0.047624	0.783734

Figure 4.12 Accuracy score on the hyperparameter tuned models

In Figure 4.12, we can see that despite our best efforts to fine-tune the hyperparameters, the tree-based approaches RF and DT, with F1 Scores of 0.98% and 0.95%, respectively, continue to perform as our most promising solution. Across all analyses, RF coupled with SMOTEENN has consistently performed as the top model.

4.5 Comparative analysis

In this section, we evaluate how well our model fared in comparison to others, both in terms of the datasets used and the results of the metrics used to evaluate the performance of each method. By comparing the AUC score, accuracy, precision, recall, and F1-score of several models, we discovered that the ensemble tree model did very well with the original dataset, the under-sampling dataset, and the over-sampling dataset.

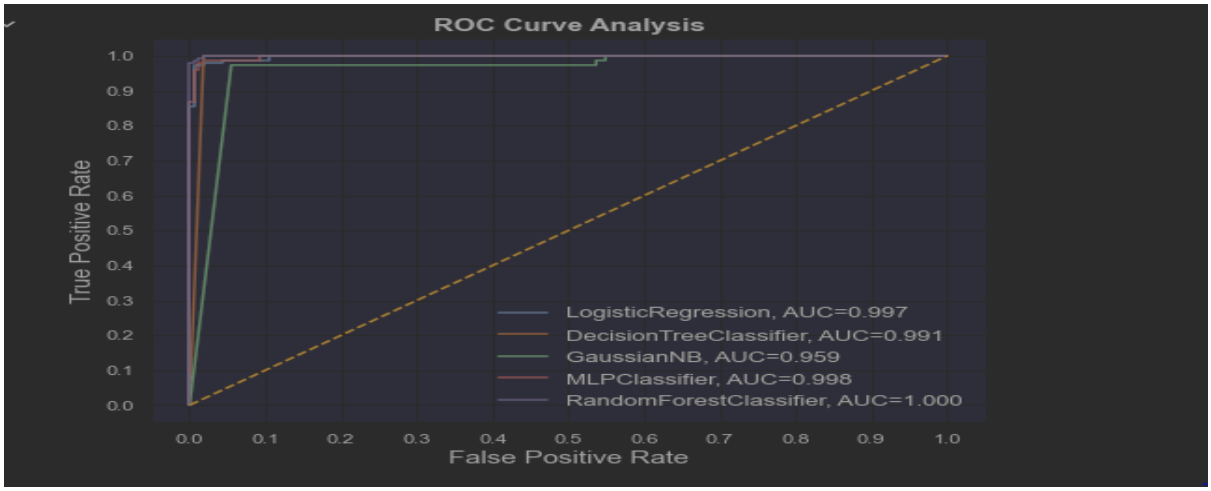


Figure 4.13 AUC on the hyperparameter tuned models

The results of each ensemble tree algorithm are compared in Figure 4.1**3**; the AUC score, precision and accuracy of RF are the highest. As a result, the RF algorithm

has been selected as the best model for estimating the likelihood of credit card fraud.

4.6 Interpretability

An important factor in establishing confidence in the models and promoting their widespread use in credit risk is the interpretability of prediction algorithms. To make ML more understandable, we have seen several techniques like feature importance and partial dependence graphs. By outlining potential explanations and highlighting significant characteristics and their influence, they shed some light on the black box and are applicable to all the learners profiled in the current paper. We identify three main approaches, which are intrinsically interpretable models and post hoc specific or agnostic methods.

Transparency and a self-explanatory structure are two characteristics of intrinsically interpretable models, which include combinations of business decision rules, DTs, LR, and linear regression. Zhuang (2020) notes that while these methods are useful for use cases subject to legal or policy constraints, they may not be precise enough for tasks like fraud detection, which have a high financial stake. This explains why it is preferable to adopt a more exact black box model if a post hoc interpretability approach is used to provide explanations for their operation or results. Post hoc details are a type of method that can only be used with certain models. The classification aggregation tablet scan (CAT scan) and the feature importance permutation metric are two examples of such techniques for sets of DTs (Breiman, 2001). For neural networks, known for their higher performance on issues but also for the complexity of its interpretation, several specialised algorithms have been developed, such as layer-wise relevance propagation or deep learning essential features Shrikumar (2017).

The fundamental drawback of the latter is that they can only be used with one kind of model, making it difficult to evaluate the performances and justifications of other models. It is possible to employ post-hoc model-agnostic strategies to overcome this drawback. They can be coarser (or local) to analyse a specific case or observation, or macro (or global) to acquire a broad picture of the model to grasp it in its whole.

Visual interpretation strategies have been put into practise because visualisation is one of the most important techniques for understanding models. The most important ones are individual conditional expectation curves (ICE) Goldstein (2014) for local approaches. These techniques show how a model's variable has an impact. Knowing which variables have the most effects on predictions and are therefore crucial for the model can help you choose the variables to plot more effectively. For this, Fisher (2019) developed the model dependence measure, which is an agnostic variant of the permutation feature importance measure (Breiman, 2001).

One more time-honored method of interpretation is to follow a black box model with a more transparent one. Surrogate models are those, sometimes known as metamodels or approximation models. KernelSHAP, SHAP features importance, SHAP summary plot, and SHAP dependence plots are the most well-known local surrogate models. Tsang (2018) proposes model-agnostic hierarchical explanations (MAHE),

Ribeiro (2016) proposes locally interpretable model-agnostic explanations (LIME), Lundberg and Lee (2017) propose Shapley additive explanations (SHAP), and Lundberg and Lee (2017) and (Lundberg et al., 2019) propose the Shapley model. In addition, there are methods that have been shown to be useful that are based on "counterfactual examples." are easily digestible, as per Wachter (2018), and aid the user directly in making decisions. Bottou (2013).

Additionally, it is insufficient to see the impacts of each variable separately if the variables in a model interact. The effects of interactions between variables can be analysed and interpreted using techniques like variable interaction networks (VIN) Hooker (2004) or interaction strength Friedman and Popescu, (2008).

Others prefer to use advanced black box models and then utilise post hoc interpretability techniques, while others, like Rudin (2019), claim that it is necessary to construct interpretable models to boost accuracy in the case of fraud detection.

4.7 Accuracy under constraint

Financial and operational limitations that affect fraud detection make accuracy and interpretability of the model results essential. The trade-off between precision and interpretability, when both terms are viewed as "contradictory," is a major problem in the field of ML, especially for the most current methods like deep neural networks. The conflict between accuracy and interpretability may be seen in a variety of application sectors Yang and Bang (2019), and the most representative models' performance is evaluated in comparison tests Sahin (2020), but the validity of these tests is still up for question.

4.8 Research question addressed

In this session, we took to ensure that the research questions have been addressed. Below are the details.

Research Questions	Session Addressed
What are the ML algorithms that have proven effective in the detection of credit card fraud	Chapter two (Related Works), different scenarios addrssed
Which of the sampling method (over-sampling, under-sampling, and hybrid) will best solve the problem of the highly unbalanced dataset	This was addressed in Chapter four, the hybrid methods improved the efficiency of the model's accuracy
Are the accuracy and interpretability trade-off of the different data mining techniques measurable	This was addressed in Chapter four
What are the limitations and foreseeable future concerns of the current CCFD system	This was addressed in Chapter five
Using several statistical methods, what is the most effective ML model in the prediction of credit card fraud	This was addressed in Chapter five. The Random forest model was seen to be most effective

Table 4.2 Research question addressed

CHAPTER FIVE

5.0 Result

Several improvements were influenced by the change in technology. This study aims to enhance ML algorithms for fraud detection and interpreting the trade-off between the model as we discuss online credit card transactions that result in credit card fraud. In this paper, we provide supervised learning-based techniques for detecting fraud, including RF, DT, LR, Naive Bayes, and the multilayer perceptron deep learning algorithm. Since our dataset is highly unbalanced, we used resampling approaches such under-sampling, oversampling, SMOTE, and the hybrid method before comparing all the algorithms with various datasets. Finally, we concluded that our model would work best using RF. It can be concluded that SMOTEENN is more effective because our model is trained more effectively with less observations. In the real-world situation, SMOTEENN will be the best sampling approach because the data containing a pattern is preserved.

5.1 Broder impact

For a variety of reasons, interpretability in AI systems may be required or desirable. Below are few of them in relation to Neural Additive Models and classification models:

Safeguarding against bias: NAMs and classification models may quickly adjust for bias to produce potentially more fair models by determining whether training data is used in ways that lead to bias or discriminatory outcomes.

Improving AI system design: NAMs give developers the ability to investigate why a system performed a specific way (for instance, when a tracking system fails) and create improvements. NAMs can discover issues that could put some users at danger and that need to be fixed before the system is deployed, as well as explain results that appear to be atypical in the detection of credit card fraud.

Adhering to policy requirements: Interpretability of NAMs and classification models can be crucial in upholding legal rights related to a system, such as the well-established "right to explanation" for credit ratings in the United States. NAMs can also give people the ability to challenge model outputs, such as contesting a genuine transaction marked as being fraudulent, based on the interpretations that NAMs provide for a particular choice.

Assessing vulnerability, risk and robustness: This can be especially crucial if an AI system is implemented in an unfamiliar setting where we cannot be certain of its efficacy. For instance, NAMs for fraud detection can be studied to understand the risks associated with them or how they might fail before being implemented for customers who are not yet known.

5.2 Limitation

Once ML gets access to a vast amount of data to learn from, accuracy improves. Millions or even billions of data points in fraud detection allow the machines to develop a thorough understanding of how to discern between illegal and legal conduct. Machines must encounter as many examples of crime as they can to become

effective due to the dynamic nature of crime. In contrast to credit card fraud, where we have access to many daily transactions, the frequency of credit applications is significantly less frequent. The foundation for creating efficient anti-fraud systems is ongoing data collection.

Despite the useful business insights, the restricted number of labels that are readily available prevent a thorough assessment of our job. We have only included some segments in the results that had enough fraud cases; nevertheless, if we had been able to show the improvements of our method in more segments, the outcome would have been more credible. The top anomalies for the full dataset might potentially be presented as an option, but there would still be no objective standard to use to judge how well the model performed.

5.3 Conclusion

Humans are used in many aspects of banking operations, including fraud detection. Like a dermatologist's role, in which identifying a tumour is only one part of the job, the detection of a possible fraud is only the first step in the much longer process of fraud management. Because of the need for human inspection and corrective action against the fraudster, it is challenging to evaluate ML algorithms on their efficiency or explanatory power outside of their use by the practitioner. While rule-based methods are still often utilised in the banking industry, improved outcomes in fraud detection can be achieved by employing data-driven rule learning strategies.

When an explanation of the decision made by the black box model is not required for an investigation or legal action against the fraudster, the black box model can be utilised to optimise some procedures in fraud management. While it is yet unclear whether technique will provide the best solution, we demonstrated on a real-world example that it is possible to combine the best aspects of both approaches in the interim to create better solutions for the entire fraud management process.

5.4 Future work

For future research, the hybrid sampling technique utilised in this study and the interpretable methods will be expanded to include hybrid machine learning model for additional datasets in CCFD. Future research may concentrate on a variety of topics, beginning by recommending data preparation strategies to address the problem of missing values. The impact of various feature selection and extraction techniques on prediction precision and interpretability should also be studied in the context of credit cards. Future research should focus on the interpretability of the best suitable model among cutting-edge and hybrid ML algorithms and a hybrid sampling technique to ascertain the most accurate model.

