Option #1: Capstone Project—Final Report and Slide Presentation: U.S. Organization

Ishita Verma

MIS581– Capstone Project

Colorado State University – Global Campus

Dr. Justin Bateh

April 9, 2022

ABSTRACT

The objective of this paper is identifying fraud in credit card transactions with the help of machine learning algorithms. Since credit card fraud is increasing rapidly through the years despite of the various measures being taken in cybersecurity, this paper studies various classification and predictive models to conclude the best model which can accurately classify fraudulent and non-fraudulent transactions. A credit card owner has a certain a certain pattern or time in which they use card to shop either online or offline. This paper helps explores various machine learning techniques which can be used by banks, ecommerce industry to prevent billions of losses of money and trust from customers resulting from such frauds. The data which is used is real and is explored using R and Python libraries. Models which are built are logistic regression, decision tree, neural network and XGBoost.

I. Introduction

The rapid growth of e-commerce has led to increased use of credit cards and simultaneously complicated the issues related to frauds. Credit card fraud can lead to loss of billions of dollars for any organization and both, merchants, and customers, are affected by its consequences (Nandi, et al., 2022). Credit card fraud detection is the process to identify if a made transaction is normal or not. Moreover, fraudulent transactions can be both online as well offline. The offline fraud is when someone uses other's identity such as name, or credit card number. On the other hand, online fraud is using someone else's credit card at a market shop (Rtayli, et al., 2020). Voican (2021) stated that every person has a certain behavior pattern to spend money. They use certain operating systems, spends large amount of money in certain time, and even has a certain time to complete their transactions. Studying and understanding these patterns can help detect potential frauds. A fraud model can be built by understanding the relationship between the personal habits of card holder and his/her consumptions (Yong, et al., 2019). Fig.1 shows the credit card fraud data from year 2011 to 2018, which shows that despite of various measure to take over cybersecurity, hackers find a loop whole to make a fraud and hence there is increment each year in fraud losses.

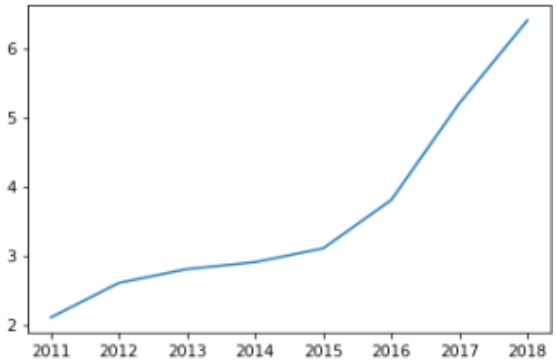


Figure 1. Varying credit card fraud data. Adapted from Uchhana, Ranjan, Sharma, Agarwal, & Punde (2021).*Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection*. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v10i6/C84000110321.pdf

The organization of this paper is as follows. An US banking organization is selected with its details on types of products and services provided by them, number of employees, a short introduction to its history and evolution through the years and its current revenue. A dataset is collected from a public website to study and predict the credit card frauds. A few potential benefits are discussed which can be gained by the analysis of dataset. In next section, research hypothesis, research methods, methodologies and methods are discussed. Along with that, security, data privacy and ethical concerns over the dataset is presented along with the methods to address them. Lastly, data analysis is conducted using various algorithms and tools and results are concluded.

II. Objective

This paper aims to study and analyze dataset of a US banking firm through various machine learning methods to predict fraudulent and genuine transactions to avoid fraud, phishing and data breach. There isn't any standard method to stop it from root cause, but it can be identified by using some methods. So, with the use of machine learning algorithms one can train the model and predict the outcome of the transaction by feeding the model with credit card fraud data and using supervised learning to classify the categories of fraud and secrecy.

III. Overview of Study

Data Analytics is the process of drawing out meaningful insights by analyzing raw data which then are converted into decisions and actions of plan. Questions like is the team structure effective enough, or "when is the right time to start a marketing campaign? And many more questions like these can be derived for successful business strategies (Stevens, 2021). There are various types of techniques and tools for data analysis, but it is highly dependent on the type of data being used and what kind of insights or questions are required. Analyzing data effectively help organizations to make quick and efficient business decisions (Donghyuk, 2020).

Some of the techniques and tools that can be used to analyze data are regression analysis, which is used to check relationships between a set if variables, K-nearest neighbor analysis, decision tree, Logistics Regression, etc. For example, in K-NN algorithm, is used for regression and classification (Shahbazi, et al., 2022). On the other hand, logistic regression is used where the output variable is qualitative and hence it is used where the objective is to predict the appearance or absence of a failure (Robles, et al., 2021). Some of the tools that are used for data analysis are R, SAS, Python, Power BI, Tableau, etc.

A Description About The Dataset Chosen

The selected data is called “Credit Card Fraud Detection” and has been picked up from a public database, named Kaggle. It contains information of transactions made by European cardholders through credit cards in September 2013. The total number of transactions that were made were 284,807 out of which many were fraudulent. Most of the variable names are not self-explanatory due to confidential issues. However, one can implement various algorithms to it to detect fraud activity.

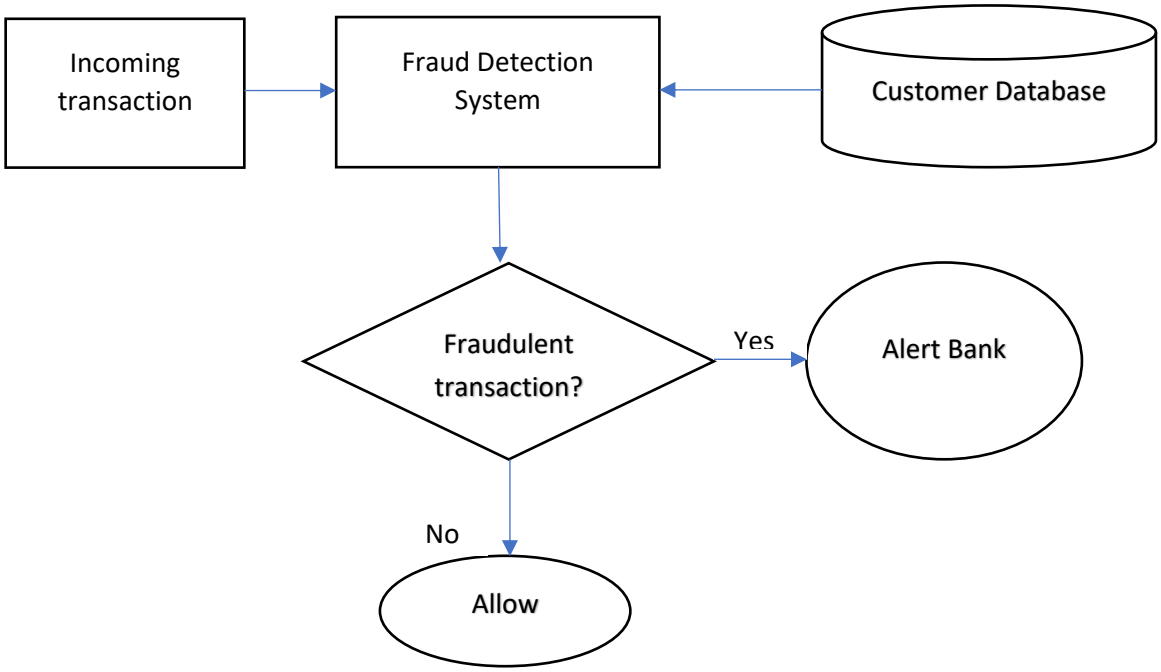
Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise. The variables V1 through V28 are all continuous variables, Time is discrete, class is binary while amount is continuous. Here, quantitative approach will be used as it is highly reliant on data (O’Leary, 2010).

Principle Component Analysis or PCA is a way to handle big data. PCA is a technique that reduces the dimensionality of the variables in a data which increase its interpretability while minimizing the loss of information. It does so by creating new uncorrelated variables that successively maximize variance (Cadima, and Jolliffe, 2016). PCA is an unsupervised learning method and has the functions of signal representation and feature selection, which is a typical method used in pre-processing of regression model (Lee, et al., 2022).

Data Dictionary

Feature	**Data Type**	**Description**
V1 - V28	continuous	28 principal components based on an unknown set of input features that contain information about each transaction
Time	discrete	contains the seconds elapsed between each transaction and the first transaction in the dataset.
Amount	Continuous	the transaction Amount, this feature can be used for example-dependent cost-sensitive learning.
Class	binary	the response variable and it takes value 1 in case of fraud and 0 otherwise**.**

Visual Model of The Data



Data Potential and Its Benefits For The Organization

Studying the data, the organization can measure how much loss has been done in a span of one month (transactions recorded for a month only). Moreover, it can also predict which cards or customers can be at the verge of potential fraud. This can definitely help to save a lot of money and can also improve customer experience and brand trust. Customers' historical data can be combined with customers' insights to create a reporting system which can readily predict any fraudulent activity and huge risks can be mitigated (Worldplay, 2019).

Another benefit of this data can be spotting of outliers and specific patterns. A fraudulent transaction is usually not linear with the non-fraudulent ones. The data can be helpful in giving insight how such outliers can be spotted to act upon (Worldplay, 2019).

The analysis of data can be beneficial in creating a solution-based service for the customers. As important it is to detect and prevent fraud, it is equally important to prevent and catch false

positives. It is highly important for a bank to not create any unnecessary inconvenience to customers and overstepping to prevent frauds. For example, a false positive can lead to decision of blocking a card which may cause inconvenience to a genuine customer. One such solution to prevent these actions is to apply customer rules. Custom rule management allows for modifications on fraud rules based on specific cardholder activity (Worldplay, 2019).

Again, focusing on the solutions to prevent the fraud, BofA can allow its users and customers to share the responsibility of fraud detection by giving them authority to manage their card(s) on mobiles. A level of control can be granted where they can regulate things like the highest amount they usually spend, or activities/areas where they frequently visit, etc. The liberty of control leads to counterfeit the fraud before it is done and also gives leverage to customers to respond in real time to them. The impact to such solutions is intangible to the brands such as BofA as well. They can decrease not only the frauds with the shared efforts but also improves relationship and loyalty with customers (Worldplay, 2019).

The combination of conclusion of analysis and the presented solutions can help BofA to fulfil their values, mission and lead them towards the direction of innovation, improved customer satisfaction and fulling the expectations of their customers.

