Machine Learning Techniques to Detect Fraudulent Credit Card Transactions

Vamshi Krushna Lakavath

Department of Applied Data Science, San Jose State University

DATA 270: Data Analytics Process

Dr. Eduardo Chan

December 10, 2021

Abstract

As the technology is growing usage of credit cards is growing it results in higher unauthorized usage of these cards. Using the latest technology, fraudsters are committing more and more frauds and making these frauds look as legit as possible. To control these frauds, and tackle the fraudsters' unpredictable behaviors we should use the latest and accurate technologies. Machine Learning provides a great solution in detecting anomalies in data. Which assures an accurate solution to this problem. So analyzing and using a large set of credit card transactions dataset we can build various machine learning techniques to detect these frauds. In this project, We have trained and built five machine learning models to find unusual behaviors of fraudsters and detect fraudulent credit card transactions using transaction data of European Union customers. These models will be useful to detect a fraudulent transaction as soon as it takes place. It helps banks and financial institutions to stop such transactions even before authorizing them, which is expected to save a lot of resources and money for banks & financial Institutions. Machine learning models like Logistic regression, Random forest, Support Vector Machine, Artificial Neural Networks, and Decision Trees have been built in this Project. All the mentioned models gave excellent results in detecting credit card fraud transactions. The performance of these models has been evaluated using different metrics. Among these models, the ANN model has given the highest accuracy of 99.8% and Precision of 99.76% with a low False Positive rate of 0.11%, So ANN can be used for detecting fraudulent credit card transactions.

Keywords: Machine Learning, Credit Card, European Union, Fraudulent Transactions, Banks, Financial Institutions, Logistic regression, Random forest, Support Vector Machine, Artificial Neural Networks, Decision Tree

Machine Learning Techniques to Detect Fraudulent Credit Card Transactions

1.1 Project Background

Credit Cards are issued by banks and financial institutions to their customer base for money transactions both online and point of sale (POS) to replace traditional cash or bank check transactions. With the growing technology, people's usage of credit cards is growing, in turn, resulting in higher unauthorized use of them (Popat & Chaudhary, 2018). These frauds are causing huge losses for both the customers and the credit card providers. These bad transactions have to be prevented and detected. Prevention helps to avoid any fraudulent attacks and detection helps in alerting and analyzing immediately after a fraud transaction takes place (Thennakoon et al., 2019). Banks and financial institutions are spending tons of resources and money on detecting and preventing these frauds.

With the increasing frauds and lack of accuracy in detecting fraudulent transactions sometimes, legit customers are getting affected. Because in order to stop these frauds banks have to block the card which is time-consuming and lots of effort is needed to obtain a new card. In this process, banks are losing customer's trust which requires immediate action and trustable solutions. To keep up the trust of its customer base banks and financial institutions have to control these fraudulent transactions before they occur and be able to detect with high accuracy in a short span of time which could increase the scope for the recovery of the fraud transaction amount.

Credit card frauds are majorly two types, card-present and card not present. In the case of a card-present physical card is required to perform a transaction which usually happens at ATMs and Point-of-Sale (POS) machines at stores (Thennakoon et al., 2019). In the other case, the physical card is not required and these transactions are either online, mobile, or mail

transactions. The information of the card like card number, date of expiry, and 3-digit card verification value (CVV) has been stolen or purchased on the dark web for card, not present transactions. Skimming, phishing, and account takeover are also a part of these transactions.

Additionally, payments fraud is getting more and more sophisticated as fraudsters are using high-end technology to appear as legit as possible. As a result, a human fraud analyst cannot prevent or detect the abnormal behavior of the fraudster. Traditionally companies relied on rules that can block fraudulent transactions. Traditionally companies relied on rules that can block fraudulent transactions which are not much effective, even though they are accurate still there is a high chances of huge loss, since the large volume of the credit card transaction taking place every day even a fraction of false rate can cause end up predicting thousands of transaction wrongly, so still there is a huge need for more accurate, effective, and adaptive fraud detection mechanism. Thus, to attain that level of accuracy and detect anomalies in the customer's transactional behavior there is a need to analyze hundreds and thousands of features that is only possible using Machine Learning algorithms.

This paper will focus on providing a Machine Learning model to detect fraudulent credit card transactions and that will eventually help in controlling and preventing. For this We would be using card not present (online) fraudulent transactions and discuss how machine learning can be used in detecting these, the latest technologies available, and the recent studies conducted on preventing these frauds. Machine learning is this generation's solution that helps work on large scale datasets that is not easy for human beings (Thennakoon et al., 2019). In this paper We will analyze machine learning models like Logistic regression, Random forest, Decision tree, Support Vector Machine (SVM), Artificial Neural Network (ANN), and K Nearest Neighbor (KNN) Model works and are accurate in detecting fraudulent transactions of the credit card.

Eventually this paper will provide the best available machine learning solution which focuses on decreasing the detection time (reactive measure) and the solution to prevent fraudulent transactions before they take place (proactive measure). Ultimately this can help mitigate and eradicate the losses occurring to banks and their customers. It will also be helpful in decreasing their resources and expenditure.

1.2 Project Requirements

For this project, the most important requirement is fraudulent credit card transactional data. That will create the scope for performing different kinds of analysis and give the best solution. These transactions are generating a large scale of data by merchants and credit card users, that data is available with banks, financial institutions, and credit unions. It is difficult to obtain these transactions because of customers' confidentiality information. For this project We are using the dataset taken from "Kaggle.com" ; it contains two-day credit card transactions of European customers, who performed these transactions through their two days purchase spendings in September 2013. So it contains a total of 284,807 records, each transaction recorded as a row, and 31 fields like, time (when the transaction had taken place), amount (amount of transaction), class (whether fraudulent or legit transaction) of transactions. This dataset is highly imbalanced so to clean and process this data a data preparation tool like Google Dataprep or Tableau Prep Builder is required. These are used to visually explore and clean the data, and also used for preparing structured and unstructured data.

To process and analyze this data hardware, software, and machine learning framework are required. To build and run a model a minimum of 16GB of RAM and an i5 processor is required. under software requirements TensorFlow can be used, it is a free open source python

library used as an artificial intelligence for swift computing, batch and stream data processing, classification, model building, and prediction.

Cloud computing services can also be used. Google Cloud Products are one of the best available open source resources. that provides software as a service (SaaS) and Platform as a Service (PaaS). They are cost effective and easily scalable and measurable services. Dataproc and Dataflow are an online computing service which offers analytics engines like Spark and Hadoop in a unified form. These are helpful for batch and streaming data processing, also helpful for querying and Machine Learning.

1.3 Project Deliverables

The deliverables of the Project will be weekly project progress reports in the form of draft Submissions. This includes an introduction, data collection, data preparation, data processing, model selection, a comparative summary of recent research papers in this field, and a Project management plan.

At the end of the timeline of the project, a final project presentation will be given. Also, a final project report will be provided that will have an abstract, introduction, data management, and project development plans, data engineering, model selection and model development, conclusion, sources of references. The deliverables also include dataset sources from where credit card transactions data can be obtained, tables showing the reports and results of the models that would include the accuracy and precision of the models, and some visualizations for analytical and understanding purposes.

The most significant element of the deliverables will be the most accurate and efficient machine learning models for controlling and detecting fraudulent Credit card transactions that

will ultimately detect frauds in no time and prevent fraudsters from doing the frauds, which will help banks, merchants, and customers tremendously.

1.4 Technology and Solution Survey

There are many ways to detect fraud transactions, from the beginning banks have been implementing different technologies and approaches in controlling bank and credit card related frauds but Machine Learning is most sought among all because a model can be easily build and re-model. Machine learning helps to divide, analyze, and visualize Big Data which have millions and billions of records with many fields in an easier way, which is not possible for human brain. Also, It is a cost-effective and most accurate way of detecting fraudulent transactions.

In general, there are three types of machine learning models: Supervised, Unsupervised, and Reinforcement machine learning. Out of these three, supervised machine learning models are very popular in fraud detection. Basically, it is a sub-set of machine learning and artificial intelligence. Supervised learning works on labeled data and it allows us to manage the data to control the outliers to avoid unwanted outcomes and increases the chance to get the desired output. It is further divided into regression and classification techniques (BrainStation, 2018).

A Decision tree is one of the most commonly machine learning models used for Credit card fraud detection and it has a high accuracy rate. Basically, it is a supervised machine learning algorithm used for the classification and prediction of data. In this technique the entire data which is referred to as the root node is initially split into two or more homogenous nodes. We then compare the value of the root attributes of the dataset with respect to the records' attribute and classify them accordingly. Later these nodes are further split into sub-nodes based on the decision using the above logic. Eventually, we arrive at a stage where further splitting is not possible and this node is called a terminal node. There are different approaches to partition the

dataset, the first one is depth-first greedy approach and the second one is breadth-first greedy approach, and a rule is considered efficient only when it splits the data into groups with one predominant single class (Jain et al., 2019).

The other model used for fraud transaction detection is An Artificial Neural Network (ANN) that simulates the network of neurons which acts as a human brain so the computer starts learning things and makes decisions like a human (Sadineni, 2020, p. 661). Regular computers have been extensively programmed so they can imitate as if they were interconnected brain cells to generate ANNs (Volpi, 2020, para. 2).

