Division of Computing Science and Mathematics

Faculty of Natural Sciences

University of Stirling

Development of Business Intelligence Outlier and financial crime analytics system for predicting and managing fraud in financial payment services

Guneet Kaur

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Financial Technology (FinTech)

September 2019

Abstract

Each year organizations lose 5% of revenues to fraud as highlighted in 2014 ACFE (Association of certified fraud examiners) report resulting into nearly $3.7 trillion global fraud loss if applied to 2013 estimated Gross World Product. For the year ending March 2016, 5.8 million incidents of computer misuse and fraud were reported in the Crime Survey of England and Wales (CSEW) wherein the victims were adults aged 16. Similarly, the total losses from online payment fraud for this year are estimated to be $22 billion and could go as high as $48 billion according to a study conducted by Juniper Research. Moreover, as money has evolved over time, financial payment services have a great role to play in moving money around the economy. Therefore, financial institutions need to adapt, to build brand loyalty among consumers who have more options than before to satisfy their financial needs by delivering a safe and seamless user experience. In this data-driven world, one of the ways to tackle this problem is through the application of machine learning techniques in the area of fraud detection. Thus, this project will investigate into the data mining based approach to predict and manage fraud in financial payment services.

This project aims to critically analyze the data of transactions (Cash-In, Cash-Out, Debit, Payment and Transfer) which consists of both normal as well as fraudulent customer behavior in order to build a Classification model that accurately categorize the transactions into fraudulent and genuine categories in the presence of imbalanced data. Consequently, deploying machine learning model as a tool to predict and detect fraud will improve the risk assessment capabilities of the organizations by dynamically conducting fund-flow analytics in real-time which further improve their reputation and customer loyalty.

The two statistical and two ensemble machine learning techniques such as Logistic Regression, Naïve Bayes, Random Forest and Extreme Gradient Boosting algorithms have been experimented with to find the best machine learning algorithm for the problem in question. These techniques have been explained in detail in this project along with detailed data exploration and feature engineering to aid fraud detection research in the field of financial payment services. The results achieved by exploring these supervised classification models were then presented and evaluated using test data on all type of transactions as well as using test data only on fraudulent transactions to conclude with a model which is a best classifier of accurately predicting and classifying the transactions to fraudulent and genuine ones.

The final model suggested to deploy is XGBoost classifier with 96.46% accuracy and low false positives as well as low false negatives which means that the model is capable enough not to treat most of the genuine transactions as fraudulent and real threats won't be missed out in the form of mistreated fraudulent cases. Furthermore, future work has been discussed followed by limitations and challenges in conducting this study to improve model accuracy in the future.

Attestation

I understand the nature of plagiarism, and I am aware of the University's policy on this.

I certify that this dissertation reports original work by me during my University project except for the following:

- Development of a financial mobile money simulator for recording payment transactions (section 3.2)
- Generation of dataset from simulator. **(**I have obtained it from https://www.kaggle.com/ntnu-testimon/paysim1)
- Binary encoding of isfraud and isFlaggedFraud columns in the dataset (section 4.2.2).

Acknowledgements

Foremost, I would like to thanks Almighty for giving me the strength to undertake and successfully complete the most significant academic challenge of writing dissertation.

I also acknowledge the immense support and unconditional love of my parents for supporting me spiritually during my MSc Financial Technology (FinTech) degree at the University of Stirling as well as throughout my life

I would also like to express my sincere gratitude to my supervisor Dr. Sandy Brownlee for his continuous support during the dissertation period, for his patience, enthusiasm, motivation and immense knowledge about the subject matter. His guidance helped me in all the time of research and writing this piece of work. From suggesting the good sources to improve my knowledge about my project to assisting me for future career prospects, he has been a great mentor to me. I could not have imagined a better supervisor than him.

Besides my supervisor, I would also like to thank Dr. Andrea Bracciali (programme director of MSc FinTech) for his encouragement and support to work on the exciting project and Dr. Simon Jones for making the project material available without any delay and for timely updating with all the necessary information related to the project work for the successfully completing the project on time.

Last but not the least; I would like to thank my friend Pouria Modaresi for being very kind and generous with me, always believing in me and bringing out the best in me.

Table of Contents

Abstract 2

Attestation 3

Acknowledgements 4

Table of Contents 5

List of Figures 8

List of Tables 10

1. Introduction 11

    1.1 Background and Context 11

    1.2 Scope and Objectives 14

        1.2.1 Objectives 15

    1.3 Achievements of the Dissertation 16

    1.4 Overview of the dissertation 16

2. Literature Review 17

    2.1 Introduction 17

    2.2 Previous Related Work 18

    2.3 Comparison and critical analysis of previous related work in relation to the present scenario 21

        2.3.1 Lack of research in fraud detection in financial payment services 22

3. Methodology 23

    3.1 Data Mining Framework: CRISP-DM 23

    3.2 PaySim: A financial mobile money simulator for fraud detection research 26

3.2.1 Introduction to PaySim 26
3.2.2 Scenarios and Generation of Synthetic Data 27
4. Data Understanding and Preparation 28
4.1 Data Description 28
4.2 Data Preparation and Exploration 32
4.2.1 Exploratory Data Analysis 32
4.2.2 Data Cleaning: Transforming messy data to tidy data 35
4.2.3 Feature Engineering 36
4.2.4 Data Visualization 38
5. Modelling: Teaching an Algorithm 42
5.1 Machine Learning Framework 43
5.1.1 Machine Learning for Fraud Detection 44
5.1.2 Machine Learning approaches for Fraud Detection 45
5.1.3 Towards Supervised Anomaly Detection 46
5.2 Python for fraud detection 47
5.3 Machine Learning Techniques 47
5.3.1 Disadvantages of handling Imbalanced Data using Sampling Methods 48
5.4 Statistical Techniques 49
5.4.1 Logistic Regression 49
5.4.2 Naïve Bayes 50
5.5 Ensemble Modelling Techniques 51
5.5.1 Random Forest 52
5.5.2 XGBoost Classifier 53

6. Model Training, Tuning and Results 55
6.1 Train and Test Split: 80:20 55
6.2 K-fold Cross Validation 55
6.3 Feature Selection 56
6.4 Hyperparameter Optimization 57
6.4.1 Statistical Models 58
6.4.2 Ensemble Models 60
6.5 Model Comparison and Selection of ML Algorithm 64
7. Evaluation: Measuring Fraud Detection Performance 65
7.1 Performance Metrics 65
7.1.1 Confusion Matrix 65
7.2 Evaluation of trained models 68
7.2.1 When trained models were tested on 20% test data consisting of all type of transactions 68
7.3 Analysis of confusion matrix results 71
7.3.1 When trained models were tested on 20% test data consisting only fraudulent transactions i.e. CASH_OUTs and TRANSFERs 73
7.4 Challenges and Limitations 76
8. Model Deployment: Conclusion 76
8.1 Summary 76
8.2 Evaluation 77
8.3 Future Work 77
References 79
Appendices 87

List of Figures

Figure 1: Rising fraud in payment services 12
Figure 2: Payment fraud in financial services industry 13
Figure 3: Supervised v/s unsupervised learning for fraud detection 15
Figure 4: Visual representation of Literature of review from year 2002 to 2019. 21
Figure 5: Steps involved in CRISP-DM process 24
Figure 6: Steps involved in Data Preparation stage 25
Figure 7: Use-case representation of PaySim Simulator 27
Figure 8: Statistical information about the Dataframe. 31
Figure 9: Graphical Representation of transactions 33
Figure 10: Adding new features to record errors in originating and destination accounts for each transaction. 36
Figure 11: Valid transactions over time 37
Figure 12: Fraudulent transactions over time. 37
Figure 13: Fraudulent and Valid transactions by Day and Hour 38
Figure 14: Striped and homogenous distribution over time 39
Figure 15: Same-signed fingerprints when looked over amount 40
Figure 16: Opposite polarity fingerprints over error in balance in destination accounts 40
Figure 17: Separating out genuine and fraudulent transactions using error based engineered features 41
Figure 18: Correlation heatmap of genuine and fraudulent transactions 42
Figure 19: Some examples of widely publicized machine learning applications. 43
Figure 20: A General machine learning framework 44
Figure 21: Supervised Machine Learning approach for fraud detection. 46
Figure 22: Unsupervised Machine Learning approach for fraud detection. 46
Figure 23: Hypothesis expectation in Logistic Regression 49
Figure 24: Steps involved in constructing the tree using Random Forest. 53
Figure 25: Train-test split approach 55
Figure 26: K-fold cross validation approach when K=5 56
Figure 27: Feature Importance Bar Chart 57
Figure 28: Four outcomes of a classifier in classification problem. 66

Figure 29: Error Rate in confusion matrix 66
Figure 30: Accuracy in confusion matrix 67
Figure 31: Area under ROC for Logistic Regression (left) and Naïve Bayes (right)..... 72
Figure 32: Area under ROC for Random Forest (left) and XGBoost classifier (right). 73
Figure 33: Confusion Matrix analysis of fraudulent transactions on test data using statistical methods..... 74
Figure 34: Confusion Matrix analysis of fraudulent transactions on test data using ensemble methods 74

List of Tables

Table 1: Using hybrid approach for detecting fraud in financial payment services. 15
Table 2: Scale and Scope of Research Problem. 23
Table 3: Dataset Details 29
Table 4: Definitions of columns as defined in the chosen dataset. 30
Table 5: Tabular representation of the attributes 31
Table 6: PaySim dataset statistics 32
Table 7: Comparison between Traditional rule-based V/S Machine Learning based fraud detection 45
Table 8: Hyperparameter Tuning for Logistic Regression 58
Table 9: Model Results for Logistic Regression 59
Table 10: Hyperparameter Tuning for Naïve Bayes Classifier 60
Table 11: Model Results for Naïve Bayes 60
Table 12: Hyperparameter Tuning for Random Forest 61
Table 13: Model Results for Random Forest 62
Table 14: Hyperparameter Tuning for XGBoost Classifier 63
Table 15: Model Results for XGBoost Classifier 64
Table 16: Summary of Model Performances 65
Table 17: Confusion matrix for Logistic Regression model 68
Table 18: Confusion matrix for Naïve Bayes model 69
Table 19: Confusion matrix for Random Forest model 70
Table 20: Confusion matrix for XGBoost model 70
Table 21: Evaluation metrics for Machine Learning models when all transactions trained on test data 71
Table 22: Evaluation metrics for Machine Learning models when only fraudulent transactions trained on test data 75

1. Introduction

1.1 Background and Context

In recent years, financial crime has become a great deal of attention and concern in lieu of what is it, how it occurs and why (Gottschalk, 2010) it is happening on a frequent basis which further stimulates that there is need to work on know-what, know-how and know-why in context of the increasing financial crime. When the true nature of the activities is concealed and deceived for illegal gain, normally involving breach of trust often termed out as financial crime (Pickett, Pickett, 2002). In other words, an unauthorized access and control over someone else's property for financial gain is known as financial crime. The terms such as financial crime, fraud or white-collar crime have been often used interchangeably in the financial world (Gottschalk, 2010). However, according to the Oxford English Dictionary, 'fraud means wrongful or criminal deception intended to result in financial or personal gain' (Ngai, Hu, et. al., 2011). In the financial sector, fraud affects all the key stakeholders such as the organizations, the individuals and nations (Rojas, Axelsson, 2016).

Although fraud is not a new issue as revealed by the Global Financial Crisis of 2007-08 which further portrays that the fraud mainly occurs during period of recession as compared with other normal periods of economic growth (Bănărescu, 2015). The dark side of the financial services came into picture through various frauds like Ponzi schemes perpetuated by Bernard Madoff (former NASDAQ Chairman) which led to the loss of US$50 billion worldwide. Hence, fraud detection is vital for preventing the hazardous consequences of financial fraud by minimizing the effects of unauthorized transactions upon:

- An organization's capability in terms of customer service delivery.
- Business reputation.
- Bottom line expenditure.

The extent to which a victim is impacted by the financial crime mainly depends upon various factors such as the type of crime, the amount of assets stolen and degree of trust compromised during the whole act (Deem, D.L., 2000). The financial crime can take any form like corruption, fraud, theft and manipulation. Corruption happens when improper advantage of power, office, position or assignment is taken by the fraudsters for instance, bribery. Fraud took place with unlawful or unfair gain to deprive a victim of a legal right while theft involves a forceful act to steal assets or property from the victim. Manipulation occurs when false or misleading data is provided to gain unauthorized access such as misappropriation schemes.

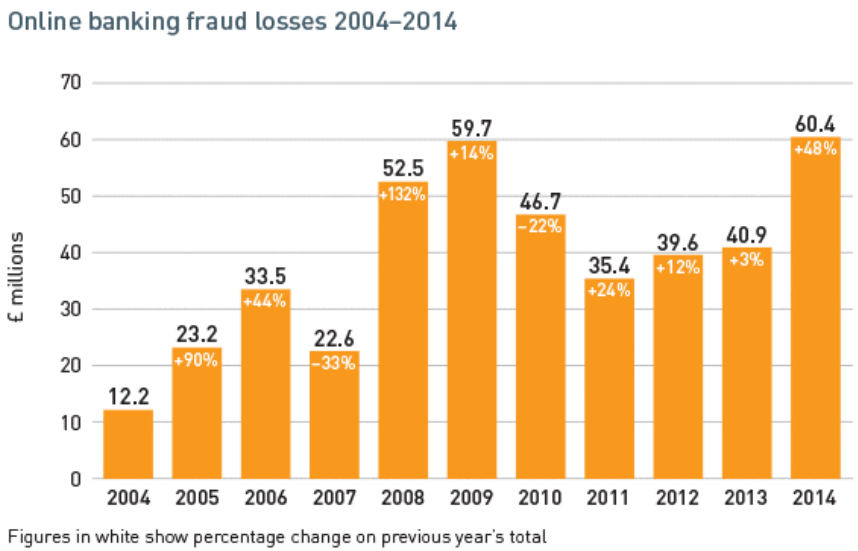

Figure 1: Rising fraud in payment services

Source: https://www.hideiptips.com/is-online-banking-safe-fraud-scam-vpn

Furthermore, the various categories of financial fraud are outlined as follows (Bangs, 2018):

1. **Bank and credit card Fraud**: It involves unauthorized ways of obtaining fund, money, assets, credits, securities from a bank, building society or any other financial institution such as money laundering, mortgage fraud and fraudulent use of plastic card details (payment fraud which is also an adjunct to identity theft).
2. **Advance fee Fraud**: It occurs during when a victim has received a communication soliciting money such as lottery scams, inheritance fraud et al.
3. **Non-Investment Fraud**: When fraudsters conned a victim to make a purchase that is subsequently found to be fraudulent in nature such as phone scams, online shopping, ticketing fraud, bogus callers and computer service fraud.
4. **Other related financial fraud**: It involves all type of record which is not recorded elsewhere such as Securities fraud, Insurance fraud, mass- marketing, corporate fraud, fake prizes, Investment fraud et al.

This project investigates into the Bank and credit card fraud category in general and in depth analysis of fraud in financial payment services specifically. Financial payment services have a vital role in moving money around the economy. Since ancient times many ways have been developed to transfer value between two or more people from barter system of exchange to precious metals (like gold), to paper based money

and now electronic value transfer systems (Bollen, 2010). The concept of money and payment are inter-related and has evolved over time. One of the emerging way provided by financial companies to facilitate commercial exchange between two or more parties is mobile money service i.e. using mobile phones to send and receive funds. But such services (card-based mobile money payment model), along with being an efficient tool for bank transactions, also poses a great deal of threat in terms of financial fraud both for the service providers and the customers.

Furthermore, Payment card fraud (Debit, Credit and Charge cards) falls into two categories: *Application Fraud* and *Behavioral Fraud (Bolton, Hand, 2001)*. In application fraud, fraudsters use false information to obtain new cards from financial institutions whereas in behavioral Fraud card details of a victim have been obtained fraudulently to made sales on a 'Cardholder Not present' basis. Also, fraud in financial payment services can take any of the forms (Sakharova, 2012) (Appendix1). The financial losses from payment card fraud totaled $21.84 billion in 2015, $24.71 billion in 2016 and $27.69 billion in 2017 and the actual amount of losses will further increase by 2020 (Robertson, 2017).

Figure 2: Payment fraud in financial services industry

Source: https://www.paymentssource.com/opinion/as-payment-fraud-spikes-involving-customers-can-aid-prevention

This problem highlights the need for the financial institutions especially banks to focus on developing an effective fraud detection system to reduce the damages caused by the fraud as it counts for million dollar business worldwide. Thus, this project aims to develop a business intelligence system to predict and manage the fraud in financial payment services to reduce the impact of financial crime on all the key stakeholders in the digital banking age.

Business Intelligence is a very important tool to aid organizations in improved decision making in an era of tidal wave of emerging technologies to reduce the stress on their bottom-line and the cost of compliance. It supports businesses by examining data in terms of architectures,

databases and equipment to assist the performance of forecasting, decision support systems, statistical analysis, analytical processing and data mining (Smiles, Kumar, 2018). To achieve the aim of this project, data mining or machine learning will be used to make effective use of the synthetic dataset generated using the simulator called PaySim (It will be explained in later section). The use of data mining dedicated to fraud analytics provide extensive and in-depth analysis of the phenomena of the *fraud*
and build data-driven financial crime detection system for financial payment services.

1.2 Scope and Objectives

The fast base of technological development and with the advent of new technologies, new opportunities have been created to commit fraudulent acts which further impose various challenges for organizations at operational, financial and psychological levels. Along with monetary losses, fraud also has a staggering impact on organization's goodwill and customer relations. According to global economic survey of 2018 conducted by PWC, 49% of the total 7200 companies surveyed by them had experienced some kind of fraud.

Therefore, many organizations try to devise various strategies and implement variety of techniques to fight against fraud, both for fraud prevention and fraud detection but couldn't retain consumer's confidence in electronic payments and card-issuer's reputation.

To address this issue, this project will look at various data mining techniques to perform exploratory and predictive analysis on the given dataset. It helps to identify the fraud before it becomes material and manage the fraud with an objective assessment of the risk involved in the fraudulent transactions. Usually, data comes in two formats: structured data and unstructured data. Data in its pre-defined form is known as structured data and it is easy to extract meaningful information out of it such as the data that resides in spreadsheets and SQL databases while the unstructured data has no pre-defined data model and comes in many forms as texts, images, audio, video and other multimedia content (Baars, Kemper, 2008). For this project, the given dataset is in structured form and can be directly processed with the computing language such as Python.

Hybrid Approach	
Exploratory Analysis	**Predictive Modelling**
Used to gain knowledge about the patterns that are yet not known and carry a high financial crime risk.	Data-driven detection of financial crime using machine learning to identify complex financial crime patterns.
Eg. Outlier detection, data visualization.	Eg. Regression analysis, decision trees, Neural networks.

Table 1: Using hybrid approach for detecting fraud in financial payment services.

Furthermore, data mining techniques for fraud detection are classified into following two types:

a) **Supervised Learning for fraud detection**: This method uses labeled data and the output is predicted when algorithms learn to predict it from the input data. For instance, classification of available records as 'fraudulent' and 'non-fraudulent'.
b) **Unsupervised Learning for fraud detection**: This method uses unlabeled data to infer the natural structure after the algorithms learn from the input data.

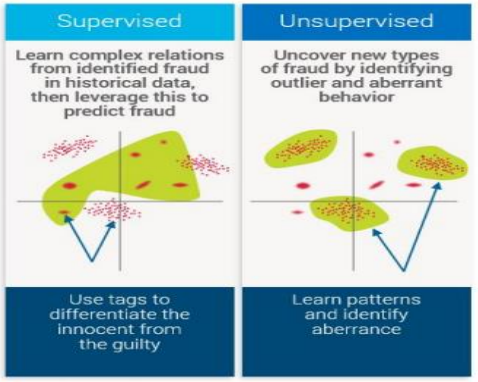

Figure 3: Supervised v/s unsupervised learning for fraud detection
Source: https://s3.amazonaws.com/assets.datacamp.com/production/course_10246/slides/chapter1.pdf

As the dataset under consideration for this project is labeled one, hence supervised machine learning techniques will be used.

The scope of this project lies in developing a Business Intelligence (BI) system (classification model) using data mining applications that involves supervised learning techniques to perform the analysis on the structured data which will be tested on existing dataset.