IV. Research Hypothesis

Hypothesis is concept or a statement which must be tested to prove its credibility, or it can be defined as an idea which can be tested based on the facts available. The most common types of hypotheses are Simple hypothesis, complex hypothesis, null hypothesis, alternative hypothesis, logical hypothesis, empirical hypothesis, and statistical hypothesis (Betts, n.d.). This paper explores null and alternative hypothesis. The null hypothesis (H0) is a statement regarding population that is either believed to be true or is put to argument unless it can be proved wrong or incorrect. Null hypothesis is the most widely used statistical way to make inferences about

population effects (Stunt, Grootel, Bouter, Trafimow, Hoekstra, & Boer (2021). Null hypothesis is also called as conjecture, and in other words, assumes that a difference between two or more variables or features is just by chance. Alternative hypothesis, on the other hand, is the claim to challenge the null hypothesis.

A hypothesis test is carried out at a stated level of significance, and this may be described as "We can't be certain about rejecting the null hypothesis but there is a probability of being wrong that we are prepared to accept and that is the significance level" (Porkess, & Mason (2012). Hayes (2022) states that there are four steps to test statistical hypothesis. Firstly, the two hypotheses are stated. Secondly, formulation of analysis plan is decided where evaluation of data is outlined. Next step involves execution of the plan and analysis of data. Lastly, the final step studies the results and either accept or reject the null hypothesis. The paper follows these steps while examining the credit card fraud dataset.

Hypothesis For the Data

The hypothesis statement for the dataset will be:

Null Hypothesis (H0)- All the transactions are authorized or made by card holders.

Alternative Hypothesis (H1)- All the transactions are not made by card holders and are fraud.

V. Literature Review

Popat and Chaudhary (2018) surveyed on credit card fraud done in two ways, one is credit card fraud virtually while second is when used physically. They also used techniques such as regression, logistic regression, classification, neural network, decision tree. They also used K-NN, support vector machine (SVM), Naïve Bayes etc. After their study through various machine learning processes, they concluded that these mentioned algorithms could provide high accuracy

to detect the fraud cases, and hence can be used to decrease cost and increase profit in various industries such as banking.

Varmedja, et al. (2019) also used various machine learning methods to detect credit card frauds. They used Logistic regression, Random Forest, Multilayer Perception (ANN) and Naïve Bayes. Their ANN model consisted of four hidden layers with accuracy of 99.93%. The accuracy score of logistics regression came out to be 97.46% containing 98 detections of fraud out of 56962 data samples. For the same dataset Naïve Bayes and Random Forest, accuracy score is 99.23% and 99.96% respectively.

Randhawa et, al. (2018) also worked with twelve machine learning models to detect credit card frauds transactions. They used hybrid models by using AdaBoost and majority voting methods. The algorithms that were used are Naïve Bayes, Random Forest, Decision Tree, Gradient Boosted Tree, Decision Stump, Random Tree, Neural Network, Linear Regression, Deep Learning, Logistic Regression, SVM, Multilayer Perceptron. As a result, when standard algorithms used with AdaBoost and majority voting methods under benchmark data the best accuracy and sensitivity acquired by Random Forest algorithm 95% and 91% respectively. When experimented with real-world data the accuracy rate is still above 90% even with 30% noise in the dataset.

VI. Research Design

Methodology

Selecting a research methodology may not be an easy task but can be determined by which approach will answer the research questions. The common types of research methodologies are qualitative and quantitative. Qualitative research attempts to answer research questions through the perceptions or behaviors of the target population while quantitative

research uses statistical data to test hypotheses (O'Leary, 2017). For this paper, only quantitative methodology is used.

Research Method Used

Research methods help defining the approach of how data is collected and evaluated. There are two types of research methods, inductive and deductive (O'Leary, 2010). Inductive research is when a mixture of theories and observation is applied to carry out the research while deductive method uses creating hypothesis statement which is tested using detailed strategy. This paper follows deductive method and analyzes the data to conduct the research.

After standardizing the entire dataset, it is split into training and test set in split ratio of 0.80, hence 80% dataset is train dataset while other 20 is the test dataset. Following are some methods and their results which were completed for the analysis. Tools used were R as well as Python. R and Python, both are open-source programming languages that can be used for statistical purposes, data analysis, graph plotting and machine learning (CSU-Global, 2022).

Techniques And Tools Used

The tools used are R and Python. In this paper total four algorithms are used. These are logistic regression, decision tree, XGBoost classifier and artificial neural network. XGBoost classifier increases the accuracy of classification. This is done by resampling and varying the weights of weak learners in order to reduce the errors, which ultimately increases the accuracy of a model (Bowd, et al., 2020). Using these techniques in R, a classifier is built which helps to detect the fraudulent transactions. Also, Python is being used to carry out the same algorithms.

Logistic Regression Model

Logistic regression is one of the statistical techniques which is used in research designs that analyzes the relationship between one dependent variable with one or more than one

independent variables, when dependent variable has only two categories. For example, dependent variable can be 0 or 1, yes or no, etc. (Salkind, 2010). Here since the dependent variable is 'class', logistic regression is applied on it to predict the value. Logistic regression is a multivariate statistical analysis (MSA) method which assess the influence of all the independent variables on dependent variable and assess correlations among all factors (Tehrany, Shabani, Jebur, Haoyuan, Wei, & Xiaoshen, 2017).

Decision Tree

Decision tree is a non-parametric supervised leaning method used for classification and regression. It enables one to weigh or evaluate possible actions based on probabilities, costs, or benefits by starting with single node and then branching to several probable outcomes. To implement this model, rpart.plot() function is used to work in R language and DecisionTreeClassifier() function in Python. Decision tree is used to classify or predict future observations (test data in this case) on the basis of set of decision rules. IBM (2021) states that decision tree model is also known as rule induction and has several advantages including the logic behind the decision is clear unlike black box model, tree ignore the attributes which do not contribute and hence low noise, and lastly, the model can easily be converted to an if-then rules which is a comprehensive form and easy to understand. The results of decision tree are assessed by calculating Area under curve (AUC) and ROC curve and the null hypothesis is rejected.

Artificial Neural Network

Artificial Neural Network (ANN) are a type of machine learning algorithm that is based on and is similar to human nervous system. The ANN models learns patterns using given data, and hence is supervised method which is used to conduct classification and predictions. Nufer and Muth (2022) states that a multilayer perception is a common form of artificial neural

network. It allows and processes multiple variables at input, which is then related to one or more than one independent variables in an output layer. Also, hidden layer(s) between the input and output allows to identify nonlinear relationships between the variables as well. To implement this model, neuralnet package is imported. Then using plot function, plot is built. Now, in the case of Artificial Neural Networks, there is a range of values that is between 1 and 0. A threshold as 0.5 is set, that is, values above 0.5 will correspond to 1 and the rest will be 0 (Kaggle, 2022). The end result is evaluated based on the AUC.

XGBoost

XGboost is used for classification as well as regression tasks. It comprises of multiple underlying ensemble models like decision tress (Data Flair, 2022). All the decision trees are combined to form a model of gradient boosting which is stronger and have greater accuracy. Applying the model to produce plot for Bernoulli deviance vs iteration, area under curve was around 95.55 which resulted in rejection of null hypothesis. Bernoulli deviation is just a discrete distribution of probability, i.e. having only two outcomes (For e.g. 0.5 probability of getting heads or tails while tossing a coin) (GeeksforGeeks, 2021).

VII. Limitations

While this paper couldn't reach out goal of 100% accuracy in fraud detection, it ended up creating a system that can, with enough time and data, get very close to that goal. As with any such project, there is some room for improvement here. The very nature of this project allows for multiple algorithms to be integrated together as modules and their results can be combined to increase the accuracy of the final result. This model can further be improved with the addition of more algorithms into it. However, the output of these algorithms needs to be in the same format as the others. Once that condition is satisfied, the modules are easy to add as done in the code. This provides a great degree of modularity and versatility to the project. More room for

improvement can be found in the dataset. It is evident that the precision of the algorithms increases when the size of dataset is increased. Hence, more data will surely make the model more accurate in detecting frauds and reduce the number of false positives. However, this requires official support from the banks themselves.

VIII. Ethical considerations

With several types of data on different applications and systems, data security is a necessary act of keeping data safe from unwanted access. To determine security techniques to protect data, it is important to understand how data is being used, stored and how it is accessed. The lack of security skills, security measure being taken, and data breaches are some of the challenges which come along with big and sensitive data. Data breaches can result in alteration and corruption of data, and stealing which can impact identities of individuals, fine to organizations due to failure in keeping data safe and following privacy mandates like General Data Protection Regulation (GDPR). Hence data security policies are written to explain how to handle such challenges (CSU- Global, 2022). Recently, WhatsApp released a privacy policy which forces its customers to share their data with Facebook by accepting the terms. Amidst such rising concerns over privacy, companies need to ensure accountability and transparency with customers (Bradley, 2020). From a privacy and security perspective, the challenge is to ensure that data subjects (i.e., individuals) have sustainable control over their data, to prevent misuse and abuse by data controllers (i.e., big data holders and other third parties), while preserving data utility, i.e., the value of big data for knowledge/ patterns discovery, innovation, and economic growth. Fraud is one of ethical issues whose complexity is increasing day by day. This paper explores some of the security risks in credit card payments, privacy rights of customers and challenges and concerns related to them and the data analysis.

Challenges While Performing Data Analysis

Oracle (2002) shared some of the fundamental requirements of data security, which are confidentiality, integrity, and availability. Confidentiality allows one to see only the data which they are supposed to see. Confidentiality has several aspects like privacy of communications, which is ability to control the sharing of private information. This information can be related to health, credit card, or employment for an individual and trade secrets, etc., for businesses. Second aspect of confidentiality is sensitive data storage, that is ensuring that data remain private after collection. Thirdly, authentication of users is another aspect. This allows rights to see data to different individuals and organizations and hence implies decision making abilities to the holder. Lastly, granular access control is the amount of data that should be used/seen by user.

Integrity is data protection from being corrupted or deleted while it is in database or while its being transmitted. Data is protected against viruses, eavesdropping, and maintaining relationships between the values present in database. Integrity is also control of privilege and control access where only authorized users are able to make changes in data and should be a concern especially when the data is access remotely (Guangwei, Shan, Miaolin, Yanglan, Xiangyang, Qiubo, Li Li, & Wei Li, (2022)

Lastly, availability is making data available to authorized users on time and without delay. For example, denial of service attacks is prohibition of legible users to access data. Some of the aspects of availability are resistance to deliberate attacks, scalability as per demand, flexibility in managing the user population and ease of use or ease of ability provided to users to complete their work.

Ethical conduct is essential in inspiring confidence in public and simultaneously overcoming the concerns and reluctance arising from analyzing their data (Choma, 2014).

According to Taylor and Pagliari (2018), the greatest challenge for researchers are ethical ones, for example an unclear boundary between private or public data and ensuring privacy of data subjects on a social media platform.