Support vector machine, which is one of the powerful supervised classification algorithms commonly used as a fraud detection model. In this technique, the data points are transformed into vectors in high dimensional space and hyperplanes are drawn to divide the space into various classes with separate behavioral characteristics (Rtayli & Enneya, 2020, p. 944). This method provides a solution to complex non-linear problems like credit card fraud detection using linear classification by leveraging kernel functions. (Popat & Chaudhary, 2018, p. 1123) Depending on the dataset and classification goal we can apply kernels like Gaussian radial basis function, polynomial function.

Logistic regression is a statistical analysis method that is used for solving problems where required solution is predicting the binary classes. It is also one of the effective models in credit card fraud detection. It considers a linear association among its input variables and the output variables. The better the linear association among the input variables gives a more accurate model. Logistic regression is helpful while guessing the chances of happening of a circumstance by comparing data into a logit function (Ramesh, 2017).

In addition to the above-mentioned models, Random forest is also used for fraud detection. It is also called a random decision forest. Random forest is extremely helpful while solving problems such as classification and regression (Praveen, 2020, p. 662). It is one of its biggest advantages. It is an aggregation of a set of decision trees and every individual tree makes a class prediction (Praveen, p.662, 2020).

1.5 Literature Survey of Existing Research

There are many studies conducted and being conducted in the area of fraud detection and prevention. Among those, Credit card fraud detection is one of the complex and significant areas. We also noticed that machine learning techniques have seen tremendous advancement in the prediction and classification area. It also promises better solutions for controlling these frauds than ever. The literature survey on some of the research papers we would be referring for this project are discussed below.

Maniraj et al. (2019) provided a solution to detect credit card frauds using machine learning models. Their primary objective is to detect fraudulent transactions with 100% accuracy while minimizing incorrect fraud classification. They analyzed and pre-processed datasets and deployed Machine learning algorithms such as Isolation Forest (IF) and Local Outlier Factor (LOF) algorithms on credit card transactions data. While considering one-tenth of the dataset their algorithm reached over 99.6% accuracy with 28% precision and with the whole dataset, the precision increased to 33% (Maniraj et al., 2019).

According to Sadgali et al. (2019), the virtualization and technological development makes bank transactions digitalized and that allows online transactions to go under various frauds. Their major objective is to find the best fraudulent transaction detecting model to implement in their future work. For their analysis, they have used supervised machine learning

techniques on a single generic dataset. Those techniques are K-nearest neighbor, random forests, decision tree, and support vector machines. Out of many performance indicators they have chosen two indicators that are Mean squared error MSE and accuracy. Support vector machine technique has given the least MSE value with highest accuracy of 99.7% that has proven to be the best among other techniques according to their analysis (Sadgali et al., 2019).

According to Popat & Chaudhary, "There are four types of Credit Card Frauds : Card not present, Skimming, Phishing, Lost/Stolen Card" (2018) . They proposed that there are various machine learning techniques to deal with credit card frauds like deep learning, logistic regression, Naive Bayesian, SVM, Neural Network, K nearest neighbor (KNN), etc, (Popat & Chaudhary, 2018). For their research, they have studied different supervised machine learning techniques for fraud detection. From their study, Popat & Chaudhary (2018) concluded that machine learning techniques are best in comparison to predictive, clustering, and outlier detection for credit card fraud detection as they have high accuracy and detection. Suggested that financial organizations can select one of the methods from their models to increase profits and decrease losses.

Another research by Jain et al. (2019) explained various types of credit card fraud and existing techniques in their research paper. They have conducted research on Decision Tree, Logistic regression, SVM, Artificial Neural Networks (ANN), Bayesian Network, KNN, and Fuzzy Logic Based System to detect fraud transactions. As a result of their study, they found that Naive Bayesian Network and Neural Network have given the higher accuracy and Fuzzy Logic system and logistic regression gave the lowest accuracy whereas KNN and SVM gave medium accuracy. Artificial Neural Network (ANN) has attained the highest accuracy of 99.71% with a 99.68% detection rate (Jain et al., 2019), they also concluded that all these techniques have

shown a major drawback that they don't give the same results in all environments (Jain et al., 2019). Jain et al. (2019) proposed solution for this gap is to create a hybrid of these techniques to cancel out the limitations and get better performance.

All of the above-mentioned approaches gave excellent results with high accuracy rates. The first approach posted an accuracy of 99.6% using Local Outlier Factor and Isolation Forest algorithms (Maniraj et al., 2019), the second approach gave a result with 99.7% accuracy using the support vector machines technique (Sadgali et al., 2019). The third approach proposed to use their models according to the requirements, while the last approach provided the Artificial Neural Network technique giving the highest accuracy with the highest precision of 99.71% and 99.68% respectively (Jain et al., 2019). These results show how accurate machine learning models are in detecting fraudulent credit card transactions that guarantee a solution in controlling these frauds while saving a lot of resources and money that banks and credit card providers are spending to overcome this problem.

Data and Project Management Plan

2.1 Data Management Plan

Data Management is one of the most important processes of any project. It involves different tasks like data collection, data cleaning, data preparation, data transformation, data storage, data backup, etc. Analyzing the data requirements and collecting data for the project is the first and important step of any project. Data collection is the most difficult and time taking process. Especially collation of data like Credit card transactions is the toughest part. Because of confidentiality and customer privacy availability of these kinds of datasets is very rare. Even though they are available most of these datasets are transformed to eliminate the leakage of customer confidentiality information. The dataset that we will be using in this research project is collected from Kaggle.com. Kaggle is a data science and machine learning community. It keeps track of credit card transactions. The dataset that we are using for this project is the purchase spendings of some of the card users in the month of September 2013, the data has been created by their two days purchase spendings. The Only 492 of the 284,807 transactions in the sample are fake. The information in the dataset is quite valuable. The dataset available in Kaggle is imbalanced. Data collection techniques involve Analyzing the usage patterns on each card and determining if there is any variation from normal spending habits is the most effective technique of disclosing CCF. An individual's normal spending patterns tend to be of a certain type (Sadineni, 2020). Each credit card holder can be interpreted as a set of patterns using this behavioristic profile. Details such as purchasing categories, spending patterns, frequency of purchases, and so on may be included in the behavioristic profile. Certain trends that change indicate a potential hazard to the transaction system. In order to obtain a more accurate result, it is critical to use genuine data rather than prepared data. The authorities who give the data set, on

the other hand, are bound by the law to protect the privacy of customers. Before sharing the information, it is anonymized. Examining each card's usage trends and determining whether there is any variation from "normal" spending patterns. An individual's normal spending habits tend to be of a certain type. Each credit card holder can be interpreted as a set of patterns using this behavioristic profile ((Dornadula & Geetha, 2019). Details such as purchasing categories, spending patterns, frequency of purchases, and so on may be included in the behavioristic profile. The occurrence of certain patterns indicates a potential hazard to the transaction system. Data mining capabilities such as pattern recognition, data classification, and data model identification can be used to detect CC in advance.

First and foremost, We got this dataset from Kaggle.com website, a data analysis service that offers free and open source datasets. This dataset has total 31 columns, with 28 of them labeled from v1 to v28 to preserve sensitive information. Time, Amount, and Class are represented by the other columns. The time difference between the first and subsequent transactions is shown in this graph. The amount of money exchanged is referred to as the amount. The Class field has two values class1 and class2. Class1 means a genuine transaction and class0 means a fraud transaction. The data set has total of 284,807 transactions in it and it is a size of 143MB (Mega Bytes) while using as comma separated values format (.CSV) and approximately 65.9MB while zipped, and it is an open source dataset no permission is required to share and preserve. While the actual real time credit card transactions data is only available with banks and credit card providers to access, analyze, share and store that data the permission and consent of banks and customers is required.

Metadata of my data set consists the information of the dataset like the description of the fields and formats of each field in the dataset. It also includes the dataset size, dataset source.

Metadata file is stored in the Google cloud storage bucket which we are using for this project. Which will be useful to get the quick glimpse of the dataset. This metadata will be stored in a text file it requires a storage of less than 30KB (Kilo Bytes).

Data storage is one of the important steps of this research paper. It is because to avoid any downtime or losing data. It is most important to safely securing the cleaned and prepared data, splitting data for training and testing purpose for model building because if we lose data at any stage of the project it will cause to halt the process and sometimes we might need to start the whole data related process from the beginning which will result in delay or failure of project. For my project we are using Google cloud Storage to store and secure my data. Cloud storage charges for storing data but it provides high availability and durability.

We are also backing up my dataset and data related to the project in two more places. One is my local computer and another is a GitHub repository. Which will provide extra security to my datasets and help in not delaying the project due to data loss at any stage of the project. And, we are making sure that this data will be available at least for a period of 3 years. All the datasets and data related to this project is handled and managed by me because we are solely working on this project. All data access and activity permissions will be under my control. Which we can control under Google IAM roles for Cloud Storage. Which will ask permission to others whoever want access to the data under my storage bucket. We can control the settings of access provide and forbid.

2.2 Project Development Methodology Data Description

For my research project we choose to follow the CRISP-DM methodology which stands for Cross Industry Standard Process for Data Mining. Which has six standard steps to complete a Data Mining related projects. Those are Business Understanding, Data Understanding, Data

Preparation, Modeling, Evaluation, and Deployment. These steps will guide and make clear cut path for a model building project. Some of these steps are inter-connected. This process is iterative and adaptive. In addition to these steps would like to add another step that is Deliverables. The description of the different stages of the CRISP-DM.