1.2.1 Objectives

This project aims to detect fraud in financial payment services at an early stage when fraudsters just started intruding the financial system using a synthetic financial dataset for fraud detection generated by a mobile money payment simulator called PaySim.

The goal is to:

Build a Classification model by analyzing the transactional data (Cash-In, Cash-Out, Debit, Payment and Transfer) that consists of both normal customer behavior and fraudulent behavior to correctly categorize the transactions into fraudulent and non-fraudulent.

This goal will be achieved by investigating into following research questions with respect to the available dataset in order to build a model with the highest possible success rate:

i. What types of transactions are actually fraudulent in nature?
ii. What determines whether or not the feature 'isFlaggedFraud' (illegal) gets implemented?
iii. Are expected merchants accounts accordingly labelled?
iv. Are there account labels common to fraudulent transactions?

1.3 Achievements of the Dissertation

The set objectives of this project as defined in section 1.2 were successfully achieved after carefully understanding the problem statement and extensive data analysis conducted using python programming language. Various machine learning techniques were researched on and best techniques were selected to build classification models. Through rigorous experimentation, XGBoost model was discovered to be the best one as it passes all the tests conducted upon it using different methods (section 8.2.1) (section 8.2.2) to evaluate the accuracy of the model. Besides the set objectives, this project has helped me to improve my knowledge regarding financial services domain in terms of compliance and regulatory measures, risks involved and use of technology to improve risk management. Through extensive research and continuously improving my knowledge on the subject matter, helped me to contribute into this emerging field of research and lead the project in the best possible direction.

1.4 Overview of the dissertation

The project work for this study has been organized into eight chapters as follows:

- **Chapter 1: Introduction**
  This chapter gives primary insight into the problem statement, background & context, scope & objectives of the project. It also highlights the achievements of this dissertation.
- **Chapter 2: Literature Review**

This chapter describes the domain knowledge of financial crime related to payment services, previous related work and highlighted that there is lack of research in the area of fraud in payment services along with the motivation to conduct in this area using machine learning techniques.

- **Chapter 3: Methodology**
  It explains about the CRISP-DM methodology which has been opted for the smooth running of the project. It also describes about the PaySim simulator used as a case study to conduct this project.
- **Chapter 4: Data Preparation and Exploration**
  This chapter contains information about detailed data summary, exploratory data analysis, data cleaning, feature engineering and data visualization in order to make data suitable for machine learning algorithms.
- **Chapter 5: Modelling: Teaching an algorithm**
  This chapter introduces to the various machine learning techniques used in this project along with answering why data mining is important for fraud detection research.
- **Chapter 6: Model Training, Tuning and Results**
  This chapter gives insights into the hyperparameter optimization to train the models in order to obtain the desired results.
- **Chapter 7: Evaluating fraud detection performance**
  This chapter highlights the importance of evaluating the classifier based on some performance metrics and compares the confusion matrix results of all the models. It also describes the challenges and limitations of this study.
- **Chapter 8: Model Deployment: Conclusion**
  This chapter covers the summary and Evaluation followed by Future work to improve research in the area of fraud detection in financial payment services.

2. Literature Review

2.1 Introduction

One of the common threats to financial services is the 'fraud' which is as old as humankind. With the expansion in channels and services provided by the financial institutions to the customers, there is significant hike in transaction volumes too because the ease provided by the channel/service, makes us transact more. For Instance, when we transacted more when we switched from branch-based physical banking to tele-banking and even more when we further switched from tele-banking to mobile banking. Thus, we are digital consumers in a way that we does not only live in a technological and digital world but we live for the technology and by the technology with a wide range of access to various

form of digital technology devices such as cell phones, laptops, televisions, Wi-Fi enabled smart appliances et al. This leads to a huge surge in industry wide digital revolution (Carminati et al., 2015) giving fraudsters' fresh opportunities to be confrontational and financial services industry has always been fraudsters' prime and favorite target.

For 44% of financial industry professionals, one of the major concerns for this year is the risk posed by payment fraud according to a recent survey by TD bank. The online and mobile payment fraud takes place when a fraudster steals identity and payment information. The total losses from online payment fraud for this year are estimated to be $22 billion and could go as high as $48 billion according to a study conducted by Juniper Research. Thus, financial institutions need to adapt, to build brand loyalty among consumers who have more options than before to satisfy their financial needs. To gain a competitive edge over the competitors, financial institutions need to deliver a safe and seamless user experience by developing a robust model to detect fraud.

Financial fraud has been addressed by many different techniques including traditional methods such as transactional monitoring system implemented by banks (Hawlova, 2013), anomaly detection and various rule based methods as well as more advanced and elaborated techniques such as data-mining based detection which enhances the capability of traditional methods when analytics is added to traditional approaches as reported in the literature.

2.2 Previous Related Work

Some relevant works in the domain of financial fraud detection are reviewed next:

Below is the summarization of some relevant works in this area before drawing broader conclusions about the current state of art:

a) Common data mining approaches for fraud detection

Yue et al. (2007), Bolton, Hand (2002) and Wang (2010) were the first ones who conducted surveys in the field of fraud detection. Bolton, Hand (2002) statistically researched on the fraud associated with money laundering, computer intrusion, credit cards, and telecommunications and medical and scientific fraud using supervised and unsupervised fraud detection tools. Their findings show that there is lack of flagged fraud data in this field and to decrease the losses, speed at which one can detect the fraud matters in the banking fraud. Yue et al. (2007) through their study provides a comprehensive view on financial fraud detection (FFD) process using data mining techniques such as regression, statistical tests and neural networks. Wang (2010) focused more specially on the application of data mining techniques in detecting fraud in financial statements and pointed out that there is lack of mature methods and lack of access to data to discover fraud. Also, there is difference between the datasets, methods and the evaluation techniques. Bhowmik (2008) used Naïve Bayesian classifier to predict fraud and looked at performance metrics

derived from the confusion matrix such as accuracy, recall and precision to construct ROC (Relative Operating Characteristic) for detailed analysis. Sahin et al. (2013) investigated into cost sensitive decision tree approach for fraud detection as compared to traditional data mining methods.Gray, Debreceny (2014) studied the application of data mining techniques primarily on quantitative data and secondarily on text data to detect fraud in financial statements audit. Carcillo et al. (2019) proposed a hybrid approach that combines supervised and unsupervised techniques to improve accuracy in fraud detection particularly in credit card fraud.

b) Signature based architectures for fraud detection

Edge, Sampaio (2009) critically examined signature based architectures, models and fraud applications with respect to their proactive capabilities for detecting fraud within streaming financial data.

c) Statistical Techniques and Artificial Neural Networks approach for fraud detection

Pal, Jamal (2015) introduced two primary techniques for fraud detection using data mining as statistical techniques and artificial intelligence and found that decision tree is the most powerful technique for credit card fraud detection. . Maes et al. (2002) applied two machine learning techniques: Artificial neural networks and Bayesian Neural networks on the real world financial data to investigate the credit card fraud. Patidar et al. (2011) investigated into fraudulent credit card transaction using neural network technique along with genetic algorithm. Akhilomen (2013) implemented anomaly detection algorithm along with pattern recognition based on neural networks to detect fraud in real-time transaction on the internet and thereby, classifying the transactions as legitimate, suspicious fraud and illegitimate transaction. Olszewski et al. (2013) applied user accounts visualization and classification threshold-type detection based on Self-Organizing Maps (SOM Grid and UMatrix) in the field of telecommunication fraud detection. Ngai et al. (2011) proposed a conceptual framework for classifying the data mining techniques such as Logistic model, Neural Networks, Bayesian Belief Network to Financial fraud detection.

d) Unsupervised methods for credit card fraud detection

Bolton, Hand (2002) discussed unsupervised credit card fraud detection to detect changes in unusual transactions through behavioral outlier detection techniques such as Peer group analysis (detects the behavior of object which behaves differently than it used to behave before) and Break Point analysis (identifies spending behavior in a single account based on the transaction information). Dutree and Hofland (2017) implemented a single layer neural network using fraud oversampling and focal loss-function for detecting fraud in financial payments. Weston et al. (2008) applied peer group analysis method (unsupervised method) on the real credit card transaction data to identify the transactions that deviate from their peer group and flagged them as potentially fraudulent in nature.

e) Cognitive approach for fraud detection

Grazioli et al. (2006) provided a cognitive approach to detect fraudulent behavior in financial statements audit by looking at psychological phenomenon i.e. the detection of deception which means a deliberate attempt to mislead others and analyzing the thinking process of auditors particularly when they made errors while evaluating documents created by others.

f) Money laundering detection in mobile money & Cryptocurrency transactions

Rieke et al. (2013) applied a predictive security analysis (PSA) tool at a run time to detect money laundering patterns in mobile money transactions. They analyzed synthetic process behavior of the transactions based on properties captured from real-world data to propose a system that will raise an alert if abnormal transaction happens. Brenig et al. (2015) presented a study that focuses on economic analysis of Money Laundering using Cryptocurrencies and accentuated that the hike of public interest in Cryptocurrencies in turn, leads to an increase in the amount of fraudulent activities and scams posing challenges for financial systems in general and specifically for AML programs across the globe.

g) Beneish m-score model, Benford's law and Hidden Markov Model for fraud detection

Herawati et al. (2015) showed that Beneish m-score model can be used an effective tool for data mining of fraud-committed companies. Geyer (2010) studied the fraudulently reported financial data using Benford's law which tell us about the use of expected distribution of significant digits in naturally occurring datasets for fraud detection. Khandare (2016) proposed a Hidden Markov /model (HMM) to detect unobserved (hidden) activities on credit cards. It works by maintaining a database of past transactions and sending an alert message to the card holder if any unusual transaction takes place.

h) Organization specific fraud detection tools & models

Chen et al. (2015) introduced a big data based fraud prevention product called AntBuckler built up by Alibaba to identify bad users and transactions to prevent fraud. Spathis et al. (2002) examined published data to develop a model for detection of false financial statements for Greek firms. They used Logistic regression to identify factors associated with the false financial statements aiding accounting and auditing fraud detection research.

i) Survey on Big data techniques for fraud detection

Ahmed et al. (2016) presented a survey on clustering based techniques to detect anomalous (abnormal) behavior for further analysis such as fraudulent activity. They also highlighted the issue of scarcity of real data to work in fraud detection area and discussed on the synthetic data generation process. Omolara et al. (2018) presented a survey on the application of big data techniques such as data-mining, clustering, neural networks, genetic algorithms and fuzzy support vector machine models in the area of financial fraud detection particularly Bank Fraud, securities

fraud, Insurance fraud, Computer- Intrusion fraud and other related financial fraud and highlighted the need to look into other issues such as hardware equipment, maintenance, software licensing requirements for big data infrastructure.

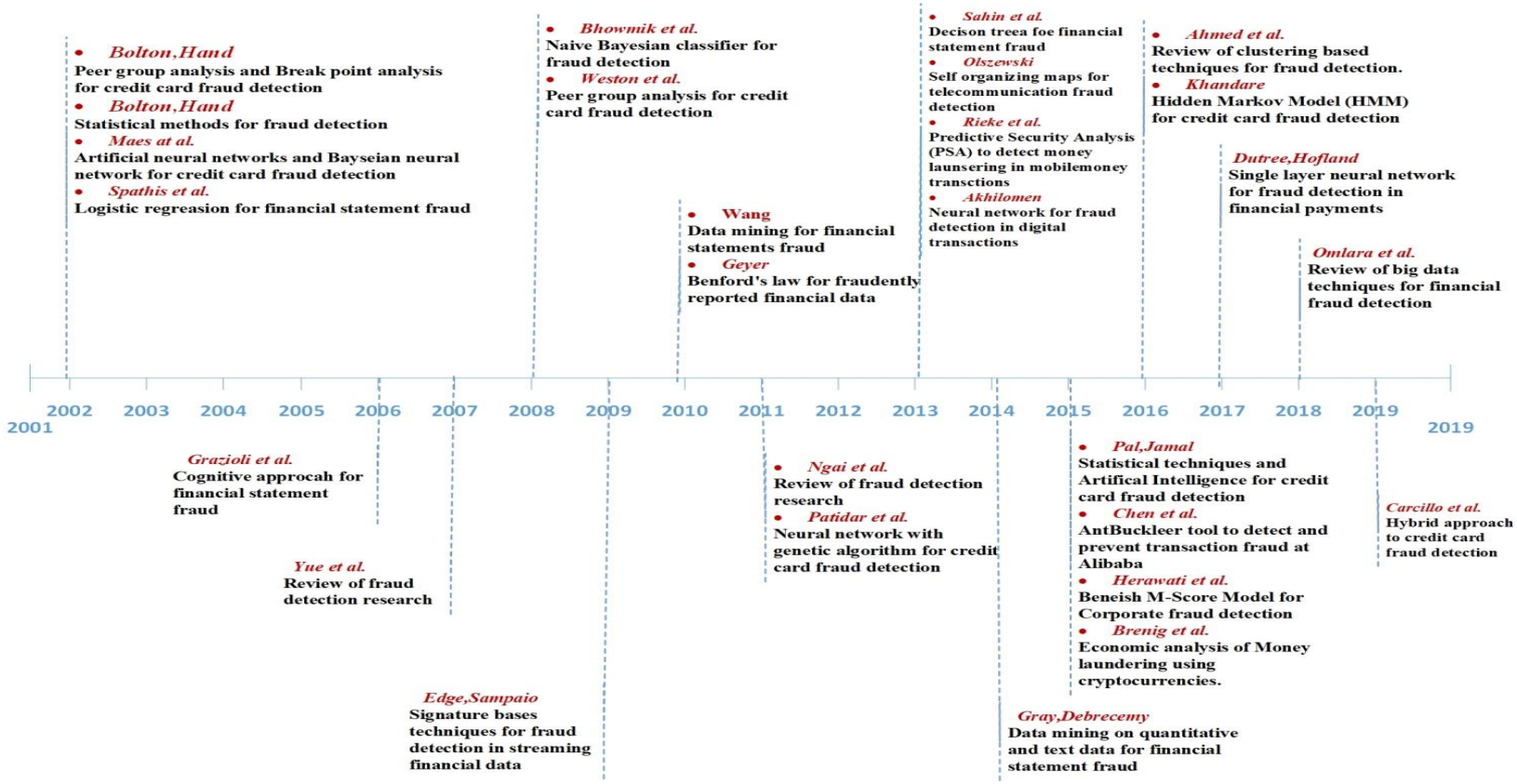

Figure 4: Visual representation of Literature of review from year 2002 to 2019.

2.3 Comparison and critical analysis of previous related work in relation to the present scenario

The purpose of this review was to scrutinize the type of financial fraud investigated into and the techniques applied in terms of detecting the fraud within the past few years from 2002 to 2019. *From a story of financial fraud recounted by Aristotle in sixth century BC Greece where Greek sculptors committed fraud by carving signatures of Phidias and Praxiteles into their works, that was prepared for export to Roman collectors*

(Gong et al, 2016) to modern frauds such as the shell game, Ponzi and Pyramid schemes et al., financial fraud has become a worldwide crime. Fraud impacts human lives from raising the price we pay for acquiring goods and services to pulling out resources from innovation, it touches every area of our lives. Therefore, detection of fraud is a worthwhile endeavor.

It is evident from the literature that from using fragmented approach to prevent fraud such as using business rules to look for anomalies in the datasets, to applying more than standard analytics techniques such as predictive analytics – including a form of AI known as data mining or machine learning, real-time transaction monitoring has become a baseline requirement for organizations (which was once a luxury). Despite of the numerous efforts made by the prominent researchers in the past in the area of financial fraud detection, one of the shortcomings in prior literature is found to be the imbalance between the demand-supply sides of fraud, due to more focus on the supply side of frauds i.e. fraudsters, without endogenizing the behavior of demand side i.e. victims. The research does not reveal the behavior of the victims after the fraud happens and what kind of proactive or reactive measures were already in place for immediate action. Also, Olszewski et al. (2013) in their study do not classify the accounts into fraudulent and non-fraudulent accounts, which if included would offer much better results. Though Akhilomen (2013) provided an efficient way to discover fraud in real time transactions but he does not take into account the fact that IP addresses used to identify the location of the transactions can easily be altered using varied proxy servers that disclosed huge gap in the research. The study conducted by Khandare (2016) demonstrated that the HMM approach cannot be generalized to be well suited on the global fraud detection problem. Additionally, the existing literature shows the attempts to deal with class imbalance problem in other domains such as telecommunications, fraudulent telephone calls and text classification (Chawla et al, 2002) and credit card fraud detection using under-sampling methods (Carneiro, 2017) which makes the approach restricted to generalize well on real data. *These papers are the most relevant ones in relation to the set objectives (section 1.2.1) and the solution will be proposed in terms of the research gap examined in these papers.* Furthermore, there is limited research in financial payment services fraud field particularly into fraud in mobile financial services (in relation to dataset under consideration) because of limited deployment of such services globally (Mudiri, 2013). The proper treatment of extreme class imbalance in the field of fraud analytics is very important to take into account in order to come up with better fraud models. Therefore, objective is to find out a ML algorithm that works well with unbalanced data to make the approach suitable for real time data.

2.3.1 Lack of research in fraud detection in financial payment services

Furthermore, the published literature talks more about predicting and preventing fraud in credit cards and financial statements and focused more on common supervised machine learning algorithms such as regression, Naïve Bayes, Sector Vector Machines (SVM), neural networks and decision trees and unsupervised machine learning algorithms such as k-means clustering because of ease in interpreting the results obtained from them, neglecting the scope and scale of problem in question. Besides this, it is apparent from the literature that there is lack of research in

the area of fraud detection in financial payment services. Due to ever growing usage of mobile money as a mode of executing payments, likelihood of criminals to perform fraudulent activities has also increased which fosters the need to probe into potential security pitfalls with the ultimate goal to develop a system to predict and prevent fraud in financial payment services (Digital risk management). Another obstacle which hampered the research in this area is in the lack of publicly available financial data sets because of private nature of transactions.

Hence, the work presented in this dissertation is an effort to address various concerns in the domain of detecting fraud in financial payment services using data mining or machine learning approach and providing convincing solution to such concerns as outlined below:

Problems Highlighted in Existing Literature	**Approach used to address the problem**
Lack of publicly available dataset	Use of Synthetic dataset for fraud detection research in financial payment services. PaySim Simulator (to obtain data) is used as a case study to propose an alternative to publicly available datasets for research in financial services industry.
Issue of fraud detection under extreme class imbalance	Find an ML algorithm which works well with highly skewed data.
Use of common supervised and unsupervised machine learning algorithms in the existing research, resulting into conflicting model performance.	Ensemble methods will be used to combine various machine learning techniques into one predictive model to improve fraud detection in order to optimize model performance. Ensemble methods will also be compared with statistical machine learning approaches.

Table 2: Scale and Scope of Research Problem.

3. Methodology

3.1 Data Mining Framework: CRISP-DM

The growth of large databases is the motivating stimulus behind the industry wide adoption of data mining as a means of uncovering valuable information from these large databases (i.e. knowledge discovery in databases) (Hand, 2006). Data mining's foundation is an amalgamation of three intertwined disciplines: statistics (studies numerical relationship between data), artificial intelligence (software/machines displaying human-like intelligence), and machine learning (predictions are made when algorithms learn from the data). Data mining is a discipline that keeps evolving to keep pace with affordable computing power and limitless potential of Big Data. Various methodologies such as SEMMA

(Sample, Explore, Modify, Model, Assess) developed by SAS Institute, DMAIC (Define, Measure, Analyze, Improve and Control) specifically used in Six Sigma practice have been described in the literature for successful implementation of data mining projects. However, a generic framework i.e. CRoss-Industry Standard Process for Data Mining (CRISP-DM) which is one of such streamlined approach and an industry proven way to implement data mining projects successfully will be used to fulfill the . This hierarchical model aims to analytically solve a business problem by translating them to data mining tasks (Wirth, Hipp, 2000).