In recent years, the idea of interpretable ML has gained popularity. Although sophisticated, nonlinear function mapping is a useful tool for problem resolution, it is challenging to understand. To persuade domain experts to trust on these algorithms' detections of fraud, it is essential to be able to pinpoint the factors that influence outcomes. In this study, we discussed using SHAP and some other explanations in interpreting the accuracy of model applications. The business professionals supported the use of such procedures in the financial industry because

they make it easier to understand the suggested models and give them the ability to assess generated anomalies more quickly.

Appendices

Appendix 1. Data Preprocessing

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 975036 entries, 0 to 975035
Data columns (total 22 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   trans_date_trans_time  975036 non-null  object
 1   credit_card_number     975036 non-null  int64
 2   merchant               975036 non-null  object
 3   category               975036 non-null  object
 4   amount                 975036 non-null  float64
 5   first_name             975036 non-null  object
 6   last_name              975036 non-null  object
 7   gender                 975036 non-null  object
 8   street                 975036 non-null  object
 9   city                   975036 non-null  object
 10  state                  975036 non-null  object
 11  zip_code               975036 non-null  int64
 12  latitude               975036 non-null  float64
 13  longitude              975036 non-null  float64
 14  city_population        975036 non-null  int64
 15  job                    975036 non-null  object
 16  day_of_birth           975036 non-null  object
 17  trans_number           975036 non-null  object
 18  unix_time              975036 non-null  int64
 19  merchant_lat           975036 non-null  float64
 20  merchant_long          975036 non-null  float64
 21  fraud                  975036 non-null  int64
dtypes: float64(5), int64(5), object(12)
memory usage: 163.7+ MB
```

Dataset Information