2.2.1 Business Understanding

Business Understanding is the starting step of the CRISP-DM. Which focuses on the understanding the business that is Credit card frauds in our case. It is one of the first and most important phase of this project under this phase Understanding credit card fraud, types of credit card frauds, problems facing due to these frauds, and demand for the solution to deal with these frauds are observed. To detect and prevent fraudulent transactions we need an accurate fraud detection Machine Learning model and that is the main objective of this project. These models will help in detecting the fraudulent credit card transactions in no time. That will help recover the amount of fraud transactions from fraudsters. That will ultimately save a lot of money and resources of the banks, merchants, and customers.

2.2.2 Data Understanding

The second phase of this project is understanding the data, to build a model to detect fraudulent credit card transactions we need to have credit card transactions and at least some of those transactions should have fraud transactions. A huge dataset is required and data collection is one of the most difficult processes especially when it comes to financial issues like credit card theft because of customers personal information so it is hard to access real time data from banks or credit card providers. We have collected an open source fraudulent credit card transactions dataset from Kaggle.com. This dataset has 284,807 transactions and few of the are fraudulent transactions. It has total 31 columns. Some of these fields are time, amount, and class of

transaction whether fraud or genuine etc. there are no missing values and the quality of the dataset is good to get accurate results.

2.2.3 Data Preparation

Data Preparation is the third phase of my project. In this stage we will prepare my Fraudulent Credit Card transactions data that is creditcard.csv. We will be using this prepared dataset for model building in the next phase. This is one of the important phases of this project because any machine learning project must have cleaned and complete data. Data Preparation and transformation will help in cleaning, formatting and improving the data quality and integrity. It will finally provide a clean and complete data that is further required to split data for training and testing for model building. Time field of this data is in the seconds format. It is not in actual Time format because that will not much helpful in for training my models. And It will further increase the process to calculate time duration between each transaction. So, Time field will provide the duration between the first transaction and rest of the transactions. It will help in evaluating how frequently transactions are taking place. Likewise amount field is formatted to USD currency. And the Class field is created in binary format that will ease the process of distinguishing fraud and genuine transactions. This will be helpful while training and testing the model by dividing the data into 70 percent and 30 percent respectively. The other field V1 to V28 are converted in the decimal format that is because of securing customers' confidentiality. After Data Preparation we will collect and store this Fraudulent credit card transactions dataset to a storage bucket. We will be using Google Cloud Storage for storing the data related to this project in the bucket called Vamshi-CCD-Frauds-270.

2.2.4 Model Building

In this phase my ultimate goal is to build few models for detecting Fraud Credit Card Transactions. Model Building is the forth phase of this project also a fourth step in the CRISP-DM project development methodology. The models we will be building are Random Forest, Logistic regression, Support Vector Machine (SVM), Decision tree, Artificial Neural Network (ANN), and K Nearest Neighbor (KNN). To build these models we need a cleaned dataset, that will already be cleaned and prepared in the prior step. First, we will gather dataset from my Cloud Storage bucket Vamshi-CCD-Frauds-270 and divide that dataset for training and testing purpose. And then train each model with the training dataset. If some of these models require additional data preparation we will go back to the Data Preparation and Prepare the Data according to that model. That is one of the best flexibilities in using the CRISP-DM methodology. After building these models we will proceed with the testing these models' accuracy using Testing dataset.

2.2.5 Model Evaluation

Model Evaluation is next step after Building the Model. It also a fifth step of the CRISP-DM method. In the above phase we will use only 70 percent of data for training purpose of the Model, here we will use the rest of the 30 Percent of the Data for Testing the above Built Models. In this phase we can evaluate the accuracy of the all models and calculate the precision. This way we can find the best accurate model for Detecting Fraudulent Credit card transactions. We will Compare all the models and provide the best accurate and suitable model for Fraud detection. In this phase we can evaluate my models by comparing them in a way that which model is most powerful in solving Fraud Credit card transactions issue. we will also make the

time and cost evaluation that will also helpful While implementing the model in the actual world of credit card frauds.

2.2.6 Deployment

This is the final phase of the CRISP-DM methodology. At this stage we will be deploying the most accurate model to detect the fraudulent credit card transactions. Banks, Financial Institutions, and Credit card providers can use these models to detect their customers credit card transactions and find a fraudulent transaction. Before deploying the models in their system users of these models should check their model deployment environment and make sure that they meet all the data, software, and hardware requirements. And will analyze their requirements and objectives. Like only detecting these transactions or also preventing them before they actually take place. After Deploying and using these models can collect results and reviews of the Model. That will be helpful in further development of the Model.

2.2.7 Deliverables

Deliverables is the last step of this project. In this phase we will be providing an accurate Credit Card Fraud Detection Model, Presentation of the project, Dataset source, and final Project report.

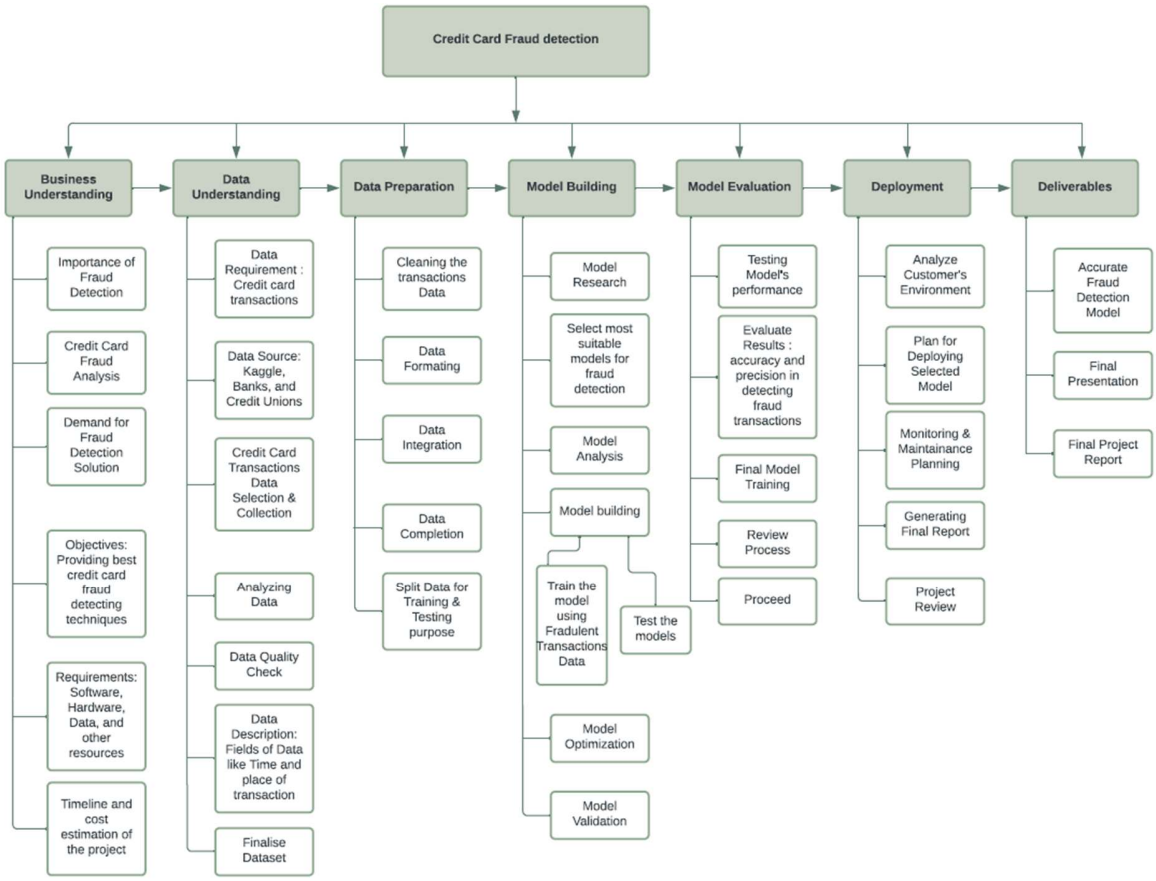
2.3 Project Organization Plan

For every project organization plan should be there, because it will help in dividing the important phases of the project and implementing those tasks in a planned manner. For our project we are using Work Breakdown Structure (WBS). It breaks down the important phases of the project in a smaller task and leads to the final project deliverables. So for our project the first important stage is Business understanding, under this we have included sub tasks like importance of the fraud detection, the analysis of credit card frauds, the demand for the solution for credit

card frauds, objectives like providing the best credit card fraud detection models, project requirements, and timeline and costs associated with the project, In the second phase we have included task and subtask related to the Data Understanding, in this stage the subtasks of our projects are finding data requirements to analyze and build a credit card fraud detection model, finding out and deciding the data sources, collecting data, checking the quality of data, providing data description and finalizing the dataset for next stage. The third stage our project is Data Preparation, under this we perform the subtasks like, cleaning the CreditCard.csv dataset, formatting the datasets, checking and attaining the data integration, completion and finally splitting the data for training, testing, and validation purposes for building the models like SVM, ANN, Decision tree and Random forest. So the fourth stage of our project is building the model. At this stage we take care of tasks like researching about the models and selecting few suitable models for detecting credit card frauds, model analysis and building the model using the training dataset then optimizing those using the testing and validation datasets. At the fifth stage we evaluate the Models by testing the models using testing dataset and validate the models using validation dataset, evaluating the results and metrics of the models and doing final model training the proceeding the model for Deployment at the sixth stage. Under the Deployment stage, our focus is to deploy the model to be used by the customers so we check the environment of the customer, in our case our customers are the banks and Financial Institutions, so we ensure that their environment and platform support our models and plan to deploy our models and monitoring and maintaining the models. At the last stage which is Deliverables of our project, at this stage we will provide the final accurate credit card fraud detection model, giving a final presentation, and submitting the final project report will end our project.

