Finanicial Fraud Detection using Artificial intelligence

MINI PROJECT REPORT

Submitted by

SANTOSH KUMAR	**(310520104111)**
RAMESH GIRI	**(310520104096)**
VIVEK CHAUHAN	**(310520104140)**
RANJIT KUMAR	**(310520104098)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DHANALAKSHMI SRINIVASAN

COLLEGE OF ENGINEERING AND TECHNOLOGY

MAMALLAPURAM, CHENNAI – 603 104

ANNA UNIVERSITY :: CHENNAI - 600 025

MAY 2023

ANNA UNIVERSITY :: CHENNAI – 600 025

BONAFIDE CERTIFICATE

Certified that this project report " **FINANCIAL FRAUD DETECTION USING ARTIFICIAL INTELLIGENCE MECHANISM"** is the bonafide work of **" SANTOSH KUMAR (310520104111), RAMESH GIRI (310520104096), VIVEK CHAUHAN (310520104140) and RANJIT KUMAR(31052010409)"** who carried out the project work under my supervision.

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. P. Malathi, Ph. D,
Associate Professor,

Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan College of Engineering & Technology, Mamallapuram, Chennai.

SIGNATURE

SUPERVISOR

Dr. P. Malathi, Ph. D,
Associate Professor,

Department of Computer Science and Engineering,

Dhanalakshmi Srinivasan College of Engineering & Technology, Mamallapuram, Chennai.

Submitted for the project viva voce examination held on_____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

First of all, we thank, Our **Almighty** for his blessings upon us to strengthen our mind and soul to take up this project. We owe a great many thanks to a great many people who helped and supported us in this project.

We thank Our **Chairman, Thiru. A. Srinivasan** who allowed us to do the project.

We extremely thank to Our **Director, Thiru. P. Mani** for his constant support in selecting the project.

We are also thankful to Our **Principal**, **Dr. R. Saravanan, Ph.D** for his constant support to do project.

We are very thankful to Our **Vice Principal**, **Dr. V Janakiraman, Ph.D** for his constant support to do project.

We are grateful to Our **Head of the Department**, **Dr. P. Malathi**, **Ph.D** who expressed his interest and guide in our work and supplied with some useful ideas.

We thank Our **Guide, Dr. P. Malathi, Ph.D** for following our project with interest and for giving me constant support. She taught us not only how to do the project, but also how to enjoy project.

We wish to extend our grateful acknowledgement and sincere thanks to our project coordinators, **Dr. P. Malathi** and **Mr. S. Niresh Kumar** for their constant encouragement and kind support in completing the project

Furthermore, we would like to thank all our **Teaching Faculty and Non-teaching Faculty** for their timely help in solving any project queries.

Finally, we would like to thank our **Parents** for their blessings, support and encouragement throughout our life.

Abstract

Fraud detection on bank payments is a crucial aspect of ensuring the safety and security of financial transactions. With the increasing use of electronic payment methods, the risk of fraudulent activities has also increased. The abstract of this topic aims to highlight the importance of detecting and preventing fraud in bank payments. The paper will discuss various methods and techniques used to detect fraud, such as data analytics, machine learning, and artificial intelligence. The study will also focus on the challenges and limitations of these methods and suggest possible solutions. The goal is to develop an effective fraud detection system that can prevent financial losses and protect the integrity of financial transactions.Fraudulent behavior can be seen across many different fields such as e-commerce, healthcare, payment and banking systems. Fraud is a billion-dollar business and it is increasing every year.

Financial fraud is a global problem that affects individuals, businesses, and governments. It involves using deception, misrepresentation, or manipulation to obtain financial gain. Financial fraud detection is the process of identifying and preventing fraudulent activities in financial transactions. The goal of financial fraud detection is to protect the financial system from fraudsters, maintain trust in financial institutions, and safeguard the investments of individuals and businesses. This requires a combination of domain expertise, data analytics, and machine learning. Financial institutions and regulatory bodies use various techniques to identify suspicious activity and prevent fraud. With the increasing complexity of financial systems and the sophistication of fraudsters, financial fraud detection has become a critical area of focus for the financial industry and governments globally.

Table of contents

List of Figures

List of Tables

Chapter 1

1.1 Introduction

Financial transactions make the most of this world as people exchange goods, services and more. These transactions usually occur following an agreement made between two or more people upon performing the said services (Westermeier, 2020). Therefore, they range from a small amount to large business transactions involving huge sums. Examples of financial transactions include the reception of cash, deposits, purchases, invoices, charges of services, expenses and more. The currencies range depending on the agreement of the parties involved. The banking industry is also involved in the bulk of financial transactions as most individuals and businesses seek to engage in such services. With the wide variety of types of transactions and amounts involved, financial transactions are tempting to some people to commit fraud.

The data involved in financial transactions is often huge and could uniquely identify and classify each one. Detecting fraud cases is an emerging field in the Machine Learning area, which utilizes large datasets to detect these anomalies. The application of such procedures is successful because transactions have trends which can be pointed out by ML models. Any transaction standing out can be a good example of a fraud transaction warranting additional investigations. Banks, insurance companies and more are some of the largest customers of fraud detection because of the delicate nature of their work. However, fraud detection is not as straightforward in most cases as the parameters of the transactions might be many and the case too complex to detect (Carminati, et al., 2018). Approaches ranging from regression analysis, checking the probability of distribution, using statistical approaches, and more are useful in revealing the anomalies associated with the fraud. Most detection tools rely on a few key models

which analyze the data and raise any suspicious behavior. The level of threshold in terms of anomalies can be adjusted to differentiate legitimate transactions from fraudulent ones.

The rising use of machine learning has prompted it to be utilized in many areas. The field focuses on using computers and huge datasets to seek patterns, knowledge, and trends that are otherwise hard to detect from manual procedures. Technology is placing itself as a key part of most modern actions and is turning out to be an ever-important area.

1.2 Background

Different types of fraud exist depending on the area they represent. Some of these fraud types include credit card fraud, tax evasion or malpractices, insurance fraud, banking fraud (Repousis, Lois, & Veli, 2019), and more. The fraud cases center around misuse, misappropriation, or violations involving funds. The perpetrators usually masquerade the transactions as legitimate ones to avoid some truth, regulations, or laws.

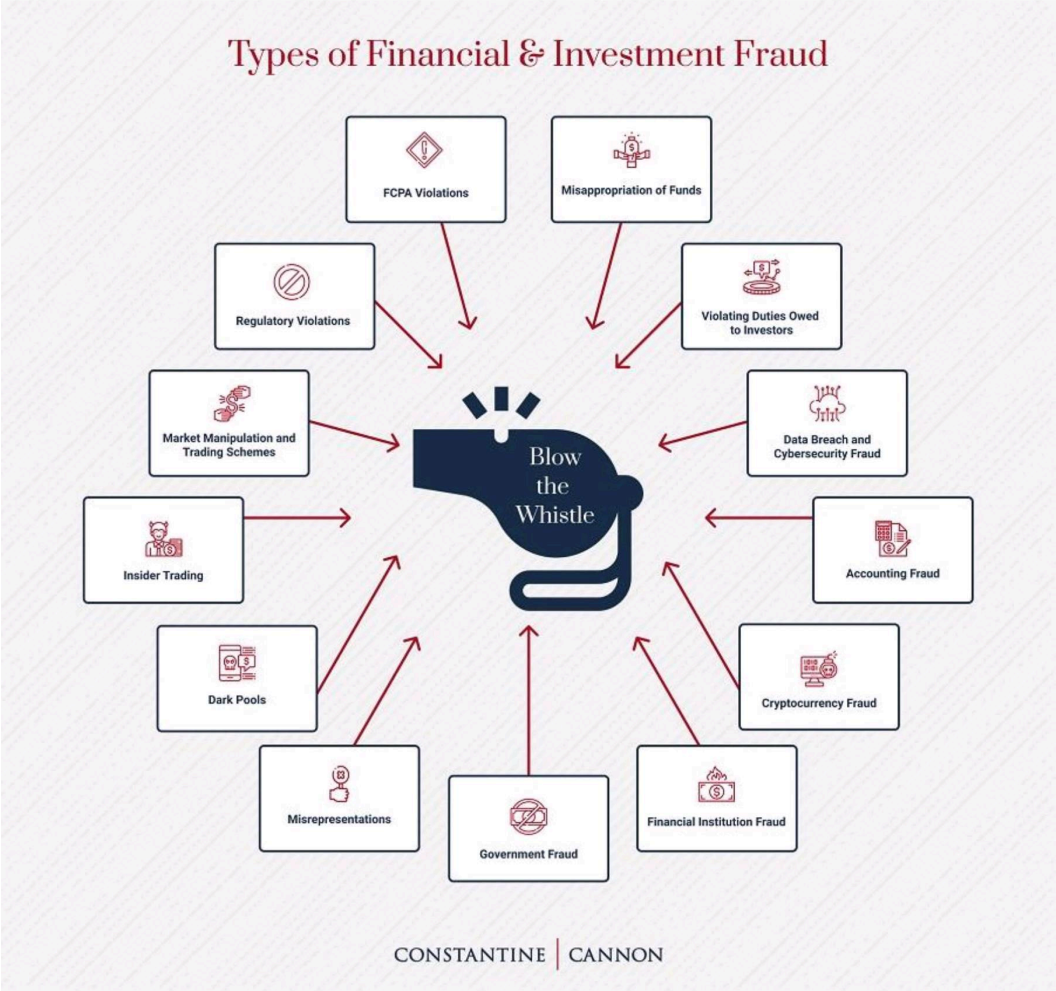


Figure 1: Financial Fraud Types

This image represents a few types of financial and investment fraud categories that transactions can fall into.

1.2.1 History of Fraud

Financial fraud has been in existence for a long period since the times of the Greeks. With their times involving the exchange of goods for currency, there were examples where merchants would use their wit to avoid paying loans, misrepresent their services and more (Hajdari, 2021). Fraud has since evolved to involve investors, traders, businesses, and large corporations. Cases like the collapse of Lehman Brothers investment bank are an example of bad

things happening after ignoring sound financial operations. Insider trading is also a troubling case of fraud where a market such as the stock market is manipulated to provide a benefit to someone buying or selling them.

1.2.2 Laws on Fraud

All countries take fraud cases seriously and impose harsh punishments for perpetrators. The United States has codes associated with fraud in its list of laws. These provide strict measures of how transactions should be conducted and the consequences of going against them (Ermakova & Frolova, 2019). The United Kingdom is also serious with financial fraud, with cases receiving steep sentences of up to ten years. Different nations have also come up with regulations to handle fraud and deal with the perpetrators.

1.2.3 Fraud Detection Models

Detecting fraud in credit card transactions has been a major part of modern banking. It is typical for one to fall prey to such cases where criminals obtain the information and utilized it to use funds on one's account without authorization. The approach utilizes supervised learning, where labelled cases are used to train the model and create a model to classify new data (Khatri, Arora, & Agrawal, 2020). These models can also be utilized on real-time data where the detection happens on the go. Suspicious cases are then sent as an alert to the appropriate people.

1.3 Goals and Objectives

The main objective of this exercise is to develop a detection model on a dataset containing transactions labelled as fraud or legitimate cases. This model can then be used by an institution to classify new data depending on its performance. The modelling process will involve a detailed data analysis procedure that will consider the data types, quality, and more.

Objectives

- To investigate the cases of fraud, fraud detection and changes in technology that can handle them.
- To create a working fraud detection machine learning model on a chosen fraud dataset.
- To clean up the dataset and check its eligibility in creating a working fraud detection model.
- To test the results of the fraud detection model on new data and determine its performance.
- To provide recommendations on how well to handle fraud with modern technology and the created model.
- To document the detailed process of data analysis and the classification of fraud detecting cases.

Chapter 2

2.1 Literature Review

2.1.1 Finance and Banking

Banking and services involving finances have been in existence throughout the history of humankind. The accumulation of wealth and its safekeeping pushed the formation of institutions that managed money, stored it, and supported the trade (Ichinkhorloo, 2018). These institutions also allowed governments to handle and distribute wealth as needed, seek taxes from citizens, and more. Banks allow the governments, institutions, and individuals to perform these actions by standing on the transparent ground. Banking also allows the trading of countries with foreign nations, without which the exchange of goods and services would prove hard. From coins and shells to paper money, the methods of finance and banking have been changing (Ichinkhorloo, 2018). The early methods of barter trading quickly proved to be insufficient in representing fairness.