```
Index(['trans_date_trans_time', 'credit_card_number', 'merchant', 'category',
       'amount', 'first_name', 'last_name', 'gender', 'street', 'city',
       'state', 'zip_code', 'latitude', 'longitude', 'city_population', 'job',
       'day_of_birth', 'trans_number', 'unix_time', 'merchant_lat',
       'merchant_long', 'fraud'],
      dtype='object')
```

There are 22 columns in the dataset

Number of Columns in the dataset

	credit_card_number	amount	zip_code	latitude	longitude	city_population	unix_time	merchant_lat	merchant_long	fraud	year	hour	age
count	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000	975036.000000
mean	417197785652788416.000000	70.213255	48815.857405	38.534111	-90.231395	89042.679016	1345505091.924983	38.534049	-90.231559	0.005551	2021.105341	12.805828	46.756942
std	1308903587129116160.000000	160.831716	26893.952150	5.076803	13.754579	302594.108465	9320993.106374	5.111292	13.756833	0.074295	0.306992	6.817871	17.362207
min	60416207185.000000	1.000000	1257.000000	20.027100	-166.672300	23.000000	1325054544.000000	19.027785	-166.671242	0.000000	2021.000000	0.000000	17.000000
25%	180000000000000.000000	9.640000	26237.000000	34.620500	-96.798000	743.000000	1337961209.250000	34.730153	-96.900836	0.000000	2021.000000	7.000000	35.000000
50%	3520000000000000.000000	47.420000	48174.000000	39.354300	-87.476900	2456.000000	1345380799.000000	39.364304	-87.446250	0.000000	2021.000000	14.000000	47.000000
75%	4640000000000000.000000	83.010000	72042.000000	41.940400	-80.158000	20328.000000	1354320131.750000	41.958414	-80.239186	0.000000	2021.000000	19.000000	60.000000
max	4999000000000000000.000000	28948.900000	99783.000000	66.693300	-67.950300	2906700.000000	1362182379.000000	67.510267	-66.950902	1.000000	2022.000000	23.000000	98.000000

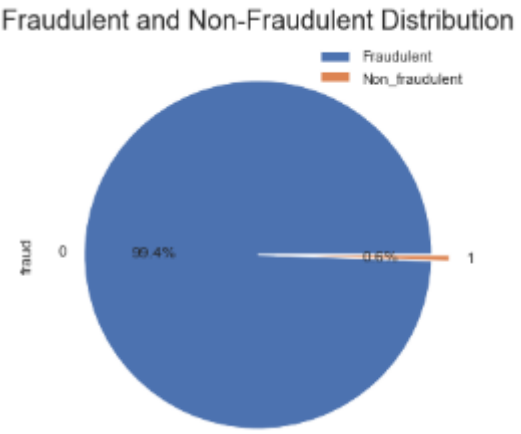
Descriptive Statistics

	trans_date_trans_time	credit_card_number	merchant	category	amount	first_name	last_name	gender	street	city	state	zip_code	latitude	longitu
0	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
1	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
2	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
3	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
4	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
975031	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
975032	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
975033	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
975034	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa
975035	False	False	False	False	False	False	False	False	False	False	False	False	False	Fa