The six phases of this well-known process model in the life cycle of a data mining project are illustrated in following figure:

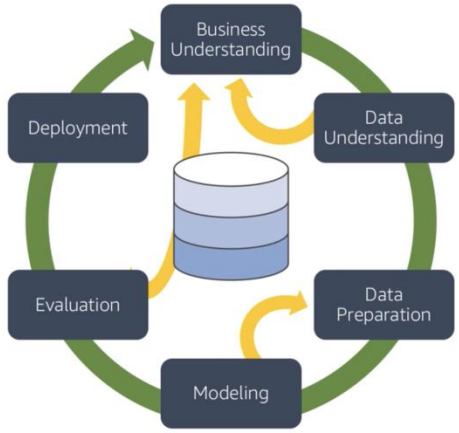

Figure 5: Steps involved in CRISP-DM process
Source: https://gist.github.com/bluekidds/cad5c0ea2e5051b638ec39810f3c4b09

The significance of each step during the implementation of data mining project is explained as follows:

a. **Business Understanding:**
   It involves understanding the objectives and requirements of a project from business perspective (Sharma et al., 2009) and then using this knowledge to define a data mining problem and a preliminary plan to achieve the stated objectives (Nadali et al., 2011).

b. **Data Understanding:**

It involves collecting the data to get properly familiar with it before proceeding further, to identify data quality problems in order to discover meaningful insights from the data and to form hypotheses for the hidden information after interesting subsets are detected from the data.

c. **Data Preparation:**

It involves all the activities from cleaning of raw data to construct final analytical data set (Sharma et al., 2012). Data preparation is very crucial step to achieve better results. Furthermore, data preparation involves various steps to apply different pre-processing techniques to make it ready for further analysis. Following diagram briefly explained the steps involved during data preparation stage of data mining:

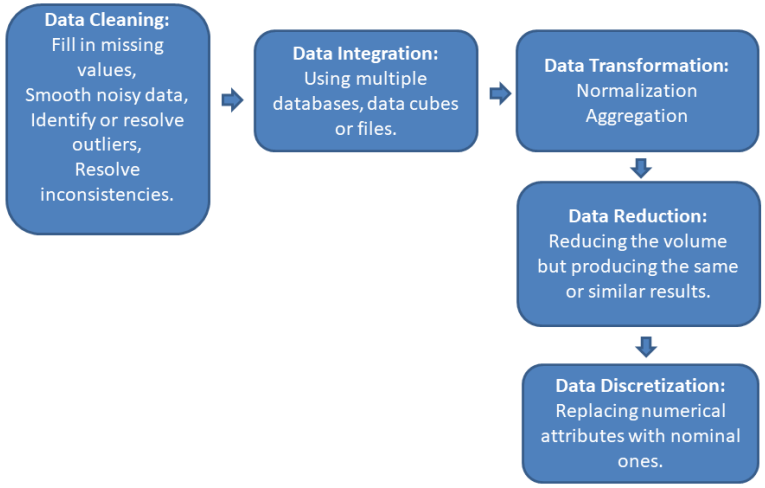

Figure 6: Steps involved in Data Preparation stage

The data preparation for this project has been explained in chapter 4.

d. **Modelling (Data Mining):**

In modeling phase, various machine learning algorithms are selected and applied on the clean data and their parameters are calibrated to order to identify the final best algorithm (see chapter 5).

e. **Evaluation**:
In this stage, the steps used to construct the model are reviewed and the model is thoroughly evaluated in terms of its performance to ensure that the stated project objectives have been successfully achieved. The key objective is to identify any deviation from the projected and the actual outcome of results and the need to sufficiently consider this issue for deciding onto using the data mining results at the end of this phase.

f. Deployment:
Mere creation of a model doesn't generally mark the end of a project because the knowledge gained from the whole data mining project needs to be properly organized and disseminated in a way that the customer can use it. This can be done by either generating a report or implementing the whole data mining process again across the enterprise under consideration.

3.2 PaySim: A financial mobile money simulator for fraud detection research

3.2.1 Introduction to PaySim

The problem of lack of publicly available financial data sets in the domain of fraud detection research is addressed by the adoption of **synthetic dataset** generated using PaySim simulator (https://www.kaggle.com/ntnu-testimon/paysim1). PaySim is a simulation tool used to generate synthetic datasets of mobile money transactions based on original dataset. It was first introduced by Lopez-Rojas and Axelsson in the wake of growth in mobile money payments. For instance, In Tanzania (one of the fastest growing economy according to the World Bank) 100 million transactions were made using mobile money during December 2013 alone, total netting a volume of $1.8 billion dollars (Rojas et al, 2016).
The sample of real transactions is extracted from the logs of a mobile money service implemented in an African country and then PaySim simulates those transactions to generate realistic synthetic dataset using Statistical analysis and Social Network Analysis (i.e. modelling the social behavior of clients/individuals using networks and graph theory). The researchers used MABS (Multi-Agent Based Simulation) toolkit called MASON version 19 (which is implemented in JAVA) for simulation. MABS is an approach that uses autonomous and interactive agents to model complex systems. In mobile money simulation, transactions (money sent or received) is represented by connections that interact with clients (who represents nodes) connected through a social network.

3.2.2 Scenarios and Generation of Synthetic Data

The synthetic dataset was obtained considering a hypothetical situation where 200 clients from 4 different cities transact with partners within or outside their city, restricted to five contacts per client within the city and two contacts outside the city. 10% of the clients connected in a network were decided to be chosen as fraudsters involved into fraud like money laundering (disguising original ownership of money). All the transactions were stored in a log file to run simulation for five times to fetch 1000 steps and then the files generated were merged together to use as an input (dataset) for machine learning algorithms.

After running five simulations, total of 486977 transactions were simulated with a total of 6006 transactions performed by 107 malicious agents and are labeled as suspicious (Lopez, 2014).

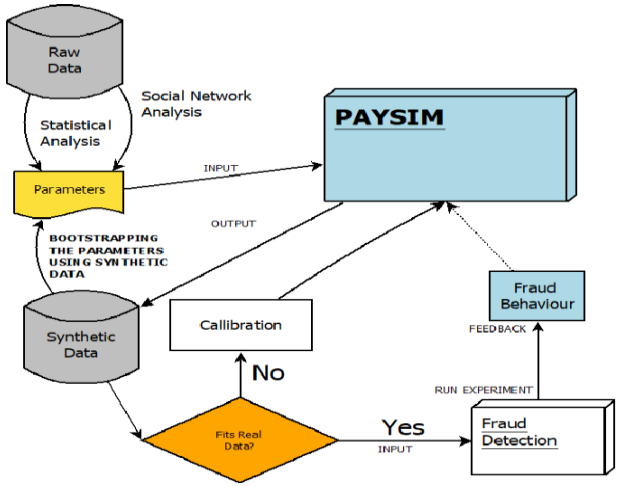

Figure 7: Use-case representation of PaySim Simulator

Source:https://www.semanticscholar.org/paper/Bootstrapping-the-Paysim-Financial-Simulator-for-Lopez-Rojas-Franke/7e8f454183a557b6189f6b552973d3b01e4082da

⇒ **Benefits of using synthetic dataset for Fraud detection in financial payment services**

Money laundering is the favorite medium of fraudsters to do illegal activities and making them look as legitimate ones in the eyes of the society. Due to failure of current countermeasures to solve this problem, this study aims to formulate the research questions under consideration as machine learning problem which involves:

Task T: Classification of transactions as: fraudulent and non-fraudulent. The aim is to find the outliers in a dataset.

Performance Measure P: Percentage of True Positives (TP) i.e. transactions correctly classified as anomalous and percentage of False Positives (FP) i.e. transactions misclassified as anomalous.

Experience E: Synthetic data generated with legal (non-fraudulent) and illegal (fraudulent) transactions.

The use of synthetic data for fraud detection research provides following benefits-

- The possibility of selecting the attributes that reduces the complexity of data structures involved which further simples data preparation process.
- To comply with different experimental setups by other researchers, the volume of data can be tuned.
- The privacy of customer is not hampered.
- No influence of political and legal policies in relation to disclosure of results.
- Different scenarios can be modeled as researcher controls the model parameters.
- Enough abnormal data can be injected to address the class imbalance problem.

4. Data Understanding and Preparation:

Data understanding and preparation are crucial steps for data mining to ensure that the analytical dataset used as an input in the Modelling stage is acceptable and is of improved quality. It requires removing noise, outliers, missing values and inconsistent data records from the given dataset. **To build reliable models, following steps have been undertaken:**

4.1 Data Description

The data for the project under consideration has been acquired from kaggle (https://www.kaggle.com/ntnu-testimon/paysim1) and is a openly distributed synthetic dataset for fraud detection created using a simulator called as PaySim. It has been downloaded as .csv file and contains 6262621x11 matrix of data where input features falls under Matrix 'x' and output features falls under matrix 'y'.

Dataset Name	Instances	Number of attributes	Data Format
PS_20174392719_1491204439457_log.csv	6362620	11	.csv

Table 3: Dataset Details

A) Types of transactions in the dataset

The types of transactions covered in the dataset and investigate into are explained as below:

a) **Cash_in:** Merchant serves as an ATM to the customers and customers can increase the balance of the account by paying in cash to the merchants.

b) **Cash_out:** Merchant serves as an ATM to the customers and customers can decrease the balance of the account by withdrawing cash from the merchants.

c) **Debit:** When customer send money from a mobile money service to a bank account, the transaction is termed as Debit. It decreases balance in the account just like Cash_out transaction.

d) **Payment:** When a customer pays to acquire goods or services from a merchant, the transaction is termed as Payment. It decreases the account balance of the sender while account balance of the receiver increases (i.e. amount got credited in his account).

e) **Transfer:** When one user sends money to another user through mobile money service platform, the transaction is termed as Transfer.

There are 11 feature labels in the chosen dataset which are explained in the table below. Also, Please note that the description of these columns has been acquired from the dataset.

Column Name	Description
Step	Step refers to the time interval that maps a unit of time with a real-world time where 1 step (total steps=744) = 1 hour (total days=30). It defines after how much time gap a transaction occurs.
Type	Type refers to the nature of the transaction. There are five types of transactions that occur in a chosen dataset- CASH_IN, DEBIT, CASH_OUT, PAYMENT OR TRANSFER.
Amount	It refers to the transaction amount (in numerical terms) in a local currency.
nameOrig	It refers to the account name of the sender who initiated the transaction.
oldBalanceOrig	It refers to the initial balance in the sender's account prior to occurrence of any transaction.
newBalanceOrig	It refers to the new balance in the sender's account after the transaction is completed.
nameDest	It refers to the recipient ID i.e. account name of the receiver of the transaction.
oldBalanceDest	It refers to the initial balance in the receiver's account prior to occurrence of any transaction.
newBalanceDest	It refers to the new balance in the receiver's account after the transaction is completed.
isFraud	The instance of all the transactions are identified as Fraudulent=1 or Non-fraudulent=0.
isFlaggedFraud	Flags suspicious transactions as fraud when a user illegally attempts to transfer more than 200.000 in a single transaction.

Table 4: Definitions of columns as defined in the chosen dataset.

The description of the 11 attributes of the dataset is stated in the below table:

Attribute Name	Type	Number of distinct classes
Step	int64	743

Type	object	5
Amount	float64	5316900
nameOrig	object	6353307
oldBalanceOrig	float64	1845844
newBalanceOrig	float64	2682586
nameDest	Object	2722362
oldBalanceDest	float64	3614697
newBalanceDest	float64	3555499
isFraud	int64	2
isFlaggedFraud	int64	2

Table 5: Tabular representation of the attributes

The other statistical information of the data frame has been displayed in the image below:

	step	amount	oldBalanceOrig	newBalanceOrig
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	2.433972e+02	1.798619e+05	8.338831e+05	8.551137e+05
std	1.423320e+02	6.038582e+05	2.888243e+06	2.924049e+06
min	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	1.338957e+04	0.000000e+00	0.000000e+00
50%	2.390000e+02	7.487194e+04	1.420800e+04	0.000000e+00
75%	3.350000e+02	2.087215e+05	1.073152e+05	1.442584e+05
max	7.430000e+02	9.244552e+07	5.958504e+07	4.958504e+07

	oldBalanceDest	newBalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
mean	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

Figure 8: Statistical information about the Dataframe.

4.2 Data Preparation and Exploration

Following steps have been undertaken during data preprocessing stage in order to ensure that the chosen machine learning models yield best results. For the purpose of the project under consideration, various libraries in python programming language have been used which are stated along with the step undertaken as below:

4.2.1 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an approach to extract meaningful information from the clean dataset in hand to summarize the important characteristics of the data. The most succinct way to gain insights into the dataset is to wrangle with the data exclusively using various Dataframe methods. The EDA checklist is based on the *research questions as stated in section 1.2.1* was explored into for identifying the main characteristics of the data before performing machine learning Modelling phase.

a) What types of transactions are actually fraudulent in nature?

In PaySim dataset, there are five types of transactions as explained in the section above. To begin with, the first task was to find out number of fraudulent and non-fraudulent transactions in total data. To do so, a function named as *fraudulent* was defined in python and it was discovered that the fraud occurs only in two of them- 'CASH_OUT' and 'TRANSFER' (Appendix2). 'CASH_OUT' is a type of transaction where merchant pays the fraudster (customer) in cash when money is sent to him and 'TRANSFER' is a type of transaction where money is sent to a fraudster (customer).

The frequency of occurrence of each transaction in Paysim dataset is outlined in the table below:

Type of Transaction	Frequency of Genuine transactions	Frequency of Fraudulent transactions	Total
CASH IN	1399284	0	1399284
CASH OUT	2233384	4116	2237500
DEBIT	41432	0	41432
PAYMENT	2151494	0	2151494
TRANSFER	528812	4097	532909
TOTAL	6354407	8213	6362620

Table 6: PaySim dataset statistics

From the above table, it can be seen that the number of fraudulent TRANSFERs i.e. 4097 is almost equal to the number of fraudulent CASH_OUTs i.e. 4116. In other words, 0.183% transactions out of total CASH_OUTs are fraudulent ones and 0.768% transactions out of total TRANSFERs are fraudulent ones. Another important derivation from above table is that the data is highly imbalanced *(see chapter 6)* with 0.13% fraudulent transactions as out of total 6362620 transactions, only 8213 transactions (0.0646% Cash_Out , 0.0643% Transfer among total transactions) which are labelled as fraudulent.

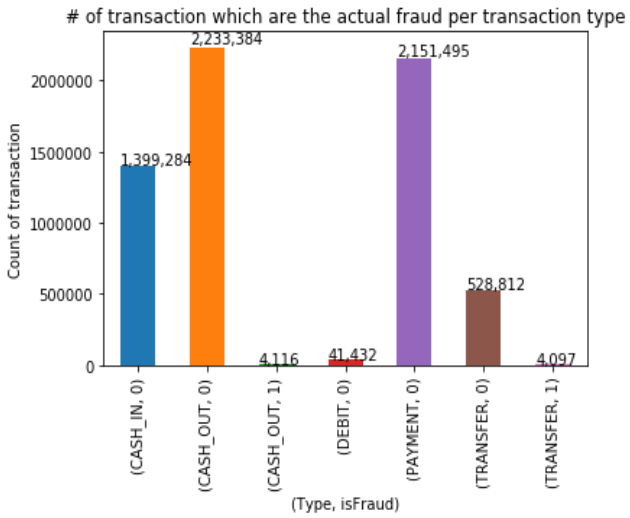

Figure 9: Graphical Representation of transactions

b) What determines whether or not the feature 'isFlaggedFraud' (illegal) gets set implemented?

The purpose of setting threshold for isFlaggedFraud is to stop the transaction from being processed once it reaches the maximum amount where it is supposed to be treated as fraudulent one.

When a customer attempts to Transfer an amount greater than 200,000 it is treated as isFlaggedFraud being implemented. Using *value_counts()* function in python, it was discovered that isFlaggedFraud is set just only 16 times *(by a simulator)* out of total 6362620 transactions. For further analysis, it was explored that *minimum amount in a TRANSFER when isFlaggedFraud is set is 353874.22* while the *maximum amount during TRANSFER when isFlaggedFraud is not set is 92445516.64* (Appendix2).

Also, for every TRANSFER transaction where isFlaggedFraud is set, oldBalanceDest=0 in all such transactions. It is interested to see that old balance is identical to new balance in origin as well as destination accounts may be because the transactions get halted when they reach the set threshold. However, in a case when isFlaggedFraud can remain not set during TRANSFER, the state of isFlaggedFraud cannot be determined as both oldBalanceDest and newBalanceDest can be 0. It is not possible to put threshold on oldBalanceOrig when isFlaggedFraud is set because the corresponding range of values overlaps with the transactions when isFlaggedFraud is not set. The *Min, Max of OldBalanceOrig when isFlaggedFraud= 1 during transfer is [353874.0, 19585040.0]* whereas *Min, Max of OldBalanceOrig when isFlaggedFraud= 0 during transfer (where OldBalanceOrig= newBalanceOrig) is [0.0, 575668.0]* (Appendix3). It is important to note that newBalanceOrig is updated only after the transaction takes place, hence it is not considered to determine whether isFlaggedFraud gets set or not because isFlaggedFraud is set before the transaction happens and will halt the process if it reaches the set threshold.

Based on a customer transacting more than once, it is vital to note that duplicate customer names exist within transactions where isFlaggedFraud is not set and no duplicates have been found where isFlaggedFraud is set which means originators of transactions have transacted only once and very few destination accounts, where isFlaggedFraud is set shows the evidence of transaction being takes place more than once (Appendix4). It means that isFlaggedFraud set is independent of whether a destination account has been used before or not.

Conclusion: Based on the above analysis, isFlaggedFraud can be treated as insignificant feature and therefore, discarded in further analysis as it does not correlate with any of the explanatory variables in the given dataset and is just set 16 times in 6262620 transactions, that too in a seemingly meaningless way.

c) Are expected merchants accounts accordingly labelled?

To find out if the merchant accounts are accordingly labelled, merchants among originator accounts during CASH_IN and, merchants among destination accounts during CASH_OUT transactions have been investigated into. Using *str.contains*() function in python, it was discovered that Merchants ('M') are not involved in CASH_IN (means being paid by a merchant as stated) transactions to customers ('C'). Similarly, there are no Merchants among destination accounts for CASH_OUT transactions (as stated involves paying a merchant) but merchants do exist for all PAYMENTs transactions among destination accounts (Appendix5). Hence, the merchants occur in an unexpected way in nameOrig and nameDest for all transactions.

d) Are there account labels common to fraudulent transactions?

The modus operandi for committing fraud states that a fraudulent transaction will include both TRANSFER (destination) and CASH_OUT (originator) i.e. first transferring the amount to a fraudulent account and then subsequently cashing it out but it was found that no such common accounts exist among 8213 fraudulent transactions (Appendix5). Further, destination accounts that originate genuine CASH_OUT for fraudulent TRANSFERs but not detected out to be fraudulent ones were investigated into and it was found that there exist 3 such accounts where such dualistic transactions were labeled as genuine ones (Appendix5). 2 out of 3 such accounts first made genuine CASH_OUT and later fraudulent TRANSFER such as genuine CASH_OUT from C423543548 occurred at step 185 while fraudulent TRANSFER occurred later at step 486 (Appendix6). Thus, nameOrig and nameDest does not indicate the presence of fraudulent transactions.

Conclusion: Based on the above analysis and noting from previous section that merchant accounts are not encoded in an unexpected way by nameOrig and nameDest, it was decided to drop these features since they are meaningless for further analysis.

4.2.2 Data Cleaning: Transforming messy data to tidy data

As the exploratory data analysis in above section shows that fraud occurs only in two types of transactions i.e. TRANSFERs and CASH_OUTs, so only the corresponding data has been taken for further analysis.

To do so, a function named as *'cleaning'* has been defined in python for adequate data cleaning. Following steps have been undertaken for data cleaning:

a) **Irrelevant Data:** Based on EDA, the columns such as 'nameOrig', 'nameDest', 'isFlaggedFraud' that proved to be irrelevant for analysis have been eliminated for further analysis (Appendix7).
b) **Binary-encoding**: For machine learning algorithms, it is very important that the data should be in numerical form. Thus, labelled data in 'TRANSFER' and 'CASH_OUT' have been encoded as [TRANSFER=0], [CASH_OUT=1] (Appendix7).
c) **Imputation of latent missing values**: The data has been analyzed to look for latent missing values in:

i. **For destination accounts:** In the destination accounts i.e. oldBalanceDest and newBalanceDest, the data has several transactions with zero balances both before and after a non-zero amount is transacted. The transactions have been analyzed to look for the transactions where zero likely denotes a missing value and it is found that the fraction of such transactions is much larger in fraudulent ones i.e. 50% as compared to genuine ones 0.06% (Appendix7).