The introduction of loaning for paybacks with interest also increased the possibilities of financing (Green, 2019). Banks soon caught up and introduced ways to obtain loans with the intention of their generation of profit from the interest. Further, since banks were designed to hold the representation of keeping funds for others, they soon got into the investment scene where they could utilize the funds for other cases. The concept of loaning also led to the introduction of seeking funds for business, personal cases, or in cases of countries running their economies.

Banking types have since expanded and subdivided into different types, including investment banks, commercial banks, retail banks, credit unions, saving and loans banks, online

banks, and community development banks (Dow, 2017), among others. Each of these types handles funds in different ways and involves different transactions on the money. The eventual goal, however, is to handle funds received from other parties and seek profits for themselves or for the clients. The typical transactions involved in finance and banking include depositing money, withdrawing it, taking loans, paying bills and utilities, paying back loans, sending money to a different person either by cheques or through wire transfer, and more (Parsaee Tabar, Abdolvand, & Rajaee Harandi, 2021). Other transactions involve handling the activities with the bank or with the bank being a third party for an agreement between two parties. Other transactions, such as investments, involve the institution receiving money from other people and making transactions that make profits on their behalf.

Banking, therefore, is a highly regulated area that needs the care to avoid any actions that might lead to the loss of anyone's funds (Begenau & Landvoigt, 2022). Despite the regulations, checks, control and overview from other parties, finance and banking are still full of financial fraud and misrepresented situations.

2.1.2 Fraud

The changes in fraud in financial transactions have been throughout history. Some of the cases are everyday small issues, but there are a set of major fraud cases which have plagued history. For example, in modern times, Wall Street has been in several scandals involving individuals and institutions committing fraud (Toms, 2019). The imbalance between regulators and the responding to issues led to the rise of corporations relying on legal loopholes to perform financial fraud. Such cases are complex in nature and involve a number of issues that constitute fraud cases. Accounting has also increased in complexity to work on keeping detailed records of financial transactions. The definition of fraud has also been broad due to the wide range of

actions that can lead to its classification as so. However, (Toms, 2019) outlines that the ethical imbalance of actions, especially involving funds, can be broadly defined as fraud. Early cases were small actions on corruption at the individual level or crimes that involved finance. Large cases of financial scandals and fraud have been on the rise in recent times. As indicated by (Toms, 2019), fraud cases saw a sharp in the late twentieth century in the United States and the United Kingdom.

The introduction of new technologies like blockchain funds is also changing the nature of fraud. Early testing of blockchain systems and currencies such as Bitcoin has shown to be a haven for fraud. However, such technologies are showing the potential to be excellent in preventing fraud if implemented correctly. Some of the modern systems present a hard time for the detection of fraudulent transactions. The rise in such cases is also indicative of the high inefficiency of modern financial systems (Tapscott & Tapscott, 2017). The researchers attribute such inefficiencies to the design of financial transactions, which were created with manual processes but have since been digitized without a redesign to ensure safety.

(Karpoff, 2021) Points out that the increase in platforms involving crowdfunding is most likely to attract more fraudulent transactions and cases. However, different factors are considered when viewing the general trends in fraud cases. Some areas integrating the blockchain more into their transactions show a decrease in fraud cases. (Karpoff, 2021) Also emphasizes the immense work necessary to detect and reduce fraud generally. The predictions might be hard to pull off given the complex nature of finance and the involved transactions. Extra research is necessary to identify and handle these areas.

2.1.3 Artificial Intelligence and Machine Learning on Fraud

Automated systems are being developed to detect and measure the risks associated with fraud cases. These systems utilize the latest technologies, including Artificial Intelligence and Machine Learning. The private insurance sector is quickly expanding and learning to utilize all available tools to combat fraudulent transactions. A framework such as the one suggested by (Dhieb, Ghazzai, Besbes, & Massoud, 2020) utilizes ML models such as Gradient boosting ones to improve high accuracy levels in detection. Compared to other approaches utilizing models such as decision trees, newer frameworks are showing an increase of around seven percent in accuracy. These models are backed up by automatic systems to collect the relevant data necessary to detect such fraudulent cases. The extensive research and application cases such as insurance present the financial industry with a good testing ground for cases of fraud.

(Psychoula, et al., 2021) Points out that full-scale adoption of tools like Machine Learning and complex models to handle fraud detection requires approaches that could be simplified and reused. The decisions, predictions and results of fraud detection models, especially the complex ones, are an area that needs to be understood for their efficient tweaking and improvements. The increase in the integration of AI and machine learning, however, is a promising move for the future of fraud detection (Psychoula, et al., 2021). Despite having a good performance and accuracy, any model is understood for its choices and classification in both fraud and legitimate cases.

The adaptation of advanced security and technology that handles data at a higher level has been forced onto organizations (Singla & Jangir, 2020). These changes are slowly becoming efficient in predicting data and handling sensitive cases at a new level. Attackers have also evolved in their approaches and tools to keep on improving their approach to financial

transactions. Modern technology has to cover areas that include detecting anomalies and contributing to the general analysis of transaction data. The tools are also allowing the side-by-side comparison of different approaches and models, which leads to the selection of the best based on a situation.

2.1.4 Emerging Technology in Handling Fraud

The progress in Machine Learning and Artificial Intelligence is still in its early days. The switch to massive data collection and the inclusion of the relevant model has been a recent move. Some methods in fraud detection are proposing the inclusion of more data to accompany the transaction and calculating the likelihood of a transaction being fraudulent (Srikanth, 2021). The researchers here propose the inclusion of suspicious activities outside finance that can help detect and predict such cases. Such a case, however, might be intrusive and not sit well with most people following its in-depth look into other areas.

For credit-card holders, fraudulent transactions can affect their trust level with their bankers or other financial-related activities. For instance, (Lebichot, et al., 2021) suggests that transactions associated with credit cards usually cover a wide range of a person's life and make it difficult to handle accurately. The changing nature of fraud activities also means that models can quickly be inefficient if a larger percentage of the fraudsters change their ways (Lebichot, et al., 2021). The researchers also show that transferring a model based on a certain case or country to a new set of data could quickly prove inefficient in accuracy. These researchers propose a model that combines different approaches and can detect the change and adapt to the new environment (Lebichot, et al., 2021). This shows the extent and future that fraud detection holds and its capabilities.

Other approaches are utilizing blockchain technology to reduce the potential occurrence of fraud cases (Attaran & Gunasekaran, 2019). However, the slow adoption rate of blockchain technology and its use cases is leading to insufficient testing results to determine the results. The costs associated with the implementation of such technologies are often putting off excellent potential use in safeguarding transactions (Attaran & Gunasekaran, 2019). The solutions need the implementation of a wide range of areas to work together in securing such financial transactions. From digital ledgers, decentralizing the accounts and data to creating permanent records for each transaction, blockchain technology is expected to change a lot in finance.

The slow adoption of new technologies has always been an issue for many institutions (Gaol, Budiansa, Weniko, & Matsuo, 2022). Despite some cases being proved to work well and handle issues related to cyber security, most financial institutions still lag behind in terms of adoption. Part of this slow approach is the hesitancy to replace working systems despite their shortcomings. Disruptive technologies are requiring the move to newer systems in finance; however, they are forcing and pushing such adoptions (Gaol, Budiansa, Weniko, & Matsuo, 2022).

2.1.5 Previous Work on Fraud Detection and CatBoosts

The application of the CatBoost algorithm in detecting fraud in the financial industry is a steeply rising case. (Chen & Han, 2021) shows a demonstration of the approach and cites the importance of utilizing CatBoost in a dynamic environment. In the comparison of CatBoost with models like Deep Neural Network, the former shows a larger performance hence a higher likelihood chance of successfully identifying these cases (Nguyen, et al., 2022). It is also important to note that areas where the finance industry plays a role such as in medical covers can also experience fraud (Hancock & Khoshgoftaar, 2020). Utilizing the CatBoost algorithm in such

cases also shows a better performance than others. Such a large application of the algorithm is evident in its good performance, particularly in fraud detection. Tools that apply the models are still on the rise, with a notable missing of their widespread use. However, most cases needing detection of fraud are often each unique on their own, which needs custom application of the algorithm with tuning and interpretation of the results to fit the case. Big data applications also exist where CatBoost is seen to apply especially when categorical variables hold more value in the data (Hancock & Khoshgoftaar, 2020). Its interdisciplinary application is also being explored for the best results in different fields.

Summary of The Literature Review

- The finance industry and banking have grown to cover many areas and brought regulatory needs.
- Loaning allows banks to generate profits from industries while securing customer funds.
- Banks vary in types, including investment, commercial, retail, saving and loan banks, and online and community development banks.
- Transactions include money deposits, withdrawals, bill payments, loan processing, and sending money.
- Fraud cases are rampant in the banking industry, given its high volume in handling money despite advances in technology.
- Detecting fraud is increasingly necessary to curb illegal practices.
- AI and ML have models that can detect and classify fraud and non-fraud cases by learning through existing data.
- Adaptive detection of fraud cases is an emerging approach to help curb crimes.
- Models like CatBoost have so far proven highly accurate in detecting fraud cases through classification.

Chapter 3

3.1 Methodology

The approach requires the obtaining of existing data on credit card transactions that have been classified alongside fraud and legitimate cases. This use of secondary data allows the application of different models to determine whether one can effectively predict such cases. The data is also descriptive, with several independent variables against the target one. The next step involves the preparation of the data to make it fit into a model without errors and issues. These preparation steps will cover the investigation of whether the data contains missing values and, if so, involve their removal or replacement with the relevant data. Missing data is a large part of data analysis as it often skews the results and results in a misinterpretation of the results. Some models also do not work when missing data is present, making dealing with it a crucial part of the analysis.

The modelling process involves the fitting of the data into a machine learning model to determine its performance. For the best results, the approach splits the data into training and testing subsets, with the training set containing more observations and is useful in providing the input of the models to determine the different classes based on the trends in the observations. The testing set allows for the classification of the determination in performance of the model and whether it can effectively classify the fraudulent transactions on blind data.

An analysis of the results will provide a look at the performance and an interpretation of whether the model can be utilized in a real-world scenario. Since machine learning models are demanding items that need a lot of effort, the results are reflective of this effort and their utilization in a different setting. The results and use of the model will be determined by the

nature of the real-world application, where similar labels and data could be utilized in future classification.

3.1.1 Rationale

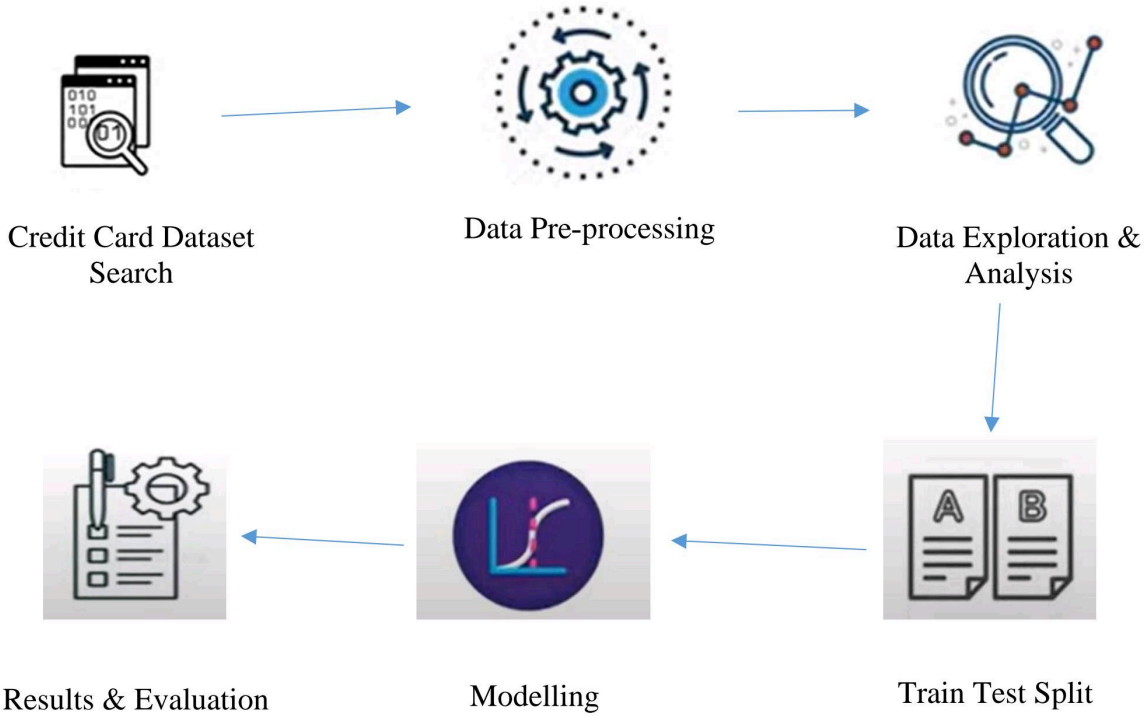
The choice for an existing dataset is to cut down the time needed to collect and organize such data for classification. Further, since most data on transactions is often closed source, utilizing an existing database allows the experimentation in an otherwise impossible task. The choice of the dataset also depends on its level of understand-ability and usefulness in modelling. Organized data is preferable due to the reduction in the time necessary to clean and organize it. The choice of using machine learning models such as CatBoost, Decision Tree, and Random Forest is to allow the comparison of performances and allowing the selection of the best for the study case. The method of determining performance will be the accuracy values, F-Scores, Precision, Recall and Support values for the models. These provide a different look into how the model handled the data and the resulting situations after training on the data. Time will not be a factor when determining the performance of the model as the study case does not need ultra-fast models. Ideally, the use case for the approach is on existing data that has been recorded and where the need for real-time prediction is not necessary.