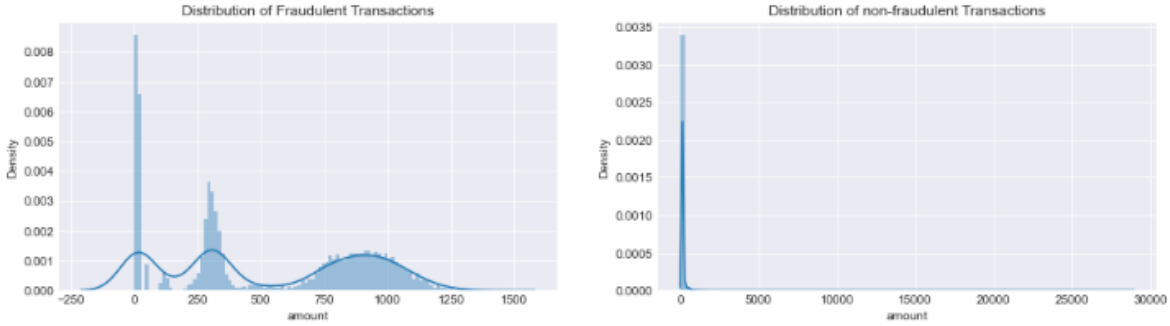
975036 rows × 22 columns

Null value in the dataset

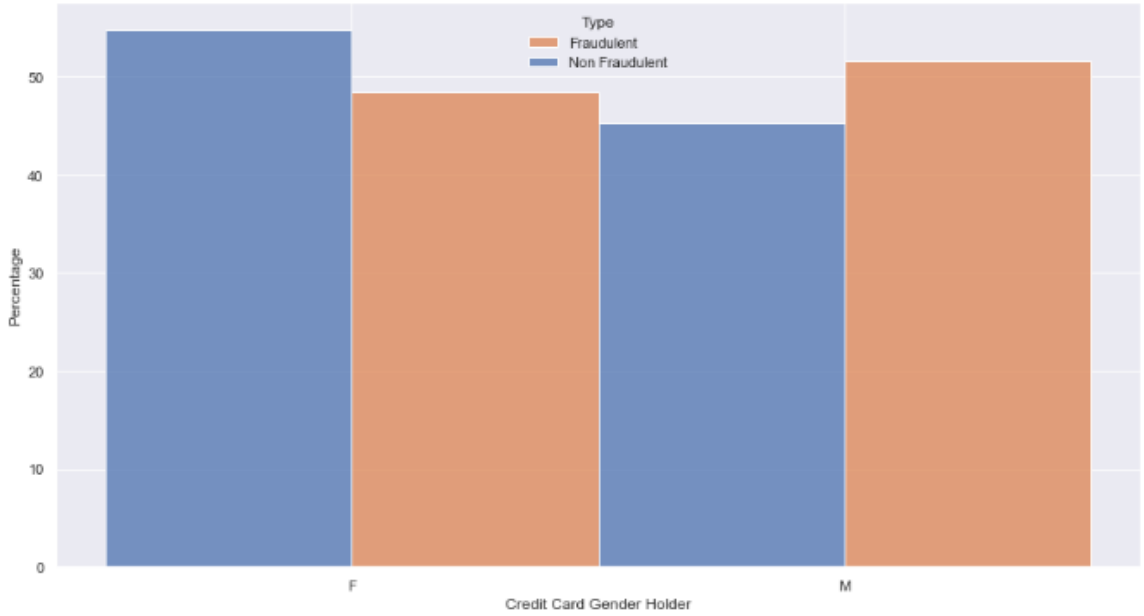
Appendix 2 Exploratory Data Analysis



Fraudulent and Non-Fraudulent Distribution



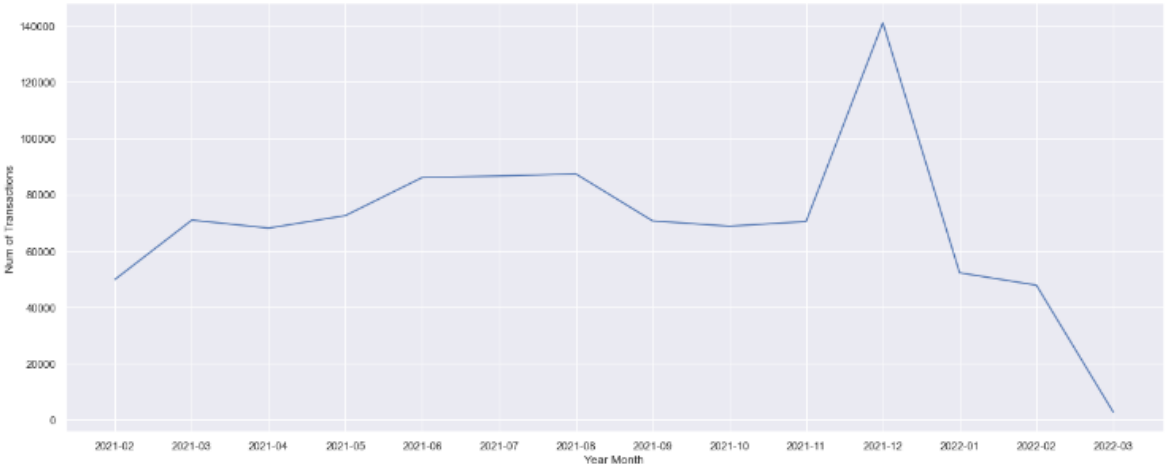
Distribution of Amount with respect to the target variable



Gender versus target variable.

	year_month	num_of_transactions	customers
0	2021-02	49866	314
1	2021-03	70939	315
2	2021-04	68078	315
3	2021-05	72532	315
4	2021-06	86064	314
5	2021-07	86596	314
6	2021-08	87359	315
7	2021-09	70652	314
8	2021-10	68758	314
9	2021-11	70421	314
10	2021-12	141060	316
11	2022-01	52202	314
12	2022-02	47791	314
13	2022-03	2718	290

Frequency of transaction per month of the year



The highest transaction took place in December 2021 with a value of 141,060 with 316 customers.

Frequency of transaction per month of the year

	Category	category_count	percent
0	entertainment	70741	7.255219
1	food_dining	68884	7.064765
2	gas_transport	98954	10.148753
3	grocery_net	34188	3.506332
4	grocery_pos	92843	9.522007
5	health_fitness	64422	6.607141
6	home	92579	9.494931
7	kids_pets	85047	8.722447
8	misc_net	47347	4.855923
9	misc_pos	60047	6.158439
10	personal_care	68354	7.010408
11	shopping_net	73261	7.513671
12	shopping_pos	87804	9.005206
13	travel	30565	3.134756

Transaction frequency with respect to the category variable.

	city	fraud	Transaction count	city_count	Transaction percentage
552	Hubbell	1	19	19	100.000000
504	Oakton	1	9	9	100.000000
1125	Seattle	1	10	10	100.000000
57	Ashland	1	10	10	100.000000
608	Kams City	1	7	7	100.000000
867	Norfolk	1	7	7	100.000000
800	Morven	1	8	8	100.000000
1064	Roland	1	11	11	100.000000
604	Kaktovik	1	12	12	100.000000
1044	Ridge Spring	1	10	10	100.000000
752	Melville	1	15	15	100.000000
876	North East	1	9	9	100.000000
637	La Grande	1	12	12	100.000000
85	Beacon	1	11	11	100.000000
982	Pleasant Hill	1	8	8	100.000000
966	Phelps	1	11	11	100.000000
575	Irvington	1	8	8	100.000000
923	Orange Park	1	10	10	100.000000
578	Isanti	1	10	10	100.000000
211	Chattanooga	1	7	7	100.000000

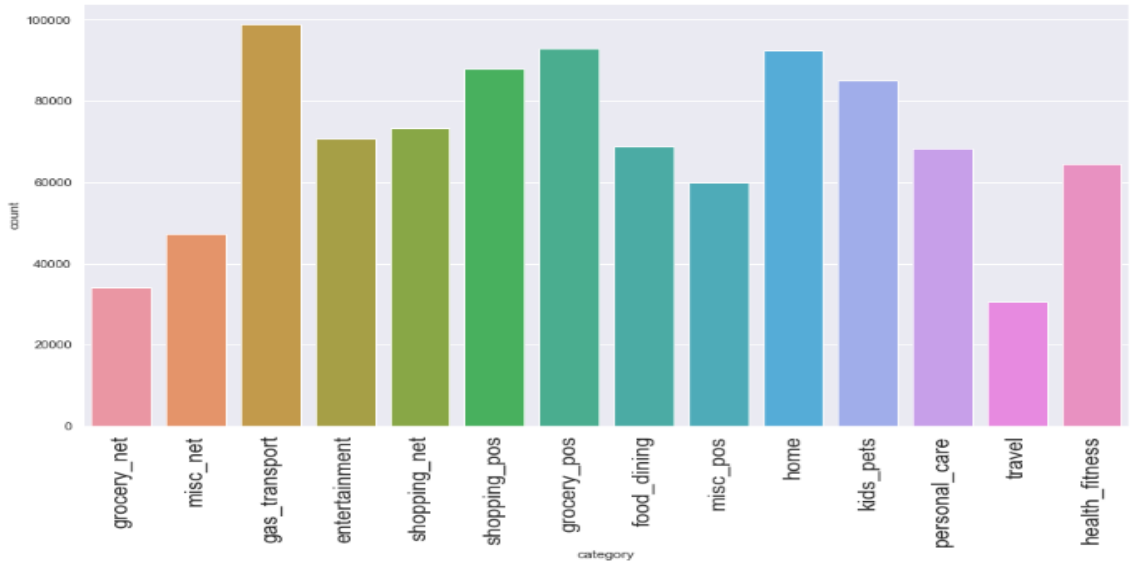
City transaction count distribution

	zip_code	fraud	Transaction count	zip_count	Transaction percentage
1443	99747	1	12	12	100.000000
1069	70447	1	11	11	100.000000
183	14532	1	11	11	100.000000
1058	72135	1	11	11	100.000000
387	28412	1	9	9	100.000000
1151	75246	1	11	11	100.000000
1162	76048	1	12	12	100.000000
404	29129	1	10	10	100.000000
647	43723	1	12	12	100.000000
1152	77038	1	15	15	100.000000
140	12508	1	11	11	100.000000
615	41101	1	10	10	100.000000