Figure 1

Work Breakdown Structure



Note. Work breakdown structure with seven steps

2.4 Project Resource and Requirement Plan

The requirements specification is a technical description of the software, hardware, resource requirements. It lists the functional, performance, and security requirements for this product.

Table 1

Resource and Cost Estimation

Function	Resource Type	Resource	Time Duration	Cost
Data Collection	Software	Kaggle	08/01/2021 - 08/15/2021	Free
Cloud Service Management Tool	Software	Gcloud CLI	09/10/2021 - 12/10/2021 (4 months)	Free
Data Preprocessing	Software	Google Data Prep	09/15/2021 – 12/10/2021 (4 months)	$24.88 ($6.22 /month/ instance for 10 hours)
Data Storage	Hardware	Google Cloud Storage Bucket	09/01/2021 - 12/31/2021 (4 months)	$ 0.12 $0.03/ GB for 25GiB
Data Processing	Hardware	GCP Dataproc (Compute Engine, Persistent Disk)	09/01/2021 - 11/31/2021 (4 months)	$24.88 ($6.22 /month/ instance for 10 hours)
Build, train, test and deploy	Software	Vertex AI	10/15/2021 - 12/15/2021(2 months)	$12.70 ($6.35/month/hour)
ML Framework	Software	Tensorflow, Keras, Scikit-learn	10/15-2021 - 12/15/2021/ (2 months)	Free
Software Development	Software	Python 3.9	09/01/2021 - 12/31/2021 (4 months)	Free
Visualizations	Software	Tableau Desktop	12/01/2021 - 12/31/2021 (1 month)	Free

Note. The functional, performance, and security requirements associated with this product.

In this project we have used several Google Cloud Services, which provide different cloud computing services at an affordable cost. Which provides all services, those required for data science or machine learning projects. For my project most of the software and hardware we will be using are provided by Google Cloud Platform. Google cloud platform provides services which are highly availability across many zones and which are cost effective so we have choosen to use Google Cloud Services. In the Cloud storage we will create a bucket named Vamshi-CCD-Frauds-270. In this bucket we have stored the Credit card fraud transactions dataset. After that for data preparation we are using Google Data Prep, which is a data cleaning and preparing tool. We have stored the cleaned data in the Cloud Storage. We have used Dataproc for Extract, Transform, and load (ETL) and data processing. Vertex AI is also developed by google research, we will use this for training and testing my dataset. We are also planning to use Tensorflow, it is a Machine learning software library provided by Google Cloud platform, that is free and open source. The programming language we will use for this project is Python 3.9 that is also free. The visualization tool, we will use for this project is Tableau Desktop that is free for students for a period of one year. So the overall estimated cost for this project is approximately 1000$ including software, hardware, and all the required resources mentioned in table 1.

2.5 Project Schedule

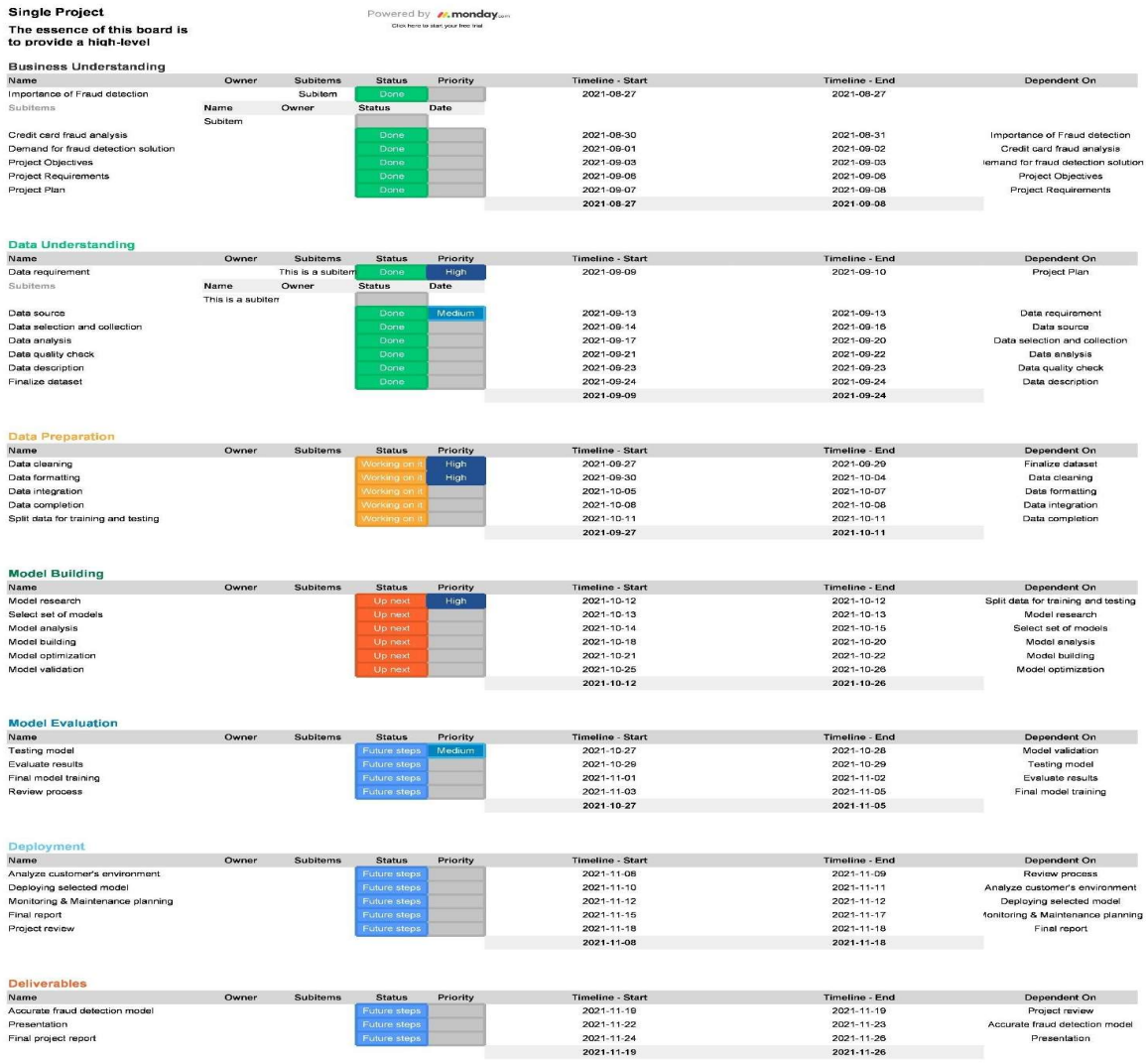
2.5.1 Gantt Chart

For this project, we have used the Gantt chart for Scheduling, dividing, and assigning our project tasks among our team members. Primarily we have divided tasks according to their priority like high, medium, and low priority tasks so that we can put extra effort and allot more time to high priority tasks, Secondarily, We have created a table to update the status of each task like done, working on it, up next, and future steps. Thirdly, we have assigned the task owner and

sub-items under that task, Then we have created a timeline for each task and subtasks by providing the start date and end time. This project starts on 27th August 2021 and ends on 26th November 2021. Lastly, we also have included that each task depends on which other tasks so keep track of the dependable tasks. A vertical black colored line on the Gantt shows the specified day's tasks of our project.

Figure 2

Gantt Chart board



Single Project

The essence of this board is to provide a high-level

Powered by monday.com

Click here to start your free trial

Business Understanding

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Importance of Fraud detection		Subitem	Done		2021-08-27	2021-08-27	
Subitems	Name	Owner	Status	Date			
	Subitem						
Credit card fraud analysis			Done		2021-08-30	2021-08-31	Importance of Fraud detection
Demand for fraud detection solution			Done		2021-09-01	2021-09-02	Credit card fraud analysis
Project Objectives			Done		2021-09-03	2021-09-03	lemand for fraud detection solution
Project Requirements			Done		2021-09-06	2021-09-06	Project Objectives
Project Plan			Done		2021-09-07	2021-09-08	Project Requirements
					2021-08-27	2021-09-08	

Data Understanding

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Data requirement		This is a subitem	Done	High	2021-09-09	2021-09-10	Project Plan
Subitems	Name	Owner	Status	Date			
	This is a subitem						
Data source			Done	Medium	2021-09-13	2021-09-13	Data requirement
Data selection and collection			Done		2021-09-14	2021-09-16	Data source
Data analysis			Done		2021-09-17	2021-09-20	Data selection and collection
Data quality check			Done		2021-09-21	2021-09-22	Data analysis
Data description			Done		2021-09-23	2021-09-23	Data quality check
Finalize dataset			Done		2021-09-24	2021-09-24	Data description
					2021-09-09	2021-09-24	