3.1.2 Why CatBoost

The use of CatBoost in this process comes from its unique characteristics and advantages fitting the process. For instance, the model has a fast inference given its application of symmetric trees. This approach on CatBoost means that it does not have to validate each tree as they mirror each other. Further, the categorical preference of the model means that it handles such variables excellently. CatBoost is typically better in complexity than other models when it comes to categorical features. The application in the finance industry on fraud detection also means that

the application will utilize the fast learning rate of the algorithm. CatBoost also allows weighting of the parameters which can come in handy when tuning on the go is needed. These advantages prompted the replacement of the Linear Regression algorithm with the CatBoost model in this study. The differences in the anticipated results are clear and could be an added advantage to real-world applications.

Workflow Process



Chapter 4

4.1 Data Pre-processing

4.1.1 Data Quality Dimensions

The 6 key data quality dimensions are represented in figure 2 below:

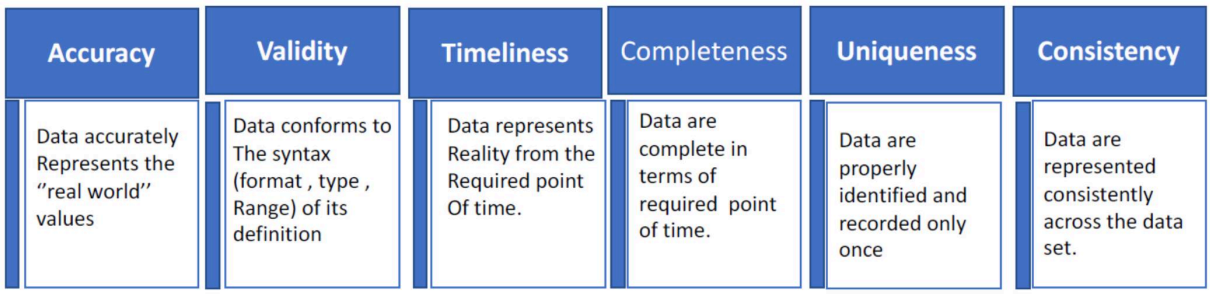


Figure 2: Data Quality Dimensions

In the data pre-processing stage, we have to check for the quality of our data based on the 6 key dimensions. Our dataset can be assessed through these dimensions to decide whether the data can be used or not. First, the data needs to be complete and no missing values in it. Second, the dataset should follow the same format when representing the values in the attributes. Third, there should be no conflicting information between the values which could mislead our analysis. Fourth, the dataset should be accurate and up to date. Fifth, search for any duplicated values in the dataset. Finally, checking for missing data or not referenced.

<table>
<tr>
<td>Completeness</td>
<td>Data is complete, no missing values.</td>
<td>

```r
cbind(lapply(lapply(Fraud, is.na), sum))
```

<table>
<tr><th></th><th>[,1]</th></tr>
<tr><td>step</td><td>0</td></tr>
<tr><td>type</td><td>0</td></tr>
<tr><td>amount</td><td>0</td></tr>
<tr><td>nameOrig</td><td>0</td></tr>
<tr><td>oldbalanceOrg</td><td>0</td></tr>
<tr><td>newbalanceOrig</td><td>0</td></tr>
<tr><td>nameDest</td><td>0</td></tr>
<tr><td>oldbalanceDest</td><td>0</td></tr>
<tr><td>newbalanceDest</td><td>0</td></tr>
<tr><td>isFraud</td><td>0</td></tr>
<tr><td>isFlaggedFraud</td><td>0</td></tr>
</table>

</td>
</tr>
<tr>
<td>Conformity</td>
<td>

- Some amounts are not following currency format
- Difficult to know which values are represented by 0 or 1 in isFraud variable.

</td>
<td>

isFraud	oldbalanceDest	newbalanceDest
1	42100000	42200000
1	41500000	40900000
0	41400000	41500000
0	41400000	41300000

oldbalanceOrg	newbalanceOrig	amount
38900000	38900000	6419835
38600000	38900000	6072832
38400000	38600000	5860863
38400000	38400000	5677662

</td>
</tr>
<tr>
<td>Consistency</td>
<td>No conflicting information</td>
<td></td>
</tr>
<tr>
<td>Accuracy</td>
<td>All data is up to date</td>
<td></td>
</tr>
<tr>
<td>Duplicates</td>
<td>There are no duplicates in our dataset</td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td>No missing data</td>
<td></td>
</tr>
</table>

4.2 Data Exploration & Analysis

The data has eleven variables with varying information on transactions such as the type, balance changes, names of the accounts, the step in the transaction, and whether or not the transaction is a fraud.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figure 3: Overview of Variables

In Figure 3, we can see an overall view of the variables in our dataset.

	step	amount	oldbalanceOrg	newbalanceOrig	oldbalanceDest	newbalanceDest
count	6362620	6362620	6362620	6362620	6362620	6362620
mean	243.40	179861.90	833883.10	855113.67	1100701.67	1224996.4
std	142.33	603858.23	2888242.67	2924048.50	3399180.11	3674128.9
min	1.00	0.00	0.00	0.00	0.00	0.0
25%	156.00	13389.57	0.00	0.00	0.00	0.0
50%	239.00	74871.94	14208.00	0.00	132705.66	214661.4
75%	335.00	208721.48	107315.18	144258.41	943036.71	1111909.2
max	743.00	92445516.64	59585040.37	49585040.37	356015889.35	356179278.9

Figure 4: Summary of Statistics of Numeric Variables

Using R studio program, we are able to find the statistics of the numeric variables of our dataset as shown in Figure 4. Finding the mean, minimum, maximum and standard deviation helps us in better understanding the dataset at the initial stage.

```
> names(fraud)
 [1] "step"          "type"           "amount"
 [4] "nameOrig"      "oldbalanceOrg"  "newbalanceOrig"
 [7] "nameDest"      "oldbalanceDest" "newbalanceDest"
[10] "isFraud"       "isFlaggedFraud"
```

Figure 5: Attributes in The Dataset

Figure 5 shows the names of the attributes in the dataset. We have a total of 11 names and each name represents an attribute in our dataset.

- 'step' = maps to a unit of time in the real world
- 'type' = is the type of transaction made
- 'amount' = amount of money transferred
- 'nameOrig' = person who initiated the transaction
- 'oldbalanceOrg' = the amount before the transaction
- 'newbalanceOrg' = the amount after the transaction
- 'nameDest' = the person who is the recipient of the transaction
- 'oldbalanceDest' = initial balance of the recipient before the transaction
- 'newbalanceDest' = new balance of the recipient after the transaction
- 'isFraud' = the transaction is fraud
- 'isFlaggedFraud' = the transaction is flagged fraud

```
> str(fraud)
spec_tbl_df [1,048,575 × 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ step          : num [1:1048575] 1 1 1 1 1 1 1 1 1 1 ...
 $ type          : chr [1:1048575] "PAYMENT" "PAYMENT" "TRANSFER" "CASH_OUT" ...
 $ amount        : num [1:1048575] 9840 1864 181 181 11668 ...
 $ nameOrig      : chr [1:1048575] "C1231006815" "C1666544295" "C1305486145" "C840083671"
 ...
 $ oldbalanceOrg : num [1:1048575] 170136 21249 181 181 41554 ...
 $ newbalanceOrig: num [1:1048575] 160296 19385 0 0 29886 ...
 $ nameDest      : chr [1:1048575] "M1979787155" "M2044282225" "C553264065" "C38997010" .
 $ oldbalanceDest: num [1:1048575] 0 0 0 21182 0 ...
 $ newbalanceDest: num [1:1048575] 0 0 0 0 0 ...
 $ isFraud       : num [1:1048575] 0 0 1 1 0 0 0 0 0 0 ...
 $ isFlaggedFraud: num [1:1048575] 0 0 0 0 0 0 0 0 0 0 ...
```

Figure 6: Columns in the dataset

The *str()* command gives us the columns in the dataset as shown in Figure 6.

```
> head(fraud)
# A tibble: 6 × 11
   step type     amount nameOrig oldba…¹ newba…² nameD…³ oldba…⁴ newba…⁵ isFraud isFla…⁶
  <dbl> <chr>     <dbl> <chr>      <dbl>   <dbl> <chr>      <dbl>   <dbl>   <dbl>   <dbl>
1     1 PAYMENT    9840. C123100…  170136  160296. M19797…       0       0       0       0
2     1 PAYMENT    1864. C166654…   21249   19385. M20442…       0       0       0       0
3     1 TRANSFER    181  C130548…     181       0  C55326…       0       0       1       0
4     1 CASH_OUT    181  C840083…     181       0  C38997…   21182       0       1       0
5     1 PAYMENT   11668. C204853…   41554   29886. M12307…       0       0       0       0
6     1 PAYMENT    7818. C900456…   53860   46042. M57348…       0       0       0       0
# … with abbreviated variable names ¹oldbalanceOrg, ²newbalanceOrig, ³nameDest,
#   ⁴oldbalanceDest, ⁵newbalanceDest, ⁶isFlaggedFraud
```

Figure 7: Structure of the dataset

The *head()* command showed us the structure of the dataset for better understanding as shown in figure 7.

```
> table(paymenttype)
paymenttype
 CASH_IN CASH_OUT    DEBIT  PAYMENT TRANSFER
  227130   373641     7178   353873    86753
```

Figure 8: Payment types & Occurrences

Through the *table()* command we found out the types of transactions and the number of their occurrences as shown in figure 8.

4.2.1 Preparation and Data Visualization

It is also notable that the dataset has no missing values, which further makes it easier to perform the modelling. There are more legitimate transactions than there are fraudulent, with the fraud cases being 1142 while the legitimate ones being more than one million.

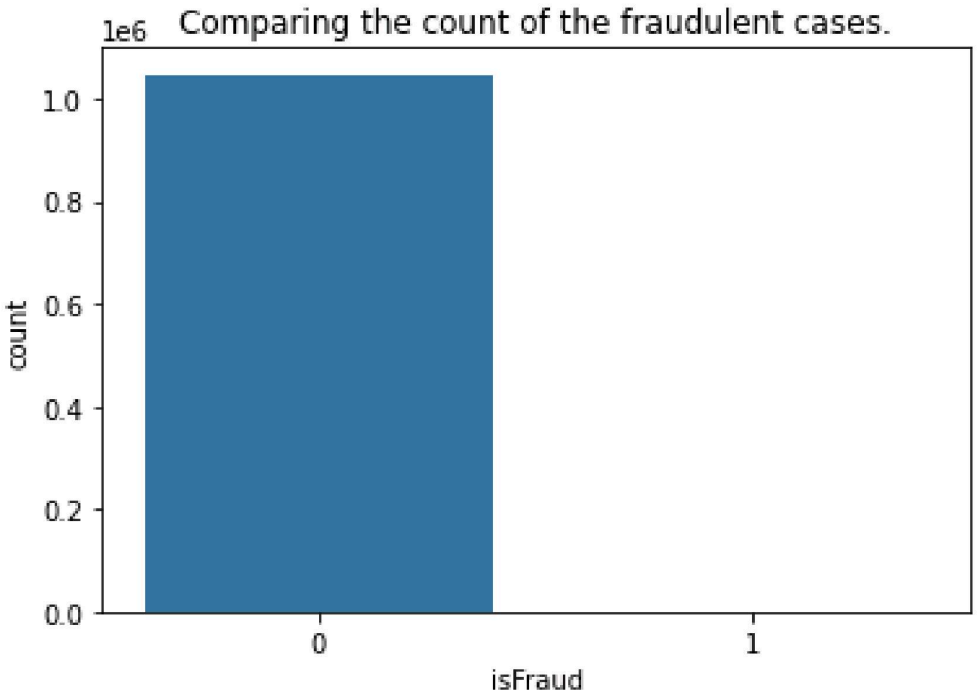


Figure 9: Fraud vs Non-Fraud Cases

The image shows the fraud against the non-fraud cases, with a clear indication that the two classes are not balanced in terms of number.