131	12207	1	11	11	100.000000
115	11944	1	10	10	100.000000
114	11763	1	9	9	100.000000
113	11747	1	15	15	100.000000
1245	80019	1	11	11	100.000000
381	28119	1	8	8	100.000000
1068	70065	1	10	10	100.000000
101	10553	1	11	11	100.000000

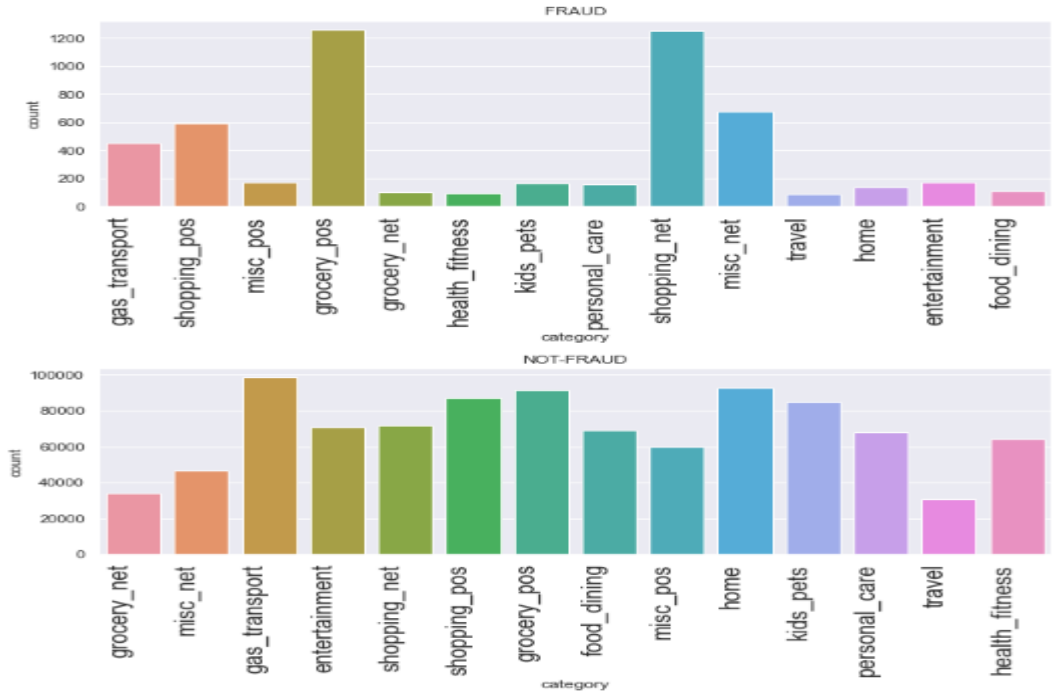
Zip transaction count distribution

	Category	fraud	count	category_count	percent	percent_grp
4	gas_transport	0	98504	98954	10.148753	99.545243
5	gas_transport	1	450	98954	10.148753	0.454757
8	grocery_pos	0	91584	92843	9.522007	98.643947
9	grocery_pos	1	1259	92843	9.522007	1.356053
13	home	1	137	92579	9.494931	0.147982
12	home	0	92442	92579	9.494931	99.852018
25	shopping_pos	1	595	87804	9.005206	0.677646
24	shopping_pos	0	87209	87804	9.005206	99.322354
14	kids_pets	0	84883	85047	8.722447	99.807165
15	kids_pets	1	164	85047	8.722447	0.192835
23	shopping_net	1	1250	73261	7.513671	1.706228
22	shopping_net	0	72011	73261	7.513671	98.293772
1	entertainment	1	168	70741	7.255219	0.237486
0	entertainment	0	70573	70741	7.255219	99.762514
3	food_dining	1	105	68884	7.064765	0.152430
2	food_dining	0	68779	68884	7.064765	99.847570
20	personal_care	0	68200	68354	7.010408	99.774702
21	personal_care	1	154	68354	7.010408	0.225298
11	health_fitness	1	96	64422	6.607141	0.149017
10	health_fitness	0	64326	64422	6.607141	99.850983
18	misc_pos	0	59875	60047	6.158439	99.713558
19	misc_pos	1	172	60047	6.158439	0.286442
16	misc_net	0	46672	47347	4.855923	98.574355
17	misc_net	1	675	47347	4.855923	1.425645
7	grocery_net	1	100	34188	3.506332	0.292500
6	grocery_net	0	34088	34188	3.506332	99.707500
26	travel	0	30478	30565	3.134756	99.715361
27	travel	1	87	30565	3.134756	0.284639

Category variable with respect to the target variable

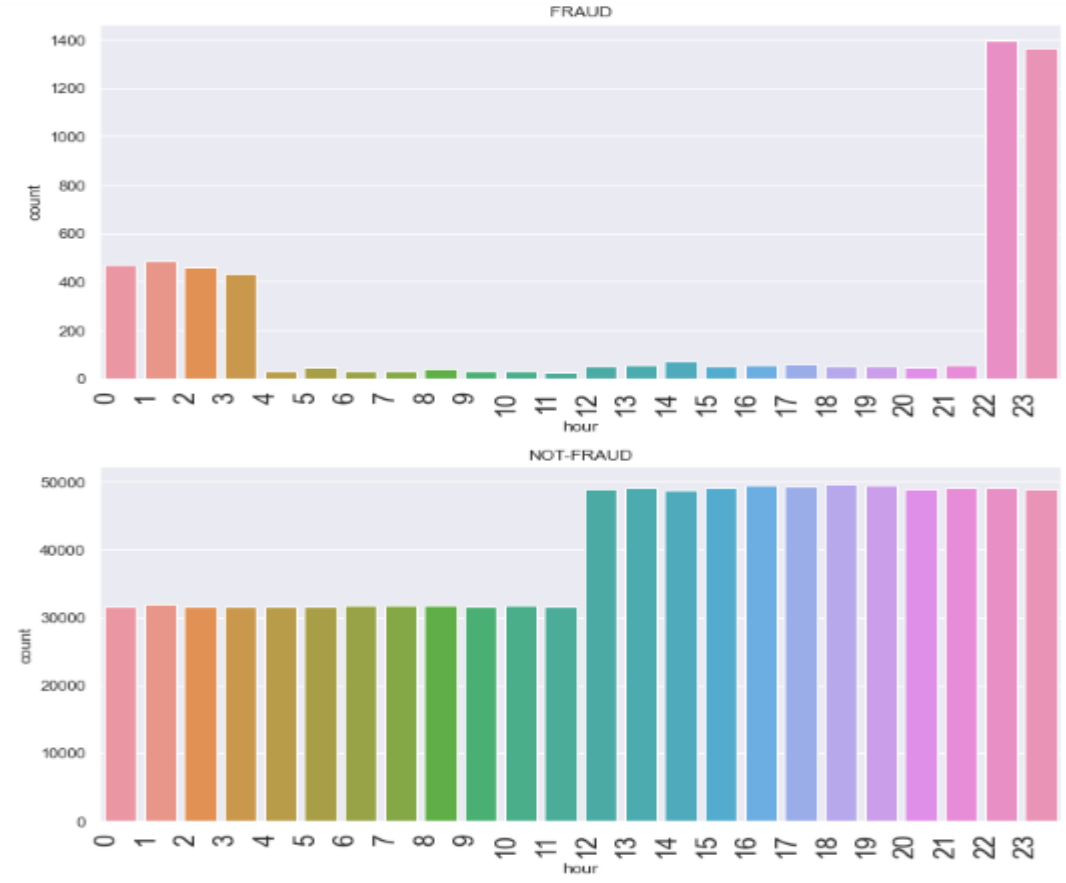


Category variable with respect to the transaction frequency

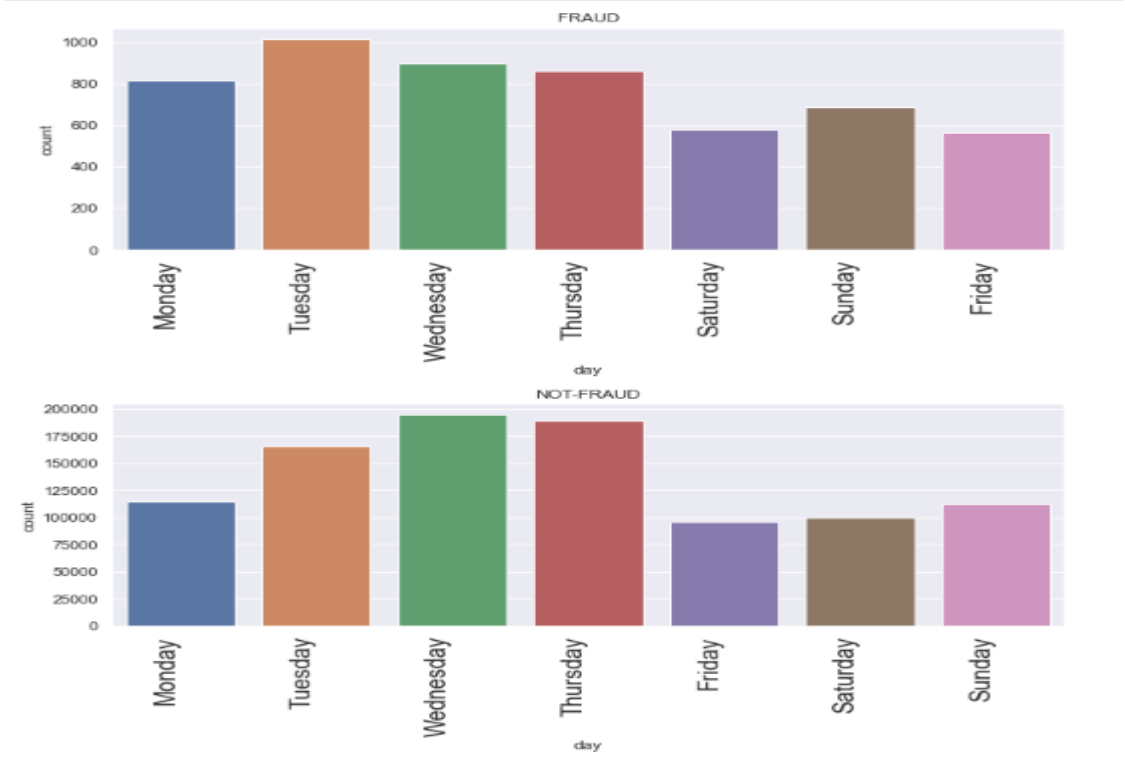


Analysis of the category variable

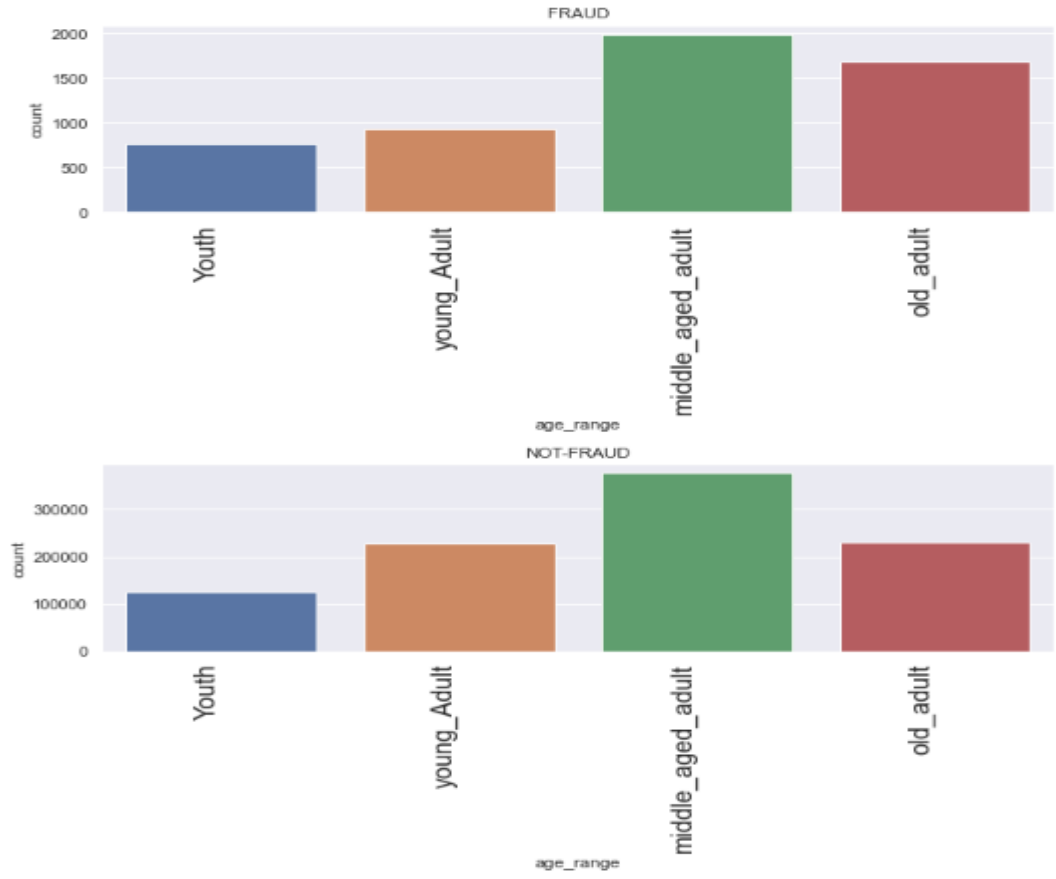
Distribution of the target variable per month



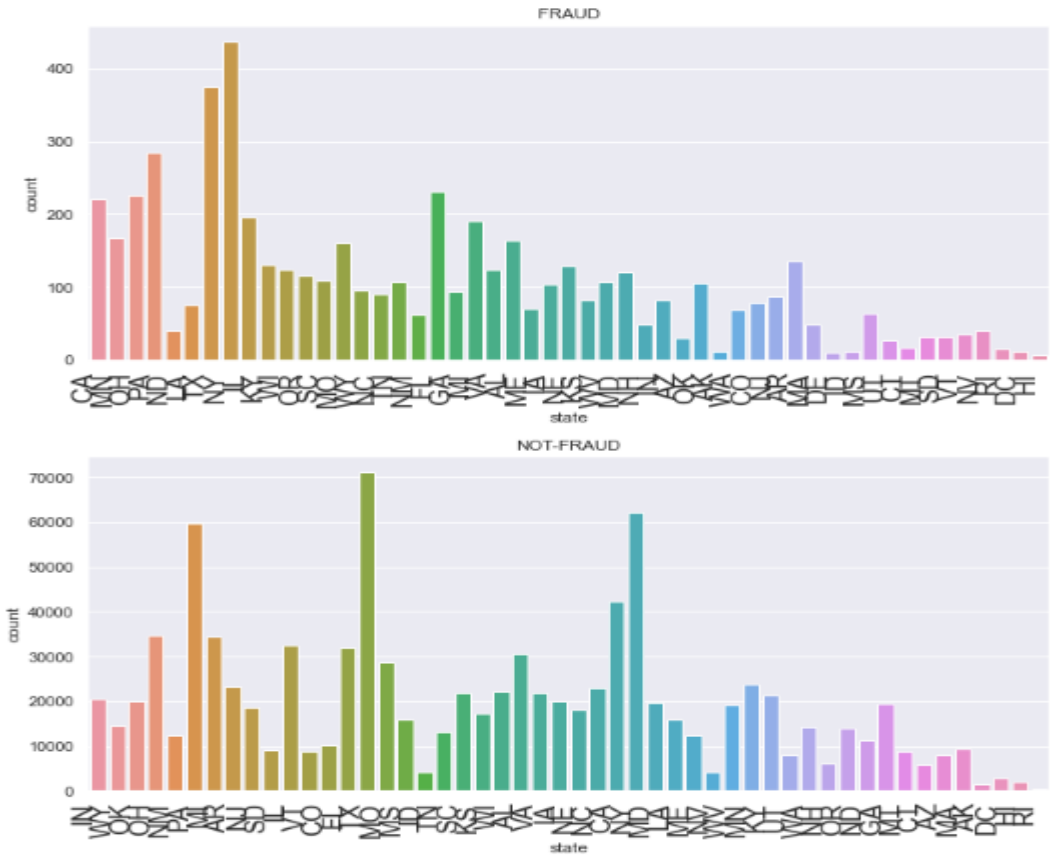
Distribution of the target variable per hour



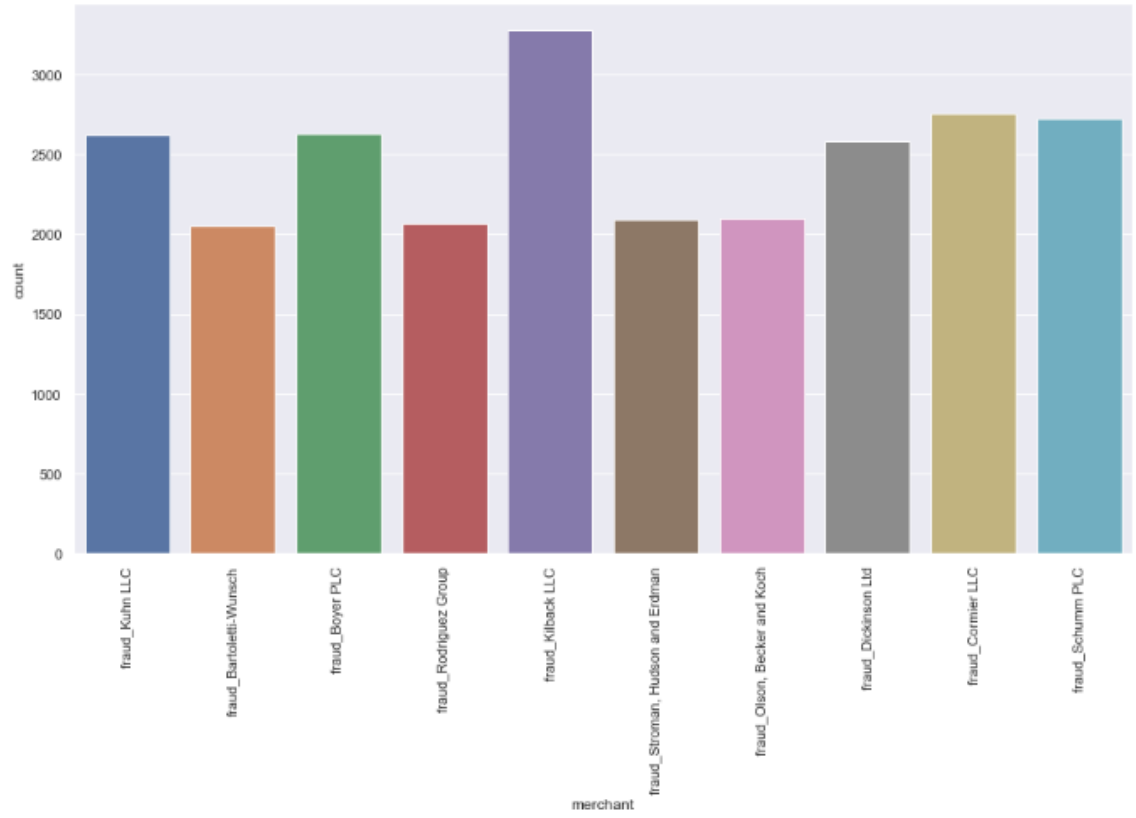
Distribution of the target variable per day



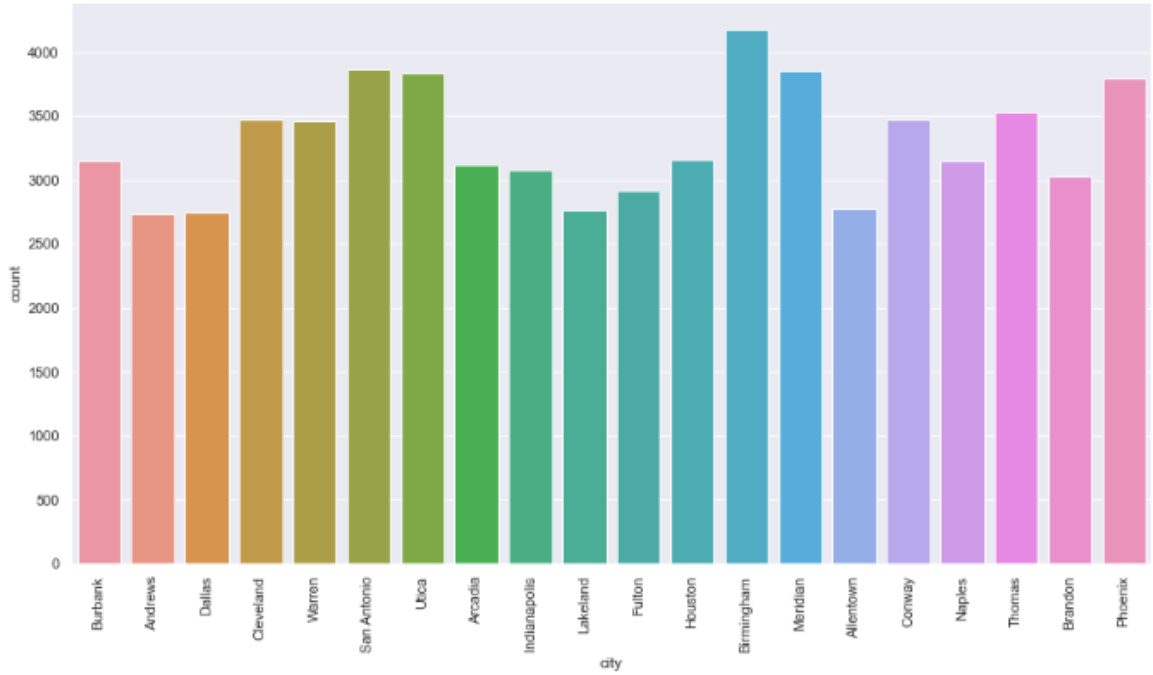
Distribution of the target variable per age range



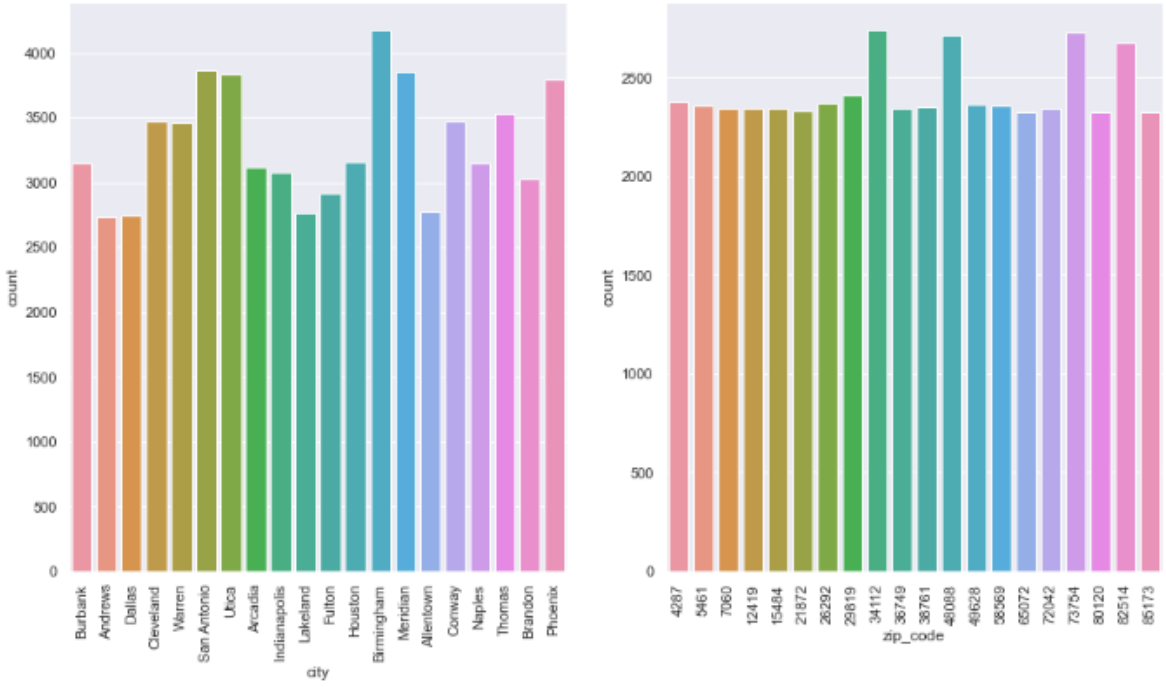
Distribution of the target variable per state



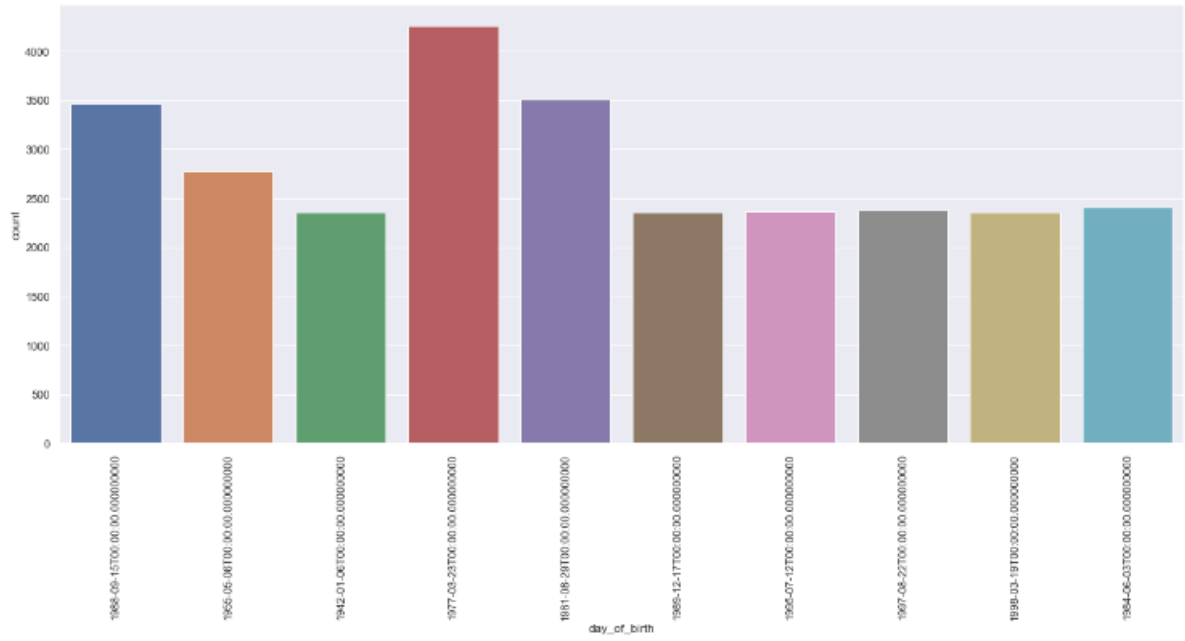
Top 10 jobs with respect to transaction count



Top 20 states with respect to transaction count



City and zip code per transaction frequency



Top 10 date of birth with respect to transaction count

Appendix 3 Feature Encoding