The dataset used contains information like name of credit card holder, card number, or CVV code, etc. Hence the challenges of confidentiality, maintaining integrity and availability is critical in this case. Moreover, with data mining solutions, an extra security layer is required for this sensitive dataset. This will ensure that data is protected against internal as well as external threats of data breaches.

Plans, Tools, And Techniques To Address The Challenges

Some of the ways data can be protected against cyber-attacks, data theft or even human errors are data discovery and classification, data encryption, dynamic data masking (DDM), user and entity behavior analytics (UEBA), change management and auditing, identity, and access management (IAM) and backup and recovery (Tierney, 2021).

Data classification is a process of labelling data and giving them tags which protects data according to its regulatory requirements and values while data discovery is scanning sensitive data repositories. Logrippo, (2021) states that labelling is a time-honored method which assign entities to security levels. The same process is conventionally used in high security areas such as military. Typically, labels are tuples with elements are ordered domains or ordered sets.

Further, data encryption is making data unreadable. This is done is two ways, one is software data encryption, and other is hardware encryption. In software encryption data is stored in SSD after using software to encrypt it while in hardware a sperate processor is used to encrypt and is stored on devices like USB. DDM involves masking data while not changing the original data. Another method is UEBA which indicates unnormal activity in order to identify threat.

Lastly, other plans like regularly keeping backup, having strict change management helps in securing data.

To ensure data security and privacy for this dataset, the features in the dataset are renamed and given pseudo names to maintain integrity and privacy. The values in each row and column are converted using principle component analysis. This permits confidentiality of data, hence providing availability and integrity (ethical) at the same time. This encryption is done to improve data security and to eliminate potential threats during transmission of data (Wu, Dai, Maseleno, Yuan, & Balas, 2020).

Privacy and Ethical Concerns In Visualizations

Visualizations are a powerful tool to influence decision making which also connects duties and ethical responsibilities which come along while presenting them. It is important to understand and be aware of the fact that data is not unbiased. It is always been gathered and processed with a an objective. Some of the ethical challenges faced by people during visualization are visibility, privacy and power (Github, 2019).

With visibility, designers must struggle to make invisible visible. They should make visualizations compatible with the literacy of audiences to be addressed. Managing complexity is, therefore, a virtue in design that can be in direct opposition with the desire to visualize the invisible.

Privacy is again gathering data with empathy. Collecting the required type and amount of data directly impacts quality and scope of the analytical results. Hence small but good data collection is encouraged. Each datum is considered useless if there is no context and its not associated with metadata. Collection of such data can be easily misused (Rafferty, Whitehill, Romero, Cavalli-Sforza, & International Educational Data Mining Society, 2020).

Power is the controlling oneself to deliver ethics. Hence, the goal is to deliver the truth without any bias or suppressing the falsehood. This requires expertise and practice to achieve.

Till now, the paper covered the objective of this paper, i.e., analyzing a raw credit card transactional data to identify and predict fraudulent transactions. Several potential benefits of the same has been discussed including saving money, increasing customers' trust, proactive and strategic plans of actions that can be avoided with early detection of frauds etc. Hypothesis statement was then tested using quantitative methodology and several methods like logistic regression, decision tree, ANN, and XGBoost. Ethical concerns regarding data and ways to protect the privacy of data were also discussed. Techniques like PCA and encryption were used to achieve the data privacy and to avoid data breach.

IX. FINDINGS

This section will share the results and conclusions for rejecting or accepting the null hypothesis. Below are all the models that were built and their results in R as well as in Python. The complete project can be found on Github with link here

Data Upload and Exploration

Data is uploaded and explored. The head function in Fig. 2 shows the first six rows of the data and str() in Fig. 3 shows that there are total of 248807 observations and 31 variables.

Figure 2

Head of dataset

```
> head(credit_card)
  Time         V1          V2        V3         V4          V5          V6          V7          V8         V9
1    0 -1.3598071 -0.07278117 2.5363467  1.3781552 -0.33832077  0.46238778  0.23959855  0.09869790  0.3637870
2    0  1.1918571  0.26615071 0.1664801  0.4481541  0.06001765 -0.08236081 -0.07880298  0.08510165 -0.2554251
3    1 -1.3583541 -1.34016307 1.7732093  0.3797796 -0.50319813  1.80049938  0.79146096  0.24767579 -1.5146543
4    1 -0.9662717 -0.18522601 1.7929933 -0.8632913 -0.01030888  1.24720317  0.23760894  0.37743587 -1.3870241
5    2 -1.1582331  0.87773675 1.5487178  0.4030339 -0.40719338  0.09592146  0.59294075 -0.27053268  0.8177393
6    2 -0.4259659  0.96052304 1.1411093 -0.1682521  0.42098688 -0.02972755  0.47620095  0.26031433 -0.5686714
         V10        V11         V12        V13        V14        V15        V16         V17         V18         V19
1  0.09079417 -0.5515995 -0.61780086 -0.9913898 -0.3111694  1.4681770 -0.4704005  0.20797124  0.02579058  0.40399296
2 -0.16697441  1.6127267  1.06523531  0.4890950 -0.1437723  0.6355581  0.4639170 -0.11480466 -0.18336127 -0.14578304
3  0.20764287  0.6245015  0.06608369  0.7172927 -0.1659459  2.3458649 -2.8900832  1.10996938 -0.12135931 -2.26185710
4 -0.05495192 -0.2264873  0.17822823  0.5077569 -0.2879237 -0.6314181 -1.0596472 -0.68409279  1.96577500 -1.23262197
5  0.75307443 -0.8228429  0.53819555  1.3458516 -1.1196698  0.1751211 -0.4514492 -0.23703324 -0.03819479  0.80348692
6 -0.37140720  1.3412620  0.35989384 -0.3580907 -0.1371337  0.5176168  0.4017259 -0.05813282  0.06865315 -0.03319379
         V20         V21         V22         V23         V24        V25         V26         V27         V28 Amount
1  0.25141210 -0.018306778  0.277837576 -0.11047391  0.06692807  0.1285394 -0.1891148  0.133558377 -0.02105305 149.62
2 -0.06908314 -0.225775248 -0.638671953  0.10128802 -0.33984648  0.1671704  0.1258945 -0.008983099  0.01472417   2.69
3  0.52497973  0.247998153  0.771679402  0.90941226 -0.68928096 -0.3276418 -0.1390966 -0.055352794 -0.05975184 378.66
4 -0.20803778 -0.108300452  0.005273597 -0.19032052 -1.17557533  0.6473760 -0.2219288  0.062722849  0.06145763 123.50
5  0.40854236 -0.009430697  0.798278495 -0.13745808  0.14126698 -0.2060096  0.5022922  0.219422230  0.21515315  69.99
6  0.08496767 -0.208253515 -0.559824796 -0.02639767 -0.37142658 -0.2327938  0.1059148  0.253844225  0.08108026   3.67
  Class
1     0
2     0
3     0
4     0
5     0
6     0
```

Figure 3

Str() function on dataset

```
> str(credit_card)
'data.frame':   284807 obs. of  31 variables:
 $ Time  : num  0 0 1 1 2 2 4 7 7 9 ...
 $ V1    : num  -1.36 1.192 -1.358 -0.966 -1.158 ...
 $ V2    : num  -0.0728 0.2662 -1.3402 -0.1852 0.8777 ...
 $ V3    : num  2.536 0.166 1.773 1.793 1.549 ...
 $ V4    : num  1.378 0.448 0.38 -0.863 0.403 ...
 $ V5    : num  -0.3383 0.06 -0.5032 -0.0103 -0.4072 ...
 $ V6    : num  0.4624 -0.0824 1.8005 1.2472 0.0959 ...
 $ V7    : num  0.2396 -0.0788 0.7915 0.2376 0.5929 ...
 $ V8    : num  0.0987 0.0851 0.2477 0.3774 -0.2705 ...
 $ V9    : num  0.364 -0.255 -1.515 -1.387 0.818 ...
 $ V10   : num  0.0908 -0.167 0.2076 -0.055 0.7531 ...
 $ V11   : num  -0.552 1.613 0.625 -0.226 -0.823 ...
 $ V12   : num  -0.6178 1.0652 0.0661 0.1782 0.5382 ...
 $ V13   : num  -0.991 0.489 0.717 0.508 1.346 ...
 $ V14   : num  -0.311 -0.144 -0.166 -0.288 -1.12 ...
 $ V15   : num  1.468 0.636 2.346 -0.631 0.175 ...
 $ V16   : num  -0.47 0.464 -2.89 -1.06 -0.451 ...
 $ V17   : num  0.208 -0.115 1.11 -0.684 -0.237 ...
 $ V18   : num  0.0258 -0.1834 -0.1214 1.9658 -0.0382 ...
 $ V19   : num  0.404 -0.146 -2.262 -1.233 0.803 ...
 $ V20   : num  0.2514 -0.0691 0.525 -0.208 0.4085 ...
 $ V21   : num  -0.01831 -0.22578 0.248 -0.1083 -0.00943 ...
 $ V22   : num  0.27784 -0.63867 0.77168 0.00527 0.79828 ...
 $ V23   : num  -0.11 0.101 0.909 -0.19 -0.137 ...
 $ V24   : num  0.0669 -0.3398 -0.6893 -1.1756 0.1413 ...
 $ V25   : num  0.129 0.167 -0.328 0.647 -0.206 ...
 $ V26   : num  -0.189 0.126 -0.139 -0.222 0.502 ...
 $ V27   : num  0.13356 -0.00898 -0.05535 0.06272 0.21942 ...
 $ V28   : num  -0.0211 0.0147 -0.0598 0.0615 0.2152 ...
 $ Amount: num  149.62 2.69 378.66 123.5 69.99 ...
 $ Class : int  0 0 0 0 0 0 0 0 0 0 ...
>
```

Fig. 4 shows the summary part of the statistical summary for each feature. The summary includes minimum, maximum values, standard deviation, mean, medium, etc. Further, Fig. 5 shows that there are no missing values in the dataset.