The account balances of destination accounts before the transaction is made was not either imputed with a statistic or with a subsequent adjustment for the amount transacted from a distribution because the destination account balances being zero strongly indicates the presence of fraud. If such values would have chosen to be imputed, it will result in masking this indicator of fraud and making fraudulent transactions appear

as genuine ones. ***Hence, instead of replacing such values with 0, it was decided to replace them with -1 which will be more suitable for a machine learning algorithm to detect fraud*** *(Appendix6).*

ii. **For originating accounts:** In the originating accounts i.e. oldBalanceOrigin and newBalanceOrigin, the data has several transactions with zero balances both before and after a non-zero amount is transacted. Based on the analysis, it was found that that the fraction of such transactions is much smaller in fraudulent ones i.e. 0.3% as compared to genuine ones 47%. **Hence, *instead of replacing such values with a numerical value, it was decided to replace the value with 0 with a null value (Appendix6).***

4.2.3 Feature Engineering

It involves the task of transforming raw data into features that describe the inherent structures in the data. The data imputation conducted in this section has been used to create 2 new features (columns) (Appendix7) to record errors in originating and destination accounts for each transaction. The computation of these features is inspired from the possibility that zero-balances serve to differentiate fraudulent transactions from genuine ones. Furthermore, these two new features turn out to better represent the underlying problem to the machine learning models used (see results section).

These two new features have been built using following formulas:

errorBalanceOrig= newBalanceOrig+amount-oldBalanceOrig
errorBalanceDest= newBalanceDest+amount-oldBalanceDest

	step	type	amount	oldBalanceOrig	newBalanceOrig	oldBalanceDest	newBalanceDest	errorBalanceOrig	errorBalanceDest
2	1	0	181.00	181.0	0.0	-1.0	-1.00	0.00	181.0
3	1	1	181.00	181.0	0.0	21182.0	0.00	0.00	21363.0
15	1	1	229133.94	15325.0	0.0	5083.0	51513.44	213808.94	182703.5
19	1	0	215310.30	705.0	0.0	22425.0	0.00	214605.30	237735.3
24	1	0	311685.89	10835.0	0.0	6267.0	2719172.89	300850.89	-2401220.0

Figure 10: Adding new features to record errors in originating and destination accounts for each transaction.

Based on the summary statistics on the errorBalanceOrig, it is indicated that the most negative error is -7.450581e-09 which is very small and close to 0, 3rd quartile is also 0 (i.e. 75% of the data is between (-7.450581e-09 and 0) and the largest error is 10,000,000. However, a large proportion of the data have an error of 0 or close to 0. On the other hand, large errors can be seen in valid transactions such as about 75% of the data have errors exceeding 52,613.43 and the largest error is 92,445,520. Based on the summary statistics on the errorBalanceDest, it is discovered that there exist large positive and negative errors in the accounts where money has been moved to for both fraudulent and valid transactions. These distinctions make both of these new features as potentially effective ones.

Two more features have been constructed named as 'DayofWeek' and 'HourOfDay'. To construct the features, time patterns were investigated into *(see images below)* and it was found that the number of transactions is dispersed over the course of a month, so it was decided to construct 'DayofWeek' and 'HourOfDay' features.

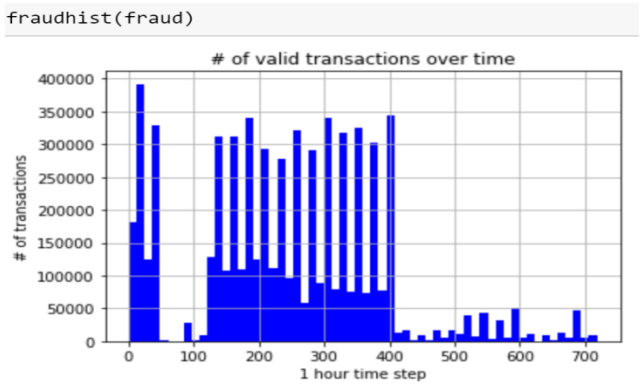

Figure 11: Valid transactions over time

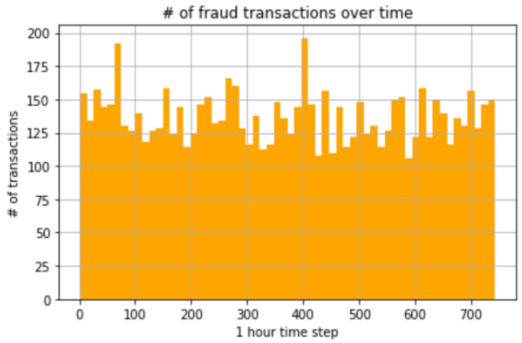

Figure 12: Fraudulent transactions over time

The time patterns in above images show that most of the valid transactions occur around 0th and 60th time step ***(one step= 1hour and defines a time interval after the transaction happens)*** *and 110th and 410th time steps. The other observation is that the frequency of occurrence of fraudulent transactions does not seem to change much over time.*

To get hours and days of the week, a function named as 'DayHour' was defined using python. Also, two variables as num_days=7 and num_hours=24 were defined and following calculations were made and visualized as follows:

- fraud_days = X.step % num_days
- fraud_hours = X.step % num_hours
- valid_days = Y.step % num_days
- valid_hours = Y.step % num_hours

```
DayHour(fraud,valid)
```

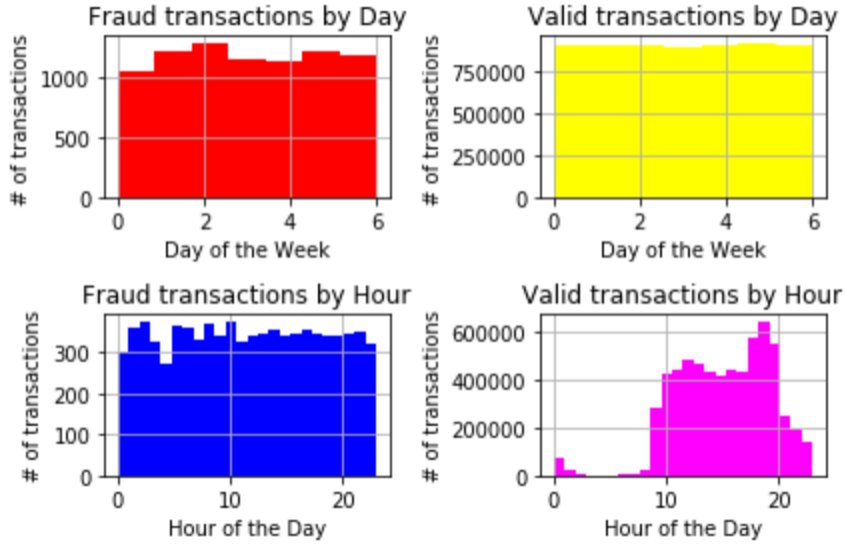

Figure 13: Fraudulent and Valid transactions by Day and Hour

From graphs above, it can be observed that both valid and fraudulent transactions are uniformally distributed over the Days of the Week. Although there is slightly more variation for fraudulent transactions it does not appear to be enough to act as a strong indicator for our model. Also, there is strong evidence in terms of valid transactions that they mostly occur from hour 0 to hour 9 (inclusive) and fraudulent transactions also occur at similar rates during any hour of the day except hour 0 to hour 9 (inclusive).

4.2.4 Data Visualization:

The differences between fraudulent and genuine transactions were visualized in order to find more suitable patterns and/or trends, detect outliers to make data-driven decisions. Such differences have been visualized in several ways:

a) Looking for dispersion over time

When the dispersion of fraudulent and genuine transactions has been viewed over time using plot below, it was found that they yield different fingerprints. It can be seen that fraudulent transactions are more homogenously distributed over time in comparison to genuine ones. Also, there is a balanced distribution between CASH_OUTs and TRANSFERs in genuine transactions. The 'jitter' parameter in the plotStrip function has been used to define the width of each fingerprint. This parameter attempts to make separation between the transactions occurring at the same time with different abscissae.

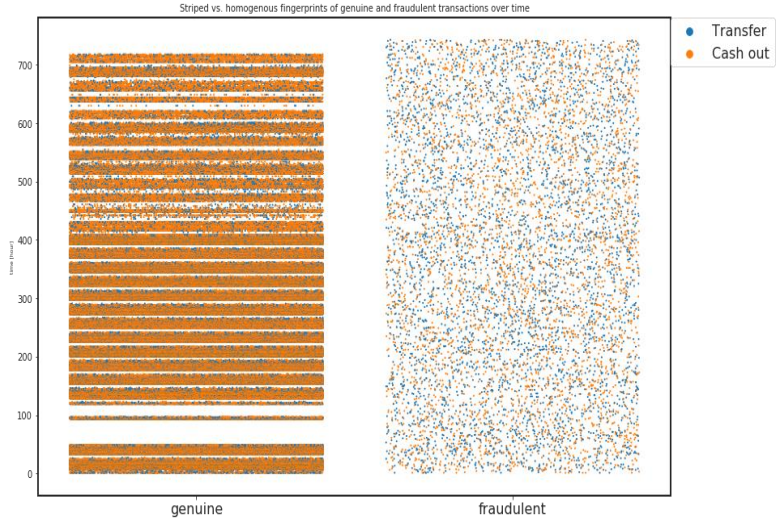

Figure 14: Striped and homogenous distribution over time

b) Looking for dispersion over amount

When the dispersion of fraudulent and genuine transactions has been viewed over time using plots below, it was found that new *errorBalanceDest* feature is more effective than original *amount* feature to reveal the presence of fraud in a transaction.

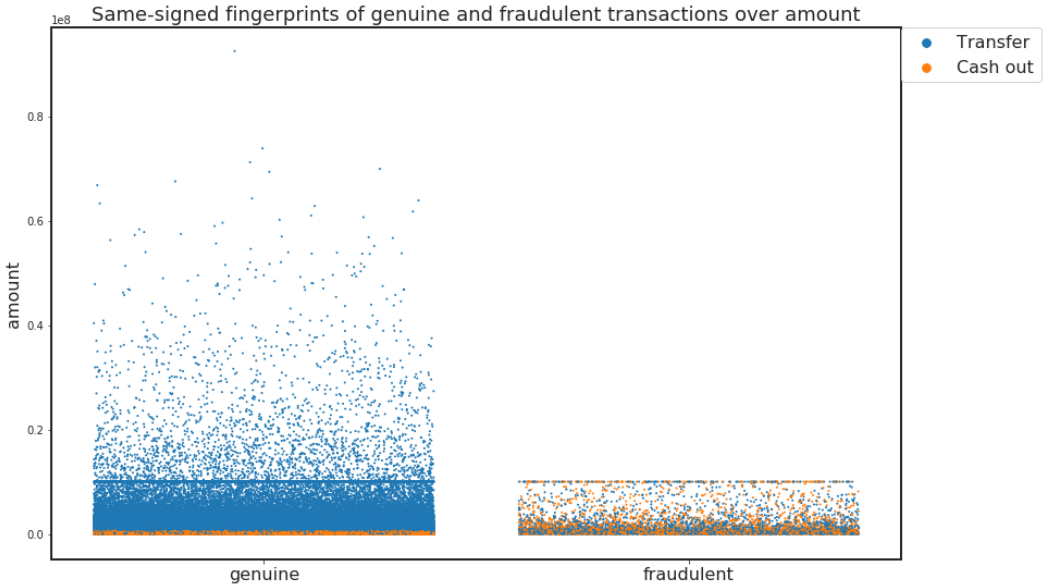

Figure 15: Same-signed fingerprints when looked over amount

c) Looking for dispersion over error in balance in destination accounts

When the dispersion of fraudulent and genuine transactions has been viewed over error in balance in destination accounts using plots below, opposite polarity fingerprints have been plotted and it is discovered that typically fraudulent transactions are “Cash_OUT”.

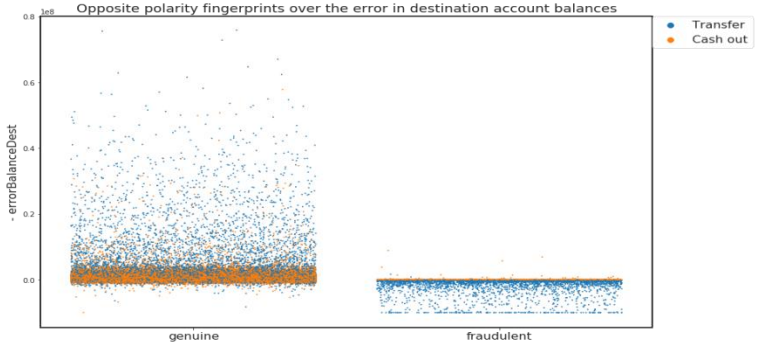

Figure 16: Opposite polarity fingerprints over error in balance in destination accounts

d) Separating out fraudulent and genuine transactions using error-based engineered features

The fraud and non-fraud data has been distinguished using 3D plot below which shows that the original step feature is ineffective in make such distinction.

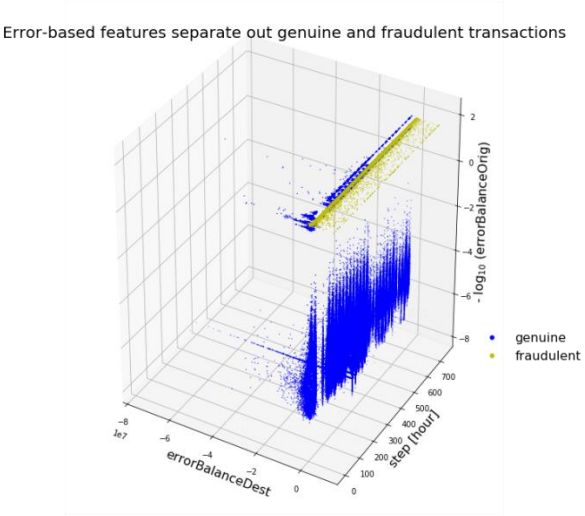

Figure 17: Separating out genuine and fraudulent transactions using error based engineered features

e) Obtaining difference between fraudulent and genuine transactions using correlation heatmap

The correlation heatmaps below provide a comprehensive evidence of difference between fraudulent and genuine transactions. For instance, in case of genuine transactions amount has high correlation with errorBalanceOrig while in case of fraudulent transactions, amount has high correlation with oldBalanceOrig. The correlation heat map also highlights the important features and it can be found that type and newly created feature 'HourofDay' are not important for the output variable.

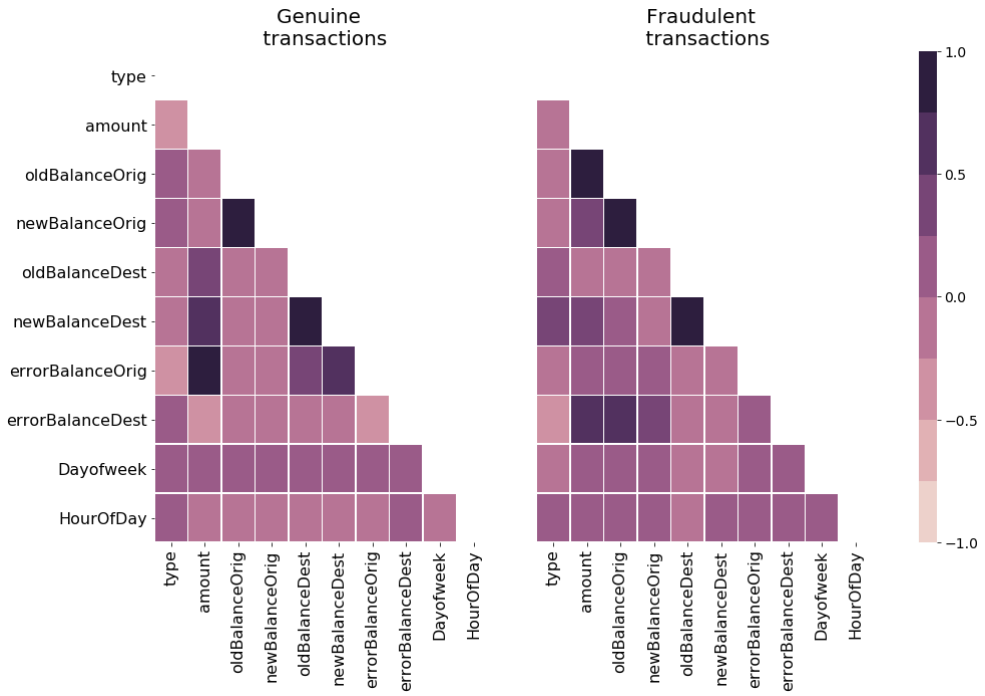

Figure 18: Correlation heatmap of genuine and fraudulent transactions

5. Modelling: Teaching an Algorithm

In this stage of CRISP-DM process, machine learning algorithms are trained to predict labels from the features, tuned them for the business/project need and then validating them on holdout data. Finally, the trained model is the output generated from Modelling which can be used for *inference*, making predictions on new data points and is capable of deploying it on the live data.

To build models for detecting fraud in financial payment services, the task is boiled down to outlier detection i.e. to scan the dataset in order to find potential anomalies in the data. In the past, employees used to handle this task manually but the automation of this process becomes feasible with the rise of machine learning, deep learning and artificial intelligence and other relevant fields of information technology. It helps to save lots of time and intensive amount of labor that is used for fraud detection in financial payment services.

In the following sub-sections, my *Machine Learning* based *Pythonic* approach is explained.

5.1 Machine Learning Framework

For Banks and other financial institutions, Fraud detection is one of the top priorities which can be addressed using Machine Learning. Machine Learning is a branch of artificial intelligence that automates analytical model building through data analysis. The idea behind machine learning is that systems can learn from the data and, then can identify patterns to make decisions without more human intervention. This field has evolved from just pattern recognition to models independently adapting to newly fed data into the system. While we have already witnessed many machine learning algorithms in practice since a long time, the recent development lies in the capability to automatically apply complex mathematical calculations to massive amount of big data.

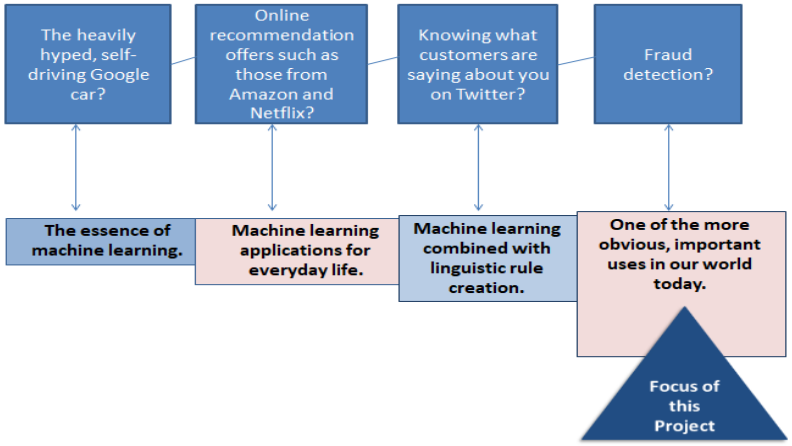

Figure 19: Some examples of widely publicized machine learning applications.

<u>The general framework for machine learning is outlined below:</u>

a) **Prediction Engineering:** It involves the task of defining a project/business need in order to transform it to the machine learning problem for generating labeled examples from a dataset.
b) **Feature Engineering:** It refers to extracting predictor variables for each of the labels from the raw data.
c) **Modelling:** In this step, a machine learning model is trained on the features, tuned for the business/project need so as predictions can be validated before deploying to the new data.