```
#one-hot encoding the category variable
category_onehot = pd.get_dummies(df.category, prefix='category', drop_first=True)
#one-hot encoding the gender variable
gender_onehot = pd.get_dummies(df.gender, prefix='gender', drop_first=True)
#one-hot encoding the day_of_week variable
day_of_week_onehot = pd.get_dummies(df.day, prefix='day', drop_first=True)
#one-hot encoding the age variable
age_onehot = pd.get_dummies(df.age_range, prefix='age_range', drop_first=True)
```

```
data = pd.concat([df, category_onehot,gender_onehot,day_of_week_onehot,age_onehot], axis=1)

data.head()
```

	trans_date_trans_time	credit_card_number	merchant	category	amount	first_name	last_name	gender	street	city	state	zip_code	la
0	2021-02-01 00:02:00	4750000000000	fraud_Kling-Grant	grocery_net	19.460000	Carrie	Washington	F	6114 Adams Harbor Suite 096	Kingsford Heights	IN	46346	41.4
1	2021-02-01 00:03:00	4330000000000000	fraud_Huel-Langworth	misc_net	13.010000	Scott	Martin	M	7483 Navarro Flats	Freedom	WY	83120	43.0
2	2021-02-01 00:05:00	4720000000000000000	fraud_Streich, Hansen and Veum	gas_transport	50.020000	Robert	Drake	M	463 Willie Estates	Burbank	OK	74633	36.6
3	2021-02-01 00:06:00	180000000000000	fraud_Johns Inc	entertainment	6.110000	Jared	Camacho	M	4257 Perez Mall	Canton	OH	44702	40.8
4	2021-02-01 00:08:00	4540000000000000000	fraud_Spinka Inc	grocery_net	32.140000	Nathan	Mendoza	M	767 Adam Mill Apt. 115	Espanola	NM	87533	35.9

Feature Encoding

```
data.drop(['merchant','street','city','state','job',
           'category','gender','day',
           'age'],axis=1, inplace=True)
data.columns
```