Data Preparation

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Data cleaning			Working on it	High	2021-09-27	2021-09-29	Finalize dataset
Data formatting			Working on it	High	2021-09-30	2021-10-04	Data cleaning
Data integration			Working on it		2021-10-05	2021-10-07	Data formatting
Data completion			Working on it		2021-10-08	2021-10-08	Data integration
Split data for training and testing			Working on it		2021-10-11	2021-10-11	Data completion
					2021-09-27	2021-10-11	

Model Building

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Model research			Up next	High	2021-10-12	2021-10-12	Split data for training and testing
Select set of models			Up next		2021-10-13	2021-10-13	Model research
Model analysis			Up next		2021-10-14	2021-10-15	Select set of models
Model building			Up next		2021-10-18	2021-10-20	Model analysis
Model optimization			Up next		2021-10-21	2021-10-22	Model building
Model validation			Up next		2021-10-25	2021-10-26	Model optimization
					2021-10-12	2021-10-26	

Model Evaluation

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Testing model			Future steps	Medium	2021-10-27	2021-10-28	Model validation
Evaluate results			Future steps		2021-10-29	2021-10-29	Testing model
Final model training			Future steps		2021-11-01	2021-11-02	Evaluate results
Review process			Future steps		2021-11-03	2021-11-05	Final model training
					2021-10-27	2021-11-05	

Deployment

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Analyze customer's environment			Future steps		2021-11-08	2021-11-09	Review process
Deploying selected model			Future steps		2021-11-10	2021-11-11	Analyze customer's environment
Monitoring & Maintenance planning			Future steps		2021-11-12	2021-11-12	Deploying selected model
Final report			Future steps		2021-11-15	2021-11-17	Monitoring & Maintenance planning
Project review			Future steps		2021-11-18	2021-11-18	Final report
					2021-11-08	2021-11-18	

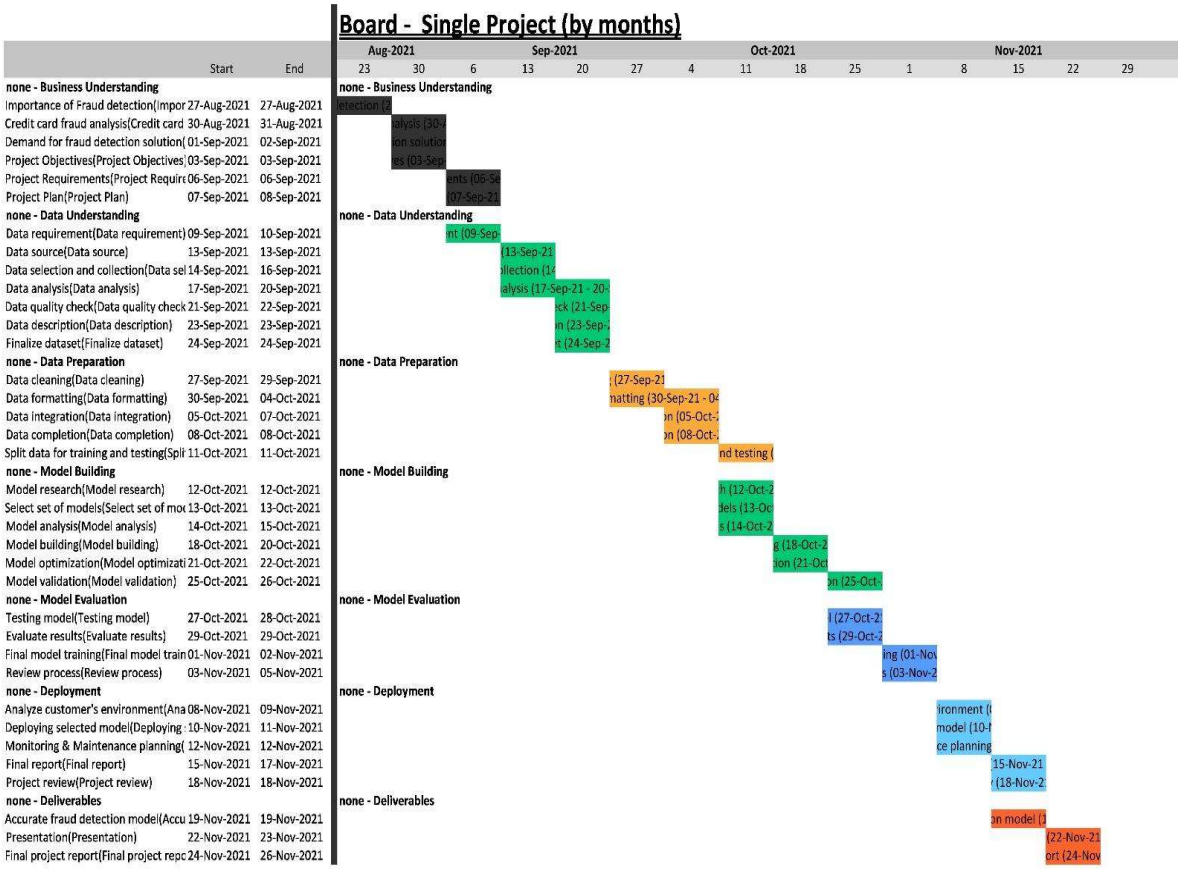
Deliverables

Name	Owner	Subitems	Status	Priority	Timeline - Start	Timeline - End	Dependent On
Accurate fraud detection model			Future steps		2021-11-19	2021-11-19	Project review
Presentation			Future steps		2021-11-22	2021-11-23	Accurate fraud detection model
Final project report			Future steps		2021-11-24	2021-11-26	Presentation
					2021-11-19	2021-11-26	

Note. Gantt chart board showing task status and timeline Figure 3

Figure 3

Gantt Chart



Note. Gantt chart showing all tasks, timeline, and deadline.

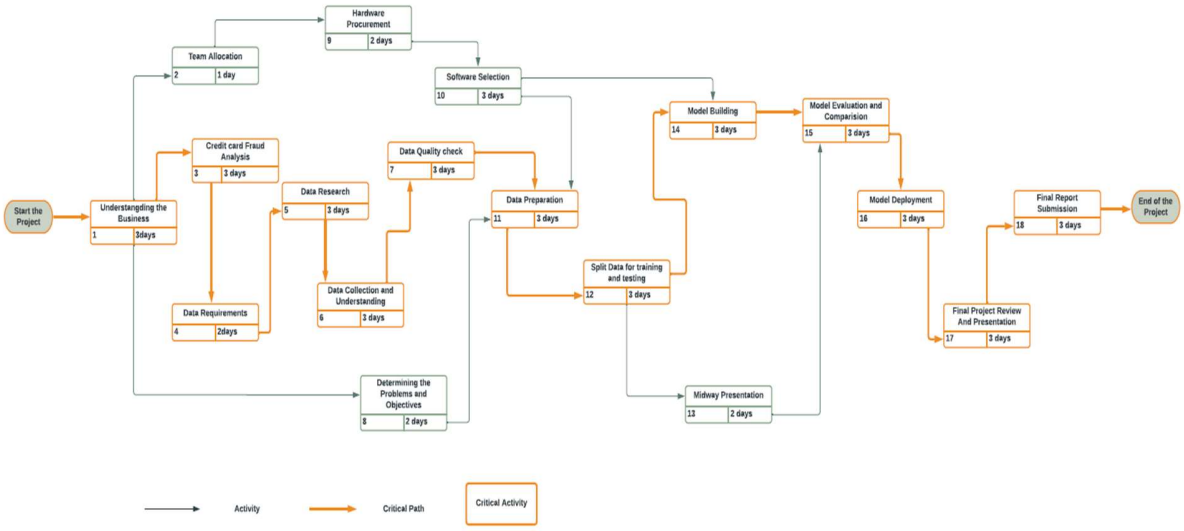
2.5.2 PERT Chart

PERT chart is a visual representation of different tasks associated with the project, PERT chart created before starting the project by analyzing all the tasks to do for the successful completion of the project, it also includes the number of days each task takes to complete. PERT chart consists of sequential as well as parallel paths from starting to the end. Among all the paths the path which is taking the highest time and has important tasks is the Critical Path. For our project PERT chart starts with Understanding the Business and Ends with Final Report

Submission. The Critical path of our PERT chart takes 38 days. The Critical path and critical tasks are shown in the Orange color.