The types of transactions are distributed among cash outs, payments, cash-ins, transfers of funds, and debit transactions. For an easy process in handling the model, these types are one hot encoded by converting each category into a numerical representation.

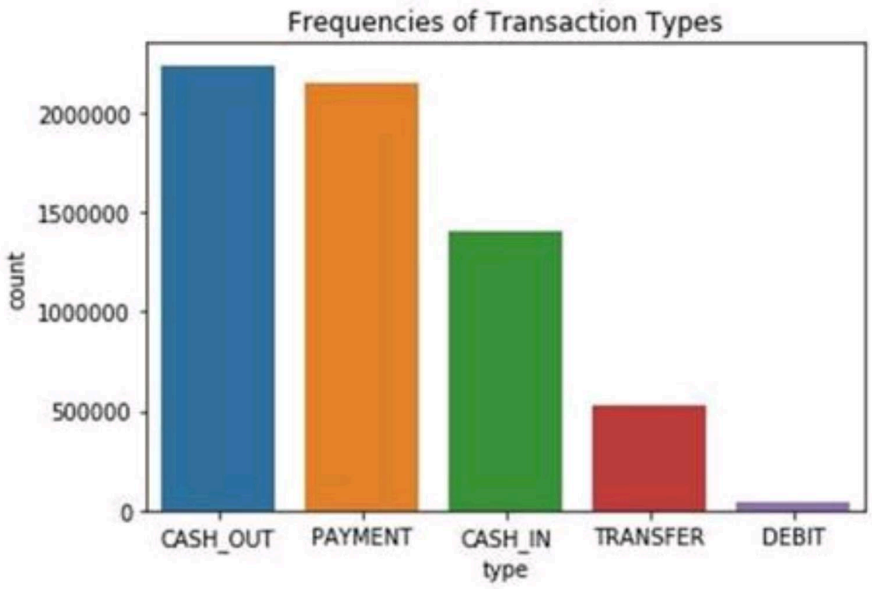


Figure 10: Type of transactions

The transaction types also differ in type. In the figure above, cash-out transactions are leading, followed by payments, then cash-in ones, transfers and very few transactions on debit. It would be evident that the higher-level transaction types are more likely to be targeted for fraud.

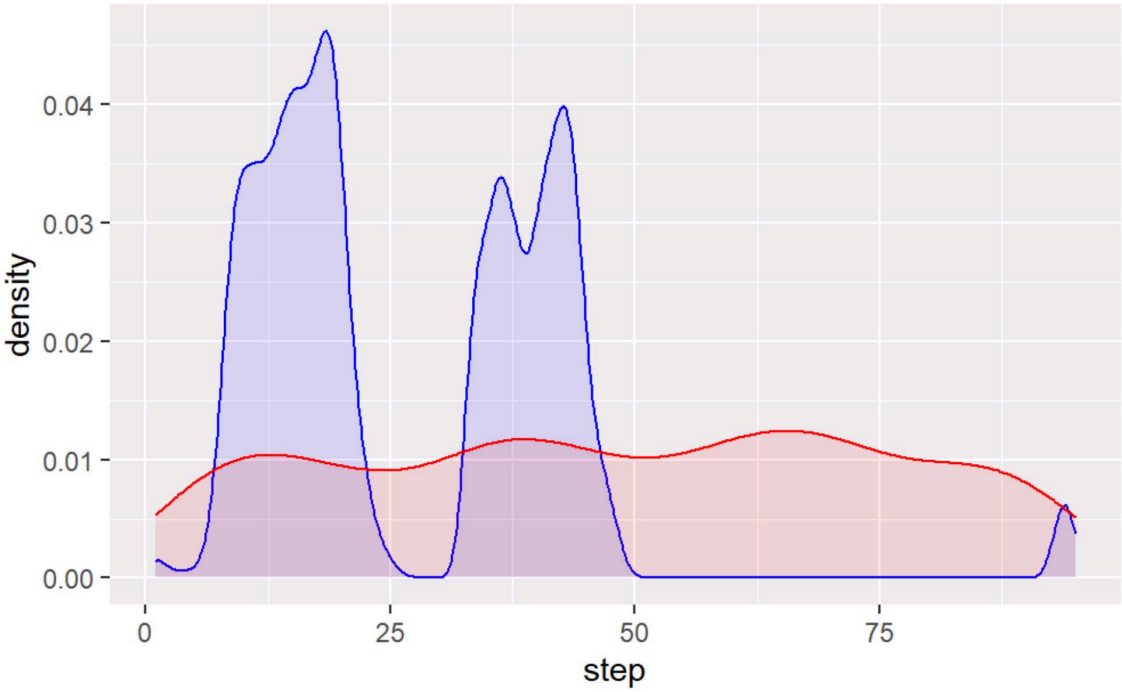


Figure 11: Density Plot of Fraud Cases

The density plot in figure 8 shows the separation between the fraud and non-fraud cases over time period of one month. The blue plot refers to the “Fraud” and the red plot refers to “Non-Fraud”. As the plot shows that at the beginning of the month there is an increase in the fraud cases compared to the end of the month.

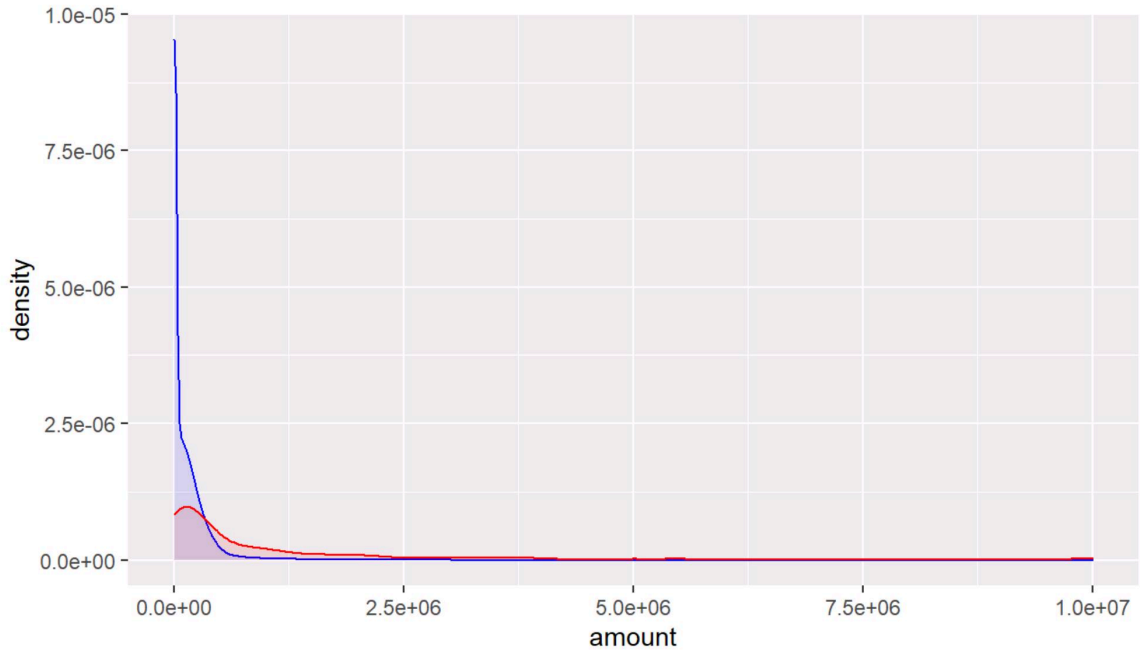


Figure 12: Density Plot of The Amount Transferred

In figure 12, the density plot shows that when the amount of transaction is high there is a higher probability that the transaction if fraud.

4.3 Modelling

```
In [5]: df1.head(5)
```

Out[5]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Figure 13: Dataset Overview

The dataset is split into different subsets which contain either the target variable, 'isFraud', and the features variables, 'isFraud', 'nameOrig', and 'nameDest'. The subsets are split with a test/target ratio of 20/80. This allows a larger training size for the model and a small set of values to determine its performance. There is also a need for the dataset to be balanced to avoid any skewed results. This is achieved by up sampling where values with zero values are inserted among the existing samples to raise their rate. The result is a representation of legitimate transactions being 837934 while the fraud cases increase to 648200. This means that the sets are ready for feeding into the machine learning model.

4.3.1 Overall View of Results

Model	Train Accuracy	Test Accuracy	Train F1-Score	Test F1-Score	Train AUR-ROC	Test AUR-ROC
CatBoost	0.9998	0.9993	0.9997	0.7286	0.9997	0.9279
Decision Tree	1.0	0.9997	1.0	0.8257	1.0	0.8878
Random Forest	0.9826	0.9895	0.9799	0.1557	0.9814	0.9603

Table 1: Results of Models

4.4 CatBoost Classifier

This is a model that utilizes gradient boosting to boost the output of decision trees. It is excellent for classification and options which require ranking. The model utilizes five hundred iterations and takes three minutes to finish training.

4.4.1 Results

```
Training Dataset
Accuracy =: 0.999762470948111
F1_Score =: 0.9997277816207096
AUR_ROC =: 0.9997893628853824

Validation Dataset
Accuracy =: 0.9993383878120307
F1_Score =: 0.7286063569682152
AUR_ROC =: 0.927904353232756
```

Getting the confusion_matrix results of the CatBoost Classifier. This report indicates the TP,TN,FP,FN.

```
In [21]: print(confusion_matrix(target_valid, predictions_cat_valid))
```

```
[[167512     86]
 [    25    149]]
```

Getting the classification_report results of the CatBoost Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
In [22]: print(classification_report(target_valid,predictions_cat_valid, digits=3))
```

```
              precision    recall  f1-score   support

           0      1.000     0.999     1.000    167598
           1      0.634     0.856     0.729       174

    accuracy                          0.999    167772
   macro avg      0.817     0.928     0.864    167772
weighted avg      0.999     0.999     0.999    167772
```

Figure 14: CatBoost Classifier Results

The result indicates an accuracy of 99.9%, 72.8% F1-score and an Area Under Curve ROC of 92.8% on the validation dataset. The precision shows that for the dataset, the model has a chance of 1 in predicting legitimate transactions, and a 0.634 chance of predicting fraud transactions. The recall results indicate that the CatBoost model has a chance of 1 in going through the non-fraud cases but a 0.856 chance in going through the fraud transactions.

		Predicted	
		Positive (0)	**Negative (1)**
Actual	**Positive (0)**	167512	86
	Negative (1)	25	149

Table 2: CatBoost Confusion Matrix

The confusion matrix shows that 167512 transactions were correctly classified as legitimate cases and 149 ones were correctly labelled as fraud cases. These are the true positives and true negative predictions of the dataset from the CatBoost model. The misclassifications were 25 and 86 for false positive and false negatives respectively.

4.5 Decision Tree

The decision tree classifier is a model that utilizes a set of outcomes that are probable and predicts their chance of occurrence. This creates a tree-like structure with every possible combination of outcomes and their probabilities. The model is a supervised learning tool and is popularly used in classifying items.

4.5.1 Results

```
Training Dataset
Accuracy =: 1.0
F1_Score =: 1.0
AUR_ROC =: 1.0

Validation Dataset
Accuracy =: 0.9996602532007725
F1_Score =: 0.8256880733944953
AUR_ROC =: 0.8878773345579385
```

Getting the confusion_matrix results of the DecisionTree Classifier. This report indicates the TP,TN,FP,FN.