```
Index(['trans_date_trans_time', 'amount', 'first_name', 'last_name',
       'zip_code', 'latitude', 'longitude', 'city_population', 'day_of_birth',
       'unix_time', 'merchant_lat', 'merchant_long', 'fraud', 'trans_date',
       'year', 'month', 'hour', 'age_range', 'category_food_dining',
       'category_gas_transport', 'category_grocery_net',
       'category_grocery_pos', 'category_health_fitness', 'category_home',
       'category_kids_pets', 'category_misc_net', 'category_misc_pos',
       'category_personal_care', 'category_shopping_net',
       'category_shopping_pos', 'category_travel', 'gender_M', 'day_Monday',
       'day_Saturday', 'day_Sunday', 'day_Thursday', 'day_Tuesday',
       'day_Wednesday', 'age_range_young_Adult', 'age_range_middle_aged_adult',
       'age_range_old_adult'],
      dtype='object')
```

Dropping Columns

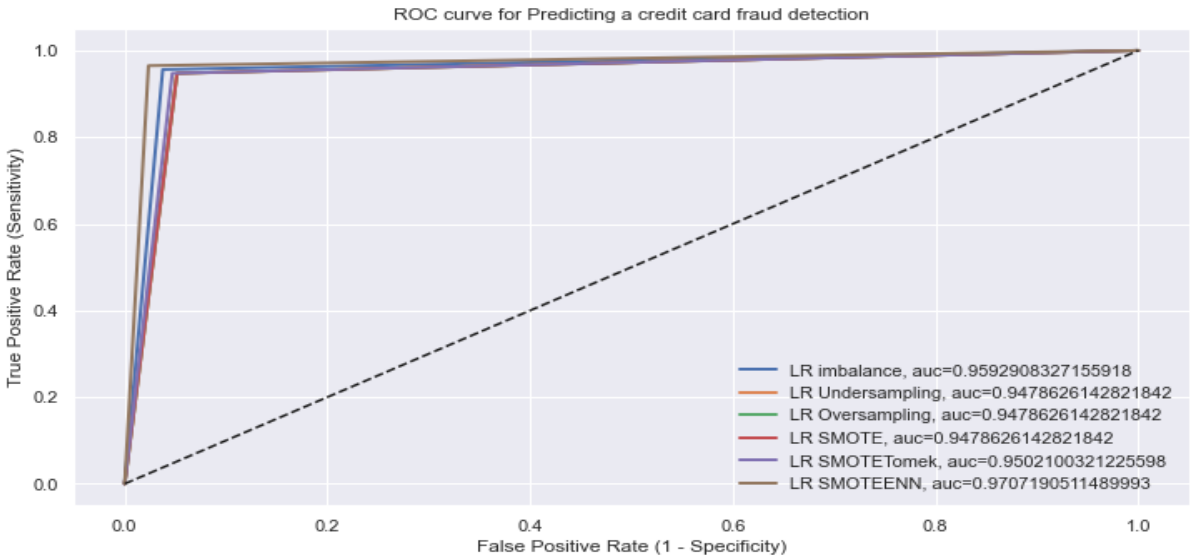
Appendix 4 Summary of result

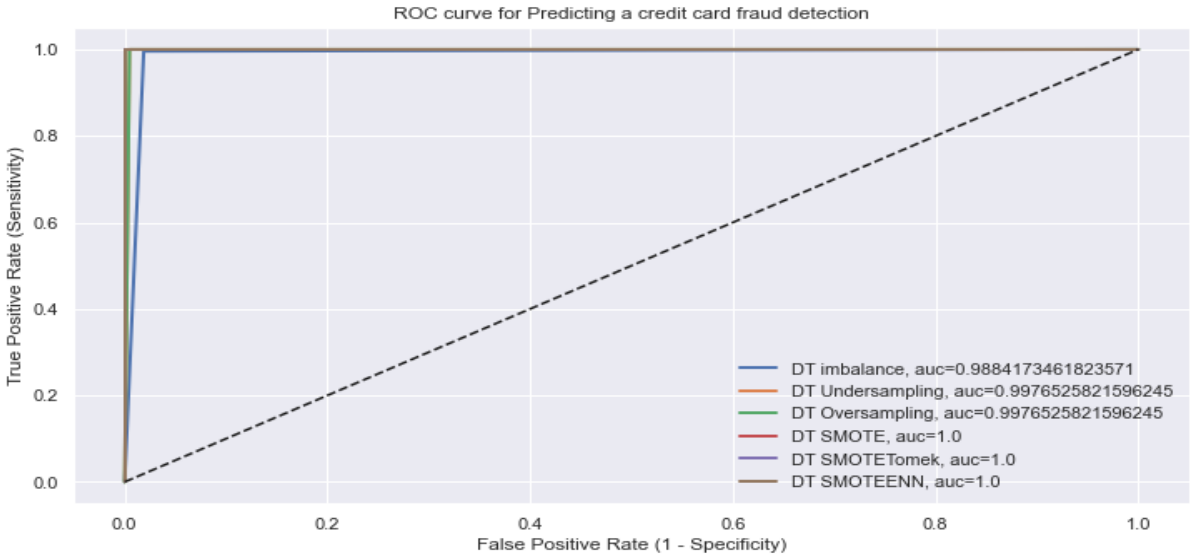
	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
0	Logistic Regression - Before Resampling	0.994065	0.994103	0.994103	0.991605	0.113636	0.009242
1	Logistic Regression - Random Under Sampling	0.807596	0.817995	0.865573	0.922796	0.031539	0.781885
2	Logistic Regression - Random Over Sampling	0.802472	0.815738	0.860568	0.919903	0.030567	0.785582
3	Logistic Regression - SMOTE	0.803053	0.817167	0.865932	0.923003	0.031551	0.780037
4	Logistic Regression - SMOTETomek	0.828598	0.808761	0.867409	0.923854	0.031966	0.781885
5	Logistic Regression - SMOTEENN	0.951841	0.938326	0.865625	0.922826	0.031900	0.791128
6	Decision Tree - Before resampling	0.998621	0.998287	0.998287	0.998206	0.908661	0.749838
7	Decision Tree - Random Under Sampling	0.981376	0.961683	0.962904	0.976806	0.127603	0.974122
8	Decision Tree - Random Over-sampling	0.981986	0.960899	0.948607	0.969044	0.095695	0.977819
9	Decision Tree - SMOTE	0.984735	0.953943	0.954381	0.972169	0.105752	0.968577
10	Decision Tree - SMOTETomek	0.993584	0.965812	0.963960	0.977373	0.129948	0.964880
11	Decision Tree - SMOTEENN	1.000000	0.966960	0.957796	0.973942	0.104296	0.870610

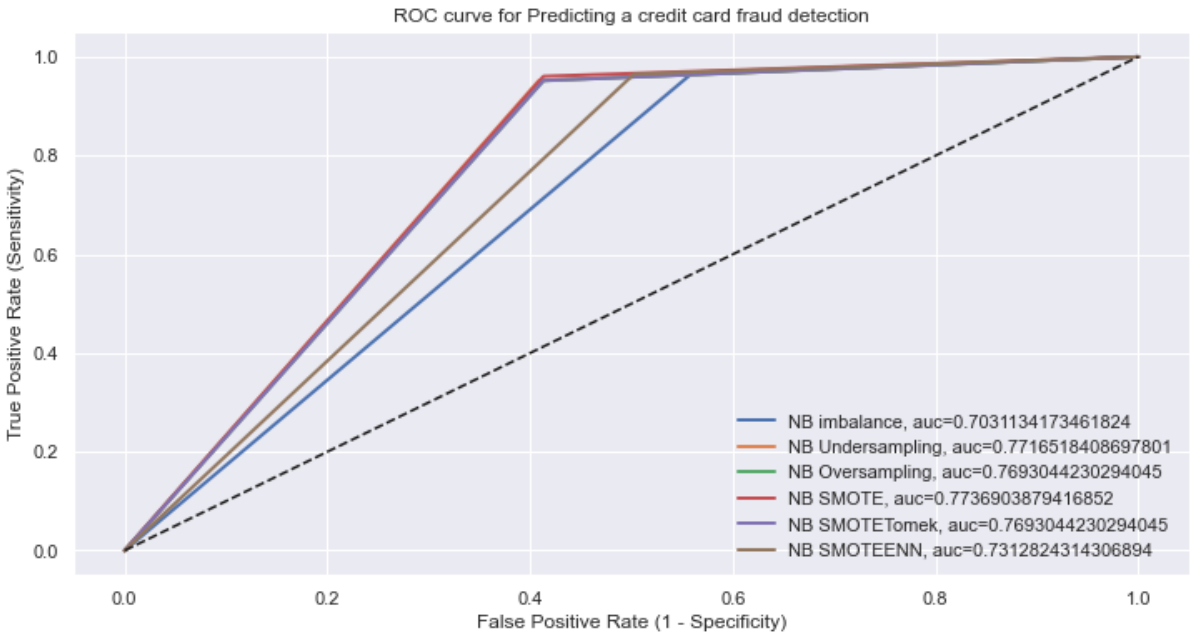
	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
12	Naive Bayesian - Before resampling	0.899867	0.900862	0.900862	0.943015	0.034955	0.670565
13	Naive Bayesian - Random Under Sampling	0.683531	0.663360	0.543665	0.698819	0.010011	0.829945
14	Naive Bayesian - Random Over Sampling	0.690616	0.684262	0.602843	0.746780	0.011189	0.807763
15	Naive Bayesian - SMOTE	0.692607	0.678297	0.588095	0.735152	0.010961	0.820702
16	Naive Bayesian - SMOTETomek	0.683776	0.655983	0.585244	0.732908	0.010694	0.805915
17	Naive Bayesian - SMOTEENN	0.745042	0.737885	0.519379	0.678053	0.009571	0.835490
18	Neural Network (MLPC) - Before Resampling	0.519089	0.519724	0.519724	0.678682	0.008865	0.814815
19	Neural Network (MLPC) - Random under sampling	0.983869	0.944235	0.958268	0.974271	0.113666	0.959335
20	Neural Network (MLPC) - Random Over sampling	0.988479	0.943793	0.942258	0.965588	0.084707	0.959335
21	Neural Network (MLPC) - SMOTE	0.988626	0.934403	0.944515	0.966804	0.086883	0.946396
22	Neural Network (MLPC) - SMOTETomek	0.997709	0.954060	0.934689	0.961466	0.074734	0.946396
23	Neural Network (MLPC) - SMOTEENN	1.000000	0.993392	0.879799	0.930942	0.035717	0.794824
24	Random Forest - Before resampling	0.999979	0.998178	0.998178	0.998090	0.896688	0.738791
25	Random Forest - Random Under Sampling	1.000000	0.965104	0.972750	0.982217	0.166456	0.975970
26	Random Forest - Random Over Sampling	1.000000	0.969697	0.968811	0.980037	0.148086	0.972274
27	Random Forest - SMOTE	1.000000	0.959525	0.968319	0.979761	0.145520	0.966728
28	Random Forest - SMOTETomek	1.000000	0.965812	0.962863	0.976772	0.126214	0.961183
29	Random Forest - SMOTEENN	0.999056	0.977974	0.962863	0.976772	0.126214	0.826248
30	Random Forest - SMOTEENN [Hyperparameter Tuned]	1.000000	0.980176	0.979642	0.985913	0.192504	0.835490
31	Decision Tree - SMOTEENN [Hyperparameter Tuned]	0.990557	0.955947	0.939612	0.964068	0.070798	0.815157

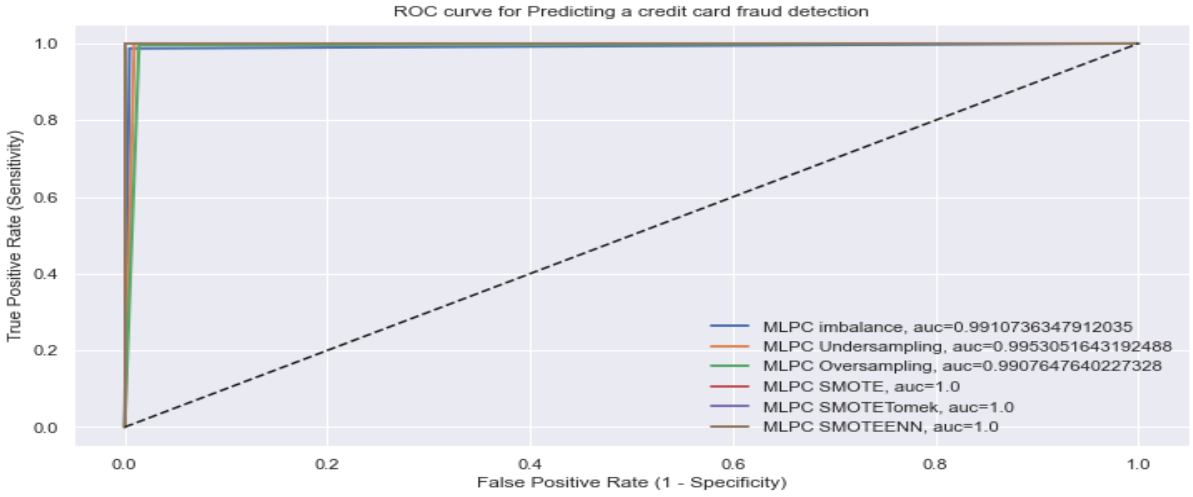
	Model title	Training Score	Testing Score	Accuracy	F1 Score	Precision	Recall
32	Logistic Regression - SMOTEENN [Hyperparameter...	0.951841	0.929515	0.869471	0.925039	0.032530	0.783734
33	MLP - SMOTEENN [Hyperparameter Tuned]	1.000000	0.993392	0.883009	0.932765	0.036908	0.800370

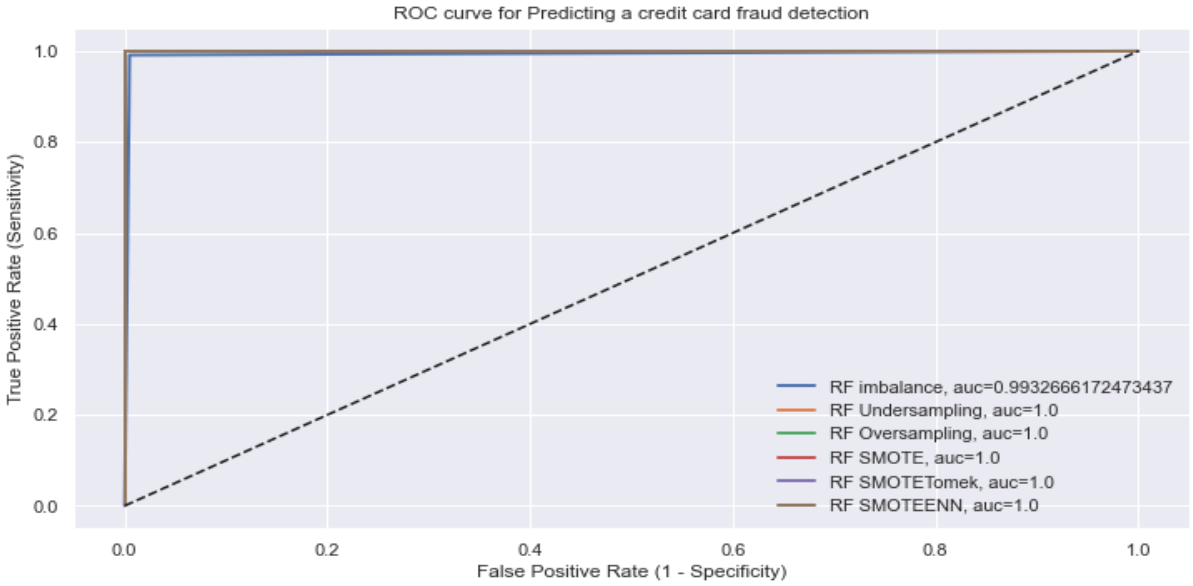
Performance Evaluation



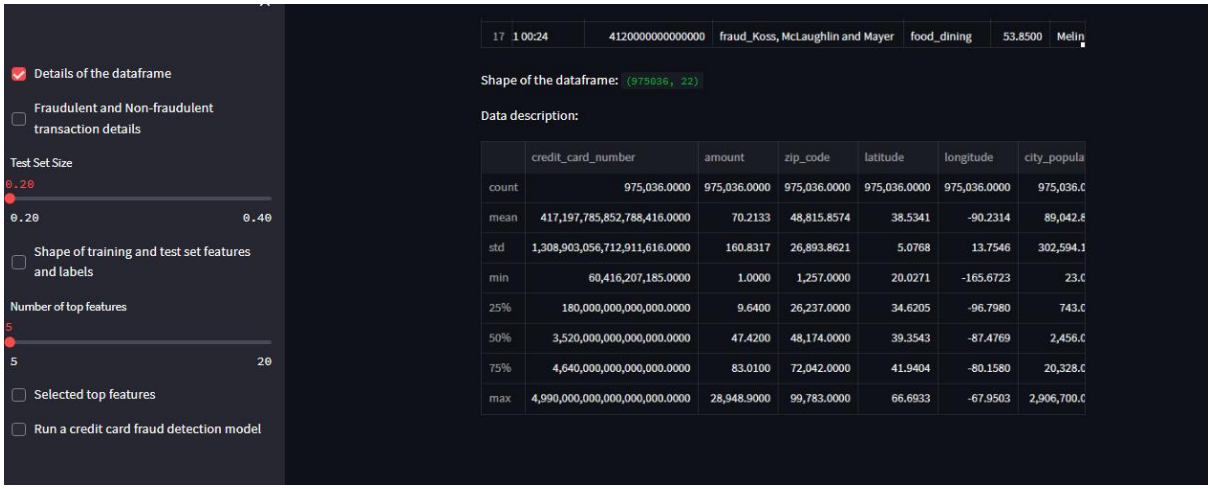


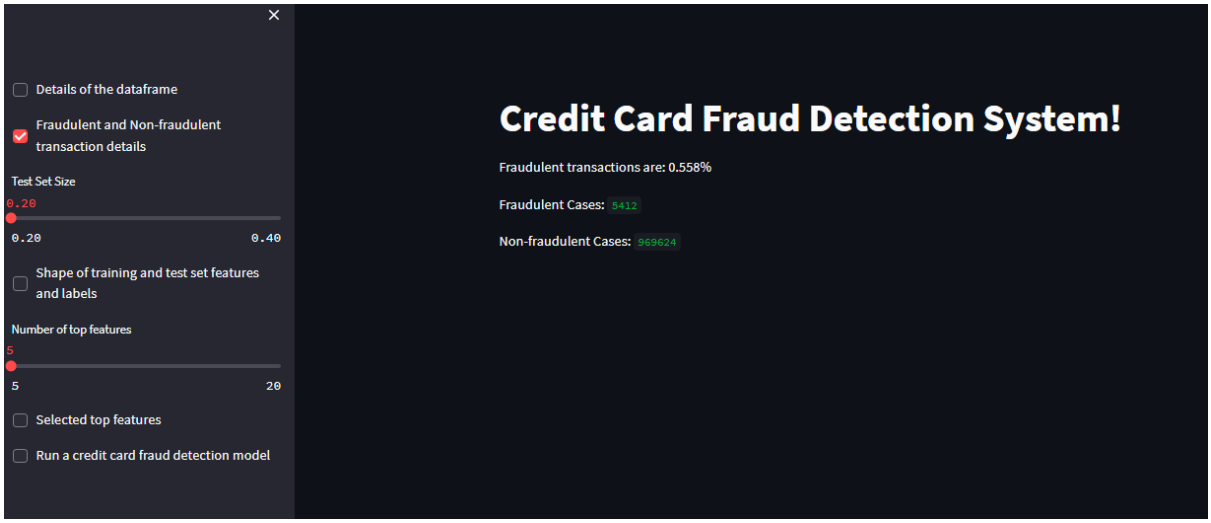

ROC curve for Predicting a credit card fraud detection
True Positive Rate (Sensitivity)
False Positive Rate (1 - Specificity)
NB imbalance, auc=0.7031134173461824
NB Undersampling, auc=0.7716518408697801
NB Oversampling, auc=0.7693044230294045
NB SMOTE, auc=0.7736903879416852
NB SMOTETomek, auc=0.7693044230294045
NB SMOTEENN, auc=0.7312824314306894

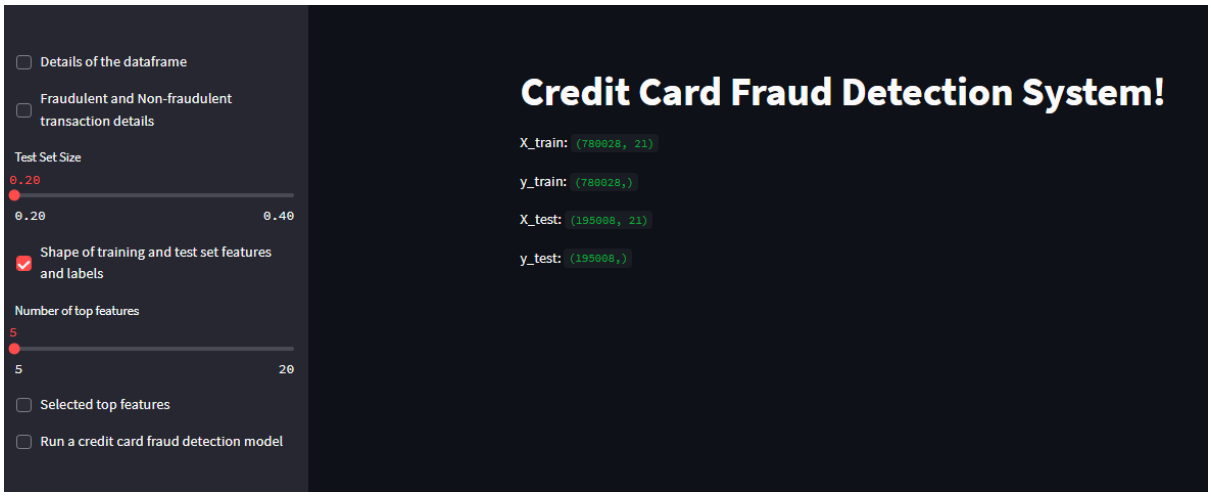

ROC curve for Predicting a credit card fraud detection
True Positive Rate (Sensitivity)
False Positive Rate (1 - Specificity)
MLPC imbalance, auc=0.9910736347912035
MLPC Undersampling, auc=0.9953051643192488
MLPC Oversampling, auc=0.9907647640227328
MLPC SMOTE, auc=1.0
MLPC SMOTETomek, auc=1.0
MLPC SMOTEENN, auc=1.0


ROC curve for Predicting a credit card fraud detection
True Positive Rate (Sensitivity)
False Positive Rate (1 - Specificity)
RF imbalance, auc=0.9932666172473437
RF Undersampling, auc=1.0
RF Oversampling, auc=1.0
RF SMOTE, auc=1.0
RF SMOTETomek, auc=1.0
RF SMOTEENN, auc=1.0

Appendix 5 Streamlit Visualisation







Appendix 6 Ethics Approval

Project status

Status

Approved

Actions

Date	Who	Action	Comments
20:44:00 04 September 2022	Jarutas Andritsch	Supervisor approved	
21:59:00 01 July 2022	Olabisi Dabi	Principal investigator submitted	

Ethics release checklist (ERC)

Project details

Project name: The accuracy versus interpretability trade off in the application of machine model for predicting fraudulent credit card transaction

Ethics Approval

References

A. Saxena and H. Ponnapalli, (2012) U.S. Patent Application No. 13/109,946.

S. Meredith, D. Kent, D. Patterson, and E. Abrahamian. (2018) “Fraud detection via mobile device location tracking.” U.S. Patent No. 9,858,575.

S. Rajasekaran, and R. Varadarajan, (2005) U.S. Patent No. 6,908,030. Washington, DC: U.S. Patent and Trademark Office.

J. Essebag, S. Pochic, and C. Lalo, (2018) U.S. Patent No. 10,032,169. Washington, DC: U.S. Patent and Trademark Office.

Y. Li and X. Zhang, (2004) “A security-enhanced one-time payment scheme for a credit card,” 14th International Workshop Research Issues on Data Engineering: Web Services for e-Commerce and e-Government Applications, Proceedings, Boston, MA, USA, pp. 40-47, doi: 10.1109/RIDE.2004.1281701.

G. McDonald, (2010) U.S. Patent Application No. 12/408,325.

S. Gupta and R. Johari, (2011) “A new framework for credit card transactions involving mutual authentication between cardholder and merchant,” International Conference on Communication Systems and Network Technologies, pp. 22- 26. 10.1109/CSNT.2011.12.

N. Trivedi, (2020) “An efficient credit card fraud detection model based on machine learning methods.” International Journal of Advanced Science and Technology vol. 29, no.5. pp. 3414-3424.

S. Gupta, and R. Johari. (2011) “A new framework for credit card transactions involving mutual authentication between cardholder and merchant.” International Conference on Communication Systems and Network Technologies.

E. Altam, G. Macro, and F. Varetto, (1994) Corporate distress diagnosis: a comparison using linear discriminant analysis and neural networks. Journal Banking and Finance, vol. 18, pp. 505-529.

Y. Sahin, and E. Duman, (2011) “Detecting credit card fraud by ANN and logistic regression.” In 2011 International Symposium on Innovations in Intelligent Systems and Applications, pp. 315-319.

A. Ng and M.Jordan,(2002) “On discriminative vs. generative classifiers: A comparison of logistic regression and naïve Bayes,” Advances In Neural Information Processing Systems, vol. 2, pp. 841-848.

A. Shen, R. Tong and Y. Deng, (2007) “Application of classification models on credit card fraud detection,” Service Systems and Service Management 2007, pp. 1-4.

Y. Sahin and E. Duman, (2011) “Detecting credit card fraud by ANN and logistic regression,” Innovations in Intelligent Systems and Applications, pp. 315-319.