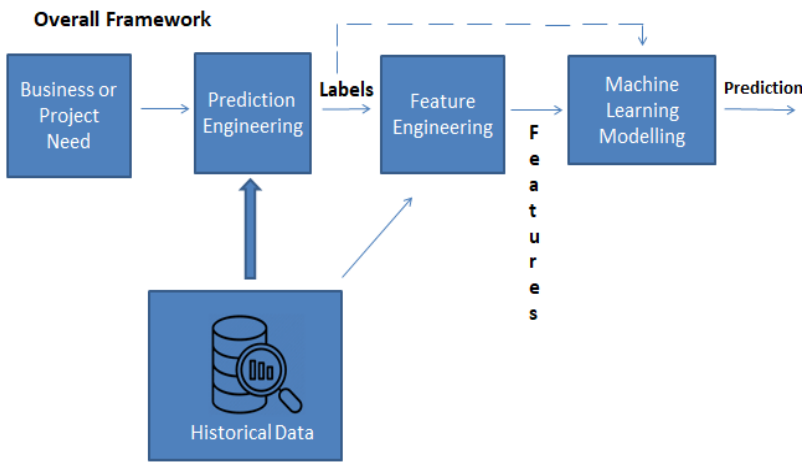

Figure 20: A General machine learning framework

5.1.1 Machine Learning for Fraud Detection

Financial Fraud detection is a challenging problem as criminals are crafty. They have learnt to change their tactics to hamper the organizations. Though fraudulent transactions are rare, but this small fraction of activity can turn into big billion dollar losses in the blink of an eye. Thus, it is very important for financial services organizations to implement right tools and systems in place. Traditionally, companies used to reply upon rule based systems (manual based) for detecting fraud. Such systems can hardly process or adapt to the real time data which is critical for the digital space. But the good news is that with advances in machine learning, systems can identify hidden correlation between user behavior and the likelihood of fraudulent actions.

Rule-based fraud detection	**ML-based fraud detection**
Catching obvious fraudulent scenarios.	Finding hidden and implicit correlations in data.
Requires much manual work to enumerate all possible detection rules.	Automatic detection of possible fraud scenarios.
Multiple verification steps that harm user experience.	The reduced number of verification measures.

Long-term processing.	Real-time processing.

Table 7: Comparison between Traditional rule-based V/S Machine Learning based fraud detection.

Hence, machine learning improves fraud detection accuracy by offering combination of predictive and behavioral analytics techniques that let machines recognize patterns & trends and, then generating predictions based on those observed patterns and trends.

The benefits of machine learning in the area of fraud detection research are as follows:

- **Adjustable to new inputs**: The models built using machine learning adapt to the data and, thus can change over time without the need to repetitively build a model.
- **Making decisions in-time**: Sophisticated analytics should also be built to provide speed and eases business decisions.
- **Discovering new patterns**: New patterns can be uncovered using machine learning techniques to detect changes in fraud behaviors.
- **Less disruption to genuine claims**: Machine learning helps to deter the fraudulent claims as it aims at lowering false positive rate to yield less interruption to genuine claims.
- **Reducing operation cost**: Operational efficiency will be improved with a reduction of the payout on fraudulent claims.

Capgemini claimed that fraud investigation has minimized by 70% as well as detection accuracy has improved by 90% using machine learning.

5.1.2 Machine Learning approaches for Fraud Detection

There are two common approaches in machine learning to bridge a gap between identifying nefarious transactions and maintaining quality customer services. These are divided into:

a) Supervised Learning for Fraud Detection:

This approach involves the task of training an algorithm using labelled historical data. The target variables have already been marked in the existing dataset and the goal of training is to make the system capable of predicting these variables in the future data.

Figure 21: Supervised Machine Learning approach for fraud detection.

Source: https://campus.datacamp.com/courses/fraud-detection-in-python/introduction-and-preparing-your-data?ex=8

b) Unsupervised Learning for Fraud Detection:

This approach features no labels and process unlabeled data. The task is to classify the unlabeled data into clusters to detect hidden relationships between variables in data items.

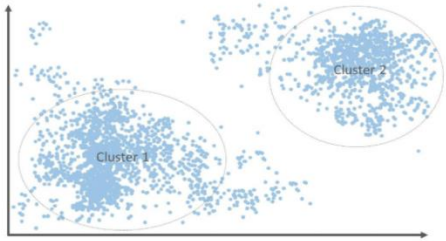

Figure 22: Unsupervised Machine Learning approach for fraud detection.

Source: https://campus.datacamp.com/courses/fraud-detection-in-python/introduction-and-preparing-your-data?ex=8

5.1.3 Towards Supervised Anomaly Detection

Predicting fraud is often related to detecting anomalous behavior in the given data. However, anomaly detection is being regarded as unsupervised learning task as they stem from unlikely events. But, there is mismatch between the required detection rates and the predictive performance of purely unsupervised anomaly detection methods (Görnitz et al., 2013). Therefore, it implies that labeled data is required to guide the model generation and improve the detection accuracy.

For this project, supervised machine learning techniques (explained in sub-section 'Techniques') will be applied on labelled dataset using python (explained in next section).

5.2 Python for fraud detection

Python is most commonly used language for scientific computing. It comprises of mature ecosystem of scientific libraries making it an appealing choice for algorithmic development. For the dataset in hand, following library packages have been used and installed:

a) *NumPy*: it refers to numerical python and is used for basic mathematical functions and belongs to SciPy stack.
b) *Pandas*: it also belong to SciPy stack and is used to read into the CSV file to create data frame (a data structure with both rows and columns in a tabular form).
c) *SkLearn*: it is a free machine learning library for Python. It features various algorithms such as classification, regression and clustering and also supports NumPy and SciPy (Python numerical and scientific libraries).
d) *Matplotlib*: it is used for plotting data and other 2D visualizations.
e)*Seaborn*: It is based on Matplotlib and used for interactive statistical graphics.

5.3 Machine Learning Techniques

Supervised Machine Learning is categorized into Regression and Classification Problem. The difference between the two lies in terms of their output variable i.e. for regression the output is numerical (or continuous) and for Classification, it is categorical (or discrete). In machine learning, problems like detecting fraud in financial payment services are usually framed as classification problems- Given a data observation, the task is to predict a discrete class label output. The classification problems involve creating models that can intelligently categorise the transactions into fraudulent or genuine ones based on transactions details. But modelling fraud detection as a classification problem involves a challenge that fraud data is mostly skewed towards non-fraudulent observations (Besenbruch, 2018) the majority of transactions are legal transactions. Thus, the data is highly imbalanced data as it contains many more samples from one class and few samples from rest of the class (Ganganwar, 2012). The problem of highly imbalanced data can be handled by using sampling methods to make the classifier less sensitive to class imbalances. The several possible techniques are:

i) **Over-sampling:** To achieve balanced distribution, Over-sampling duplicates minority class instances which can lead to the possibility of overfitting the classifier i.e. classifier works well on training data but fails to adequately perform on new data. Eg. SMOTE (Synthetic minority over-sampling technique)

ii) Under-Sampling: To achieve balanced distribution, Under-Sampling reduces the number of observations in the majority class (non-fraudulent transactions) which leads to loss of classification performance because of ambiguity of decision boundary between the classes. E.g. Random under-sampling.

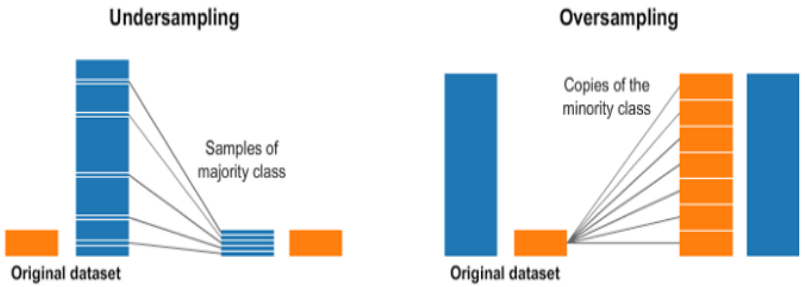

Figure: Over-sampling V/S Under-sampling techniques
Source: https://medium.com/sfu-big-data/winning-against-imbalanced-datasets-14809437aa62

5.3.1 Disadvantages of handling Imbalanced Data using Sampling Methods

In case of random under-sampling method, though it solves the memory problems by reducing the number of training data samples and improves the runtime of the model but necessary information about the data is discarded which could be necessary for building rule-based classifiers like Random Forest. Also, the sample chosen using this method may be a biased sample and will not be treated as an accurate representation of the population in that case. Therefore, it can cause the classifier to poorly perform on real unseen data. Similarly, SMOTE generates artificial minority fraudulent class instances from existing ones rather than duplicating existing instances from the data (Chawla et al., 2003). It works in the feature space instead of the data space. It finds the K-nearest neighbours of each minority instance such that, they also belong to the same class where, *K= (SMOTe %) / 100* and randomly selects one of them in order to create a synthetic instance and then a new minority instance in the neighbourhood is produced by calculating linear interpolations. Though it doesn't result in any information loss but is not very practical for high dimensional data because it avoids considering neighbouring examples that may belong to other classes which further leads to overlapping of classes and creates additional noise.

As using either of above mention method, the model will not adequately perform on real-world skewed test data (see Model Training and Tuning section). So, the objective here is to build a model which deals well with highly skewed data.

5.4 Statistical Techniques

5.4.1 Logistic Regression

Logistic Regression is an extension of linear regression model and is a machine learning algorithm for classification problems. It is based on the concept of probability and is a predictive analysis algorithm. Instead of fitting a straight hyperplane like in linear regression, it uses more complex cost function defined as **'Sigmoid function'** or **'Logistic function'**. Sigmoid function is an S-shaped curve and is used to map predictions to probabilities. The hypothesis of logistic regression tends it to limit the cost function between 0 and 1 where S-shaped curve can take any real-valued number and map it between 0 and 1, but never exactly at those limits.

$$0 \leq h_\theta(x) \leq 1$$

Figure 23: Hypothesis expectation in Logistic Regression
Source: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

In Logistic regression, output value (y) is predicted when input values (x) are combined linearly using weights or coefficient values (referred to as Beta –the Greek capital letter). The output value being modeled in logistic regression is a binary value (0 or 1) instead of a numerical value. It uses following equation as the representation:

y = e^(b0 + b1*x) / (1 + e^(b0 + b1*x))

Where, y= predicted output
b0= the bias or intercept term
b1= the coefficient for single input value (x)

In the input data, each column has an associated b coefficient. The b coefficient is a constant real value that must be learned from your training data and is the actual representation stored in a memory or in a file.

The coefficients or beta values are estimated from the input data using **maximum-likelihood function**. It is a minimization algorithm used to optimize the best values for the coefficients in your training data. The best coefficients are the ones that minimize the error in the probabilities

predicted by the model as compared to those in the data and results in a model that predicts a value very close to 1 for the default class and a very close to 0 for the other class.

Data Preparation for Logistic Regression: To ensure that model is robust and performs well, following assumptions are made:

- **Binary Output Variable:** The probability belonging to a default class will be snapped into a 0 or 1 classification.
- **Remove Noise:** It removes noise in the output variable (y) by considering outliers and misclassified instances from your training data.
- **Gaussian distribution:** For more reliable predictions, the input and output variables must have a Gaussian distribution.
- **Remove Collinearity:** Multiple highly-correlated inputs will overfit your data, so it is necessary to remove them.
- **Fail to converge:** The expected likelihood estimated process can fail to converge because of highly correlated or sparse data (i.e. lots of zeroes in the input data).

5.4.2 Naïve Bayes

A Naïve Bayes algorithm is a probabilistic machine learning model which is based on the Bayes Theorem and is mainly used for classification task. In machine learning, the goal is to select the best hypothesis (h) given data (d) and in classification problem; our hypothesis (h) may be the class to assign for a new data instance (d) using our prior knowledge. Using Bayes' theorem we can use our prior knowledge to calculate the probability of a hypothesis.
Bayes' Theorem is defined as:

$$P(h|d) = (P(d|h) * P(h)) / P(d)$$

Where, the probability of hypothesis h given the data d is represented by **P(h|d)** which is called the posterior probability, **P(d|h)** means that the probability of data d given that the hypothesis h was true, **P(h)** refers that the probability of hypothesis h is true (regardless of the data) which is called the prior probability of h and **P(d)** is the probability of the data (regardless of the hypothesis).

The hypothesis with the highest probability will be selected after calculating posterior probability for a number of different hypotheses. It is called as maximum a posteriori (MAP) hypothesis (the maximum probable hypothesis).

This can be written as:

MAP(h) = max(P(d|h) * P(h))

In classification, the probability of each class (e.g. P(h)) will be equal, if we have an even number of instances in each class in our training data. It is a constant term and could be dropped and the equation will be stated as:

MAP(h) = max(P(d|h))

The probabilities can be calculated in a simple way for each hypothesis to make it tractable, that's why it is called as naïve Bayes or idiot Bayes. They are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on.

Representation for Naive Bayes Models:

For naive Bayes, the representation used is probabilities which are stored to file for a learned naïve Bayes model.

This includes:

- ⇒ Class Probabilities: The probabilities of each class in the training dataset are known as class probabilities.
- ⇒ Conditional Probabilities: It means probabilities of each input value given each class value.

Learning from naïve Bayes model is fast because no coefficients need to be fitted using optimization procedures. It involves calculation of class and conditional probabilities and for new data, predictions can be made using Bayes theorem. The class probabilities are calculated by dividing the frequency of instances (that belong to each class) by the total number of instances. The conditional probabilities are calculated are the frequency of each attribute value for a given class value dividing by the frequency of instances with that class value.

5.5 Ensemble Modelling Techniques

In machine learning, individual models may suffer from various errors due to noise, bias or variance. The solution to such problem is the use of Ensemble methods. Ensemble learning is a method in which multiple machine learning models are combined and constructed strategically to solve a particular problem. Ensemble models tend to be more flexible (less bias) and less data sensitive (less variance).

There are two popular ensemble methods:

- **Bagging:** Bootstrap Aggregation or Bagging is a method in which a bunch of individual models are trained by a random subset of data in a parallel way. E.g. Random Forest, Bagged decision trees, extra trees et al.
- **Boosting:** Bootstrap Aggregation or Bagging is a method in which a bunch of individual models are trained in a sequential way where each model learns from mistakes made by the previous model. E.g. AdaBoost, Stochastic Gradient Descent, XGBoost et al.

For the chosen dataset in hand and to accomplish the goals of this project, Random Forest (Bagging algorithm) and XGBoost (Boosting algorithm) have been used to build models (explained in below section).

5.5.1 Random Forest

Random Forest is an ensemble model in which bagging is featured as an ensemble method and decision tree is represented as the individual model. In this method, the trees are constructed in a way that reduces the correlation between individual classifiers by taking samples of the training dataset with replacement. For each split, subsets of features are selected randomly rather than greedily choosing the best split point while constructing the tree.

*A random Forest model for classification problem is constructed using **RandomForestClassifier** class.*

Framework of Random Forest:

Step 1: From the training set, randomly select n subsets.

Step 2: Train n decision trees.

- One decision tree is trained by one random subset.
- The optimal splits are based on random subset of features and are not selected greedily. (e.g. out of 20 features in total, randomly select 10 to split).

Step 3: The records/candidates in the test set are independently predicted by each individual tree.

Step 4: Final prediction is made by using the class with the majority vote.

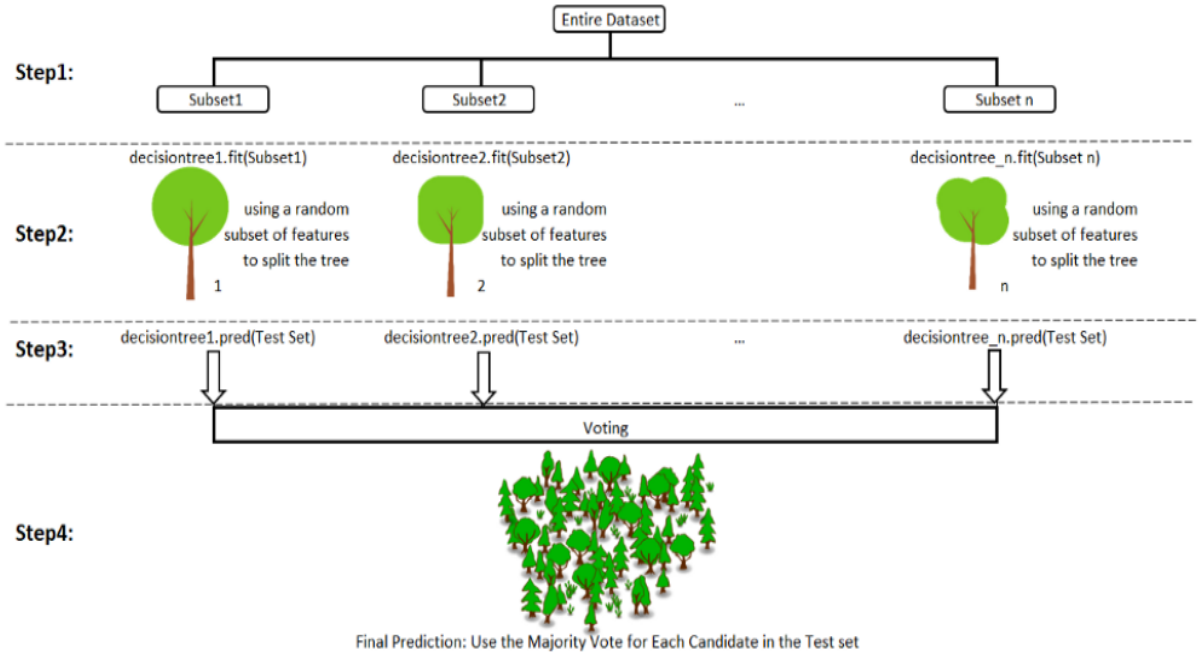

Figure 24: Steps involved in constructing the tree using Random Forest.

Source:https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725

Pros: Random forest is a simplified method to construct a model speedily and can be used with different types of data such as dates, postal codes, credit card numbers, transaction type, IP addresses. They are well known for predicting precisely with datasets that have missing records.

Cons: Random forest sometimes suffer from the problem of overfitting means the model over-remembers the patterns in the training dataset and fails to predict well on future data. Also, if data is highly imbalanced, the accuracy may decrease.

5.5.2 XGBoost Classifier

XGBoost stands for extreme gradient boosting which is similar to framework of Gradient Boosting i.e. model learns from the mistake directly instead of updating the weights of data points but XGBoost is more efficient and more faster than existing gradient boosting algorithm. It has additional features for finding important variables and doing cross validation.

To deal with structured data regardless of type of prediction at hand: Regression or Classification; it has become the 'state-of-the-art' machine learning algorithm.It is important to note that XGBoost works with numeric vectors only. So, all other forms of data need to be converted into numeric vectors. One simple method of converting categorical variable into numeric vector is One Hot Encoding where integer coded variables are removed to add new binary values (0 or 1) for each unique integer value.

Framework of XGBoost Classifier: XGBoost inherited its framework from Gradient Boosting algorithm which involves following steps:
Step 1: It involves training a decision tree.

Step 2: In this step, the recently trained tree is applied for prediction.

Step 3: It involves calculating residual of tree constructed in step 2 and save residual errors as the new y.

Step 4: Repeat step 1 until the number of trees set to train are reached at.

Step 5: In this step, the final prediction is made by simply adding up the predictions of all the trees.

Hyperparameters used in XGBoost: These are categorized into:

a) **General Hyperparameters:** It refers to the type of booster such as tree or linear model used for boosting.
b) **Booster Hyperparameters:** It depends upon which booster has been used for boosting i.e. tree specific or linear specific.
c) **Learning task Hyperparameters:** These Hyperparameters decide on the learning scenario.

The hyperparameters used for this algorithm are explained in Model Training and Tuning section.

Pros: The execution speed of XGBoost is faster than other ensemble classifiers. It has harness the power of multi-core computers as it is parallelizable making it feasible to train on very large datasets. It has wide variety of tuning Hyperparameters for regularization, cross-validation, missing values, scikit-learn compatible APIs, tree parameters, user defined objective functions etc.

Cons: XGBoost classifier requires more computational power than other machine learning algorithms.

6. Model Training, Tuning and Results

From the analysis made in Data Preparation and Exploration section, it is evident that the data now contains adequate information to be used for further process. The steps taken in this section are explained below:

6.1 Train and Test Split: 80:20

To build reliable models which perform better on real-world data, the data was divided into 80:20 ratio as training dataset : test dataset. The purpose of doing so is to test the performance of the trained model on unseen data. This kind of approach helps in getting rid of two common machine learning diseases: Over-fitting (when ML algorithm can remember patterns) and Under-fitting (when ML algorithm cannot remember correlations). This task has been performed using test_train_split method from sklearn library in python.