```
In [24]: print(confusion_matrix(target_valid, predictions_dt_valid))
[[167580     18]
 [    39    135]]
```

The interpretation of the confusion matrix.
Actual Values
TP = 167580
FP = 18\

Predicted Values
FN = 39
TN = 135

Getting the classification_report results of the DecisionTree Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
In [25]: print(classification_report(target_valid,predictions_dt_valid, digits=3))
              precision    recall  f1-score   support

           0      1.000     1.000     1.000    167598
           1      0.882     0.776     0.826       174

    accuracy                          1.000    167772
   macro avg      0.941     0.888     0.913    167772
weighted avg      1.000     1.000     1.000    167772
```

Figure 15: Decision Tree Classifier Results

Implementing the model and viewing the results show a 99.9% accuracy result, 82.6% F1-Score, and 88.8% score of the AUC-ROC against the validation dataset. This translates to a chance of 1 in predicting legitimate transactions and a 0.882 chance of predicting the other

fraudulent cases. The recall results also show that the model has a chance of 1 for checking all legitimate records but a smaller 0.776 chance in going through the fraudulent ones.

		Predicted	
		Positive (0)	**Negative (1)**
Actual	**Positive (0)**	167580	18
	Negative (1)	39	135

Table 3: Decision Tree Confusion Matrix

The confusion matrix displayed shows the overview performance of the model. 167580 transactions were correctly classified correctly as a legitimate transaction, with 135 being classified as negative, or fraud cases correctly. 39 cases were false positives while 18 cases were false negatives.

4.6 Random Forest

This algorithm is an approach that combines several decision trees to build a set of trees that are more likely to perform well. The choice for the final set of trees is considered through a voting process and the intention is to result in a high-performing combination of items. Ideally, the decision tree outcome should be an improvement of each of the individual trees. The approach here combines different sets of tree depths and several estimators and considers their results for the best combination.

4.6.1 Results

```
Training Dataset
Accuracy =: 0.982555408866226
F1_Score =: 0.9798613403763619
AUR_ROC =: 0.9814738343310192

Validation Dataset
Accuracy =: 0.9895274539255656
F1_Score =: 0.15569437770302738
AUR_ROC =: 0.9603113320009167
```

Getting the confusion_matrix results of the RandomForest Classifier. This report indicates the TP,TN,FP,FN.

```
In [29]: print(confusion_matrix(target_valid, predictions_rf_valid))
[[165853   1745]
 [    12    162]]
```

The interpretation of the confusion matrix
Actual Values
TP = 165853
FP = 1745

Predicted Values
FN = 12
TN = 165

Getting the classification_report results of the RandomForest Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
In [30]: print(classification_report(target_valid,predictions_rf_valid, digits=3))
              precision    recall  f1-score   support

           0      1.000     0.990     0.995    167598
           1      0.085     0.931     0.156       174