G. John, and P. Langley. (2013) “Estimating continuous distributions in Bayesian classifiers.” arXiv preprint arXiv:1302.496.

J. Awoyemi, O. Adetunmbi, and S. Oluwadare, (2017) “Credit card fraud detection using machine learning techniques: A comparative analysis, International Conference on Computing Networking and Informatics (ICCNI).

K. Sherly, (2012) “A comparative assessment of supervised data mining techniques for fraud prevention,” Int. J. Sci. Tech. Res, vol. 1, no. 16.

J. Pun and Y. Lawryshyn, (2012) “Improving credit card fraud detection using a metaclassification strategy,” International Journal of Computer Applications, vol. 56, no. 10.

R.Quinlau, (1986) “Induction of decision trees,” Machine learning, vol. 1, no. 1, pp. 1- 106.

P. Save, P. Tiwarekar, K. Jain, and N. Mahyavanshi, (2017) “A novel idea for credit card fraud detection using a decision tree.” International Journal of Computer Applications, vol. 161, no. 13.

S. Kalyanakrishnan, D. Singh, R. Kant, 2014, “On building decision trees from large-scale data in applications of online advertising,” Proceedings of the 23rd ACM International Conference on Information and Knowledge Management.

J. Gaikwad, A. Deshmane, H. Somavanshi, S. Patil, and R. Badgujar, (2014) “Credit card fraud detection using decision tree induction algorithm.” International Journal of Innovative Technology and Exploring Engineering (IJITEE), vol.4, no. 6.

S. Lakshmi, and S. Kavilla, (2018) “Machine learning for credit card fraud detection system.” International Journal of Applied Engineering Research vol. 13, no. 24, pp.16819-16824.

S. Maes, K. Tuyls, B. Vanschoenwinkel and B. Manderick, (2002) “Credit card fraud detection using Bayesian and neural networks,” Proceedings of the 1st International Naiso Congress on Neuro Fuzzy Technologies, pp. 261-270.

T. Behera, and S. Panigrahi, (2015) “Credit card fraud detection: A hybrid approach using fuzzy clustering and neural network.” In 2015 Second International Conference on Advances in Computing and Communication Engineering, pp. 494-499.

Batista G, Prati R and Monard M-C. (2004) A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6, 20–29.

Hart P (1968) The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14(3), 515–516.

Tomek I (1976) Two modifications of CNN. IEEE Transactions on Systems, Man, and Cybernetics, 6(11), 769–772.

Wilson DL (1972) Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics 2(3), 408–421.

Widmer G and Kubat M (1994) Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101. https://doi.org/10.1007/BF00116900

Laurikkala J (2001) Improving identification of difficult small classes by balancing class distribution. In Quaglini S, Barahona P and Andreassen S (eds), Artificial Intelligence in Medicine. Berlin, Heidelberg: Springer.

G. McDonald, U.S. Patent Application No. 12/408,325. 2010.

Sadineni, Praveen Kumar. (2020). Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms. 659-660. 10.1109/ISMAC49090.2020.9243545.

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

Sadineni, Praveen Kumar. (2020). Detection of Fraudulent Transactions in Credit Card using Machine Learning Algorithms. 659-660. 10.1109/ISMAC49090.2020.9243545.

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

Drummond, C. and Holte, R. (2003). C4.5, class imbalance, and cost sensitivity: why undersampling beats over-sampling. Workshop on Learning from Imbalanced Datasets II.

Varmedja D, Karanovic M, Sladojevic S, Arsenovic M, Anderla A. (2019) Credit card fraud detection-machine learning methods. In: 18th international symposium INFOTEH-JAHORINA (INFOTEH)

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

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

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

OGWUELEKA, F.N., 2011. DATA MINING APPLICATION IN CREDIT CARD FRAUD DETECTION SYSTEM. Journal of engineering science & technology, 6(3), 311–322

Kotsiantis, S. (2007). Supervised machine learning: A review of classification techniques.

Guo, C. and Berkhahn, F. (2016). Entity embeddings of categorical variables.

Russac, Y., Caelen, O., and He-Guelton, L. (2018). Embeddings of categorical variables for sequential data in fraud context. International Conference on Advanced Machine Learning Technologies and Applications.

Carcillo, F., Pozzolo, A. D., Borgne, Y.-A. L., Caelen, O., Mazzer, Y., and Bontempi, G. (2018). Scarff; a scalable framework for streaming credit card fraud detection with spark. Information Fusion.

Pozzolo, A. D. (2015). Adaptive machine learning for credit card fraud detection. PhD Thesis

Bhattacharyya, S., Jha, S., Tharakunnel, K., and Westland, J. C. (2011). Data mining for credit card fraud: A comparative study. Decision support systems.

Noghani, F. F. and Moattar, M. H. (2015). Ensemble classification and extended feature selection for credit card fraud detection. Journal of AI and Data Mining.

Shapley, L. S. (1953). A value for n-person games. Annals of Mathematics Study, Princeton University Press.

Fossi, L. and Gianini, G. (2019). Managing a pool of rules for credit card fraud detection by a game theory-based approach. Future Generations Computer Systems.

Fawcett and Provost, 1997 Adaptive fraud detection. Data mining and knowledge discovery

Qi, Z.; Zhang, Z. (2020) A hybrid cost-sensitive ensemble for heart disease prediction. pages 21, 73.

Annalisa Appice, Pedro Pereira Rodrigues, Vıtor Santos Costa, Carlos Soares, Jo˜ao Gama, and Alıpio Jorge, editors, (2015) Machine Learning and Knowledge Discovery in Databases, pages 200–215. (Cited on p. 17.)

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 321–357. (Cited on p. 17.)

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. (2005). Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In International Conference on Intelligent Computing, pages 878–887.

DAVID NOVOA-PARADELA, ÓSCAR FONTENLA-ROMERO and BERTHA GUIJARRO-BERDIÑAS, 2020. Adaptive Real-Time Method for Anomaly Detection Using Machine Learning. Proceedings, 54(38), 38–

DE MORAIS, R.F.A.B. and G.C. VASCONCELOS, 2019. Boosting the performance of over-sampling algorithms through under-sampling the minority class. Neurocomputing (Amsterdam), 343, 3–18

WANG, Z. et al., 2019. SMOTETomek-based Resampling for Personality Recognition (July 2019). IEEE access, 7, 1–1

JOHN, O.A., A. ADEBAYO and O. SAMUEL, 2018. EFFECT OF FEATURE RANKING ON THE DETECTION OF CREDIT CARD FRAUD: COMPARATIVE EVALUATION OF FOUR TECHNIQUES. I-manager's Journal on Pattern Recognition, 5(3), 10–

S. Kiran, (, 2018) Credit card fraud detection using naive bayes model based and knn classifier," International Journal of Advance Research, Ideas and Innovations in Technoloy, vol. 4, no. 3.

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

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

ZHUANG, H. et al., 2020. Interpretable Learning-to-Rank with Generalized Additive Models

SHRIKUMAR, A., P. GREENSIDE and A. KUNDAJE, 2017. Learning Important Features Through Propagating Activation Differences

FAN, J., M.E. NUNN and X. SU, 2009. Multivariate exponential survival trees and their application to tooth prognosis. Computational statistics & data analysis, 53(4), 1110–1121

GOLDSTEIN, A. et al., 2014. Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation. ArXiv.org

Fisher A, Rudin C and Dominici F, 2019. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. Journal of Machine Learning Research 20(177), 1–81

YU, Y. and Z.X. CHENG, 2013. The Application of Artificial Intelligence in Ocean Development: In the View of World Expo 2010. Applied Mechanics and Materials, 347-350, 2335–2339

SAHIN, E.K., 2020. Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. SN Applied Sciences, 2(7)

TSANG, M. et al., 2018. Can I trust you more? Model-Agnostic Hierarchical Explanations

GUIDOTTI, R. et al., 2019. A Survey of Methods for Explaining Black Box Models. ACM computing surveys, 51(5), 1–42

SCOTT M LUNDBERG, GABRIEL G ERION and SU-IN LEE, 2019. Consistent Individualized Feature Attribution for Tree Ensembles. ArXiv.org.

LUNDBERG, S. and S. LEE, 2017. A Unified Approach to Interpreting Model Predictions. ArXiv.org

WACHTER, S., B. MITTELSTADT and C. RUSSELL, 2018. COUNTERFACTUAL EXPLANATIONS WITHOUT OPENING THE BLACK BOX: AUTOMATED DECISIONS AND THE GDPR. Harvard journal of law & technology, 31(2), 841–

BOTTOU, L. et al., 2012. Counterfactual Reasoning and Learning Systems