Figure 4

PERT Chart



Note. PERT chart to find critical activity path and timeline

Data Engineering

3.1 Data Process

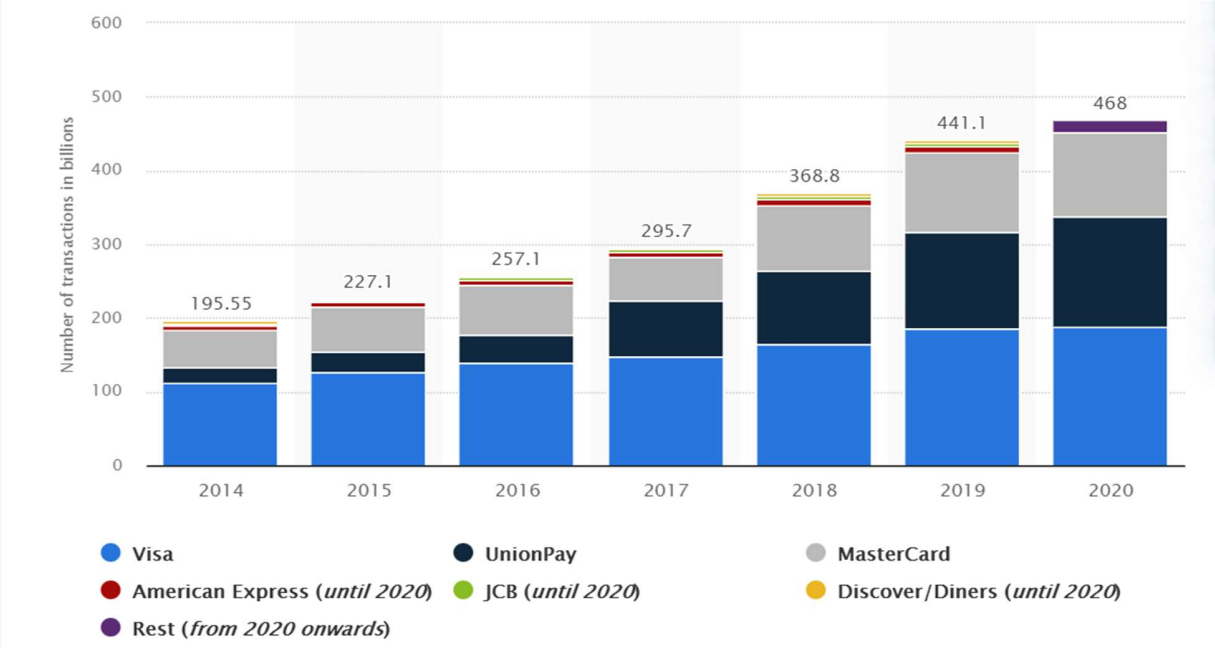
To build a credit card fraud detection machine learning model we need to have Credit Card Transactions data, so transaction data is one of the most important requirements of this project. Credit card transactions data can be either stored data that can be historical transactions or streaming data that is current transactions of the credit card users. So for this project, We have collected and used the historical transactions of credit card users for my model training, testing, and building. This data can be availed from credit card providers, payment processors, or open-source data sources. We have collected data from Kaggle.com that is an open-source data provider which is free. Collected data has been stored in the Cloud storage services and in my local machine. After that, We have analyzed the dataset to check its quality and usability by exploring the data using a Jupyter notebook. In this process, we came to know the formats, types of fields, missing, and ambiguous values in the dataset. It will further provide scope to take the necessary steps to clean, format, and transform the dataset according to the model requirements. For analyzing the raw dataset and performing necessary changes we are planning to use Jupyter Notebook. Using the Jupyter Notebook data quality check, data cleaning, Null value replacement, data formatting, and data transformation will be done. After this process, It helps to identify the important features of the dataset which will be used for building the models like Logistic regression, Random forest, Support Vector Machine, Artificial Neural Networks, and Decision Trees. Now, we have decided to use a part of this dataset to train and rest is for testing the models, which is an important step in building a model.

3.2 Data Collection

In this section, we want to focus on the data collection process for collecting required data for building an accurate credit card fraud detection model. Training and testing is a process of model creation so to train and test the model some of the transactions of this dataset must be fraudulent. The credit card transactions data is generated by the usage of credit cards by its user. The number of transactions done by credit card users in a year is provided in figure 5.

Figure 5

Visa, MasterCard, UnionPay Transaction Volume 2020



Note. From A number of purchase transactions on global general-purpose card brands American Express, Diners/Discover, JCB, Mastercard, UnionPay, and Visa from 2014 to 2020 (in billions), by Best, R. de. (2021, July 9), (https://www.statista.com/statistics/261327/number-of-per-card-credit-card-transactions-worldwide-by-brand-as-of-2011/).

From figure 5, we can notice that every year hundreds of billions of credit card transactions are taking place and a huge amount of big data is being generated by credit card

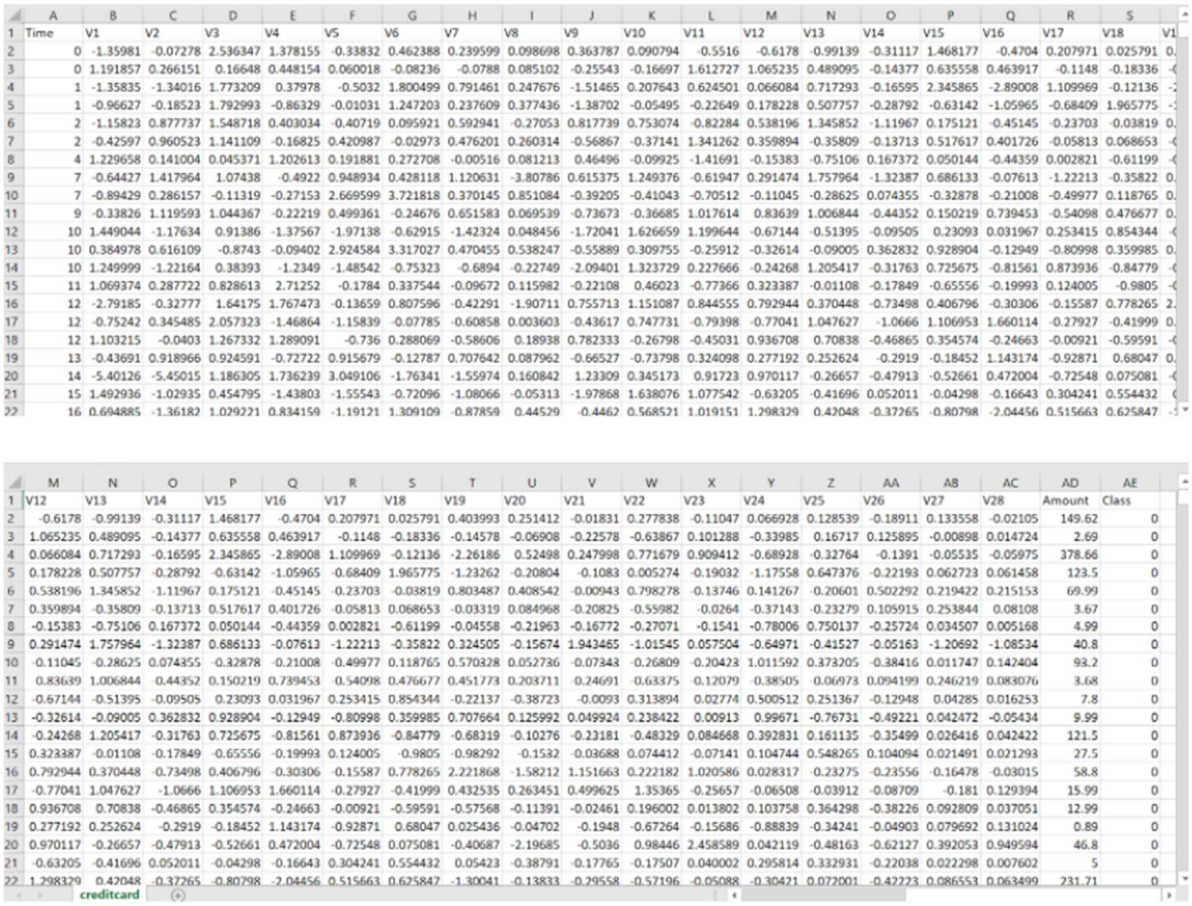
users. It also gives information about the number of credit cards transactions taken place in recent years, 468 billion transactions were done by credit card users in 2020 alone. The bar graph also states that every year the number of credit card transactions is increasing drastically. The credit card transactions doubled in 2019 as compared to 2015. But, this data is not available to everyone because it holds a lot of personal, private, and confidential information of credit card users. The data might consist of a 16 digit Credit Card number, address of the user, balance or credit limit of the user, phone number, etc. If credit card providers provide their customers' transactions data to everyone, Fraudsters can use that data to obtain important information about credit card users and that will facilitate them to commit more frauds very easily than ever because of that reason no bank or credit card provider is very skeptical about sharing their customers' credit card transactions to others. So this data is available with banks, credit card providers, and payment processors. When data is generated by users it is stored and saved by the credit card providers and payment processors. For Example, Visa, MasterCard are payment processors that connect merchants and credit card issuer banks like WellsForgo, and financial institution companies like American Express. Each transaction is recorded as a row like that every day hundreds of millions of transactions are converted and saved in rows by payment processors and banks in their local database. This data can be used to build the model. This dataset consists of details like Transaction ID, Transaction Time, Card Number, Customer Name, Card Holder address, Amount of Transaction, Location of Transaction, Merchant ID, Merchant Name, Merchant Address, and a few more details regarding the transaction. The transactional data consist of qualitative and quantitative data. Qualitative data represents the descriptive fields like customer name, merchant name, product details, etc. Whereas quantitative data consist of numeric values like the number of transactions, amount of transactions. The sample dataset

columns consist of details of the credit card transactions. Like, TransactionID states that unique id for each transaction, TX_DATETIME represents the date and time when a transaction has taken place, CUSTOMER_ID means unique Identification Number provided individual customers when they open a credit card account with the bank, TERMINAL_ID represents the terminal where the credit card was used for the transaction, TX_AMOUNT is amount of each transaction, and TX_FRAUD provides the information of the transaction whether the transaction is legit or not in this Zero represents a legit transaction and One represents a Fraudulent transaction.