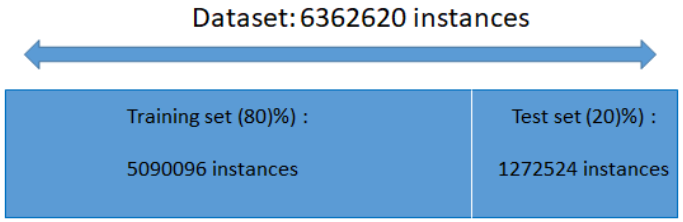

Figure 25: Train-test split approach

6.2 K-fold Cross Validation

Train and Test split approach may suffer from high variance as number of instances in the test set can be increased/ decreased to improve the testing accuracy. To solve such kind of issues, splitting dataset into K-equal folds (number of groups) is a better idea. Out of these groups/folds, one fold is treated as hold-out or test set and the remaining sets as training set. Then testing accuracy of the model is calculated and this process of choosing train and test set continued until all folds are considered. Then training error is computed K-times. At the end, average training accuracy is used as the estimate of the model. In this approach, the value of K can be 3,5,10 and so on. *For the project under consideration, the K=5 has been used.*

The dataset contains 6362620 rows, and fold size is 5 i.e. 1272524 rows in each fold that will iterate 5 times to get training accuracy as:

Training accuracy = (error1+……. +error5) / 5

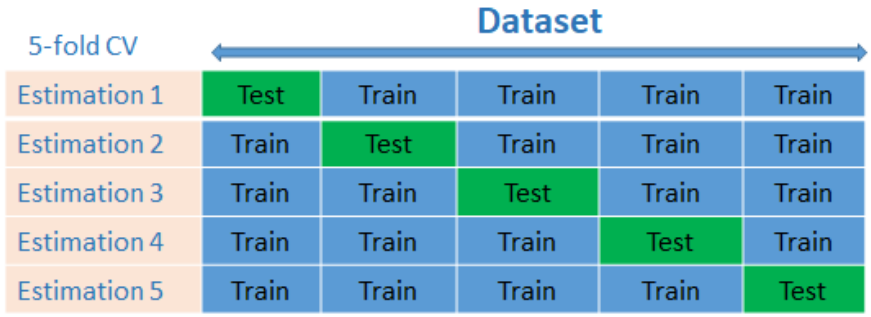

Figure 26: K-fold cross validation approach when K=5

6.3 Feature Selection

Irrelevant features negatively impact the model importance. So, it is very important to find out the important features. The one way to do is by using the feature importance property of the model using a function called plot_importance() (inbuilt class that comes with tree based classifiers). This property order features in terms of their importance towards the output variable. The higher the score, the more important the feature is more the model.

The figure below shows that errorBalanceOrig (a new feature) is the most relevant one followed by Step, newBalanceOrig, oldBalanceDest, NewBalanceDest, Amount, errorBalanceDest and oldBalanceOrigin and type is not important for the output variable.

The above mentioned features other than type have been considered to build the models throughout this section.

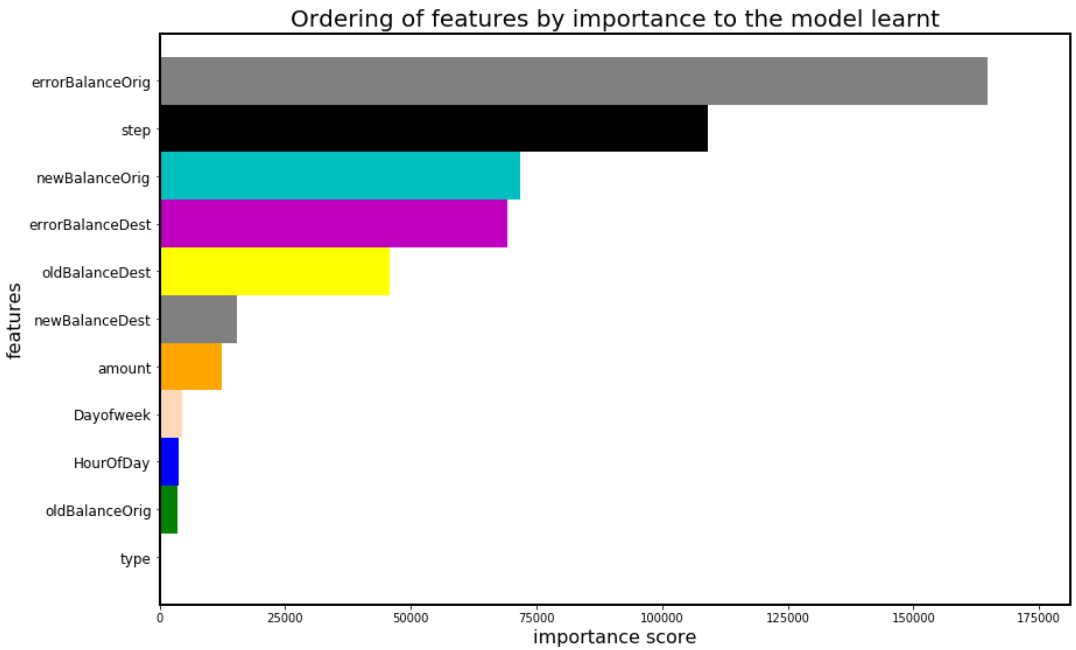

Figure 27: Feature Importance Bar Chart

6.4 Hyperparameter Optimization

In machine learning, a hyperparameter is a model specific property which needs to be tuned before building and testing a model. The purpose of hyperparameter optimization is to achieve high precision and accuracy. Two commonly used techniques for hyperparameter tuning are:

a) **Grid Search-** In this method, a combination of hyperparameter values is tried and accuracy is noted. Once, the evaluation is done for all combinations, the model with the set hyperparameters which gives the best accuracy is retained and considered to be the best.

b) **Random Search**-In this method, some random combination of hyperparameter values is tried to find out the model with best hyperparameter values and best accuracy.

For this project, Grid Search method has been taken into consideration to find out the best hyperparameters for the models.

6.4.1 Statistical Models

As explained in the Modelling section, two statistical machine learning models have been used to train the data and build a model after optimum hyperparameter optimization.

6.4.1.1 Logistic Regression

The model has been trained considering various hyperparameters. The hyperparameters along with the information regarding their corresponding values used for Logistic Regression model are explained below:

- **Penalty**: It specifies which type of regularization to use- L1 or L2 where L1 also known as Lasso regression adds 'squared magnitude' while L2 also known as Ridge regression adds 'absolute value of magnitude' of coefficient as penalty term to the loss function
- **C**: It determines the regularization strength and is actually the inverse of regularization strength (lambda).

Hyperparameters	Values
Penalty	L1, L2
C	-3, 3, 9

Table 8: Hyperparameter Tuning for Logistic Regression

The optimized values of the hyperparameters were obtained using Grid search method with 87% accuracy. The penalty term used to train Logistic Regression is L1 and the numerical value of 3 was used to train this model.

Model Results:

When the model was trained using Logistic Regression, the training accuracy of 0.9969395 was achieved and when it was tested on unseen dataset the accuracy of 0.9969376 was achieved. Also, on validation data, it gives accuracy of 0.893587. The accuracy implies that the model got 89% of the predictions right i.e. it is correctly predicting fraud and non-fraud transactions.

Model	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Accuracy (mean)
Logistic Regression	0.9969395	0.9969376	0.893587

Table 9: Model Results for Logistic Regression

6.4.1.2 Naïve Bayes

The hyperparameters and their corresponding values used for this model have been explained below:

- Multinominal Naïve Bayes: It makes sure that each p(fi. | c) is a multinominal distribution and comes with alpha and fit_prior function which control the form of the model.

- tf-idf pipeline: It stands for term frequency-inverse document frequency which rates importance of a word inside a document and the purpose of pipeline is to cross validate the steps taken to set different hyperparameters. In python,scikit-learn comes with built-in TfidfVectorizer which comes with max_df, binary and norm functions to choose from.

Hyperparameters	Values
Mnb_alpha	np.linspace(0.5, 1.5, 6)

Mnb_fit_prior	[True, False]
tfidf_pip__tfidf_vectorizer__max_df	np.linspace(0.1, 1, 10)
'tfidf_pip__tfidf_vectorizer__binary	[True, False]
'tfidf_pip__tfidf_vectorizer__norm	[None, 'l1', 'l2']

Table 10: Hyperparameter Tuning for Naïve Bayes Classifier

The optimized values of the hyperparameters were obtained using Grid search method with 92% accuracy. The optimized values used to train the model are: 'mnb__alpha'=0.5, 'mnb__fit_prior'=True, 'tfidf_pip__tfidf_vectorizer__max_df'=0.75, 'tfidf_pip__tfidf_vectorizer__binary'=True, 'tfidf_pip__tfidf_vectorizer__norm'=l1.

Model Results:

When the model was trained using Naïve Bayes Classifier, the training accuracy of 0.98497 was achieved and when it was tested on unseen dataset the accuracy of 0.98493 was achieved which shows that the model performs slightly better than Logistic Regression model. Also, on validation data, it gives accuracy of 0.9424 (better than logistic regression model). The accuracy implies that the model got 94% of the predictions right i.e. it is correctly predicting fraud and non-fraud transactions.

Model	**Train-Test Split Performance**		**K-fold (k=5)**
	Train Accuracy	**Test Accuracy**	**Accuracy (mean)**
Naïve Bayes	0.98497	0.98493	0.9424

Table 11: Model Results for Naïve Bayes

6.4.2 Ensemble Models

6.4.2.1 Random Forest

The hyperparameters along with their corresponding values used for this model are explained below:

- n_estimators = It defines how many trees exist in the forest.
- max_features = It measures the max number of features to be considered for splitting a node.
- max_depth = It measures max number of levels in each decision tree.
- min_samples_split = It measures the min number of data points should be placed in a node before splitting the node.
- min_samples_leaf = It measures the min number of data points that should be allowed in a leaf node.
- bootstrap = It is a method for sampling data points (with or without replacement).

Hyperparameters	Values
bootstrap	[True]
max_depth	[80, 90, 100, 110]
max_features	[2, 3]
min_samples_leaf	[3, 4, 5]
min_samples_split	[8, 10, 12]
n_estimators	[100, 200, 300, 1000]

Table 12: Hyperparameter Tuning for Random Forest

The optimized values of the hyperparameters were obtained using Grid search method with 89% accuracy. The optimized values used to train the model are: 'max_depth'=100, 'max_features'=2, 'min_samples_leaf'=3, 'min_samples_split'=10, 'n_estimators'=200.

Model Results:

When the model was trained using Random Forest, the training accuracy of 0.9643 was achieved and when it was tested on unseen dataset the accuracy of 0.9558 was achieved but when experimented using validation data, it gives accuracy of 0.7980. The accuracy implies that the model

got 79% of the predictions right out of 100% predictions i.e. the classifier is not properly identifying fraud and genuine cases. This model doesn't performance better than the two statistical models built.

Model	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Accuracy (mean)
Random Forest	0.9643	0.9558	0.7980

Table 13: Model Results for Random Forest

6.4.2.2 Extreme Gradient Boosting

The hyperparameters and their corresponding values used for this model are explained below:

- N_estimators: The default value is 100. The number of rounds or tree in the forest.
- subsample: The default value is 1 and the range is 0 to 1. During construction of a tree, subsample ratio of the training instance need to be specified to prevent overfitting.
- learning_rate: It is used to control the weights of new trees getting added to the model. It ranges between 0.0001 to 0.1 on a log10 scale.
- max_depth: The default value is 6 and the range is 0 to ∞. It defines the maximum depth of a tree.
- colsample_bytree: The default value is 1 and the range is 0 to 1. During construction of a tree, subsample ratio of columns need to be specified.
- min_child_weight: The default value is 1 and the range is 0 to ∞. It is required to specify minimum sum of instance weight needed in a child.

Hyperparameters	Values
Learning_rate	stats.uniform(0.01, 0.6)
Subsample	stats.uniform(0.3, 0.9)
max_depth	[3, 4, 5, 6, 7, 8, 9]
colsample_bytree	stats.uniform(0.5, 0.9)
min_child_weight	[1, 2, 3, 4]
n_estimators	stats.randint(150, 1000),

Table 14: Hyperparameter Tuning for XGBoost Classifier

The optimized values of the hyperparameters were obtained using Grid search method with 93.34% accuracy. The optimized values used to train the model are: 'n_estimators'=100, 'learning_rate'=0.01, subsample=0.3, 'max_depth'=5, 'colsample_bytree'=0.5, 'min_child_weight'=3.

Model Results:

When the model was trained using XGBoost Classifier, the training accuracy of 0.9995 was achieved and when it was tested on unseen dataset the accuracy of 0.9997 was achieved. Also, on validation data, it gives accuracy of 0.9646. The accuracy implies that the model got 96% of the predictions right i.e. the prediction about number of fraudulent and genuine transactions is more accurate than Logistic Regression, Naïve Bayes and Random Forest. To conclude with the best model, accuracy was evaluated in terms of confusion matrix and other performance metrics as explained in next section (chapter 7).

Model	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Accuracy (mean)
XGBoost Classifier	0.9995	0.9997	0.9646

Table 15: Model Results for XGBoost Classifier

6.5 Model Comparison and Selection of ML Algorithm

Based on the model performance of the above model trained, tested and validated subsequently to correctly identify fraudulent and genuine transactions, it can be discovered that Extreme gradient boosting method *(see table below)* which is an ensemble of decision trees not only performs slightly better than other two statistical models i.e. Logistic Regression and Naïve Bayes but also performs much better than other ensemble technique i.e. Random forest.

Model	Train-Test Split Performance		K-fold (k=5)
	Train Accuracy	Test Accuracy	Accuracy (mean)
Logistic Regression	0.9969395	0.9969376	0.893587
Naïve Bayes	0.98497	0.98493	0.9424
Random Forest	0.9643	0.9558	0.7980

Extreme Gradient Boosting	0.9995	0.9997	0.9646

Table 16: Summary of Model Performances

As accuracy can not only be the sole determinant of the model performance in case of class-imbalanced classification problem, so in order to conclude with the best model, accuracy was evaluated in terms of *performance metrics beyond accuracy* as explained in next section.

7. Evaluation: Measuring Fraud Detection Performance

7.1 Performance Metrics

Performance metrics refers to some metric or a measure used to find out the effectiveness of a trained model using test dataset after doing the adequate feature engineering, model selection and generating the output in probability or class form in relation to problem under consideration. Accuracy (correct predictions/total predictions *100) is one of such measures to assess the model against the test dataset but it may hide the actual details that are important to get diagnosed to understand the performance of a model. That's why the standard metric used to assess the performance of classification problem is Confusion Matrix. Confusion matrix is a reliable measure for to measure the correctness of a classification task (Sokolova et al., 2009)

7.1.1 Confusion Matrix

The confusion matrix is a table that contains the prediction results produced by a binary classifier on a classification problem in question. It consists of two dimensions which display 'Actual' and 'Predicted' results and set of "classes" in both dimensions. The columns in confusion matrix represent 'actual classifications' while rows represent 'predicted classifications'. It overcomes the limitation of accuracy by giving meaningful insights into the type of errors being made.

The classification test dataset produces four possible outcomes which are explained as follows:

i. **True Positives:** In this case, the label which was *predicted positive* (true=1) is *actually positive* (true=1).
ii. **False Positives**: In this case, the label which was *predicted as positive* (true=1) is *actually negative* (false=0).
iii. **True Negatives:** In this case, the label which was *predicted negative* (false=0) is *actually negative* (false=0).

iv. **False Negatives:** In this case, the label which was *predicted as negative* (false=0) is *actually positive* (true=1).

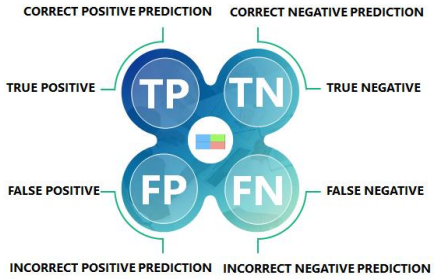

Figure 28: Four outcomes of a classifier in classification problem.

A. Basic Metrics-

On the basis of above four outcomes, two basic measures derived from confusion matrix are: **Error Rate** and **Accuracy**.

- **Error Rate (ERR):** It is calculated as total number of incorrect predictions (FN + FP) divided by the total predictions in a dataset (TP+TN+FP+FN) or (P + N).

Figure 29: Error Rate in confusion matrix
Source: https://classeval.wordpress.com/introduction/basic-evaluation-measures/

- **Accuracy (ACC):** It is calculated as total number of correct predictions (TN + TP) divided by the total predictions in a dataset (TP+TN+FP+FN) or (P + N).

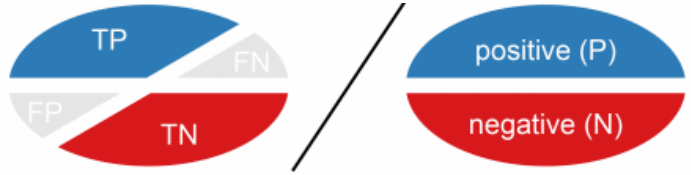

Figure 30: Accuracy in confusion matrix

Source: https://classeval.wordpress.com/introduction/basic-evaluation-measures/

B. Metrics Beyond Error rate & Accuracy-

The more informative measures derived from confusion matrix are:

- **Precision (positive predictive value):** It is calculated as number of correct positive predictions (TP) divided by total number of positive predictions (TP+FP). The best value for precision is 1.0 and the worst one is 0.0.
  ⇒ *Precision = TP / (TP+FP)*
- **Sensitivity (Recall or True positive rate**): It is calculated as number of correct positive predictions (TP) divided by total number of positives (TP+FN) or (P).
  ⇒ *Sensitivity = TP / (TP+FN) or (P).*
- **F1 measure:** It measures Recall and Precision at the same time. It represents harmonic mean of these two measures.
  ⇒ *F1 measure = 2*Precision*Recall / (Precision + Recall).*
- **Specificity (true negative rate)**: It is calculated as number of correct negative predictions (TN) divided by total number of negatives (TN+FP) or (N). The best value for precision is 1.0 and the worst one is 0.0.
  ⇒ *Specificity = TN / (TN+FP) or (N).*
- **Matthew Correlation Coefficient (MCC):** It makes use of all four outcomes in confusion matrix. It is a good evaluation measure for imbalanced dataset.
  ⇒ $$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}$$
- **Area under ROC curve:** An ROC (Receiver operating characteristic) curve plots TPR v/s FPR on a graph that displays the performance of a classification algorithm.

- **Area under Precision-Recall curve:** This curve plots Precision against Recall on a graph to show the performance of a classification algorithm.

7.2 Evaluation of trained models:

The models are evaluated in terms of confusion matrix along with the informative measures derived from this matrix. Please note that throughout the below sub-sections, TP and TN refer to the observations that are correctly predicted (in green colour) and FP and FN means that actual class contradicts with predicted class (in red colour).

7.2.1 When trained models were tested on 20% test data consisting of all type of transactions

The models were firstly evaluated on test data consisting of all type of transactions as follows:

7.2.1.1 Confusion Matrix for Logistic Regression

The outcome of confusion matrix for logistic regression model is explained below:

- **True Positive (TP):** The value of predicted genuine transactions and actual genuine transactions is 689855.
- **True Negative (TN):** The value of predicted fraudulent transactions and actual fraudulent transactions is 1063.
- **False Positives (FP):** The value of predicted genuine transactions is 1008 but in reality, these 1008 are fraudulent transactions which are wrongly classified as genuine ones.
- **False Negative (FN):** The value of predicted fraudulent transactions is 677 but in reality, these 677 are genuine transactions which are wrongly classified as fraudulent ones.

		Predicted Class	
		Class = Genuine	Class = Fraudulent
Actual Class	Class = Genuine	689855 (TP)	677 (FN)
	Class = Fraudulent	1008 (FP)	1063 (TN)

Table 17: Confusion matrix for Logistic Regression model

7.2.1.2 Confusion Matrix for Naïve Bayes

The outcome of confusion matrix for Naïve Bayes model is explained below:

- **True Positive (TP):** The value of predicted genuine transactions and actual genuine transactions is 818283.
- **True Negative (TN):** The value of predicted fraudulent transactions and actual fraudulent transactions is 980.
- **False Positives (FP):** The value of predicted genuine transactions is 1466 but in reality, these 1466 are fraudulent transactions which are wrongly classified as genuine ones.
- **False Negative (FN):** The value of predicted fraudulent transactions is 10394 but in reality, these 10394 are genuine transactions which are wrongly classified as fraudulent ones.