Figure 4

Summary of each feature in the dataset

```
> summary(credit_card)
      Time              V1                  V2                 V3                  V4                  V5
 Min.   :     0   Min.   :-56.40751   Min.   :-72.71573   Min.   :-48.3256   Min.   :-5.68317   Min.   :-113.74331
 1st Qu.: 54202   1st Qu.: -0.92037   1st Qu.: -0.59855   1st Qu.: -0.8904   1st Qu.:-0.84864   1st Qu.:  -0.69160
 Median : 84692   Median :  0.01811   Median :  0.06549   Median :  0.1799   Median :-0.01985   Median :  -0.05434
 Mean   : 94814   Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.0000   Mean   : 0.00000   Mean   :   0.00000
 3rd Qu.:139321   3rd Qu.:  1.31564   3rd Qu.:  0.80372   3rd Qu.:  1.0272   3rd Qu.: 0.74334   3rd Qu.:   0.61193
 Max.   :172792   Max.   :  2.45493   Max.   : 22.05773   Max.   :  9.3826   Max.   :16.87534   Max.   :  34.80167
       V6                  V7                  V8                  V9                 V10                 V11
 Min.   :-26.1605   Min.   :-43.5572   Min.   :-73.21672   Min.   :-13.43407   Min.   :-24.58826   Min.   :-4.79747
 1st Qu.: -0.7683   1st Qu.: -0.5541   1st Qu.: -0.20863   1st Qu.: -0.64310   1st Qu.: -0.53543   1st Qu.:-0.76249
 Median : -0.2742   Median :  0.0401   Median :  0.02236   Median : -0.05143   Median : -0.09292   Median :-0.03276
 Mean   :  0.0000   Mean   :  0.0000   Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.00000   Mean   : 0.00000
 3rd Qu.:  0.3986   3rd Qu.:  0.5704   3rd Qu.:  0.32735   3rd Qu.:  0.59714   3rd Qu.:  0.45392   3rd Qu.: 0.73959
 Max.   : 73.3016   Max.   :120.5895   Max.   : 20.00721   Max.   : 15.59500   Max.   : 23.74514   Max.   :12.01891
      V12                 V13                 V14                 V15                 V16                 V17
 Min.   :-18.6837   Min.   :-5.79188   Min.   :-19.2143   Min.   :-4.49894   Min.   :-14.12985   Min.   :-25.16280
 1st Qu.: -0.4056   1st Qu.:-0.64854   1st Qu.: -0.4256   1st Qu.:-0.58288   1st Qu.: -0.46804   1st Qu.: -0.48375
 Median :  0.1400   Median :-0.01357   Median :  0.0506   Median : 0.04807   Median :  0.06641   Median : -0.06568
 Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.0000   Mean   : 0.00000   Mean   :  0.00000   Mean   :  0.00000
 3rd Qu.:  0.6182   3rd Qu.: 0.66251   3rd Qu.:  0.4931   3rd Qu.: 0.64882   3rd Qu.:  0.52330   3rd Qu.:  0.39968
 Max.   :  7.8484   Max.   : 7.12688   Max.   : 10.5268   Max.   : 8.87774   Max.   : 17.31511   Max.   :  9.25353
      V18                 V19                 V20                 V21                 V22
 Min.   :-9.498746   Min.   :-7.213527   Min.   :-54.49772   Min.   :-34.83038   Min.   :-10.933144
 1st Qu.:-0.498850   1st Qu.:-0.456299   1st Qu.: -0.21172   1st Qu.: -0.22839   1st Qu.: -0.542350
 Median :-0.003636   Median : 0.003735   Median : -0.06248   Median : -0.02945   Median :  0.006782
 Mean   : 0.000000   Mean   : 0.000000   Mean   :  0.00000   Mean   :  0.00000   Mean   :  0.000000
 3rd Qu.: 0.500807   3rd Qu.: 0.458949   3rd Qu.:  0.13304   3rd Qu.:  0.18638   3rd Qu.:  0.528554
 Max.   : 5.041069   Max.   : 5.591971   Max.   : 39.42090   Max.   : 27.20284   Max.   : 10.503090
```

Figure 5

Missing values in dataset

```
> colSums(is.na(credit_card))
  Time     V1     V2     V3     V4     V5     V6     V7     V8     V9    V10    V11    V12    V13    V14    V15    V16
     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0
   V17    V18    V19    V20    V21    V22    V23    V24    V25    V26    V27    V28 Amount  Class
     0      0      0      0      0      0      0      0      0      0      0      0      0      0
>
```

Fig. 6 shows that the dataset is completely imbalanced with genuine transactions as 288315 and fraudulent as just 492. It is evident that transactions which are genuine are much higher than the transactions which are fraudulent. Clearly the dataset is very imbalanced with 99.8% of cases being non-fraudulent transactions. A simple measure like accuracy is not appropriate here as even a classifier which labels all transactions as non-fraudulent will have

over 99% accuracy. An appropriate measure of model performance here would be AUC (Area Under the Precision-Recall Curve). The AUC is beneficial in imbalanced dataset and can provide insights to a model's performance in classification (Sofaer, et al., 2019). Brownlee (2020) suggested some of the metrics which can be used to calculate performance of models in case of imbalanced data. These metrics are threshold metrics which includes sensitivity-specificity metrics, precision-recall metrics, Ranking metrics such as ROC curve analysis, and lastly probabilistic metric. For this project, ROC analysis is used.

Figure 6

Number of fraudulent and non-fraudulent transactions

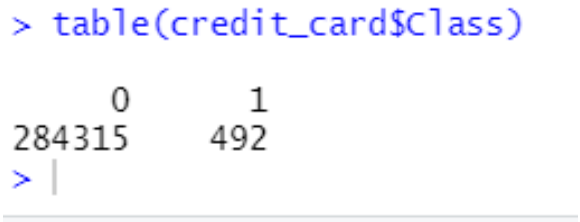


Fig. 7 shows the distribution of Time by class variable. It is evident that the fraudulent transactions are uniform throughout the time and no certain pattern. No, if the amount feature is checked against the class, it is evident that the with genuine transactions, the amount is much higher while for fraudulent transactions, the amount is much more. This is depicted in Fig. 8.

Figure 7

Distribution of Time of transaction by class in R

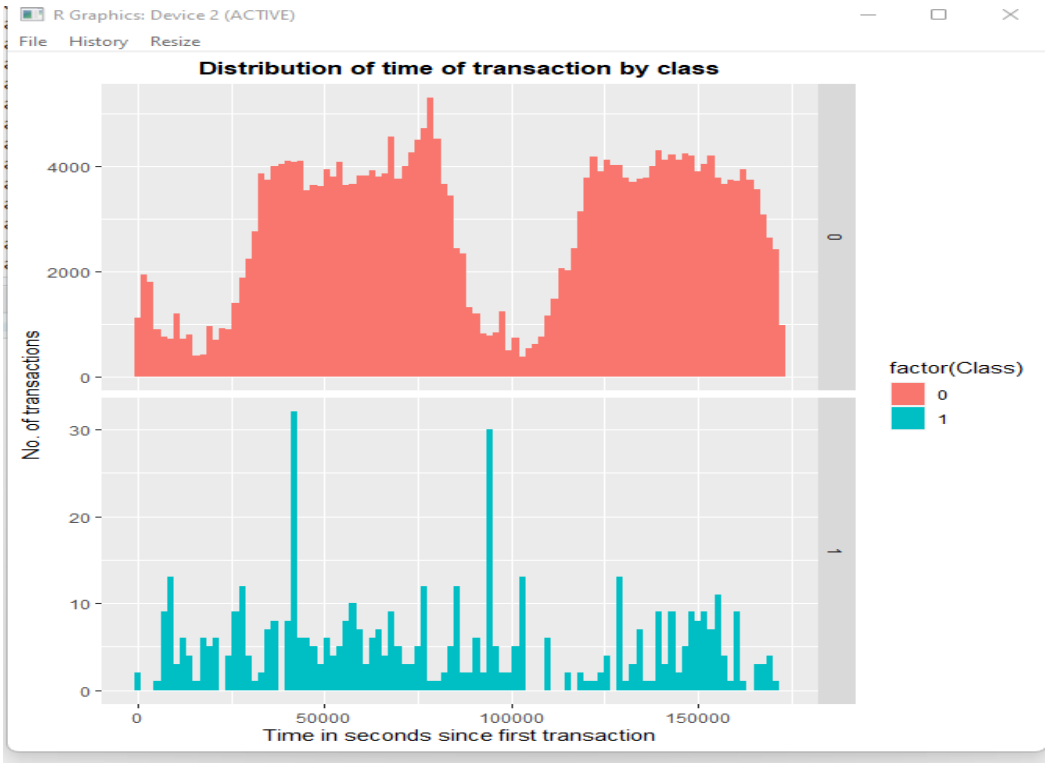
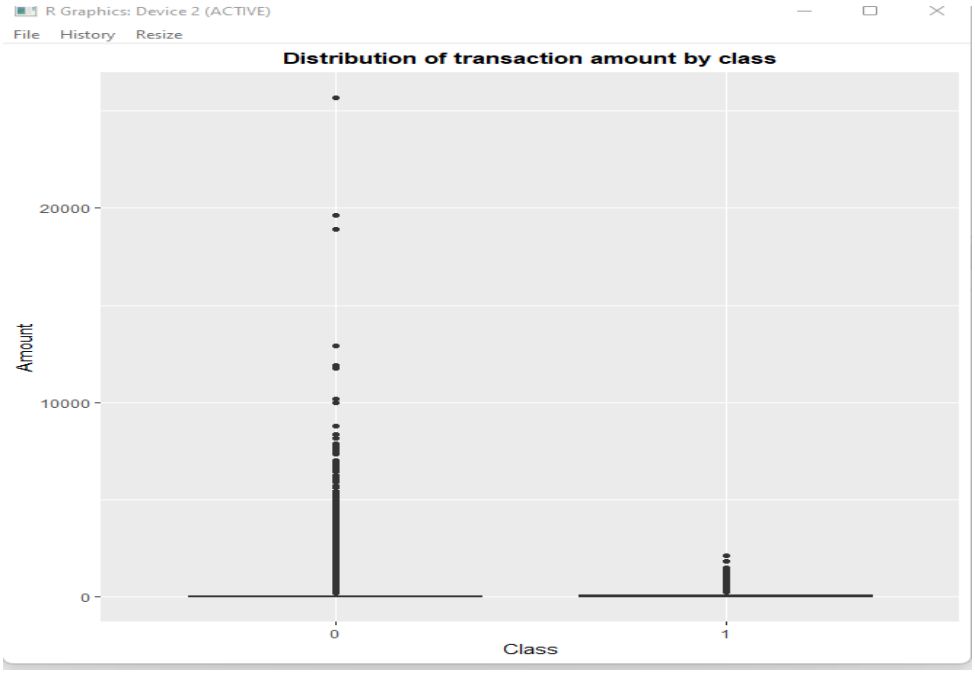


Figure 8

Distribution of transaction by Amount in R



Now for further exploration, correlation between the amount variable and other variables is explored by plotting the same as in Fig. 9. Further, Fig. 10 shows correlation heatmap in Python. The heatmap revealed strong relationships where the color is dark red or dark green. Hence the darker the color, stronger the relationship, on the other hand, lighter the color, weaker the relationship.