The data can also be collected by tracking the transactions of the customers, storing each transaction as a row and each row will consist of all the details of that particular transaction. All of these transactions will be recorded, collected, and temporarily stored in a local machine in a CSV file format, and then that CSV file will be stored in Cloud Storage for easy accessibility. It can easily be accessible for preprocessing, cleaning, preparing, future selection, training, testing, and model building. The size of the transactional dataset can be anywhere between hundreds of Megabytes (MB) to gigabytes (GB). For my project, we are using a credit card transactions dataset which is provided by Kaggle.com. Which is an open-source dataset available freely and can be used for building fraud detection models. This dataset was originally collected and provided by the Machine Learning Group-ULB (Université Libre de Bruxelles). They have shared this dataset on the Kaggle group named Credit Card Fraud Detection. The dataset name is creditcard.csv, the size of the dataset is 150.83 MB, It has 284,807 transactions and a few of those are fraud transactions. This dataset is the two days recorded transactions of European credit cardholders in September 2013. For my project, we need at least a few hours of recorded transactions so this dataset will be very useful and sufficient.

Figure 6

Credit Card Fraud Detection



Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15	V16	V17	V18	V1
0	-1.35981	-0.07278	2.536347	1.378155	-0.33832	0.462388	0.239599	0.098698	0.363787	0.090794	-0.5516	-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.
0	1.191857	0.266151	0.16648	0.448154	0.060018	-0.08236	-0.0788	0.085102	-0.25543	-0.16697	1.612727	1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-(
1	-1.35835	-1.34016	1.773209	0.37978	-0.5032	1.800499	0.791461	0.247676	-1.51465	0.207643	0.624501	0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-:
1	-0.96627	-0.18523	1.792993	-0.86329	-0.01031	1.247203	0.237609	0.377436	-1.38702	-0.05495	-0.22649	0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-:
2	-1.15823	0.877737	1.548718	0.403034	-0.40719	0.095921	0.592941	-0.27053	0.817739	0.753074	-0.82284	0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.
2	-0.42597	0.960523	1.141109	-0.16825	0.420987	-0.02973	0.476201	0.260314	-0.56867	-0.37141	1.341262	0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-(
4	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.00516	0.081213	0.46496	-0.09925	-1.41691	-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-(
7	-0.64427	1.417964	1.07438	-0.4922	0.948934	0.428118	1.120631	-3.80786	0.615375	1.249376	-0.61947	0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.
7	-0.89429	0.286157	-0.11319	-0.27153	2.669599	3.721818	0.370145	0.851084	-0.39205	-0.41043	-0.70512	-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.
9	-0.33826	1.119593	1.044367	-0.22219	0.499361	-0.24676	0.651583	0.069539	-0.73673	-0.36685	1.017614	0.83639	1.006844	-0.44352	0.150219	0.739453	-0.54098	0.476677	0.
10	1.449044	-1.17634	0.91386	-1.37567	-1.97138	-0.62915	-1.42324	0.048456	-1.72041	1.626659	1.199644	-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-(
10	0.384978	0.616109	-0.8743	-0.09402	2.924584	3.317027	0.470455	0.538247	-0.55889	0.309755	-0.25912	-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.
10	1.249999	-1.22164	0.38393	-1.2349	-1.48542	-0.75323	-0.6894	-0.22749	-2.09401	1.323729	0.227666	-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-(
11	1.069374	0.287722	0.828613	2.71252	-0.1784	0.337544	-0.09672	0.115982	-0.22108	0.46023	-0.77366	0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-(
12	-2.79185	-0.32777	1.64175	1.767473	-0.13659	0.807596	-0.42291	-1.90711	0.755713	1.151087	0.844555	0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.
12	-0.75242	0.345485	2.057323	-1.46864	-1.15839	-0.07785	-0.60858	0.003603	-0.43617	0.747731	-0.79398	-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.
12	1.103215	-0.0403	1.267332	1.289091	-0.736	0.288069	-0.58606	0.18938	0.782333	-0.26798	-0.45031	0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-(
13	-0.43691	0.918966	0.924591	-0.72722	0.915679	-0.12787	0.707642	0.087962	-0.66527	-0.73798	0.324098	0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.
14	-5.40126	-5.45015	1.186305	1.736239	3.049106	-1.76341	-1.55974	0.160842	1.23309	0.345173	0.91723	0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-(
15	1.492936	-1.02935	0.454795	-1.43803	-1.55543	-0.72096	-1.08066	-0.05313	-1.97868	1.638076	1.077542	-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	(
16	0.694885	-1.36182	1.029221	0.834159	-1.19121	1.309109	-0.87859	0.44529	-0.4462	0.568521	1.019151	1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-:

V12	V13	V14	V15	V16	V17	V18	V19	V20	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
-0.6178	-0.99139	-0.31117	1.468177	-0.4704	0.207971	0.025791	0.403993	0.251412	-0.01831	0.277838	-0.11047	0.066928	0.128539	-0.18911	0.133558	-0.02105	149.62	0
1.065235	0.489095	-0.14377	0.635558	0.463917	-0.1148	-0.18336	-0.14578	-0.06908	-0.22578	-0.63867	0.101288	-0.33985	0.16717	0.125895	-0.00898	0.014724	2.69	0
0.066084	0.717293	-0.16595	2.345865	-2.89008	1.109969	-0.12136	-2.26186	0.52498	0.247998	0.771679	0.909412	-0.68928	-0.32764	-0.1391	-0.05535	-0.05975	378.66	0
0.178228	0.507757	-0.28792	-0.63142	-1.05965	-0.68409	1.965775	-1.23262	-0.20804	-0.1083	0.005274	-0.19032	-1.17558	0.647376	-0.22193	0.062723	0.061458	123.5	0
0.538196	1.345852	-1.11967	0.175121	-0.45145	-0.23703	-0.03819	0.803487	0.408542	-0.00943	0.798278	-0.13746	0.141267	-0.20601	0.502292	0.219422	0.215153	69.99	0
0.359894	-0.35809	-0.13713	0.517617	0.401726	-0.05813	0.068653	-0.03319	0.084968	-0.20825	-0.55982	-0.0264	-0.37143	-0.23279	0.105915	0.253844	0.08108	3.67	0
-0.15383	-0.75106	0.167372	0.050144	-0.44359	0.002821	-0.61199	-0.04558	-0.21963	-0.16772	-0.27071	-0.1541	-0.78006	0.750137	-0.25724	0.034507	0.005168	4.99	0
0.291474	1.757964	-1.32387	0.686133	-0.07613	-1.22213	-0.35822	0.324505	-0.15674	1.943465	-1.01545	0.057504	-0.64971	-0.41527	-0.05163	-1.20692	-1.08534	40.8	0
-0.11045	-0.28625	0.074355	-0.32878	-0.21008	-0.49977	0.118765	0.570328	0.052736	-0.07343	-0.26809	-0.20423	1.011592	0.373205	-0.38416	0.011747	0.142404	93.2	0
0.83639	1.006844	0.44352	0.150219	0.739453	0.54098	0.476677	0.451773	0.203711	0.24691	0.63375	0.12079	-0.38505	0.06973	0.094199	0.246219	0.083076	3.68	0
-0.67144	-0.51395	-0.09505	0.23093	0.031967	0.253415	0.854344	-0.22137	-0.38723	-0.0093	0.313894	0.02774	0.500512	0.251367	-0.12948	0.04285	0.016253	7.8	0
-0.32614	-0.09005	0.362832	0.928904	-0.12949	-0.80998	0.359985	0.707664	0.125992	0.049924	0.238422	0.00913	0.99671	-0.76731	-0.49221	0.042472	-0.05434	9.99	0
-0.24268	1.205417	-0.31763	0.725675	-0.81561	0.873936	-0.84779	-0.68319	-0.10276	-0.23181	-0.48329	0.084668	0.392831	0.161135	-0.35499	0.026416	0.042422	121.5	0
0.323387	-0.01108	-0.17849	-0.65556	-0.19993	0.124005	-0.9805	-0.98292	-0.1532	-0.03688	0.074412	-0.07141	0.104744	0.548265	0.104094	0.021491	0.021293	27.5	0
0.792944	0.370448	-0.73498	0.406796	-0.30306	-0.15587	0.778265	2.221868	-1.58212	1.151663	0.222182	1.020586	0.028317	-0.23275	-0.23556	-0.16478	-0.03015	58.8	0
-0.77041	1.047627	-1.0666	1.106953	1.660114	-0.27927	-0.41999	0.432535	0.263451	0.499625	1.35365	-0.25657	-0.06508	-0.03912	-0.08709	-0.181	0.129394	15.99	0
0.936708	0.70838	-0.46865	0.354574	-0.24663	-0.00921	-0.59591	-0.57568	-0.11391	-0.02461	0.196002	0.013802	0.103758	0.364298	-0.38226	0.092809	0.037051	12.99	0
0.277192	0.252624	-0.2919	-0.18452	1.143174	-0.92871	0.68047	0.025436	-0.04702	-0.1948	-0.67264	-0.15686	-0.88839	-0.34241	-0.04903	0.079692	0.131024	0.89	0
0.970117	-0.26657	-0.47913	-0.52661	0.472004	-0.72548	0.075081	-0.40687	-2.19685	-0.5036	0.98446	2.458589	0.042119	-0.48163	-0.62127	0.392053	0.949594	46.8	0
-0.63205	-0.41696	0.052011	-0.04298	-0.16643	0.304241	0.554432	0.05423	-0.38791	-0.17765	-0.17507	0.040002	0.295814	0.332931	-0.22038	0.022298	0.007602	5	0
1.298329	0.42048	-0.37265	-0.80798	-2.04456	0.515663	0.625847	-1.30041	-0.13833	-0.29558	-0.57196	-0.05088	-0.30421	0.072001	-0.42223	0.086553	0.063499	231.71	0

creditcard

Note. Sample Credit card transaction dataset, Each row represents a single transaction, Each column represents different features of transactions. Adapted from *Credit card fraud detection* (Version 3), by Machine Learning Group - ULB (Université Libre de Bruxelles). 2018, (https://www.kaggle.com/mlg-ulb/creditcardfraud).