		Predicted Class	
		Class = Genuine	Class = Fraudulent
Actual Class	Class = Genuine	818283 (TP)	10394 (FN)
	Class = Fraudulent	1466 (FP)	980 (TN)

Table 18: Confusion matrix for Naïve Bayes model

7.2.1.3 Confusion Matrix for Random Forest

The outcome of confusion matrix for Random Forest model is explained below:

- **True Positive (TP):** The value of predicted genuine transactions and actual genuine transactions is 690503.
- **True Negative (TN):** The value of predicted fraudulent transactions and actual fraudulent transactions is 1621.
- **False Positives (FP):** The value of predicted genuine transactions is 450 but in reality, these 450 are fraudulent transactions which are wrongly classified as genuine ones.
- **False Negative (FN):** The value of predicted fraudulent transactions is 29 but in reality, these 29 are genuine transactions which are wrongly classified as fraudulent ones.

		Predicted Class	
		Class = Genuine	Class = Fraudulent
Actual Class	Class = Genuine	690503 (TP)	29 (FN)
	Class = Fraudulent	450 (FP)	1621 (TN)

Table 19: Confusion matrix for Random Forest model

7.2.1.4 Confusion Matrix for XGBoost classifier

The outcome of confusion matrix for XGBoost model is explained below:

- **True Positive (TP):** The value of predicted genuine transactions and actual genuine transactions is 828527.
- **True Negative (TN):** The value of predicted fraudulent transactions and actual fraudulent transactions is 2210.
- **False Positives (FP):** The value of predicted genuine transactions is 343 but in reality, these 343 are fraudulent transactions which are wrongly classified as genuine ones.
- **False Negative (FN):** The value of predicted fraudulent transactions is 43 but in reality, these 43 are genuine transactions which are wrongly classified as fraudulent ones.

		Predicted Class	
		Class = Genuine	Class = Fraud
Actual Class	Class = Genuine	828527 (TP)	43 (FN)
	Class = Fraud	343 (FP)	2210 (TN)

Table 20: Confusion matrix for XGBoost model

7.3 Analysis of confusion matrix results

From the above confusion matrices it can be analysed that XGBoost model is the best performer among all models as 13.03% of total genuine transactions and 26.90% of total fraudulent transactions were correctly predicted. Also, only 4.17% of total fraudulent transactions are wrongly classified as genuine ones and only 0.00067% of total genuine transactions are wrongly classified as fraudulent ones. In case of Logistic Regression, Naïve Bayes and Random forest, correctly predicted genuine transactions are 10.85%, 12.87%, 10.86% respectively and correctly predicted fraudulent transactions are 12.94%, 11.93%, 19.73% respectively which is lower than in case of XGBoost model. Also, the percentage of wrongly predicted fraudulent transactions in case of Logistic Regression, Naïve Bayes and Random forest models is 12.24%, 17.84%, 5.47% respectively and the percentage of wrongly predicted genuine transactions in case of Logistic Regression, Naïve Bayes and Random forest models is 0.0106%, 0.163%, 0.00045% respectively which is higher than in case of XGBoost model.

The Precision, Recall, F1-score and ROC derived from above confusion matrices of all models are shown in the table below:

Models	Class	Key performance measures			
		Precision	Recall	F1-score	ROC
Logistic Regression	0	1.00	1.00	1.00	0.94
	1	0.53	0.44	0.48	0.94
	Avg/ Total	1.00	1.00	1.00	-
Naïve Bayes	0	1.00	0.99	0.99	0.94
	1	0.09	0.40	0.14	0.94
	Avg/ Total	1.00	0.99	0.99	-
Random Forest	0	1.00	1.00	1.00	0.97
	1	0.98	0.78	0.87	0.97
	Avg/ Total	1.00	1.00	1.00	-
Extreme Gradient Boosting	0	1.00	1.00	1.00	1.00
	1	0.98	0.87	0.92	1.00
	Avg/ Total	1.00	1.00	1.00	-

Table 21: Evaluation metrics for Machine Learning models when all transactions trained on test data

From the above table, it can be observed that model in case of XGBoost classifier performs better than other models. The interpretation can be derived in terms of precision, recall and f1-measure values of the models built as well as by plotting ROC curve with TPR against FPR. The actual values of the dataset are represented as 'true' and 'false' whereas the values predicted by a classifier are represented as 'positive' and 'negative'. The value of precision in the above table means that classifier in case of XGBoost algorithm was incapable of incorrectly labelling a fraudulent class as a genuine one. Moreover, the precision score for all models is identical in case of genuine transactions (1.00) whereas it differs when comparison was made in terms of fraudulent transactions. Based on precision score, the possibility to incorrectly classify fraud cases as non-fraud cases in case of Logistic Regression is 0.47, Naïve Bayes' is 0.91 whereas it is much lower in case of both ensemble models which is 0.02 (low false positive rate). Recall score measure the classifier's ability to return the valid samples correctly and all models for the project under consideration return identical recall score (1.00) for genuine transactions whereas XGboost classifier outperforms other three models with a recall score of 0.87 to correctly return the fraudulent transactions. F1- score which measure trade-off between precision and recall score.

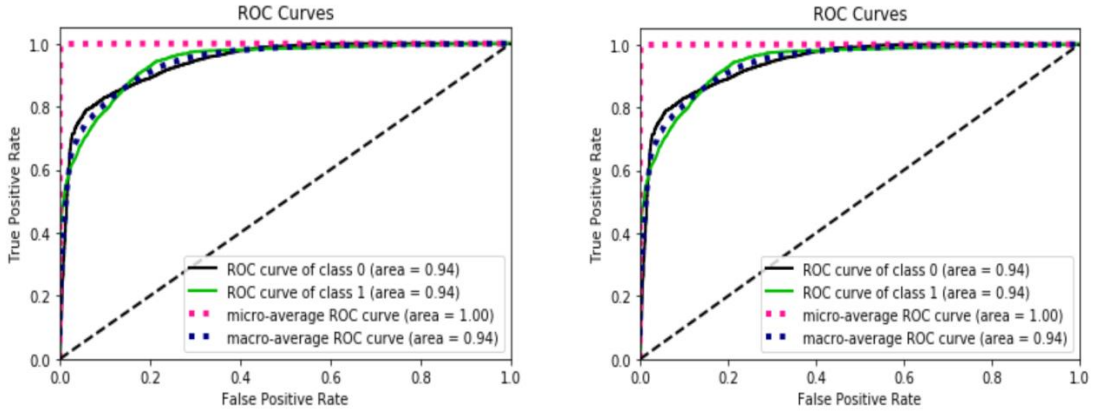

Figure 31: Area under ROC for Logistic Regression (left) and Naïve Bayes (right).

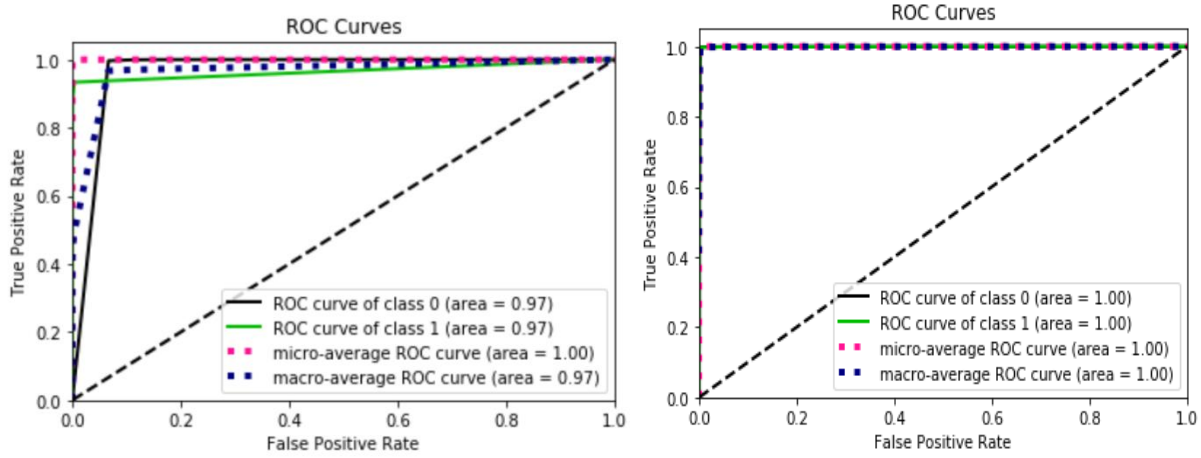

Figure 32: Area under ROC for Random Forest (left) and XGBoost classifier (right).

The area under ROC for both statistical and ensemble models indicates the proportion of genuine transactions labelled actually as 'genuine' by the classifier (true positive rate) against the proportion of fraudulent transactions labelled actually as 'genuine' (false positive rate). Based on the above figures for ROC in terms of all models, it can be interpreted that AUC for XGBoost classifier is more ideal (1.00) than other three classifiers (<1.00) i.e. all transactions predicted as genuine are actually 'genuine'.

7.3.1 When trained models were tested on 20% test data consisting only fraudulent transactions i.e. CASH_OUTs and TRANSFERs

It was also decided to test the trained models only on fraudulent transactions (CASH-OUTs and TRANSFERS) to check how accurately the model predicts the only fraudulent transactions. The accuracy of 42%, 39%, 74%, and 96% was achieved by Logistic Regression, Naïve Bayes, Random Forest and XGB classifier respectively.

7.3.1.1 Confusion Matrix Result for all models

In this case, True Positives and False Negatives are 0 because positive class i.e. genuine transactions are not considered and the results will be based only on the negative class i.e. fraudulent transactions represented in terms of False Positives and True Negatives.

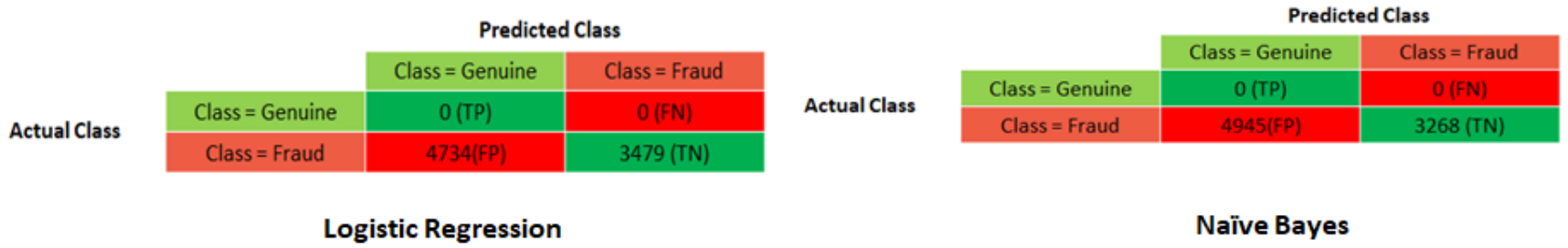

Figure 33: Confusion Matrix analysis of fraudulent transactions on test data using statistical methods.

From the above confusion matrices it can be analysed that in case of logistic regression, out of total 8213 transactions, 42.35% (3479) transactions which belong to fraudulent class were correctly predicted as fraudulent ones and 57.64% (4734) transactions which belong to fraudulent category were wrongly predicted as genuine ones i.e high false positive rate. In case of Naïve Bayes, out of total 8213 transactions, 39.79% (3268) transactions which belong to fraudulent class were correctly predicted as fraudulent ones and 60.20% (4945) transactions which belong to fraudulent category were wrongly predicted as genuine ones i.e high false positive rate. Among statistical methods, Logistic Regression is a better classifier than Naïve Bayes.

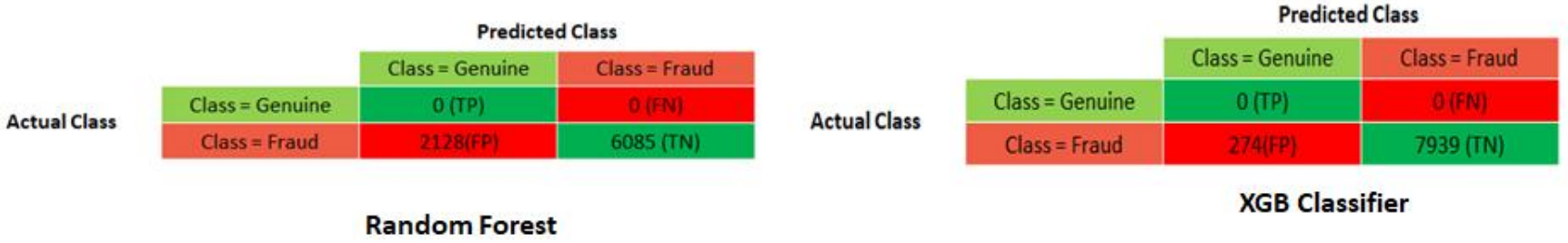

Figure 34: Confusion Matrix analysis of fraudulent transactions on test data using ensemble methods

From the above confusion matrices it can be analysed that in case of random forest, out of total 8213 transactions, 74.08% (6085) transactions which belong to fraudulent class were correctly predicted as fraudulent ones and 25.91% (2128) transactions which belong to fraudulent category were wrongly predicted as genuine ones i.e. low false positive rate than statistical methods but high false positive rate than XGB classifier. In case of XGB classifier, out of total 8213 transactions, 96.66% (7939) transactions which belong to fraudulent class were correctly

predicted as fraudulent ones and 3.33% (274) transactions which belong to fraudulent category were wrongly predicted as genuine ones i.e low false positive rate than other three models. Hence, among all four models; XGB classifier accurately predicts the negative class (true negatives) with low false positives.

The Precision, Recall, F1-score and ROC derived from above confusion matrices of all models are shown in the table below:

Models	Class	Key performance measures		
		Precision	Recall	F1-score
Logistic Regression	0	0	0	0
	1	1	0.42	0.60
Naïve Bayes	0	0	0	0
	1	1	0.40	0.57
Extreme Gradient Boosting	0	0	0	0
	1	1	0.98	0.97
Random Forest	0	0	0	0
	1	1	0.74	0.85

Table 22: Evaluation metrics for Machine Learning models when only fraudulent transactions trained on test data

From the above table, it can be observed that all models have precision score=1 whereas as Recall score differs- 0.42 for Logistic Regression, 0.40 for Naïve Bayes, 0.74 for Random forest and 0.98 for XGB classifier. It portrays than XGB classifier outperforms other classifiers and is accurately predicting the fraudulent transactions.

7.3.1.2 Comparison of Evaluation approaches

Further, the models were compared based on methods stated in section 8.2.1 and section 8.2.2

a) **Logistic Regression:** The false positives increases by 29.41% (3726) when data was tested based on method2 along with the increase in true negatives by 42.33% (2416).
b) **Naïve Bayes:** The false positives increases by 60.18% (3479) when data was tested based on method2 along with the increase in true negatives by 27.86% (2288). Among statistical methods, Naïve Bayes is not a good classifier of negative class.
c) **Random Forest:** The false positives increases by 25.90% (1678) when data was tested based on method2 along with the increase in true negatives by 54.35% (4464).

d) **XGBoost Classifier**: The false positives **decreases** by 3.33% (69) when data was tested based on method2 along with the increase in true negatives by 69.76% (5729).
From the above analysis, it can be observed that the percentage of predicting negative class increased in all the models in case of method2 along with hike in false positives except for XGB classifier. Therefore, it is concluded that XGBoost classifier performed well with both of the methods (section 8.2.1) (section 8.2.2.) and is a better classifier to achieve the set objectives.

7.4 Challenges and Limitations

The key challenge of the project was to deal with a highly skewed big data which took long computing time in terms of dealing with latent missing values, any potential outliers and creating new features. Since, the dataset used to conduct the study is a synthetic dataset which is a replica of specific properties of a real data set and when the machine learning model looks for trends to replicate, it may miss some of the random behaviors leading to biased results. Despite of this fact, synthetic data is an important tool to augment the research in the area of fraud analytics considering the privacy concerns of the financial organizations.

8. Model Deployment: Conclusion

Following CRISP-DM methodology, the final step in the process is to deploy the best model out of all the models experimented on i.e. to deploy the model with better accuracy than the other models. The primary objective of this project was to build a classification model by analysing the transactional data (Cash-In, Cash-Out, Debit, Payment and Transfer) that consists of both normal customer behaviour and fraudulent behaviour to correctly categorize the transactions into fraudulent and non-fraudulent category. Before concluding on to the power of machine learning models to accurately predict the transactions, few research questions in lieu of the problem in questions were investigated into (section 1.2.1) to come up with a reliable solution.

Following sub sections provide insight into Summary, Evaluation and Future work.

8.1 Summary

After the data was supplied to python's jupyter Notebook, it was subject to adequate data preparation through exploratory data analysis (section 4.2.1), data cleaning (4.2.2), feature engineering (section 4.2.3) and data visualization (section 4.2.4) to identify the patterns and to transform data into a form suitable for Modelling. This involved the task of performing statistical analysis on the data to provide answers to the research questions, looking for missing values, outliers; binary encoding, dropping insignificant columns and then performing feature selection for the classification model. To align with the project objectives, CRISP-DM methodology (section 3.1) was adopted. The data used for this

project is a synthetic data and was generated from a simulator called as PaySim (section 3.2). As mentioned in section 5.3 that the data is a highly imbalanced data but it was decided to obtain results without artificially rebalancing the data making the approach suitable for real-world data and keeping in view the disadvantages of using resampling methods (section 5.3.1). For Model training and tuning, statistical and ensemble machine learning techniques were used. The results were obtained by building four models: Logistic Regression & Naïve Bayes (statistical models) and Random Forest & Extreme Gradient Boosting. A grid search method was used for hyperparameter tuning and the data was splitted into 80:20 proportions to hold certain amount of data aside to check the performance of trained model on the unseen data. Further the accuracy achieved by the models was analyzed in terms of confusion matrix (section 7.2.1) (section 7.3.1).

8.2 Evaluation

As the objective of this project (section 1.2.1) was to build a classification model which correctly predict the genuine and fraudulent transactions by undertaking an hybrid approach of rigorous exploratory data analysis and then apply predictive modelling. The set objective was successfully achieved by carefully investigating the research questions. As mentioned, the chosen classification model i.e. XGBoost performed well both on training, test and validation data. The model is a better classifier than other three models as it accurately categorizes the genuine and fraudulent transactions with 96% accuracy (K-fold accuracy) and low false positive rate. Also, when the trained model was tested only on fraudulent transactions 96.66% transactions out of total 8213 transactions were correctly predicted with 3.33% false positive rate. From business point of view, it is very necessary to achieve trade-off between detecting fraudulent samples and misclassifying the genuine ones because it will increase the cost of business and hurt the goodwill and brand image of an organization involved in digital payments. Fraud detection systems also make the businesses (usually merchants) more aware of the compliance in terms of payment acceptance procedures and the need to rebuilding their strategies to adopt the more robust system capable of accurately detecting fraud. Therefore, machine learning comes into the picture and plays a crucial role for providing a platform to execute frictionless digital payment transactions.

8.3 Future Work

Given the challenges and limitations (section 7.4) associated with the project, the future potential work followed from this research is stated as follows:

- Neural networks can also be used to build and evaluate models and then comparison can be made between with statistical, neural networks and ensemble methods. But care will need to be taken regarding the imbalanced data for NNs to avoid conflicting results.

- Paysim dataset can also be interpreted as time series data and then use this property to build time series based models using algorithms like CNN (Convolutional neural networks that use perceptrons to analyze the data). Also, it is worth exploring to re-balance the data without generating spurious transactions or breaking the time series.

- The current approach deals with entire set of transactions as a whole to train the models. User specific models can be created - which are based on user's previous transactional behavior - and use them to further improve our decision making process.

- Blockchain technology could also be used for fraud detection problems as it provides a platform to cryptographically store the encrypted records of all the transactions in an open, secured and transparent environment. In such an environment activities like money laundering are difficult to commit due to a consensus protocol that establish trust between the parties. Also, as the system functions on various devices across different geographical locations, it is impossible to disable the system or to delete or forge any record.

  Despite the above mentioned piece of work not being undertaken due to time constraint for conducting this project, the set objectives of the study have been successfully achieved.