Figure 9

Correlation chart in R

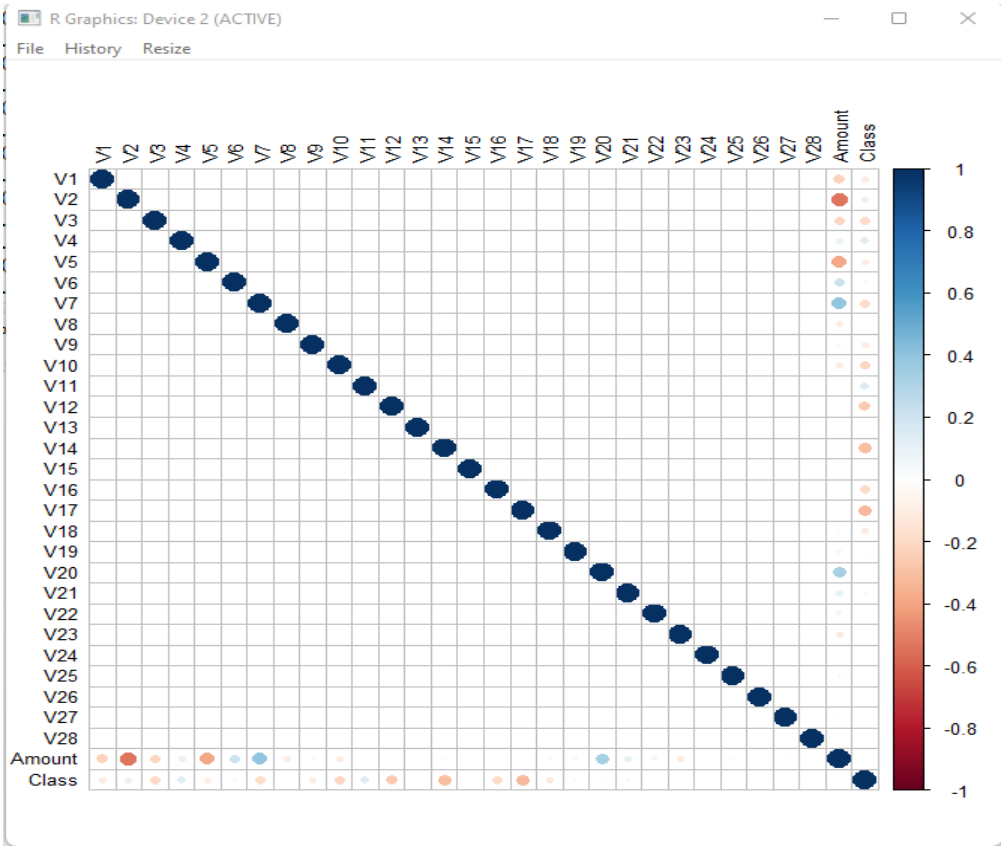
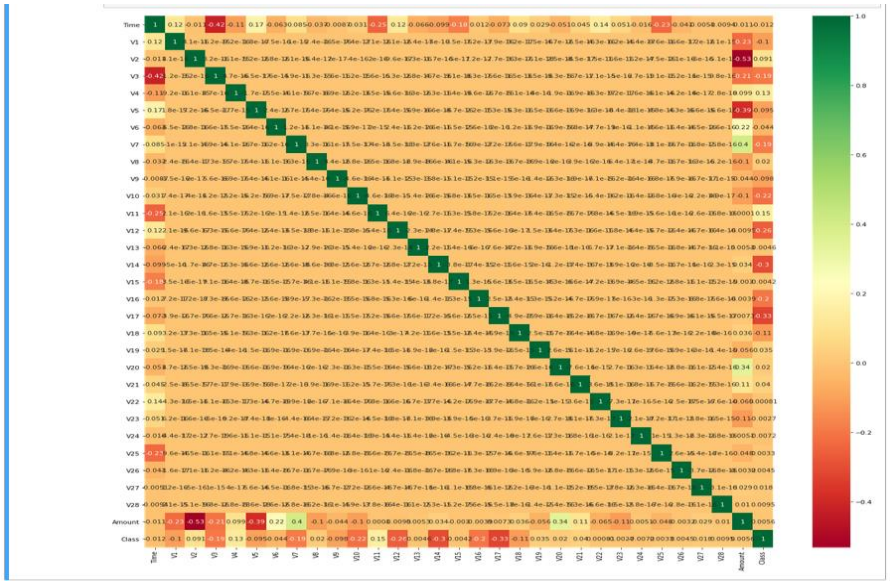


Figure 10

Correlation heat chart in Python



Model Building

Accuracy in classification refers to the performance of model and is calculated by diving number of correct predictions by total number of predictions but the scenario does not perform well when the data is skewed or imbalanced as it can be dangerously misleading (Brownlee, 2020). Standard machine learning algorithms struggle with accuracy on imbalanced data for the following reasons. One is ML algorithms struggle with accuracy because of the unequal distribution in dependent variable. This causes the performance of existing classifiers to get biased towards majority class. The algorithms are accuracy driven i.e.; they aim to minimize the overall error to which the minority class contributes very little. ML algorithms assume that the data set has balanced class distributions. They also assume that errors obtained from different classes have same cost. The methods to deal with this problem are widely known as 'Sampling Methods'. Generally, these methods aim to modify an imbalanced data into balanced distribution using some mechanism. The modification occurs by altering the size of original data set and provide the same proportion of balance.

These methods have acquired higher importance after many research have proved that balanced data results in improved overall classification performance compared to an imbalanced data set. For this dataset, methods which are used to treat the imbalance in data are undersampling, which reduces the number of observations in majority class. Second method is oversampling which replicates the observations in minority class and is also called upsampling. Lastly, Synthetic Data Generation (SMOTE and ROSE) which instead of replicating and adding the observations from the minority class, it overcome imbalances by generates artificial data. It is also a type of oversampling technique. SMOTE stands for synthetic minority oversampling technique (SMOTE) and ROSE stands for random over-sampling examples (Goswami, 2020). It

is important to note that sampling techniques should only be applied to the training set and not the testing set. A few modifications and scaling of amount is implemented onto the datasets.

Splitting data into train and test datasets

The data is split in the ration of 70-30 into train and test datasets. Now to balance the dataset in R, sampling technique is to be chosen. Fig. 11 shows the number of fraudulent and non-fraudulent transactions in train dataset. There are 199020 Not fraud and 344 fraud transactions. Applying the downsampling technique, Fig. 12 shows how Not_Fraud is reduced to 344 which is equal to Fraud transactions.

Figure 11

Fraud and Non Fraud in Train dataset

```
> table(train$Class)

Not_Fraud     Fraud
   199020       344
> |
```

Figure 12

Downsampling

```
> set.seed(9560)
> down_train <- downSample(x = train[, -ncol(train)],
+                          y = train$Class)
> table(down_train$Class)

Not_Fraud     Fraud
      344       344
> |
```

Similarly, upsampling, SMOTE and ROSE methods were applied to the dataset to balance the same. The results are shown in Fig. 13, Fig.14 and Fig. 15 respectively.

Figure 13

Upsampling

```
> set.seed(9560)
> up_train <- upSample(x = train[, -ncol(train)],
+                      y = train$Class)
> table(up_train$Class)

Not_Fraud     Fraud
   199020    199020
```

Figure 14

SMOTE

```
> set.seed(9560)
> smote_train <- SMOTE(Class ~ ., data  = train)
>
> table(smote_train$Class)

Not_Fraud     Fraud
     1376      1032
> |
```

Figure 15

ROSE

```
> set.seed(9560)
> rose_train <- ROSE(Class ~ ., data  = train)$data
>
> table(rose_train$Class)

Not_Fraud     Fraud
    99844     99520
```

To balance out the dataset, BorderlineSMOTE function is Python is used. After balancing the data, Fig. 16 shows the results of fraudulent and normal transactions.

Figure 16

Balanced dataset after applying borderline SMOTE function in Python

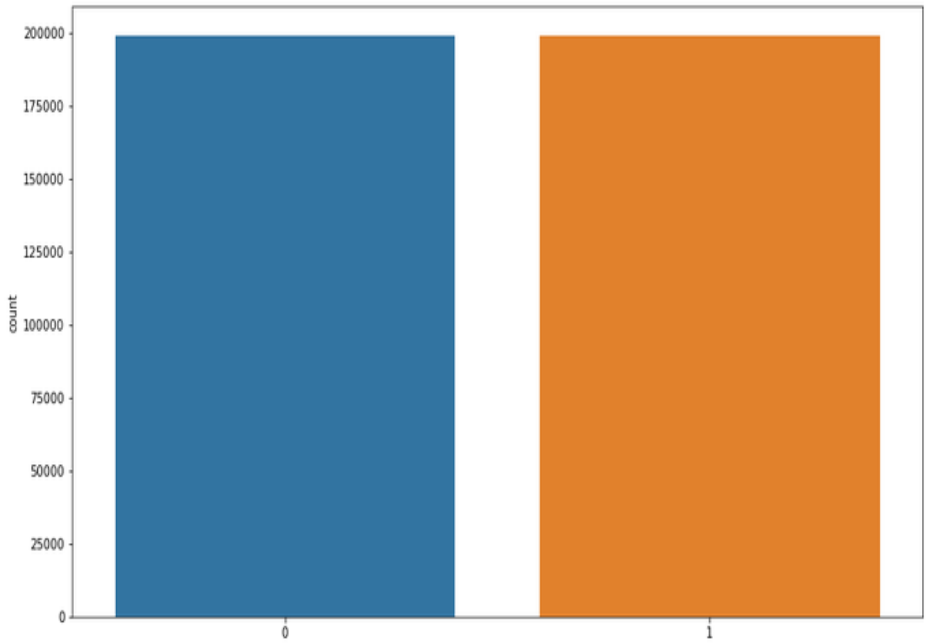
```
In [90]: from sklearn.preprocessing import StandardScaler

         scaler = StandardScaler()
         for i in X_train_Before:
             scaler = StandardScaler()
             X_train_Before[i] = scaler.fit_transform(X_train_Before[i].values.reshape(-1,1))
             X_test[i] = scaler.transform(X_test[i].values.reshape(-1,1))
```

```
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X_test[i] = scaler.transform(X_test[i].values.reshape(-1,1))
<ipython-input-90-f9deed406a1a>:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X_train_Before[i] = scaler.fit_transform(X_train_Before[i].values.reshape(-1,1))
<ipython-input-90-f9deed406a1a>:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  X_test[i] = scaler.transform(X_test[i].values.reshape(-1,1))
```

```
In [92]: borderlineSMOTE = BorderlineSMOTE(k_neighbors = 10, random_state = 42)
         X_train, y_train = borderlineSMOTE.fit_resample(X_train_Before,y_train_Before)
```

```
In [93]: sns.countplot(x=y_train)
```

```
Out[93]: <AxesSubplot:ylabel='count'>
```



Modeling

Logistic Model

The summary of the logistic model is shown in Fig. 17 where the p value for many of the features in lower than 0.005 resulting in rejecting the null hypothesis.