    accuracy                          0.990    167772
   macro avg      0.542     0.960     0.575    167772
weighted avg      0.999     0.990     0.994    167772
```

Figure 16: Random Forest Classifier Results

The model has an accuracy of 98.3% accuracy, 15.6% F1-Score and an AUC-ROC of 96% against the validation dataset. A look at the precision shows that the model classifies at a chance of 1 for the correct prediction of legitimate transactions and a chance of 0.085 for fraud cases. The recall indicates a chance of 0.99 for the model to go through all the legitimate cases, while that of 0.931 to go through the fraud cases.

		Predicted	
		Positive (0)	**Negative (1)**
Actual	**Positive (0)**	165853	1745
	Negative (1)	12	162

Table 4: Random Forest Confusion Matrix

The random forest model correctly classified 165853 legitimate transactions, and 162 as fraud cases. However, the model incorrectly classified 1745 as false negatives and 12 as false positives. This shows a poor performance when it comes to the false negatives affecting its overall score.

- True Positives (TP): These are the cases in which we predicted positive, and they are positive
- True Negatives (TN): Predicted negative and they are negative
- False Positives (FP): Predicted positive but they are negative
- False Negatives (FN): Predicted negative but they are positive

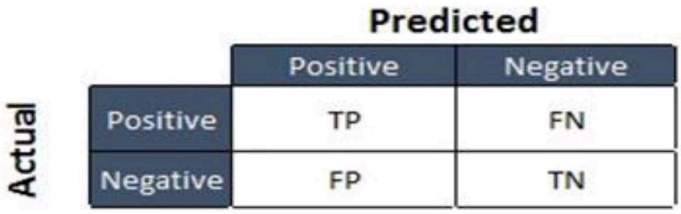


Figure 17: Confusion Matrix

- Accuracy: $\frac{(TP + TN)}{(TP+FP+FN+TN)}$, measures the number of correct predictions of total predictions as a percentage
- Precision: $\frac{TP}{(TP+FP)}$, counts the True positives cases out of the True positives plus the False positives
- Recall: $\frac{TP}{(TP+ FN)}$, measures how many true positives cases did the model succeeded in finding
- F1_Score: $2\ x\ \frac{(Precision\ x\ Recall)}{(Precision+Recall)}$, The mean of precision and recall
- Area Under the Curve _ Receiving Operators Characteristics (AUR_ROC):

ROC CURVE

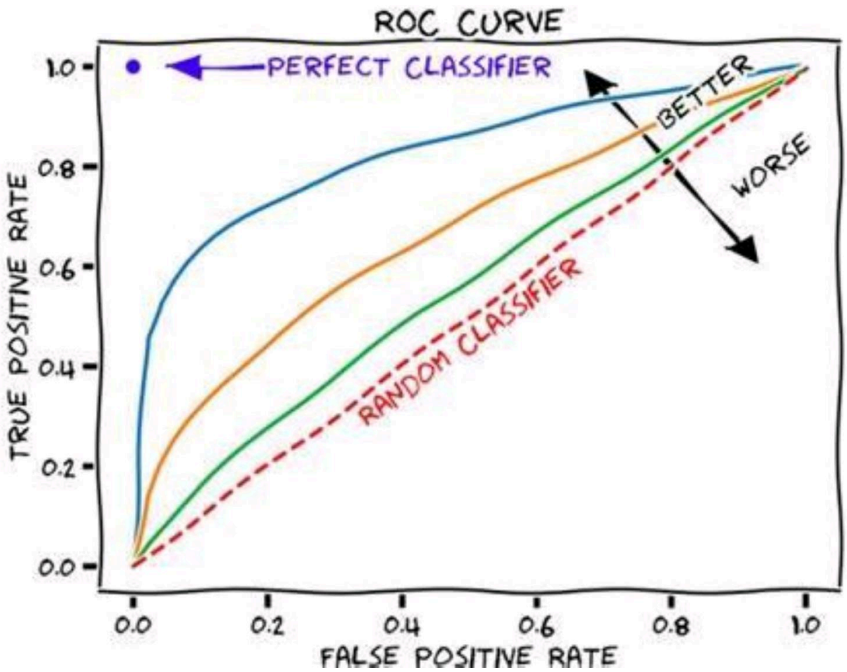


Figure 18: ROC curve

Chapter 5

5.1 Discussion

The results clearly place the CatBoost classification model at the top of the correct prediction rate followed by the Decision Tree. However, the random forest seems to perform poorly when placed against the rest of the models. A reason for this poor prediction between the two categories of fraud and non-fraud cases is the imbalance in the data. A consideration of the real-world implications of the results is to have a model that considers as many items as possible in both categories. As such, using the recall when needing to consider all options is a good way to weigh the models. Alternatively, for a generalized performance of the models, utilizing the F1 score can be an excellent way to consider whether or not the models performed well.

These models have performed quite well and can be utilized in categorizing real-world cases. In a dataset matching the information needed, the performance is expected to be high. The case for the class imbalance can raise issues in a few situations, which are solvable by expanding the dataset size. Providing more data to train on provides the models with better knowledge of fraud against non-fraud cases, thereby making them better prepared to handle new cases. The applicability of the models is high and can be utilized in detecting financial fraud in bank cases. Further tuning might also prove useful in improving the results of the models. However, tuning requires more time and resources to manually find the combination of parameters that works best with the provided datasets.

Implementing the models in the real world is a straightforward approach that takes the models, their parameters and trained state to classify new data. This is possible by wrapping it in a user interface and allowing the user to provide the necessary options for each transaction. The

information is then collected and used to classify the fraud and non-fraud cases. As the number of transactions grows, the models can keep getting better at predicting the fraud cases due to the provision of better consideration points of what distinguishes the two.

5.2 Concept of The Program

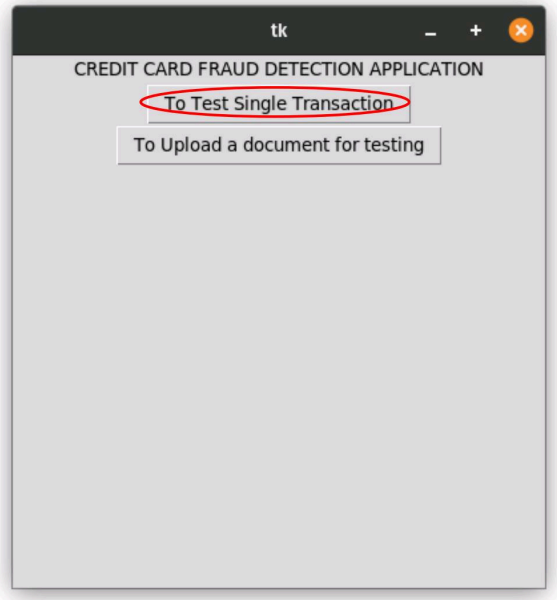


Figure 19: Program Home Page Interface

Basically, once the program runs. A platform interface output for the user to enter the required data. First, the user must choose whether the program will run for a single or multiple transaction as shown in figure 19.

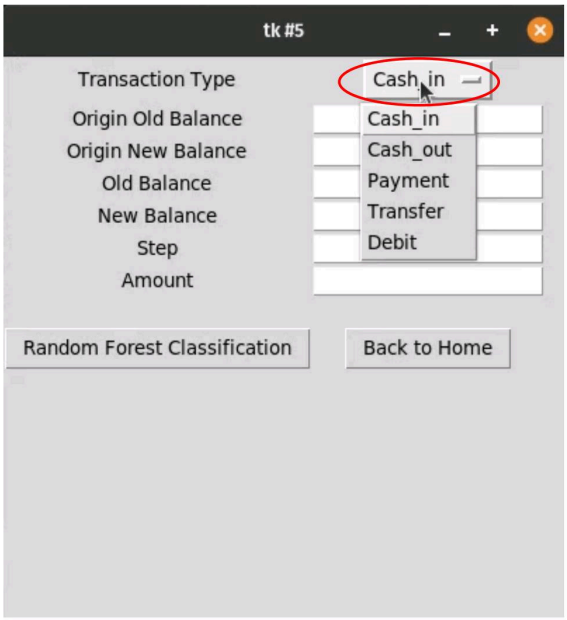


Figure 20: Types of Single Transaction Interface

If the user clicks on a single transaction figure 20 will be the output. Then, the user must choose the type of transaction whether it is cash in, cash out, payment, transfer, or debit.

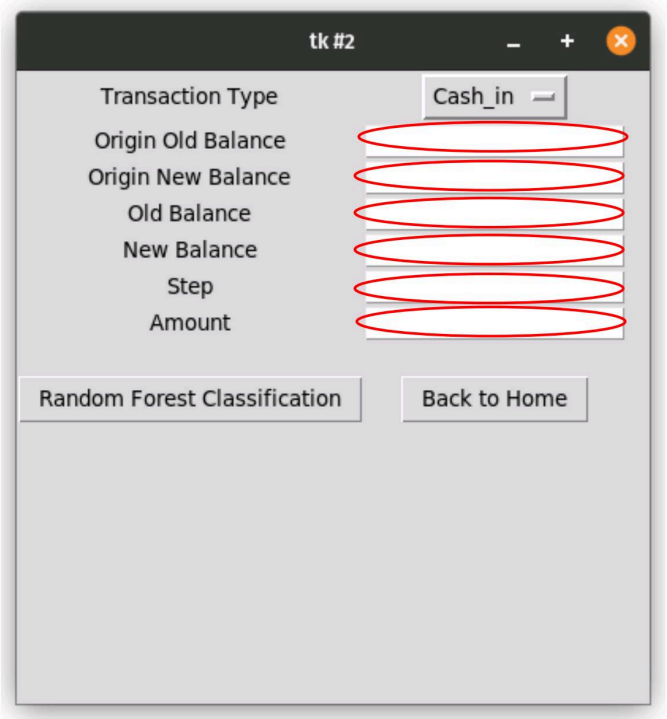


Figure 21: Type of Single Transaction Selected

After that, the user must enter several acquired data shown in figure 21.

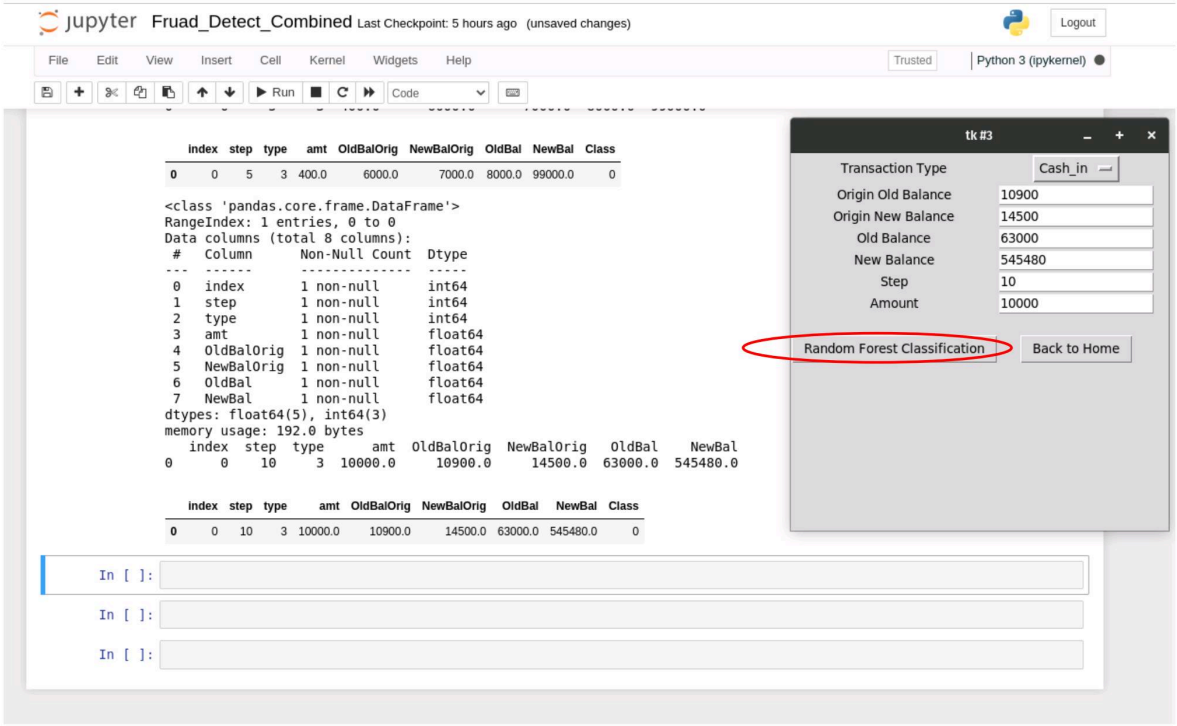


Figure 22: Single Transaction Process

Finally, the user has to click on the Random Forest Classification bottom to run the program.

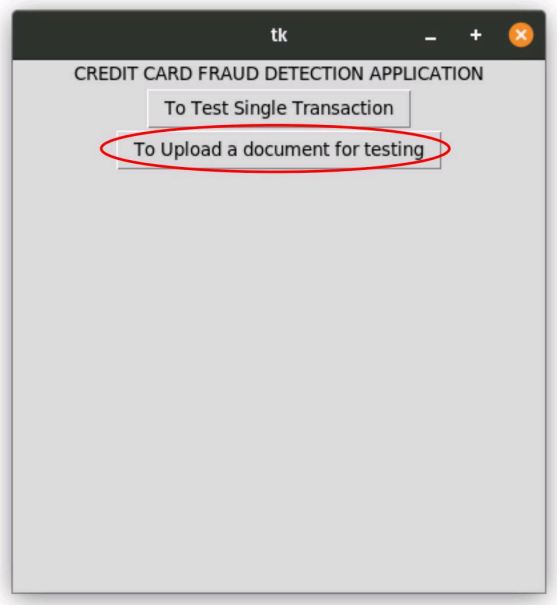


Figure 23: Program Home Page Interface

Other option is multiple transaction. The user must upload an excel file with all the required information in order to run the program.

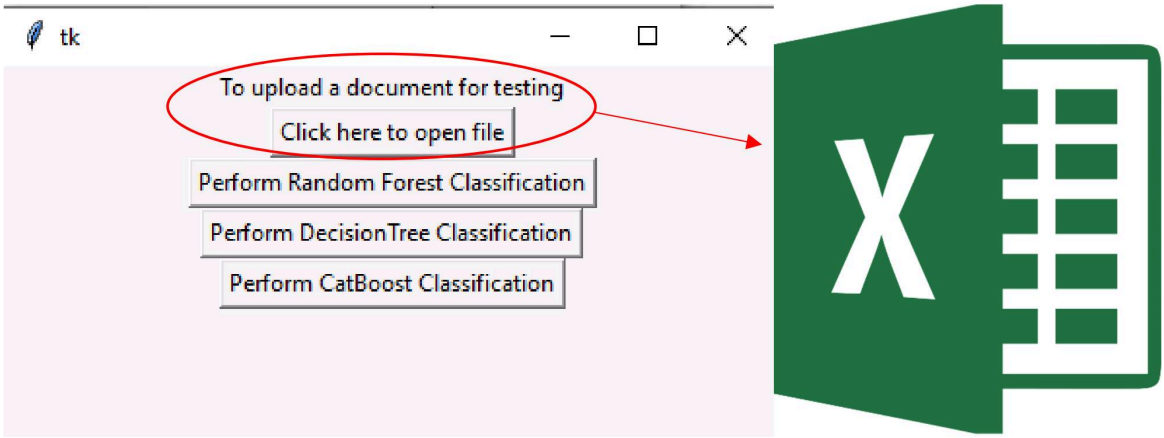


Figure 24: Multiple Transaction Interface

Once the file is uploaded, the user has the option to run one of the three models shown in figure 24. Then, the program will run the specified model on the uploaded file and return the file with the results of the transactions if they are ‘isFraud’ or ‘isFlaggedFraud’.

	A	B	C	D	E	F	G	H	I	J	K
1	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
2	1	PAYMENT	9839.64	C1231006815	170136	160296.36	M1979787155	0	0	0	0
3	1	PAYMENT	1864.28	C1666544295	21249	19384.72	M2044282225	0	0	0	0
4	1	TRANSFER	181	C1305486145	181	0	C553264065	0	0	1	0
5	1	CASH_OUT	181	C840083671	181	0	C38997010	21182	0	1	0
6	1	PAYMENT	11668.14	C2048537720	41554	29885.86	M1230701703	0	0	0	0
7	1	PAYMENT	7817.71	C90045638	53860	46042.29	M573487274	0	0	0	0
8	1	PAYMENT	7107.77	C154988899	183195	176087.23	M408069119	0	0	0	0
9	1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0	0	0	0
10	1	PAYMENT	4024.36	C1265012928	2671	0	M1176932104	0	0	0	0
11	1	DEBIT	5337.77	C712410124	41720	36382.23	C195600860	41898	40348.79	0	0
12	1	DEBIT	9644.94	C1900366749	4465	0	C997608398	10845	157982.12	0	0

Figure 25: The Uploaded Excel File

Figure 25 shows the excel file to be uploaded for multiple transactions.

J	K
isFraud	isFlaggedFraud
0	0
0	0
1	0
1	0
0	0
0	0
0	0

Figure 26: Results after running the Model

Figure 26 shows the results after running the program on the uploaded file. “1” refers to “yes” and “0” refers to “No”.

Chapter 6

6.1 Conclusion

The high accuracy results of the models are indicative of their readiness to use in a real-world setting. This performance means that the likelihood of a fraudulent case passing through is quite low. In combination with other information, an institution can further boost the outcome of the models. An example is the further scrutiny of the people or institutions performing the transactions. Suspicious combinations can be detected and examined on their legitimacy. Since such mechanisms already exist in most financial institutions, the work remains to integrate the models into the recommendations and warning systems to differentiate the cases. The exercise reveals the usefulness of modern technology and the capabilities one can have from utilizing it well. Having tools to quickly classify transactions is an excellent additional tool for a financial institution to have. Depending on the data they collect, the results can also drastically improve and present better outcomes. Additionally, further research is necessary to find more data points that distinguish fraud from non-fraud cases. The best model I would recommend is CatBoost Classifier, followed by Decision Tree and the last option would be Random Forest. This is because for highly imbalanced data accuracy cannot give reliable answers, so F_score is preferred. F_score can be separate according to the priority required. In the case of fraud detection, it is important to go through as most records as possible even if the rate of picking the fraud ones is low to avoid missing most. At least the process can be repeated but not to have a fraud case missing to be checked. That is why the highest recall is preferred.

6.2 Limitations

Caution needs to be taken when utilizing the model to predict the transactions. For instance, the remaining percentage in predicting fraud cases means that some of these transactions might pass through. The transactions might end up being classified as legitimate. Having that in mind reduces the expectations one should have on the models on a hundred percent accuracy in prediction. Further, it is also important to note that a different set of variables will need the models to be retrained, which means that a change in the information warrants an additional set of tests and analysis.

6.3 Recommendations

Adding more variables is a sure way of improving the results of the models. More variables provide better data to train the models and clearly provide distinctions among the fraud and non-fraud transactions. This is achievable by collecting more information on the nature of the transactions and feeding them to the models for the results.

An additional recommendation is to expand the size of the dataset. More data also constitutes the models being better equipped to better learn the nature of fraud and legitimate cases. Although not a guarantee of improved performance, it is the easiest and best place to start to determine whether the performance can be improved.

Using more models is also an approach that can be used to improve the results. More models provide different angles to look at the data and determine their performance. Despite more time needed to train and evaluate these models, the outcomes are worth it in terms of the knowledge of fraudulent case classification.

A combination of the ML models is also possible when implementing the case. These provide real-time results of how each one performed and allow the selection of the best or a set of the best to determine the classifications.

References

1. Attaran, M., & Gunasekaran, A. (2019). Blockchain-enabled technology: the emerging technology set to reshape and decentralise many industries. *International Journal of Applied Decision Sciences, 12*(4), 424-444.

2. Begenau, J., & Landvoigt, T. (2022). Financial regulation in a quantitative model of the modern banking system. *The Review of Economic Studies, 89*(4), 1748-1784.

3. Carminati, M., Polino, M., Continella, A., Lanzi, A., Maggi, F., & Zanero, S. (2018). Security evaluation of a banking fraud analysis system. *ACM Transactions on Privacy and Security (TOPS), 21*(3), 1-31.

4. Chen, Y., & Han, X. (2021). CatBoost for fraud detection in financial transactions. *In 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE)* (pp. 176-179). IEEE.

5. Dhieb, N., Ghazzai, H., Besbes, H., & Massoud, Y. (2020). A secure ai-driven architecture for automated insurance systems: Fraud detection and risk measurement. *IEEE Access, 8*, 58546-58558.

6. Dow, S. (2017). Central banking in the twenty-first century. *Cambridge Journal of Economics*.

7. Ermakova, E. P., & Frolova, E. E. (2019). Legal regulation of digital banking in Russia and foreign countries (European Union, USA, PRC). *Perm U. Herald Jurid. Sci., 46*, 606.

8. Gaol, F. L., Budiansa, A. D., Weniko, Y. P., & Matsuo, T. (2022). The Digital Fraud Risk Control on the Electronic-based Companies. *In Pervasive Computing and Social Networking* (pp. 741-758). Singapore: Springer.

9. Green, G. P. (2019). Rural banking. *Rural Policies for the 1990s*, 36-46.

10. Hajdari, E. (2021). The History and Origin of Fraud as a Defect in Consent in Contractual Relationships. *Brawijaya Law Journal: Journal of Legal Studies, 8*(1), 15-35.

11. Hancock, J. T., & Khoshgoftaar, T. M. (2020). CatBoost for big data: an interdisciplinary review. *Journal of big data, 7*(1), 1-45.

12. Hancock, J., & Khoshgoftaar, T. M. (2020). Medicare fraud detection using catboost. *In 2020 IEEE 21st international conference on information reuse and integration for data science (IRI)* (pp. 97-103). IEEE.

13. Ichinkhorloo, B. (2018). Collaboration for survival in the age of the market: diverse economic practices in postsocialist Mongolia. *Central Asian Survey, 37*(3), 386-402.

14. Karpoff, J. M. (2021). The future of financial fraud. *Journal of Corporate Finance, 66*, 101694.

15. Khatri, S., Arora, A., & Agrawal, A. P. (2020). Supervised machine learning algorithms for credit card fraud detection: a comparison. *In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)* (pp. 680-683). IEEE.

16. Lebichot, B., Verhelst, T., Le Borgne, Y. A., He-Guelton, L., Oblé, F., & Bontempi, G. (2021). Transfer learning strategies for credit card fraud detection. *IEEE access, 9*, 114754-114766.

17. Nguyen, N., Duong, T., Chau, T., Nguyen, V. H., Trinh, T., Tran, D., & Ho, T. (2022). A Proposed Model for Card Fraud Detection Based on CatBoost and Deep Neural Network.

18. Parsaee Tabar, A., Abdolvand, N., & Rajaee Harandi, S. (2021). Identifying the Suspected Cases of Money Laundering in Banking Using Multiple Attribute Decision Making (MADM). *Journal of Money and Economy, 16*(1), 1-20.

19. Psychoula, I., Gutmann, A., Mainali, P., Lee, S. H., Dunphy, P., & Petitcolas, F. (2021). Explainable machine learning for fraud detection. *Computer, 54*(10), 49-59.

20. Repousis, S., Lois, P., & Veli, V. (2019). An investigation of the fraud risk and fraud scheme methods in Greek commercial banks. Journal of Money Laundering Control.

21. Singla, A., & Jangir, H. (2020). A comparative approach to predictive analytics with machine learning for fraud detection of realtime financial data. *In 2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)* (pp. 1-4). IEEE.

22. Srikanth, P. (2021). An efficient approach for clustering and classification for fraud detection using bankruptcy data in IoT environment. *International Journal of Information Technology, 13*(6), 2497-2503.

23. Tapscott, A., & Tapscott, D. (2017). How blockchain is changing finance. *Harvard Business Review, 1*(9), 2-5.

24. Toms, S. (2019). Financial scandals: a historical overview. *Accounting and Business Research, 49*(5), 477-499.

25. Westermeier, C. (2020). Money is data–the platformization of financial transactions. *Information, Communication & Society, 23*(14), 2047-2063.

26. Follow me Ajitesh KumarI have been recently working in the area of Data analytics including Data Science and Machine Learning / Deep Learning. I am also passionate about different technologies including programming languages such as Java/JEE. (2020, September 2). *ROC Curve & AUC explained with python examples*. Data Analytics. Retrieved November 29, 2022, from https://vitalflux.com/roc-curve-auc-python-false-positive-true-positive-rate/.

Appendix

Code and Explanation

Python Code:

```
#pip install catboost

#pip install ipywidgets

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from catboost import CatBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import roc_auc_score
from sklearn.utils import shuffle
from sklearn.metrics import classification_report, confusion_matrix
import warnings

warnings.filterwarnings('ignore')
```

Importing the data.

```
df1 = pd.read_csv('C:\Users\khali\OneDrive\Desktop\Capstone Project\Fraud.csv')
```

Viewing the content of the dataset.

```
df1.head(5)
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	\
0	1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	
1	1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	
2	1	TRANSFER	181.00	C1305486145	181.0	0.00	
3	1	CASH_OUT	181.00	C840083671	181.0	0.00	
4	1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	

	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	M1979787155	0.0	0.0	0	0
1	M2044282225	0.0	0.0	0	0
2	C553264065	0.0	0.0	1	0
3	C38997010	21182.0	0.0	1	0
4	M1230701703	0.0	0.0	0	0

Viewing the datatypes and if there are any empty rows.

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 11 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
 0   step            1048575 non-null  int64
 1   type            1048575 non-null  object
 2   amount          1048575 non-null  float64
 3   nameOrig        1048575 non-null  object
 4   oldbalanceOrg   1048575 non-null  float64
 5   newbalanceOrig  1048575 non-null  float64
 6   nameDest        1048575 non-null  object
 7   oldbalanceDest  1048575 non-null  float64
 8   newbalanceDest  1048575 non-null  float64
 9   isFraud         1048575 non-null  int64
 10  isFlaggedFraud  1048575 non-null  int64
dtypes: float64(5), int64(3), object(3)
memory usage: 88.0+ MB
```

Checking the number of rows and columns in the data.

```
df1.shape
```

```
(1048575, 11)
```

Exploratory Analysis

Getting the number of the fraudulent and non-fraudulent cases in the dataset.

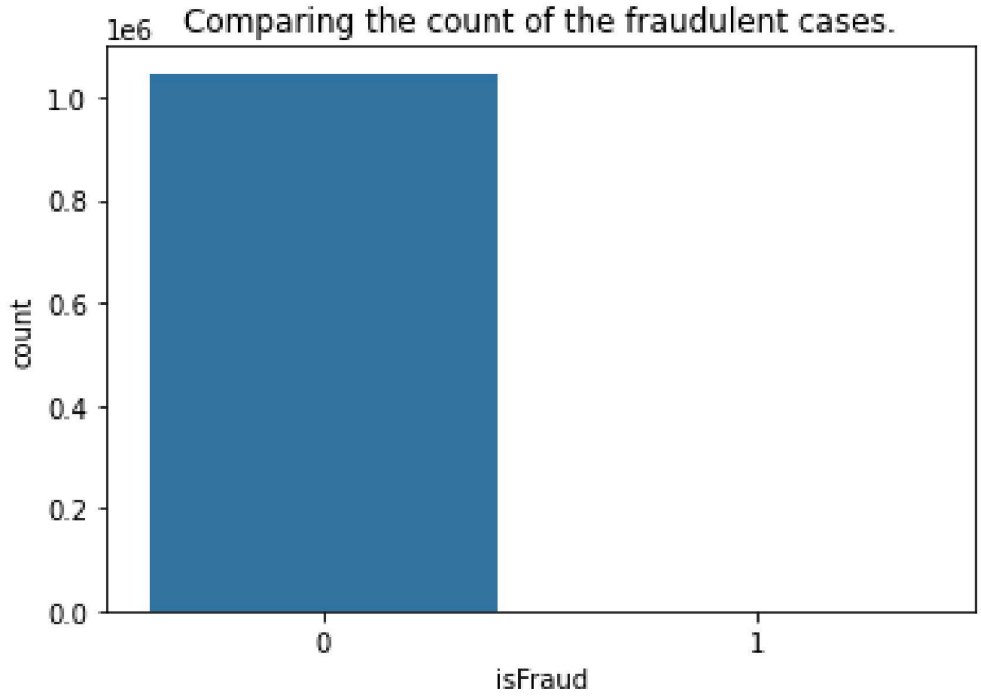
```
df1.isFraud.value_counts()
```

```
0    1047433
1       1142
Name: isFraud, dtype: int64
```

Plotting the fraudulent and fraudulent cases using countplot.

```
sns.countplot('isFraud', data=df1)
plt.title('Comparing the count of the fraudulent cases.')

plt.show()
```



This shows that there is class imbalance in the dataset.

Getting the number of cases represented by each type.

```
df1.type.value_counts()
```

```
CASH_OUT    373641
PAYMENT     353873
CASH_IN     227130
TRANSFER     86753
DEBIT         7178
Name: type, dtype: int64
```

Performing One-hot en-coding by replacing the types ith numeric numbers.

```
df1['type'].replace('CASH_OUT',1,inplace=True)
df1['type'].replace('PAYMENT',2,inplace=True)
df1['type'].replace('CASH_IN',3,inplace=True)
df1['type'].replace('TRANSFER',3,inplace=True)
df1['type'].replace('DEBIT',3,inplace=True)
```

Confirming that the changes have been made and the column is now a numerical column.

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1048575 entries, 0 to 1048574
Data columns (total 11 columns):
 #   Column          Non-Null Count    Dtype
---  ------          --------------    -----
```

```
 0   step            1048575 non-null  int64
 1   type            1048575 non-null  int64
 2   amount          1048575 non-null  float64
 3   nameOrig        1048575 non-null  object
 4   oldbalanceOrg   1048575 non-null  float64
 5   newbalanceOrig  1048575 non-null  float64
 6   nameDest        1048575 non-null  object
 7   oldbalanceDest  1048575 non-null  float64
 8   newbalanceDest  1048575 non-null  float64
 9   isFraud         1048575 non-null  int64
 10  isFlaggedFraud  1048575 non-null  int64
dtypes: float64(5), int64(4), object(2)
memory usage: 88.0+ MB
```

Viewing the updated changes on the column 'type'.

```
df1.head()
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	\
0	1	2	9839.64	C1231006815	170136.0	160296.36	
1	1	2	1864.28	C1666544295	21249.0	19384.72	
2	1	3	181.00	C1305486145	181.0	0.00	
3	1	1	181.00	C840083671	181.0	0.00	
4	1	2	11668.14	C2048537720	41554.0	29885.86	

	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	M1979787155	0.0	0.0	0	0
1	M2044282225	0.0	0.0	0	0
2	C553264065	0.0	0.0	1	0
3	C38997010	21182.0	0.0	1	0
4	M1230701703	0.0	0.0	0	0

Modelling

Splitting the dataset into target and features

```
# Split target and features
target = df1['isFraud']
features = df1.drop(['isFraud','nameOrig','nameDest'], axis = 1)

# Split into sets
features_train, features_temp, target_train, target_temp = train_test_split(features, target, test_size=0.2,ra
ndom_state=12345)
features_valid, features_test, target_valid, target_test = train_test_split(features_temp, target_temp, test_si
ze=0.2,random_state=12345)
```

Printing the length of each split set.

```
print(len(df1))
print(len(features_train))
print(len(features_valid))
print(len(features_test))
```

```
1048575
838860
167772
41943
```

Getting the length of the target column before upsampling to deal with the imbalance identified.

```
target.value_counts()
```

```
0    1047433
1       1142
Name: isFraud, dtype: int64
```

Using upsampling to balance the data.

```
def upsample(features, target, repeat):
    features_zeros = features[target == 0]
    features_ones = features[target == 1]
    target_zeros = target[target == 0]
    target_ones = target[target == 1]

    features_upsampled = pd.concat([features_zeros] + [features_ones] * repeat)
    target_upsampled = pd.concat([target_zeros] + [target_ones] * repeat)

    features_upsampled, target_upsampled = shuffle(
        features_upsampled, target_upsampled, random_state=12345)

    return features_upsampled, target_upsampled

features_upsampled, target_upsampled = upsample(features_train, target_train, 700)
```

Viewing the data after balancing the data.

```
target_upsampled.value_counts()
```

```
0    837934
1    648200
Name: isFraud, dtype: int64
```

Training the data using CatBoost Classifier model.

```
model_cat = CatBoostClassifier(iterations=500, random_seed=12345)

model_cat.fit(features_upsampled, target_upsampled, verbose=20)
```

```
Learning rate set to 0.439995
0:      learn: 0.2241539        total: 599ms    remaining: 4m 58s
20:     learn: 0.0227540        total: 9.41s    remaining: 3m 34s
40:     learn: 0.0098414        total: 19.1s    remaining: 3m 33s
60:     learn: 0.0050778        total: 28.3s    remaining: 3m 23s
80:     learn: 0.0031123        total: 38s      remaining: 3m 16s
100:    learn: 0.0021445        total: 46.4s    remaining: 3m 3s
120:    learn: 0.0018882        total: 54.2s    remaining: 2m 49s
```

```
140:    learn: 0.0015339        total: 1m 2s    remaining: 2m 38s
160:    learn: 0.0014795        total: 1m 9s    remaining: 2m 26s
180:    learn: 0.0014500        total: 1m 16s   remaining: 2m 14s
200:    learn: 0.0014490        total: 1m 22s   remaining: 2m 3s
220:    learn: 0.0014482        total: 1m 30s   remaining: 1m 54s
240:    learn: 0.0014469        total: 1m 37s   remaining: 1m 44s
260:    learn: 0.0014462        total: 1m 44s   remaining: 1m 36s
280:    learn: 0.0014460        total: 1m 51s   remaining: 1m 27s
300:    learn: 0.0014274        total: 1m 59s   remaining: 1m 18s
320:    learn: 0.0013231        total: 2m 6s    remaining: 1m 10s
340:    learn: 0.0012902        total: 2m 13s   remaining: 1m 2s
360:    learn: 0.0012637        total: 2m 20s   remaining: 54.3s
380:    learn: 0.0011777        total: 2m 28s   remaining: 46.5s
400:    learn: 0.0011527        total: 2m 36s   remaining: 38.6s
420:    learn: 0.0011331        total: 2m 44s   remaining: 30.8s
440:    learn: 0.0011328        total: 2m 51s   remaining: 22.9s
460:    learn: 0.0011322        total: 2m 59s   remaining: 15.1s
480:    learn: 0.0011312        total: 3m 5s    remaining: 7.35s
499:    learn: 0.0011304        total: 3m 13s   remaining: 0us
```

<catboost.core.CatBoostClassifier at 0x23e820e8d00>

Getting the metrics results of CatBoostClassifier.