References:

Ahmed, M., Mahmood, A.N. and Islam, M.R., 2016. A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, *55*, pp.278-288.
https://www.sciencedirect.com/science/article/pii/S0167739X15000023

Akhilomen, J., 2013, July. Data mining application for cyber credit-card fraud detection system. In *Industrial Conference on Data Mining* (pp. 218-228). Springer, Berlin, Heidelberg.
https://link.springer.com/chapter/10.1007/978-3-642-39736-3_17

Aleskerov, E., Freisleben, B. and Rao, B., 1997, March. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)* (pp. 220-226). IEEE.
https://ieeexplore.ieee.org/abstract/document/618940

Bănărescu, A., 2015. Detecting and preventing fraud with data analytics. Procedia economics and finance, 32, pp.1827-1836.
https://www.sciencedirect.com/science/article/pii/S2212567115014859

Baars, H. and Kemper, H.G., 2008. Management support with structured and unstructured data—an integrated business intelligence framework. *Information Systems Management*, *25*(2), pp.132-148.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.5544&rep=rep1&type=pdf

Besenbruch, J., 2018. Fraud Detection Using Machine Learning Techniques.
https://science.vu.nl/en/Images/werkstukbesenbruch_tcm296-910176.pdf

Bhowmik, R., 2008. Data mining techniques in fraud detection. *Journal of Digital Forensics, Security and Law*, *3*(2), p.3.
https://commons.erau.edu/cgi/viewcontent.cgi?article=1040&context=jdfsl

Bollen, R., 2010. Best practice in the regulation of payment services. Journal of International Banking Law and Regulation, 2010, p.370.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1747222

Bolton, R.J. and Hand, D.J., 2001. Unsupervised profiling methods for fraud detection. Credit Scoring and Credit Control VII, pp.235-255.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.24.5743&rep=rep1&type=pdf

Bolton, R.J. and Hand, D.J., 2002. Statistical fraud detection: A review. *Statistical science*, pp.235-249.
https://www.jstor.org/stable/3182781?seq=1#metadata_info_tab_contents

Brenig, C., Accorsi, R. and Müller, G., 2015, May. Economic Analysis of Cryptocurrency Backed Money Laundering. In *ECIS*.
https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1019&context=ecis2015_cr

Carcillo, F., Le Borgne, Y.A., Caelen, O., Kessaci, Y., Oblé, F. and Bontempi, G., 2019. Combining unsupervised and supervised learning in credit card fraud detection. *Information Sciences*.
https://www.sciencedirect.com/science/article/pii/S0020025519304451

Carneiro, N., Figueira, G. and Costa, M., 2017. A data mining based system for credit-card fraud detection in e-tail. *Decision Support Systems*, *95*, pp.91-101.
https://www.sciencedirect.com/science/article/pii/S0167923617300027

Carminati, M., Caron, R., Maggi, F., Epifani, I. and Zanero, S., 2015. BankSealer: A decision support system for online banking fraud analysis and investigation. *computers & security*, *53*, pp.175-186.
https://reader.elsevier.com/reader/sd/pii/S0167404815000437?token=323DEF01E2E2D7F998D8384083FC235CB6A95AC41FB54D9BD095D5680B5F4149CE3377817F362E1E200D4B037A1C8BC4

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, pp.321-357.
https://www.jair.org/index.php/jair/article/view/10302

Chawla, N.V., 2003, August. C4. 5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML* (Vol. 3, p. 66).
https://www.dataminingapps.com/2016/11/what-is-smote-in-an-imbalanced-class-setting-e-g-fraud-detection/

Chen, J., Tao, Y., Wang, H. and Chen, T., 2015. Big data based fraud risk management at Alibaba. *The Journal of Finance and Data Science*, *1*(1), pp.1-10.
https://www.sciencedirect.com/science/article/pii/S2405918815000021

Deem, D.L., 2000. Notes from the field: Observations in working with the forgotten victims of personal financial crimes. Journal of Elder Abuse & Neglect, 12(2), pp.33-48.
https://www.tandfonline.com/doi/abs/10.1300/J084v12n02_05

Dutrée, N. and Hofland, D., 2017. Detecting Fraud in Financial Payments.
http://cs229.stanford.edu/proj2017/final-reports/5219328.pdf

Edge, M.E. and Sampaio, P.R.F., 2009. A survey of signature based methods for financial fraud detection. *computers & security*, *28*(6), pp.381-394.
https://www.sciencedirect.com/science/article/pii/S0167404809000091

FATF (2012-2019), International Standards on Combating Money Laundering and the Financing of Terrorism & Proliferation, FATF, Paris, France,
www.fatf-gafi.org/recommendations.html

Flatley, J., 2017. Crime in England and Wales: year ending Sept 2016. Office for National Statistics-Crime Survey of England and Wales (CSEW), London.
http://www.cdeunodc.inegi.org.mx/unodc/wp-includes/js/mapa11/pais/doc/europa/CSEW2016_Resultados_ing.pdf

Ganganwar, V., 2012. An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, *2*(4), pp.42-47.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.413.3344&rep=rep1&type=pdf

Gong, J., McAfee, R.P. and Williams, M.A., 2016. Fraud cycles. *Journal of Institutional and Theoretical Economics*, *172*(3), p.544.
https://www.researchgate.net/profile/Michael_Williams62/publication/307518387_Fraud_Cycles/links/5807753508ae5ed04bfe5aa5/Fraud-Cycles.pdf

Görnitz, N., Kloft, M., Rieck, K. and Brefeld, U., 2013. Toward supervised anomaly detection. *Journal of Artificial Intelligence Research*, *46*, pp.235-262.
https://www.jair.org/index.php/jair/article/view/10802

Gottschalk, P., 2010. Categories of financial crime. Journal of financial crime, 17(4), pp.441-458.
https://www.emeraldinsight.com/doi/full/10.1108/13590791011082797

Gottschalk, P., 2010. Theories of financial crime. Journal of Financial Crime, 17(2), pp.210-222.
https://www.emeraldinsight.com/doi/full/10.1108/13590791011033908

Gray, G.L. and Debreceny, R.S., 2014. A taxonomy to guide research on the application of data mining to fraud detection in financial statement audits. *International Journal of Accounting Information Systems*, *15*(4), pp.357-380.
https://www.sciencedirect.com/science/article/pii/S1467089514000323

Grazioli, S., Johnson, P.E. and Jamal, K., 2006. A cognitive approach to fraud detection. *Journal of Forensic Accounting*, *7*(1), pp.65-88.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=920222

Hand, D.J., 2006. Data Mining. *Encyclopedia of Environmetrics*, *2*.
https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470057339.vad002.pub2

Hawlova, K., 2013. Fraud detection tools. *Journal of Systems Integration*, *4*(4), pp.10-18.
http://www.si-journal.org/index.php/JSI/article/viewFile/173/130

Herawati, N., 2015. Application of Beneish M-Score models and data mining to detect financial fraud. *Procedia-Social and Behavioral Sciences*, *211*, pp.924-930.
https://www.sciencedirect.com/science/article/pii/S1877042815054622

Khandare, N.B., 2016. Credit Card Fraud Detection Using Hidden Markov Model. *INTERNATIONAL JOURNAL*, *1*(4).
http://www.ijasret.com/VolumeArticles/FullTextPDF/37_IJASRET_7759.pdf

Lopez-Rojas, E.A., 2014. *On the simulation of financial transactions for fraud detection research* (Doctoral dissertation, Blekinge Institute of Technology).
http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A834221&dswid=-374

Lopez-Rojas, E., Elmir, A. and Axelsson, S., 2016. PaySim: A financial mobile money simulator for fraud detection. In *28th European Modeling and Simulation Symposium, EMSS, Larnaca* (pp. 249-255). Dime University of Genoa.
http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1058442&dswid=4001

Lopez-Rojas, E.A. and Axelsson, S., 2016, December. A review of computer simulation for fraud detection research in financial datasets. In 2016 Future Technologies Conference (FTC) (pp. 932-935). IEEE.
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7821715

Maes, S., Tuyls, K., Vanschoenwinkel, B. and Manderick, B., 2002, January. Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st international naiso congress on neuro fuzzy technologies* (pp. 261-270).
https://www.researchgate.net/profile/Karl_Tuyls/publication/254198382_Machine_Learning_Techniques_for_Fraud_Detection/links/555f695508ae6f4dcc926e88/Machine-Learning-Techniques-for-Fraud-Detection.pdf

Mudiri, J.L., 2013. Fraud in mobile financial services. *Rapport technique, MicroSave*, p.30.
http://www.microsave.net/files/pdf/RP151_Fraud_in_Mobile_Financial_Services_JMudiri.pdf

Nadali, A., Kakhky, E.N. and Nosratabadi, H.E., 2011, April. Evaluating the success level of data mining projects based on CRISP-DM methodology by a Fuzzy expert system. In *2011 3rd International Conference on Electronics Computer Technology* (Vol. 6, pp. 161-165). IEEE.
https://ieeexplore.ieee.org/abstract/document/5942073

Ngai, E.W., Hu, Y., Wong, Y.H., Chen, Y. and Sun, X., 2011. The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. Decision support systems, 50(3), pp.559-569.
https://www.sciencedirect.com/science/article/pii/S0167923610001302

Okutyi, E., 2012. Safaricom tightens security on m-pesa with fraud management system.
http://www.humanipo.com/news/1341/Safaricomtightens-security-on-M-Pesa-with-Fraud-Management-system

Omolara, A.E., Jantan, A., Abiodun, O.I., Singh, M.M., Anbar, M. and Kemi, D.V., 2018. State-of-the-art in big data application techniques to financial crime: a survey. Int. J. Comput. Sci. Network Secur., 18(7), pp.6-16.
https://www.researchgate.net/profile/Oludare_Omolara/publication/326942577_State-of-The-Art_in_Big_Data_Application_Techniques_to_Financial_Crime_A_Survey/links/5b6d6d24299bf14c6d98a317/State-of-The-Art-in-Big-Data-Application-Techniques-to-Financial-Crime-A-Survey.pdf

Olszewski, D., Kacprzyk, J. and Zadrożny, S., 2013, June. Employing Self-Organizing Map for Fraud Detection. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 150-161). Springer, Berlin, Heidelberg.
https://link.springer.com/chapter/10.1007/978-3-642-38658-9_14

Patidar, R. and Sharma, L., 2011. Credit card fraud detection using neural network. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(32-38).
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.301.8231&rep=rep1&type=pdf

Pickett, K.S. and Pickett, J.M., 2002. Financial crime investigation and control. John Wiley & Sons.
https://books.google.co.uk/books?hl=en&lr=&id=urOcBH4EYYwC&oi=fnd&pg=PR7&dq=related:yeeJVq0nL9gJ:scholar.google.com/&ots=pxc8hQX5nL&sig=6PUomfOvTXiX8Eey5UOMnAJXDTI#v=onepage&q&f=false

Rieke, R., Zhdanova, M., Repp, J., Giot, R. and Gaber, C., 2013, September. Fraud detection in mobile payments utilizing process behavior analysis. In *2013 International Conference on Availability, Reliability and Security* (pp. 662-669). IEEE.
https://ieeexplore.ieee.org/abstract/document/6657303

Robertson, D., 2017. The Nilson Report. HSN Consultants, Inc.
https://nilsonreport.com/upload/content_promo/The_Nilson_Report_10-17-2016.pdf

Sahin, Y., Bulkan, S. and Duman, E., 2013. A cost-sensitive decision tree approach for fraud detection. *Expert Systems with Applications*, *40*(15), pp.5916-5923.
https://www.sciencedirect.com/science/article/pii/S0957417413003072

Sakharova, I., 2012, June. Payment card fraud: Challenges and solutions. In 2012 IEEE International Conference on Intelligence and Security Informatics (pp. 227-234). IEEE.
https://ieeexplore.ieee.org/abstract/document/6284315

Sharma, S. and Osei-Bryson, K.M., 2009. Framework for formal implementation of the business understanding phase of data mining projects. *Expert Systems with Applications*, *36*(2), pp.4114-4124.
https://www.sciencedirect.com/science/article/pii/S0957417408001905

Sharma, S., Osei-Bryson, K.M. and Kasper, G.M., 2012. Evaluation of an integrated Knowledge Discovery and Data Mining process model. *Expert Systems with Applications*, *39*(13), pp.11335-11348.
https://www.sciencedirect.com/science/article/pii/S0957417412002886

Shmais, A.A. and Hani, R., Data Mining for Fraud Detection.
http://cs331project.yolasite.com/resources/Data%20Mining%20for%20Fraud%20Detection.pdf

Smiles, J.A. and Sasi Kumar, A., 2018. Application of Business Intelligence Solutions for Preventing Retail Banking Frauds. *Journal of Advanced Research in Dynamical and Control Systems*, *10*(4).
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3374732

Sokolova, M. and Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), pp.427-437.
https://www.sciencedirect.com/science/article/pii/S0306457309000259

Spathis, C.T., 2002. Detecting false financial statements using published data: some evidence from Greece. *Managerial Auditing Journal*, *17*(4), pp.179-191.
https://www.emerald.com/insight/content/doi/10.1108/02686900210424321/full/html

Sudjianto, A., Nair, S., Yuan, M., Zhang, A., Kern, D. and Cela-Díaz, F., 2010. Statistical methods for fighting financial crimes. *Technometrics*, *52*(1), pp.5-19.
https://www.tandfonline.com/doi/abs/10.1198/TECH.2010.07032

Wang, S., 2010, May. A comprehensive survey of data mining-based accounting-fraud detection research. In *2010 International Conference on Intelligent Computation Technology and Automation* (Vol. 1, pp. 50-53). IEEE.
https://ieeexplore.ieee.org/abstract/document/5522816

Weston, D.J., Hand, D.J., Adams, N.M., Whitrow, C. and Juszczak, P., 2008. Plastic card fraud detection using peer group analysis. *Advances in Data Analysis and Classification*, *2*(1), pp.45-62.
https://link.springer.com/article/10.1007/s11634-008-0021-8

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining.
In *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining* (pp. 29-39). Citeseer.
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.5133&rep=rep1&type=pdf

Yue, D., Wu, X., Wang, Y., Li, Y. and Chu, C.H., 2007, September. A review of data mining-based financial fraud detection research. In *2007 International Conference on Wireless Communications, Networking and Mobile Computing*(pp. 5519-5522). Ieee.
https://ieeexplore.ieee.org/abstract/document/4341127

Appendices

Appendix 1: Categories of Payment Card Fraud

Payment Card Fraud Categories	Description
Account Takeover	It occurs when fraudster captures personal details of a victim first to hijack his account and then requesting bank to change the address and send a new card at new address, making victim unaware of the fraud.
Lost or stolen cards	It occurs when a lost or stolen card is used to purchase goods or services offline or online.
Card-Not-Received Fraud	It occurs when a card has been stolen during its delivery to the customer (i.e. during transit) and used by a fraudster before getting it used by its legitimate user.
Counterfeit Cards	It took place when a skimmed or counterfeit card has been encoded, printed or embossed with genuine card details (which are obtained through data breaches).
ATM Fraud	It took place in two ways: firstly, 'shoulder-surfing' when a PIN is obtained by keeping an eye on a person using the ATM; secondly, 'Lebanese Loop' where a device made of bent metal or plastic is inserted into ATM machine to capture the personal details of a victim.
Card- not present transaction	It happens when a card is not physically present at the place of point-of-purchase and it proves to be fraudulent when enquired into.

```
fraudulent(df,'isFraud','type','isFlaggedFraud','TRANSFER','CASH_OUT',1,0)
```

```
 The types of fraudulent transactions are
['TRANSFER' 'CASH_OUT']

The type of transactions in which isFlaggedFraud is set:\['TRANSFER']

 The number of fraudulent TRANSFERs = 4097

 The number of fraudulent CASH_OUTs = 4116

 the Minimum amount transacted when isFlaggedFraud is:353874.22

 the Maximum  amount transacted in a TRANSFER where isFlaggedFraud is :92445516.64
```

Appendix 2:

```
minmaxbalance()
```

```
 Minimum and Maximum of oldBalanceOrig for isFlaggedFraud = 1 TRANSFERs:  [353874.0, 19585040.0]

 Minimum and Maximum of oldBalanceOrig for isFlaggedFraud = 0 TRANSFERs where oldBalanceOrig =newBalanceOrig: [0.0, 575668.0]
```

Appendix3:

```
Destination()
```

```
Have originators of transactions flagged as fraud transacted more than once? False

Have destinations for transactions flagged as fraud initiated other transactions? False

Within fraudulent transactions, are there destinations for TRANSFERS that are also originators for CASH_OUTs? False

Fraudulent TRANSFERs whose destination accounts are originators of genuine CASH_OUTs:

          step      type      amount     nameOrig  oldbalanceOrg  \
1030443    65  TRANSFER  1282971.57  C1175896731     1282971.57
6039814   486  TRANSFER   214793.32  C2140495649      214793.32
6362556   738  TRANSFER   814689.88  C2029041842      814689.88

         newbalanceOrig     nameDest  oldbalanceDest  newbalanceDest  isFraud  \
1030443             0.0  C1714931087             0.0             0.0        1
6039814             0.0   C423543548             0.0             0.0        1
6362556             0.0  C1023330867             0.0             0.0        1

         isFlaggedFraud type1  Dayofweek  HourOfDay
1030443               0    CC          2         17
6039814               0    CC          3          6
6362556               0    CC          3         18

How many destination accounts of transactions flagged as fraud have been destination accounts more than once?: 2
```

Appendix4:

```
Are there merchants among any originator accounts? False

Are there any transactions having merchants among destination accounts other than the PAYMENT type? False
```

Appendix5:

```
a='C423543548'
tranStep(a)
```

```
Fraudulent TRANSFER C423543548  occured at step: [486] whereas genuine CASH_OUT from this account occured earlier at step :
[185]
```

Appendix6:

```
def cleaning(data,column1,column2,column3,column4,column5,column6,column7,column8,column9):
    global X
    global Y
    global Xfraud
    global XnonFraud
    global fractF
    global fractG
    randomState = 5
    np.random.seed(randomState)
    X = data.loc[(data[column1] == 'TRANSFER') | (data[column1] == 'CASH_OUT')]
    Y = X[column5]
    del X[column5]
    X = X.drop([column2, column3, column4], axis = 1)
    X.loc[X[column1] == 'TRANSFER', 'type'] = 0
    X.loc[X[column1] == 'CASH_OUT', 'type'] = 1
    X[column1] = X[column1].astype(int)
    ################## Imputation of Latent Missing Values
    Xfraud = X.loc[Y == 1]
    XnonFraud = X.loc[Y == 0]
    fractF=len(Xfraud.loc[(Xfraud[column6] == 0) & (Xfraud[column7] == 0) & (Xfraud.amount)])/(1.0 * len(Xfraud))
    fractG=len(XnonFraud.loc[(XnonFraud[column6] == 0) & (XnonFraud[column7] == 0) & (XnonFraud.amount)])/(1.0 * len(XnonFra
    X.loc[(X[column6] == 0) & (X[column7] == 0) & (X.amount != 0),[column6,column7]] = - 1
    X.loc[(X[column8] == 0) & (X[column9] == 0) & (X.amount != 0),[column8,column9]] = np.nan
    print('################################################################################')
    print('\nThe fraction of fraudulent transactions with \'oldBalanceDest\' = \'newBalanceDest\' = 0 although the transacte
    print('################################################################################')
    print('\nThe fraction of genuine transactions with \'oldBalanceDest\' = \newBalanceDest\' = 0 although the transacted \'
    print('################################################################################')
```

Appendix7:

```
cleaning(df,'type','nameOrig', 'nameDest', 'isFlaggedFraud','isFraud','oldBalanceDest','newBalanceDest','oldBalanceOrig', 'ne
```

```
################################################################################

The fraction of fraudulent transactions with 'oldBalanceDest' = 'newBalanceDest' = 0 although the transacted 'amount' is non
-zero is:  0.4955558261293072
################################################################################

The fraction of genuine transactions with 'oldBalanceDest' =
ewBalanceDest' = 0 although the transacted 'amount' is non-zero is:  0.0006176245277308345
################################################################################
```

```
def Fengineering(column1,column2,column3,column4,column5):
    global newcol1
    global newcol2
    newcol1= X[column1] + X[column2]- X[column3]
    newcol2= X[column4] + X[column2 ] - X[column5]
Fengineering('newBalanceOrig','amount','oldBalanceOrig','oldBalanceDest','newBalanceDest')
X['errorBalanceOrig']=newcol1
X['errorBalanceDest']=newcol2
X.head()
```

Appendix8: Please find the project code using following link-

https://github.com/guneet94/Dissertation-Project