Figure 17

Summary of Logistic Regression Model in R

```
> summary(glm_fit)

Call:
glm(formula = Class ~ ., family = "binomial", data = up_train)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-8.4904   -0.2326    0.0000    0.0000    2.9249

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -3.756688   0.017432 -215.511  < 2e-16 ***
V1           1.320123   0.037788   34.935  < 2e-16 ***
V2           0.406317   0.060213    6.748 1.50e-11 ***
V3           0.397987   0.022940   17.349  < 2e-16 ***
V4           1.326035   0.016334   81.180  < 2e-16 ***
V5           0.871197   0.034038   25.595  < 2e-16 ***
V6          -0.560082   0.021292  -26.304  < 2e-16 ***
V7          -0.636358   0.037957  -16.765  < 2e-16 ***
V8          -0.552507   0.014420  -38.314  < 2e-16 ***
V9          -0.633473   0.017866  -35.456  < 2e-16 ***
V10         -1.175389   0.026976  -43.572  < 2e-16 ***
V11          0.769548   0.011821   65.100  < 2e-16 ***
V12         -1.152896   0.017600  -65.504  < 2e-16 ***
V13         -0.569015   0.008624  -65.977  < 2e-16 ***
V14         -1.315545   0.018062  -72.836  < 2e-16 ***
V15         -0.301143   0.008409  -35.812  < 2e-16 ***
V16         -0.531630   0.015883  -33.472  < 2e-16 ***
V17         -0.727931   0.021087  -34.520  < 2e-16 ***
V18         -0.322569   0.012627  -25.545  < 2e-16 ***
V19          0.393412   0.010958   35.903  < 2e-16 ***
V20         -0.539634   0.024409  -22.108  < 2e-16 ***
V21          0.147656   0.010142   14.558  < 2e-16 ***
V22          0.511426   0.011608   44.056  < 2e-16 ***
V23          0.160015   0.019325    8.280  < 2e-16 ***
V24          0.110532   0.009927   11.134  < 2e-16 ***
V25         -0.062140   0.010579   -5.874 4.25e-09 ***
V26         -0.095258   0.008735  -10.906  < 2e-16 ***
V27         -0.229955   0.012473  -18.437  < 2e-16 ***
V28          0.152977   0.013269   11.529  < 2e-16 ***
Amount       1.594486   0.084437   18.884  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 551801  on 398039  degrees of freedom
Residual deviance: 101925  on 398010  degrees of freedom
AIC: 101985
```

Building the logistic model in R using upsampling bore the below results as shown in Fig. 18 with receiver operating characteristic (ROC) curve and Fig. 19 shows results of AUC is 0.971.

Figure 18

ROC curve in Logistic Regression Model-Upsampling

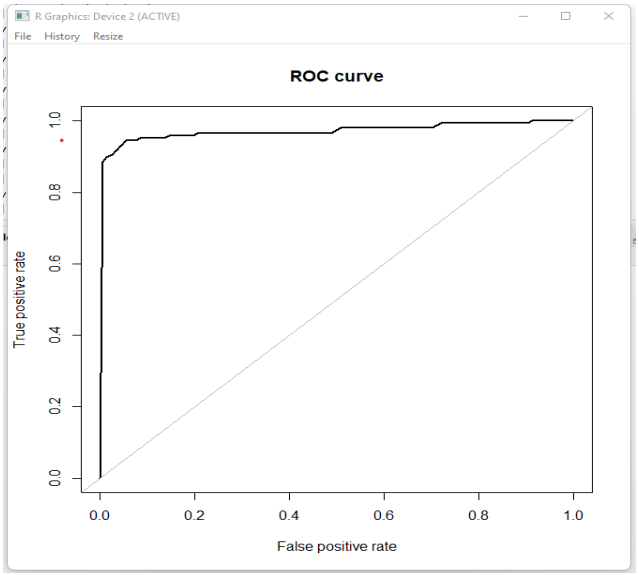


Figure 19

AUC results Logistic Regression

```
> pred_glm <- predict(glm_fit, newdata = test, type = 'response')
>
> roc.curve(test$Class, pred_glm, plotit = TRUE)
Area under the curve (AUC): 0.971
>
```

Logistic regression model in Python bore the same results as well. The Fig. 20 shows the classification report of the model where AUC is 0.953 similar to what was generated in R.

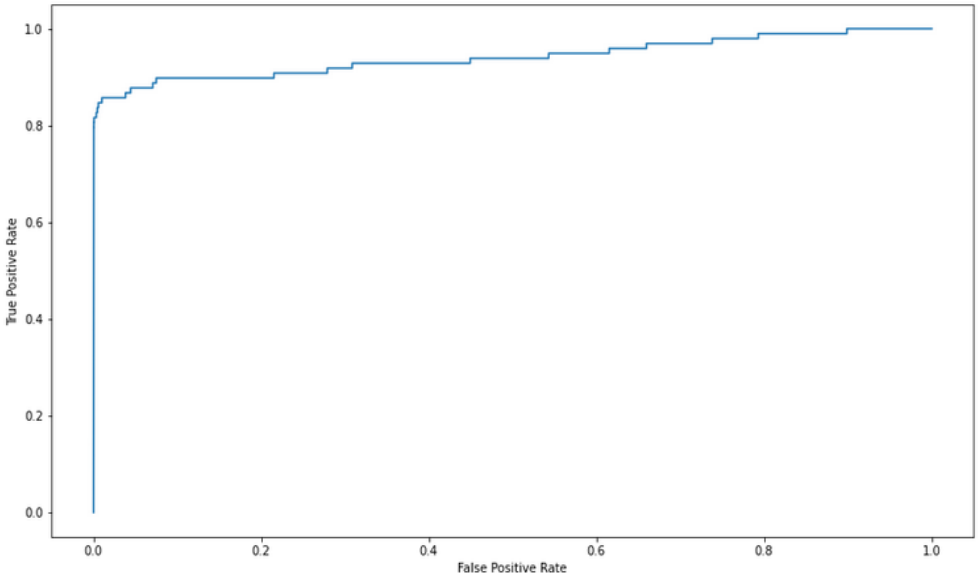
Figure 20

AUC and ROC in Logistic Regression in Python

```
In [180]:
# Logistic Regression Classifier

logReg = LogisticRegression()

logReg.fit(X_train, y_train)

log_y_pred = logReg.predict(X_test)
from sklearn import metrics
#from sklearn.metrics import classification_report
print(metrics.classification_report(y_test,log_y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     56864
           1       0.11      0.86      0.20        98

    accuracy                           0.99     56962
   macro avg       0.56      0.92      0.60     56962
weighted avg       1.00      0.99      0.99     56962
```

```
In [181]:
#define metrics
y_pred_proba = logReg.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test,  y_pred_proba)
auc = metrics.roc_auc_score(y_test, y_pred_proba)

#create ROC curve
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
print(str(auc))
```




```
0.9412005587265858
```

Since the p value for features like V4, V8, V9, V10 etc. (In Fig. 17), is less than 0.005, the association between dependent and independent features is statistically significant and hence the null hypothesis is rejected. Also, the AUC in the results is 0.97 and 0.9412 which is means that the model is predicting the values well enough.

Decision Tree

For decision tree, below model in Fig. 21 is built. To predict the model performance, AUC was built on SMOTE dataset which comes out to be 0.934. The basic feature of SMOTE is that analyze the minority samples, synthesize new samples and add it to the dataset (Zhou, et al., 2022).

Figure 21

AUC of Decision Tree

```
> pred_smote <- predict(smote_fit, newdata = test)
> print('Fitting model to smote data')
[1] "Fitting model to smote data"
> roc.curve(test$Class, pred_smote[,2], plotit = FALSE)
Area under the curve (AUC): 0.934
>
```

In Python, model was built using below commands and produced AUC of 0.9408 similar to model in R as shown in Fig. 22

Figure 22

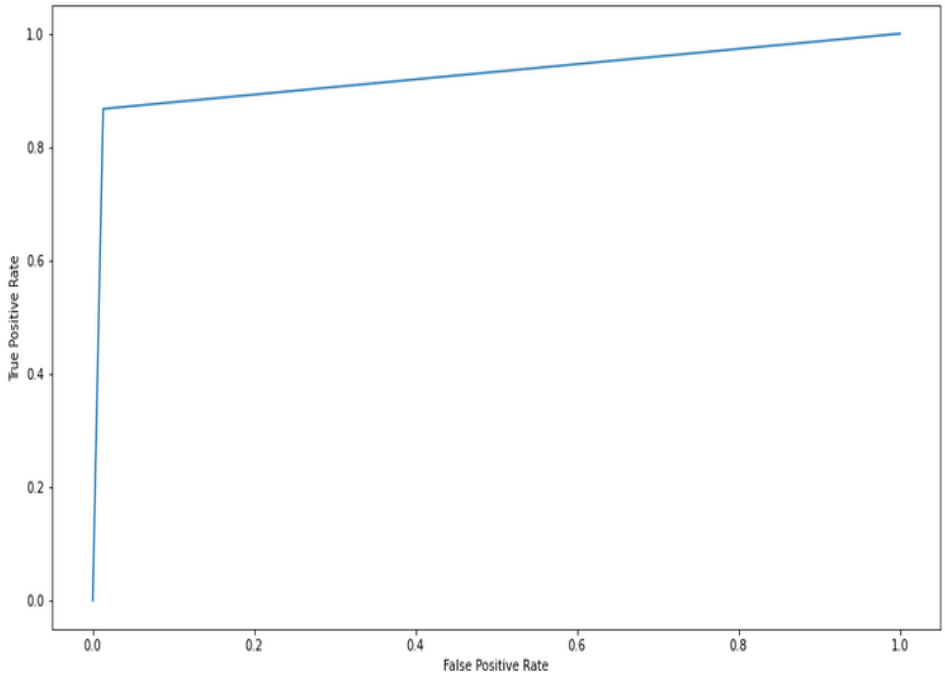
AUC of Decision Tree in Python

```
In [182]: #Decision Tree

from sklearn.tree import DecisionTreeClassifier
tree_model = DecisionTreeClassifier(max_depth = 4, criterion = 'entropy')
tree_model.fit(X_train, y_train)
predictions = tree_model.predict(X_test)
```

```
In [170]: print(type(predictions))
print(predictions.shape)
```

```
<class 'numpy.ndarray'>
(85443,)
```

```
In [183]: fpr, tpr, _ = metrics.roc_curve(y_test,  predictions)
auc = metrics.roc_auc_score(y_test, predictions)

#create ROC curve
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
print(str(auc))
```




```
0.9272106809803269
```

Results of AUC from decision tree model above also favors to reject the null hypothesis since AUC is higher than 0.5 which proves that all the transactions are not genuine transactions (Analyse-it, 2020).