```
predictions_cat_train = model_cat.predict(features_upsampled)
accuracy_cat = accuracy_score(target_upsampled, predictions_cat_train)
F1_cat = f1_score(target_upsampled, predictions_cat_train)
auc_roc_cat = roc_auc_score(target_upsampled, predictions_cat_train)

predictions_cat_valid = model_cat.predict(features_valid)
accuracy_cat_val = accuracy_score(target_valid, predictions_cat_valid)
F1_cat_val = f1_score(target_valid, predictions_cat_valid)
auc_roc_cat_val = roc_auc_score(target_valid,predictions_cat_valid)


print("Training Dataset")
print("Accuracy =:", accuracy_cat)
print("F1_Score =:", F1_cat)
print("AUR_ROC =:", auc_roc_cat)
print('')
print("Validation Dataset")
print("Accuracy =:", accuracy_cat_val)
print("F1_Score =:", F1_cat_val)
print("AUR_ROC =:", auc_roc_cat_val)
```

```
Training Dataset
Accuracy =: 0.999762470948111
F1_Score =: 0.9997277816207096
AUR_ROC =: 0.9997893628853824

Validation Dataset
Accuracy =: 0.9993383878120307
```

F1_Score =: 0.7286063569682152
AUR_ROC =: 0.927904353232756

Getting the confusion_matrix results of the CatBoost Classifier. This report indicates the TP,TN,FP,FN.

```
print(confusion_matrix(target_valid, predictions_cat_valid))
```

```
[[167512     86]
 [    25    149]]
```

Getting the classification_report results of the CatBoost Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
print(classification_report(target_valid,predictions_cat_valid, digits=3))
```

```
              precision    recall  f1-score   support

           0      1.000     0.999     1.000    167598
           1      0.634     0.856     0.729       174

    accuracy                          0.999    167772
   macro avg      0.817     0.928     0.864    167772
weighted avg      0.999     0.999     0.999    167772
```

The chances of the CatBoost model prediting correctly is 1 for the non-fraud cases and 0.634 for the fraud cases. The chances going through every record is 0.999 for the non-fraud cases and 0.856 for the fraud cases.

Training the model using DecisionTree Classifier.

```
#for max in range(1,5):
model_dt = DecisionTreeClassifier(random_state=54321, class_weight=None)

model_dt.fit(features_upsampled, target_upsampled)

predictions_dt_train = model_dt.predict(features_upsampled)
accuracy_dt = accuracy_score(target_upsampled, predictions_dt_train)
F1_dt = f1_score(target_upsampled, predictions_dt_train)
auc_roc_dt = roc_auc_score(target_upsampled, predictions_dt_train)

predictions_dt_valid = model_dt.predict(features_valid)
accuracy_dt_val = accuracy_score(target_valid, predictions_dt_valid)
F1_dt_val = f1_score(target_valid, predictions_dt_valid)
auc_roc_dt_val = roc_auc_score(target_valid,predictions_dt_valid)


print("Training Dataset")
print("Accuracy =:", accuracy_dt)
print("F1_Score =:", F1_dt)
```

```
print("AUR_ROC =:", auc_roc_dt)
print('')
print("Validation Dataset")
print("Accuracy =:", accuracy_dt_val)
print("F1_Score =:", F1_dt_val)
print("AUR_ROC =:", auc_roc_dt_val)
```

```
Training Dataset
Accuracy =: 1.0
F1_Score =: 1.0
AUR_ROC =: 1.0

Validation Dataset
Accuracy =: 0.9996602532007725
F1_Score =: 0.8256880733944953
AUR_ROC =: 0.8878773345579385
```

Getting the confusion_matrix results of the DecisionTree Classifier. This report indicates the TP,TN,FP,FN.

```
print(confusion_matrix(target_valid, predictions_dt_valid))
```

```
[[167580     18]
 [    39    135]]
```

The interpretation of the confusion matrix. Actual Values TP = 167580 FP = 18 Predicted Values FN = 39 TN = 135

Getting the classification_report results of the DecisionTree Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
print(classification_report(target_valid,predictions_dt_valid, digits=3))
```

```
              precision    recall  f1-score   support

           0      1.000     1.000     1.000    167598
           1      0.882     0.776     0.826       174

    accuracy                          1.000    167772
   macro avg      0.941     0.888     0.913    167772
weighted avg      1.000     1.000     1.000    167772
```

The chances of the DecisionTree model prediting correctly is 1 for the non-fraud cases and 0.882 for the fraud cases. The chances going through every record is 1 for the non-fraud cases and 0.776 for the fraud cases.

Training the model using RandomForest Classifier.

```
for estim in range(5,15,5):
    for depth in range(1,10):
```

```
        model = RandomForestClassifier(random_state=12345,class_weight="balanced", n_estimators=esti
m, max_depth=depth)

        predictions = pd.Series(target.mean(), index=target.index)

        model.fit(features_upsampled, target_upsampled)
        predictions_train = model.predict(features_upsampled)
        predictions_rf_valid = model.predict(features_valid)
        predictions_test = model.predict(features_test)
        auc_roc = roc_auc_score(target_valid,predictions_rf_valid)
        print('Max Depth ' + str(depth)+': ' + str(auc_roc) )
        print()
        print("n_estimators =" + str(estim)+': ' + str(auc_roc) )
```

```
Max Depth 1: 0.8101317767350528

n_estimators =5: 0.8101317767350528
Max Depth 2: 0.8319313743765356

n_estimators =5: 0.8319313743765356
Max Depth 3: 0.9129477925627456

n_estimators =5: 0.9129477925627456
Max Depth 4: 0.9277624907876854

n_estimators =5: 0.9277624907876854
Max Depth 5: 0.9443420853923449

n_estimators =5: 0.9443420853923449
Max Depth 6: 0.959952200894505

n_estimators =5: 0.959952200894505
Max Depth 7: 0.9742567498336536

n_estimators =5: 0.9742567498336536
Max Depth 8: 0.9576538029628369

n_estimators =5: 0.9576538029628369
Max Depth 9: 0.9576287018485532

n_estimators =5: 0.9576287018485532
Max Depth 1: 0.7722262137108871

n_estimators =10: 0.7722262137108871
Max Depth 2: 0.8405627628673045

n_estimators =10: 0.8405627628673045
Max Depth 3: 0.9030173185343746

n_estimators =10: 0.9030173185343746
```

```
Max Depth 4: 0.927765988483938

n_estimators =10: 0.927765988483938
Max Depth 5: 0.9516882762571028

n_estimators =10: 0.9516882762571028
Max Depth 6: 0.9620011307846238

n_estimators =10: 0.9620011307846238
Max Depth 7: 0.9567021209618582

n_estimators =10: 0.9567021209618582
Max Depth 8: 0.9567343203420664

n_estimators =10: 0.9567343203420664
Max Depth 9: 0.9554383210070403

n_estimators =10: 0.9554383210070403
```

Ftting the results after tuning the model.

```
model.fit(features_upsampled, target_upsampled)
#class_weight='balanced'
model_rf = RandomForestClassifier(n_estimators=5, random_state=12345, max_depth=8)
model_rf.fit(features_upsampled, target_upsampled)

predictions_train = model.predict(features_upsampled)
accuracy = accuracy_score(target_upsampled, predictions_train)
F1 = f1_score(target_upsampled, predictions_train)
auc_roc = roc_auc_score(target_upsampled, predictions_train)

predictions_rf_valid = model.predict(features_valid)
accuracy_val = accuracy_score(target_valid, predictions_rf_valid)
F1_val = f1_score(target_valid, predictions_rf_valid)
auc_roc_val = roc_auc_score(target_valid,predictions_rf_valid)

print("Training Dataset")
print("Accuracy =:", accuracy)
print("F1_Score =:", F1)
print("AUR_ROC =:", auc_roc)
print('')
print("Validation Dataset")
print("Accuracy =:", accuracy_val)
print("F1_Score =:", F1_val)
print("AUR_ROC =:", auc_roc_val)
```

```
Training Dataset
Accuracy =: 0.982555408866226
F1_Score =: 0.9798613403763619
AUR_ROC =: 0.9814738343310192
```

Validation Dataset
Accuracy =: 0.9895274539255656
F1_Score =: 0.15569437770302738
AUR_ROC =: 0.9603113320009167

Getting the confusion_matrix results of the RandomForest Classifier. This report indicates the TP,TN,FP,FN.

```
print(confusion_matrix(target_valid, predictions_rf_valid))
```

```
[[165853   1745]
 [    12    162]]
```

The interpretation of the confusion matrix Actual Values TP = 165853 FP = 1745

Predicted Values FN = 12 TN = 165

Getting the classification_report results of the RandomForest Classifier. This report indicates the precision- the rate of the model getting a correct answer. recall - the chances that the model will check every record. F_score is precision divided by recall.

```
print(classification_report(target_valid,predictions_rf_valid, digits=3))
```

```
              precision    recall  f1-score   support

           0      1.000     0.990     0.995    167598
           1      0.085     0.931     0.156       174

    accuracy                          0.990    167772
   macro avg      0.542     0.960     0.575    167772
weighted avg      0.999     0.990     0.994    167772
```

The chances of the RandomForest model predicting correctly is 1 for the non-fraud cases and 0.085 for the fraud cases. The chances going through every record is 0.990 for the non-fraud cases and 0.931 for the fraud cases.

```
model_rf = RandomForestClassifier(n_estimators=5, random_state=12345, max_depth=8)
model_rf.fit(features_upsampled, target_upsampled)

# Here are the predictions
y_hats = model.predict(features_test)

features_test['isFraud'] = y_hats

df.rename(columns = {'isFraud':'Class'}, inplace = True)

df.replace({'Class':{0:'Legit', 1:'Fraud'}})

df.to_csv('C:\Users\khali\OneDrive\Desktop\Capstone Project\Results', index=False)
```

R Studio Code:

```{r }
##installing libraries

library(caret)
library(rpart)
library(dplyr)
library(tidyverse)

##Data Exploration

view(fraud)

names(fraud)

## will return the columns in the dataset
str(fraud)

## will return the structure of the dataset
head(fraud)

## Checking whether there is NA values in the dataset
summary(fraud)

## Seperating Fraud from non fraud cases

fraud.true = fraud[fraud$isFraud == 0,]
fraud.false = fraud[fraud$isFraud == 1,]
```

```
isflag.true = fraud[fraud$isFlaggedFraud == 0,]
isflag.false = fraud[fraud$isFlaggedFraud == 1,]

## Density Plot for the "isFraud" vs "Step" Cases

 ggplot() + geom_density(data = fraud.true,aes(x = step), color= "blue", fill= "blue", alpha= 0.1) +

    geom_density(data = fraud.false, aes(x = step), color= "red", fill="red", alpha= 0.1)

## Density Plot for the "isFlaggedFraud" vs "Step" Cases

 ggplot() + geom_density(data = isflag.true,aes(x = step), color= "blue", fill= "blue", alpha= 0.1) +

    geom_density(data = isflag.false, aes(x = step), color= "red", fill="red", alpha= 0.1)


 ## Density Plot Fraud vs Amount

 ggplot() + geom_density(data = fraud.true,aes(x = amount), color= "blue", fill= "blue", alpha= 0.1) +

    geom_density(data = fraud.false, aes(x = amount), color= "red", fill="red", alpha= 0.1)
```