XGBoost

Now building the XGBoost Model, Fig. 23 shows the command that was used and Fig. 24 shows the results of the ROC curve. The AUC is 0.977 which means that the model is predicting the values of variable “Class” well.

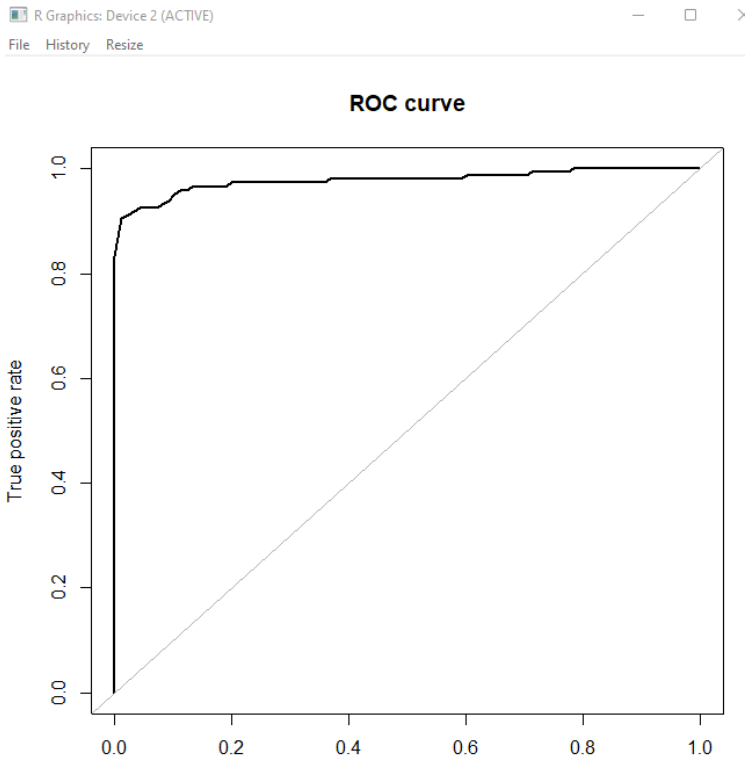
Figure 23

XGboost Model in R

```
> labels <- up_train$Class
>
> y <- recode(labels, 'Not_Fraud' = 0, "Fraud" = 1)
> set.seed(42)
> xgb <- xgboost(data = data.matrix(up_train[,-30]),
+                label = y,
+                eta = 0.1,
+                gamma = 0.1,
+                max_depth = 10,
+                nrounds = 300,
+                objective = "binary:logistic",
+                colsample_bytree = 0.6,
+                verbose = 0,
+                nthread = 7,
+ )
[22:46:22] WARNING: amalgamation/../src/learner.cc:1115: Starting in XGBoost 1.3.0,
 the default evaluation metric used with the objective 'binary:logistic' was change
d from 'error' to 'logloss'. Explicitly set eval_metric if you'd like to restore th
e old behavior.
> xgb_pred <- predict(xgb, data.matrix(test[,-30]))
>
> roc.curve(test$Class, xgb_pred, plotit = TRUE)
Area under the curve (AUC): 0.977
```

Figure 24

ROC curve for XGboost Model in R



For building XGBoost Model in Python, same concept is used to build the ROC curve which gave results as shown in Fig. 25.

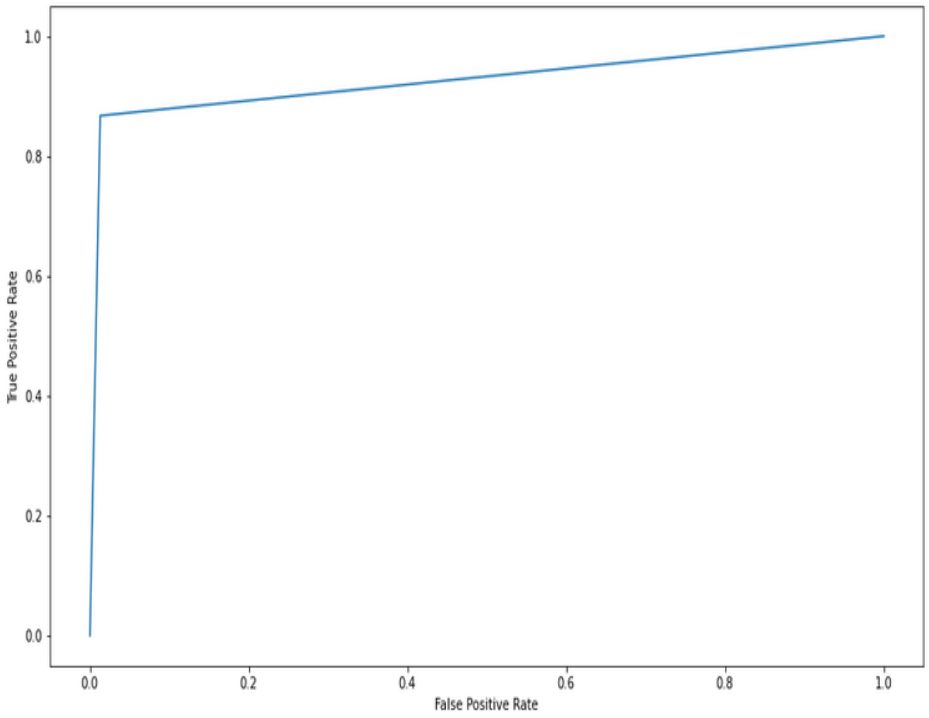
Figure 25

ROC curve for XGboost Model in Python

In [184]:
```
# XGBoost

from xgboost import XGBClassifier # XGBoost algorithm
xgb = XGBClassifier(max_depth = 10)
xgb.fit(X_train, y_train)
xgb_yhat = xgb.predict(X_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\xgboost\sklearn.py:1224: UserWarning: The use of label encoder in
XGBClassifier is deprecated and will be removed in a future release. To remove this warning, do the followin
g: 1) Pass option use_label_encoder=False when constructing XGBClassifier object; and 2) Encode your labels
(y) as integers starting with 0, i.e. 0, 1, 2, ..., [num_class - 1].
  warnings.warn(label_encoder_deprecation_msg, UserWarning)

[12:08:52] WARNING: ..\src\learner.cc:1115: Starting in XGBoost 1.3.0, the default evaluation metric used wi
th the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval_metric if you'
d like to restore the old behavior.
```

In [185]:
```
#Calculation of TPR and FPR
fpr, tpr, _ = metrics.roc_curve(y_test,  predictions)
auc = metrics.roc_auc_score(y_test, xgb_yhat)

#create ROC curve
plt.plot(fpr,tpr)
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
print(str(auc))
```




```
0.9335415757467872
```

Since the AUC is above 0.5, the XGBoost model is performing well in identifying and categorizing the two classes and hence is statistically significant leading to rejection of null hypothesis.

Neural Network

Neural network in R is created by passing argument set of label and features, dataset, number of neurons in hidden layers (3), and error calculation. Act.fct() function is a differentiable function used for smoothing the result of the cross product of the covariate or neurons and the weights (Datacamp, n.d.). The Fig 26 shows the neural network created the AUC for the same model as 0.969.

Figure 26

Neural Network in R

```
>
>
>
> library(neuralnet)
> nn=neuralnet(Class~.,data=smote_train, hidden=3, act.fct = "logistic", linear.output = FALSE)
> Predict=neuralnet::compute(nn,test)
> roc.curve(test$Class, Predict$net.result[,2], plotit = TRUE)
Area under the curve (AUC): 0.969
>
```

For neural network model in Python, tensorflow library is used. The optimizer used is the AdamOptimizer, the activation function that is used in this scenario is "Relu" which is short for rectified linear unit and sigmoid. The AUC produced by the model (Fig. 27) ranges from 0.933 to .9504 which makes the model well fitted for the prediction.

Figure 27

Neural Network Model in Python

```
In [190]: # Neural Network

          import tensorflow as tf
          from tensorflow.keras.models import Sequential
          from tensorflow.keras.layers import add, Dense, Dropout
          from tensorflow.keras.optimizers import Adam


          model = Sequential()

          model.add(Dense(32,input_dim=29,activation='relu'))
          model.add(Dense(16, activation = 'relu'))
          #model.add(Dropout(0.15))
          model.add(Dense(16,activation = 'relu'))
          #model.add(Dropout(0.15))
          model.add(Dense(1, activation = 'sigmoid'))

          opt = Adam(learning_rate=1e-4, decay=1e-6)

          model.compile(loss="binary_crossentropy",optimizer=opt,metrics=[tf.keras.metrics.AUC()])

          model.fit(X_train, y_train, epochs = 10,\
                              validation_data = (X_test, y_test))

          Epoch 1/10
          14216/14216 [==============================] - 12s 853us/step - loss: 0.0438 - auc: 0.9989 - val_loss: 0.018
          1 - val_auc: 0.9504
          Epoch 2/10
          14216/14216 [==============================] - 12s 876us/step - loss: 0.0101 - auc: 0.9996 - val_loss: 0.012
          6 - val_auc: 0.9368
          Epoch 3/10
          14216/14216 [==============================] - 12s 865us/step - loss: 0.0067 - auc: 0.9997 - val_loss: 0.010
          9 - val_auc: 0.9427
          Epoch 4/10
          14216/14216 [==============================] - 12s 869us/step - loss: 0.0053 - auc: 0.9997 - val_loss: 0.009
          0 - val_auc: 0.9430
          Epoch 5/10
          14216/14216 [==============================] - 12s 870us/step - loss: 0.0046 - auc: 0.9998 - val_loss: 0.008
          5 - val_auc: 0.9430
          Epoch 6/10
          14216/14216 [==============================] - 13s 883us/step - loss: 0.0040 - auc: 0.9998 - val_loss: 0.007
          9 - val_auc: 0.9431
          Epoch 7/10
          14216/14216 [==============================] - 12s 876us/step - loss: 0.0036 - auc: 0.9998 - val_loss: 0.006
          8 - val_auc: 0.9381
          Epoch 8/10
          14216/14216 [==============================] - 14s 970us/step - loss: 0.0032 - auc: 0.9998 - val_loss: 0.006
          4 - val_auc: 0.9383
          Epoch 9/10
          14216/14216 [==============================] - 15s 1ms/step - loss: 0.0029 - auc: 0.9999 - val_loss: 0.0063
          - val_auc: 0.9331
          Epoch 10/10
          14216/14216 [==============================] - 16s 1ms/step - loss: 0.0027 - auc: 0.9999 - val_loss: 0.0065
          - val_auc: 0.9383

Out[190]: <tensorflow.python.keras.callbacks.History at 0x1a3e8026c10>
```

Again, since the AUC for the model is above 0.5, the null hypothesis is rejected.

X. CONCLUSION

The dataset is completely imbalanced data where the number of genuine transactions is much higher than the fraudulent transactions. Due to the imbalance in data, the metric to measure the performance of model was not accuracy but area under curve (AUC). The dataset was converted into balanced one using techniques like up sampling, down sampling, SMOTE and ROSE. The ultimate models were built using SMOTE technique.

The dataset was also scaled as most of the features have numeric values as a result of PCA transformation and the amount variable had high values. For all the models generated, logistic regression, decision tree, XGBoost and Neural network, the result was statistically significant and hence the null hypothesis was rejected. The models build was Logistic regression, Decision tree, XGBoost and Neural network which bore AUC in R as .91, .934, .977 and 0.96 respectively. While in Python, AUC were .941, .92, .933 and .9504 respectively.

XI. RECOMMENDATIONS

The increasing number of frauds using credit cards has been a alarming in banking industries. It is important that strict measures should be taken to avoid such activity rather than acting upon after the loss has been done. The machine learning algorithms shared in this paper performed well and predicted the anomaly efficiently. Some of the other algorithms that can be used to predict the scenario are Random Forest, isolation forest, or local outlier factor algorithm, etc. Mensi, and Bicego (2021) states that isolation forest algorithm is a different version of random forest which is highly proficient in outlier detection. Moreover, the results of models generated will differ according to the data and hence a model which performed best with this data may not perform the same with other set of data. I recommend to add an automation system during the model deployment process which compares and calculates best performing model.

References

Analyse-it (2022). Testing the area under curve. Retrieved from https://analyse-it.com/docs/user-guide/diagnostic-performance/testing-auc#:~:text=The%20default%20null%20hypothesis%20is,is%20better%20than%20chance%20alone.

Betts, J. (n.d.). Hypothesis Examples: Different Types in Science and Research. Retrieved from https://examples.yourdictionary.com/examples-of-hypothesis.html

Bradley. T (2020). The Practical, Ethical, and Compliance Challenges of Data Privacy. Retrieved from https://securityboulevard.com/2020/01/the-practical-ethical-and-compliance-challenges-of-data-privacy/

Brownlee, J. (2020). Failure of Classification Accuracy for Imbalanced Class Distributions. Retrieved from https://machinelearningmastery.com/failure-of-accuracy-for-imbalanced-class-distributions/

Choma, A. (2014). Big Data in Finance: Ethical Challenges. Retrieved from https://www.obsfin.ch/wp-content/uploads/Document/Choma_RCP.pdf

Colorado State University-Global Campus. (2022). Module 2 - Methodology, and Tools for Analytics Project Design. In *MIS581 – Capstone-Business Intelligence and Data Analytics* (p. 2). Greenwood Village, CO: Author.

Datacamp (n.d.). neuralnet: Training of neural networks. Retrieved from rdocumentation.org/packages/neuralnet/versions/1.44.2/topics/neuralnet

Data Flair (2022). Data Science Project – Detect Credit Card Fraud with Machine Learning in R. Retrieved from https://data-flair.training/blogs/data-science-machine-learning-project-credit-card-fraud-detection/

GeeksforGeeks (2021). Bernoulli Distribution in R. Retrieved from https://www.geeksforgeeks.org/bernoulli-distribution-in-r/

Github (2019). A Reader on Data Visualization. *Chapter 5 Ethics*. Retrieved from https://mschermann.github.io/data_viz_reader/ethics.html

Goswami, S. (2020). Class Imbalance, SMOTE, borderline SMOTE, ADASYN. Retrieved from https://towardsdatascience.com/class-imbalance-smote-borderline-smote-adasyn-6e36c78d804

Guangwei Xu, Shan Li, Miaolin Lai, Yanglan Gan, Xiangyang Feng, Qiubo Huang, Li Li, & Wei Li. (2022). Verification Control Algorithm of Data Integrity Verification in Remote Data sharing. *KSII Transactions on Internet & Information Systems, 16*(2), 565–586.

Hayes, A. (2022). Null Hypothesis. *How a Null Hypothesis works.* Retrieved from https://www.investopedia.com/terms/n/null_hypothesis.asp

IBM (2021). Decision Tree Model. Retrieved from https://www.ibm.com/docs/en/spss modeler/18.1.1?topic=trees-decision-tree-models

Kaggle (2022). Recall 97% by using undersampling & Neural Network. Retrieved from https://www.kaggle.com/jdelamorena/recall-97-by-using-undersampling-neural-network

Logrippo, L. (2021). Multi-level models for data security in networks and in the Internet of things. *Journal of Information Security and Applications, 58*. https://doi.org/10.1016/j.jisa.2021.102778

Mensi, A., & Bicego, M. (2021). Enhanced anomaly scores for isolation forests. Pattern Recognition, 120. https://doi.org/10.1016/j.patcog.2021.108115

Nandi, A. K., Randhawa, K. K., Chua, H. S., Seera, M., & Lim, C. P. (2022). Credit card fraud detection using a hierarchical behavior-knowledge space model. PLoS ONE, 17(1), 1–16. https://doi.org/10.1371/journal.pone.0260579

Nufer, G., & Muth, M. (2022). Artificial Intelligence in Marketing Analytics: The Application of Artificial Neural Networks for Brand Image Measurement. *Journal of Marketing Development & Competitiveness*, *16*(1), 55–63. https://doi.org/10.33423/jmdc.v16i1.5027

O'Leary, Z (2010). *The Essential Guide To Doing Your Research Project*. London: SAGE

Oracle (2002). Oracle9i Security Overview. *Data Security Challenges.* Retrieved from https://docs.oracle.com/cd/B10501_01/network.920/a96582/overview.htm

Popat, R.R., & Chaudhary, J. (2018). A Survey on Credit Card Fraud Detection Using Machine Learning. *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 1120-1125.

Porkess, R., & Mason, S. (2012). Looking at Debit and Credit Card Fraud. *Teaching Statistics: An International Journal for Teachers*, *34*(3), 87–91.

Rafferty, A. N., Whitehill, J., Romero, C., Cavalli-Sforza, V., & International Educational Data Mining Society. (2020). Proceedings of the International Conference on Educational Data Mining (EDM) (13th, Online, July 10-13, 2020). *International Educational Data Mining Society*.

Randhawa, K., Loo, C. K., Seera, M., Lim, C. P., & Nandi, A. K. (2018). Credit card fraud detection using AdaBoost and majority voting. *IEEE Access*, *6*, 14277-14284. https://doi.org/10.1109/ACCESS.2018.2806420

Robles-Velasco, A., Cortés, P., Muñuzuri, J., & Onieva, L. (2021). Estimation of a logistic regression model by a genetic algorithm to predict pipe failures in sewer networks. *OR Spectrum*, *43*(3), 759–776. https://doi.org/10.1007/s00291-020-00614-9

Rtayli, N., & Enneya, N. (2020). Enhanced credit card fraud detection based on SVM-recursive feature elimination and hyper-parameters optimization. Journal of Information Security and Applications, 55. https://doi.org/10.1016/j.jisa.2020.102596

Salkind, N. J. (2010). *Encyclopedia of research design* (Vols. 1-0). Thousand Oaks, CA: SAGE Publications, Inc. doi: 10.4135/9781412961288

Sofaer, H. R., Hoeting, J. A., Jarnevich, C. S., & McPherson, J. (2019). The area under the precision-recall curve as a performance metric for rare binary events. Methods in Ecology and Evolution, 10(4), 565. https://doi.org/10.1111/2041-210X.13140

Stunt, J., van Grootel, L., Bouter, L., Trafimow, D., Hoekstra, T., & de Boer, M. (2021). Why we habitually engage in null-hypothesis significance testing: A qualitative study. *PLoS ONE*, *16*(10), 1–23. https://doi.org/10.1371/journal.pone.0258330

Taylor, J., & Pagliari, C. (2018). Mining Social Media Data: How Are Research Sponsors and Researchers Addressing the Ethical Challenges? Research Ethics, 14(2).

Tehrany, M. S., Shabani, F., Jebur, M. N., Haoyuan Hong, Wei Chen, & Xiaoshen Xie. (2017). GIS-based spatial prediction of flood prone areas using standalone frequency ratio, logistic regression, weight of evidence and their ensemble techniques. *Journal of the Association for Information Systems*, *18*(11), 1538–1561. https://doi.org/10.1080/19475705.2017.1362038

Tierney, M. (2021). Data Security Explained: Challenges and Solutions. Retrieved from https://blog.netwrix.com/2021/07/26/data-security/

Uchhana, N., Ranjan, R., Sharma, S., Agarwal, D., & Punde, A., (2021). Literature Review of Different Machine Learning Algorithms for Credit Card Fraud Detection. Retrieved from https://www.ijitee.org/wp-content/uploads/papers/v10i6/C84000110321.pdf

Varmedja, D., Karanovic, M., Sladojevic, S., Arsenovic, M. & Anderla, A. (2019). Credit Card Fraud Detection - Machine Learning methods. *18th International Symposium Infotech-Joharina (Infortech)*. doi: 10.1109/INFOTEH.2019.8717766.

Voican, O. (2021). Credit Card Fraud Detection using Deep Learning Techniques. *Informatica Economica*, *25*(1), 70–85. https://doi.org/10.24818/issn14531305/25.1.2021.06.

Wu, Y., Dai, X., Maseleno, A., Yuan, X., & Balas, V. E. (2020). Encryption of accounting data using DES algorithm in computing environment. *Journal of Intelligent & Fuzzy Systems*,*39*(4), 5085–5095. https://doi.org/10.3233/JIFS-179994

Yong Fang, Yunyun Zhang, & Cheng Huang. (2019). Credit Card Fraud Detection Based on Machine Learning.Computers, Materials & Continua, 61(1), 185–195.

Zhou, R., Yin, W., Li, W., Wang, Y., Lu, J., Li, Z., & Hu, X. (2022). Prediction Model for Infectious Disease Health Literacy Based on Synthetic Minority Oversampling Technique Algorithm. Computational & Mathematical Methods in Medicine, 1–6. https://doi.org/10.1155/2022/8498159