The sample of the raw dataset is provided in figure 6. The time field represents the time passed from the first transaction to the following transactions, the amount field represents the amount drawn or used in each transaction, and the Class variable represents the information like fraud or not. The fields of the dataset from V1 to V28 are modified by dimensionality reduction

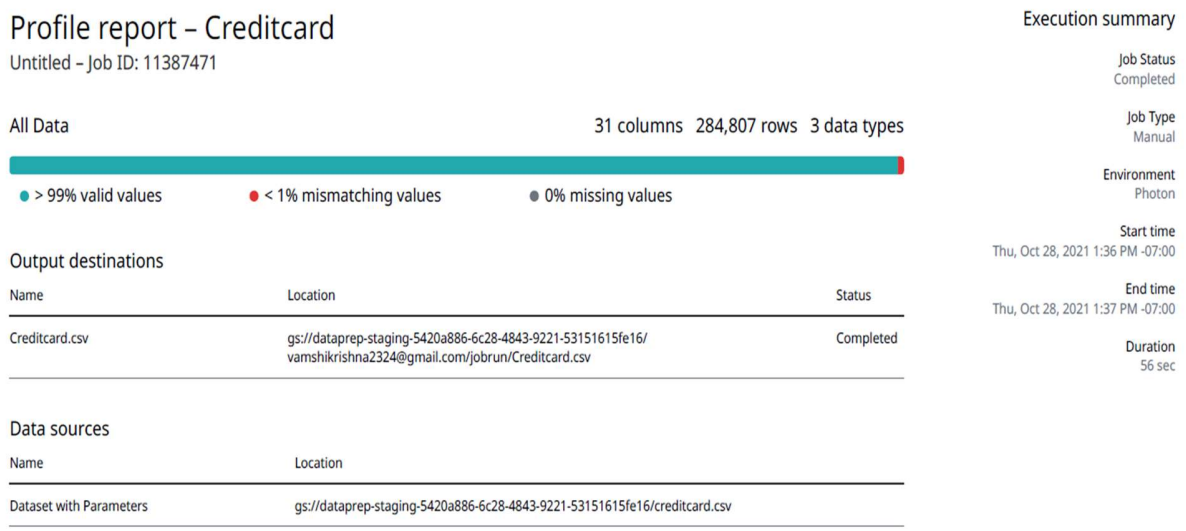
using Principal Component Analysis (PCA) transformation for hiding confidential information of credit card users to protect them from fraudsters.

3.3 Data Pre-processing

This section majorly focuses on data quality. First, we want to ensure that the quality of the data is good, which is most important for any machine learning model. Data cleaning and preparation are done in this step for use in further processes. Using the data which was stored in my cloud storage bucket in the previous step we have performed the cleaning and preparation process on the dataset. We have used Google's data preparation tool Dataprep for this project. We have checked whether there are any missing values in the dataset to replace those with the most suitable values according to the fields of the dataset using mean, median, or mode values. We have checked the uniformity and completeness of the data according to my models. Then, we have also ensured the uniformity and validity of the data.

Figure 7

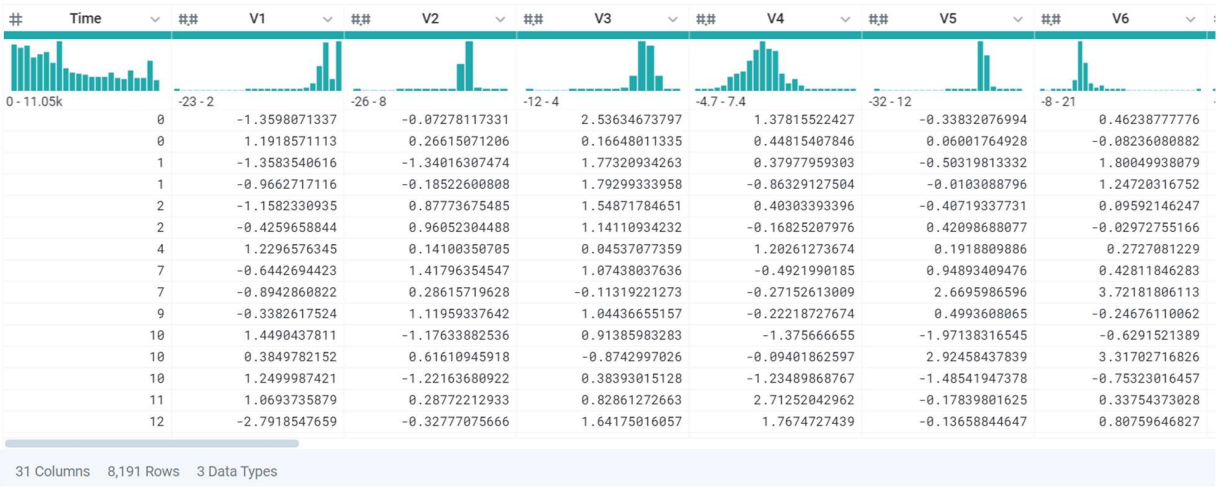
Data Preparation Profile Report



Note. Profile report generated on Dataprep after running the job

Figure 8

Sample of the Pre-processed Dataset



Time	V1	V2	V3	V4	V5	V6
0 - 11.05k	-23 - 2	-26 - 8	-12 - 4	-4.7 - 7.4	-32 - 12	-8 - 21
0	-1.3598071337	-0.07278117331	2.53634673797	1.37815522427	-0.33832076994	0.46238777776
0	1.1918571113	0.26615071206	0.16648011335	0.44815407846	0.06001764928	-0.08236080882
1	-1.3583540616	-1.34016307474	1.77320934263	0.37977959303	-0.50319813332	1.80049938079
1	-0.9662717116	-0.18522600808	1.79299333958	-0.86329127504	-0.0103088796	1.24720316752
2	-1.1582330935	0.87773675485	1.54871784651	0.40303393396	-0.40719337731	0.09592146247
2	-0.4259658844	0.96052304488	1.14110934232	-0.16825207976	0.42098688077	-0.02972755166
4	1.2296576345	0.14100350705	0.04537077359	1.20261273674	0.1918809886	0.2727081229
7	-0.6442694423	1.41796354547	1.07438037636	-0.4921990185	0.94893409476	0.42811846283
7	-0.8942860822	0.28615719628	-0.11319221273	-0.27152613009	2.6695986596	3.72181806113
9	-0.3382617524	1.11959337642	1.04436655157	-0.22218727674	0.4993608065	-0.24676110062
10	1.4490437811	-1.17633882536	0.91385983283	-1.375666655	-1.97138316545	-0.6291521389
10	0.3849782152	0.61610945918	-0.8742997026	-0.09401862597	2.92458437839	3.31702716826
10	1.2499987421	-1.22163680922	0.38393015128	-1.23489868767	-1.48541947378	-0.75323016457
11	1.0693735879	0.28772212933	0.82861272663	2.71252042962	-0.17839801625	0.33754373028
12	-2.7918547659	-0.32777075666	1.64175016057	1.7674727439	-0.13658844647	0.80759646827

31 Columns 8,191 Rows 3 Data Types

Note. Sample of cleaned and prepared data after performing all the necessary cleaning.

Figure 7 and Figure 8, are the profile reports generated after running the data cleaning and preparation steps. Which shows the type of each field of data, there is a total of three types of data in the dataset Integer, Decimal, and Boolean. The Time column is an integer, the Class column is Boolean type, the rest of all the columns are in the decimal format. There were many digits after decimal in columns V1 to V28 to ensure data uniformity. We have changed to 10 digits in the decimals. There are a total of 284,807 valid values in all columns of the dataset. All the column values are unique, all values are valid, there are no mismatched and empty/null values in the dataset after preprocessing.

3.4 Data Transformation

The required data transformation is done in this step. The format and type of all the fields have been changed. The time column represents the time elapsed from first transactions to following transactions so it was formatted to integer type instead of time format because it will be easy to calculate how much time in seconds taken by each transaction from the previous

transaction, amount filed was actually the amount of transaction, some of these transactions can be done in different countries and the amount could be in other currencies so to reduce the anomaly we have changed to a common currency USD, that will be useful in easy understanding of the value of transaction and value of the fraudulent transaction. Then for columns from V1 to V28 dimensionality is reduced by the original data provider using Principal Component Analysis (PCA) transformation for hiding confidential information of credit card users in the interest of banks and financial institutions. Principal component analysis (PCA) is widely used in making data easily understandable, reducing the noisy data, and finding the unmeasured variables, PCA will also be helpful in reducing the number of variables when the dataset has a lot of variables (Volpi, 2020). Our dataset has over 30 variables so PCA has been used to minimize the number of variables in our data. After the PCA transformation, we have checked for the smoothness of the data and presented the sample of data transformation in Figure 9.

Figure 9

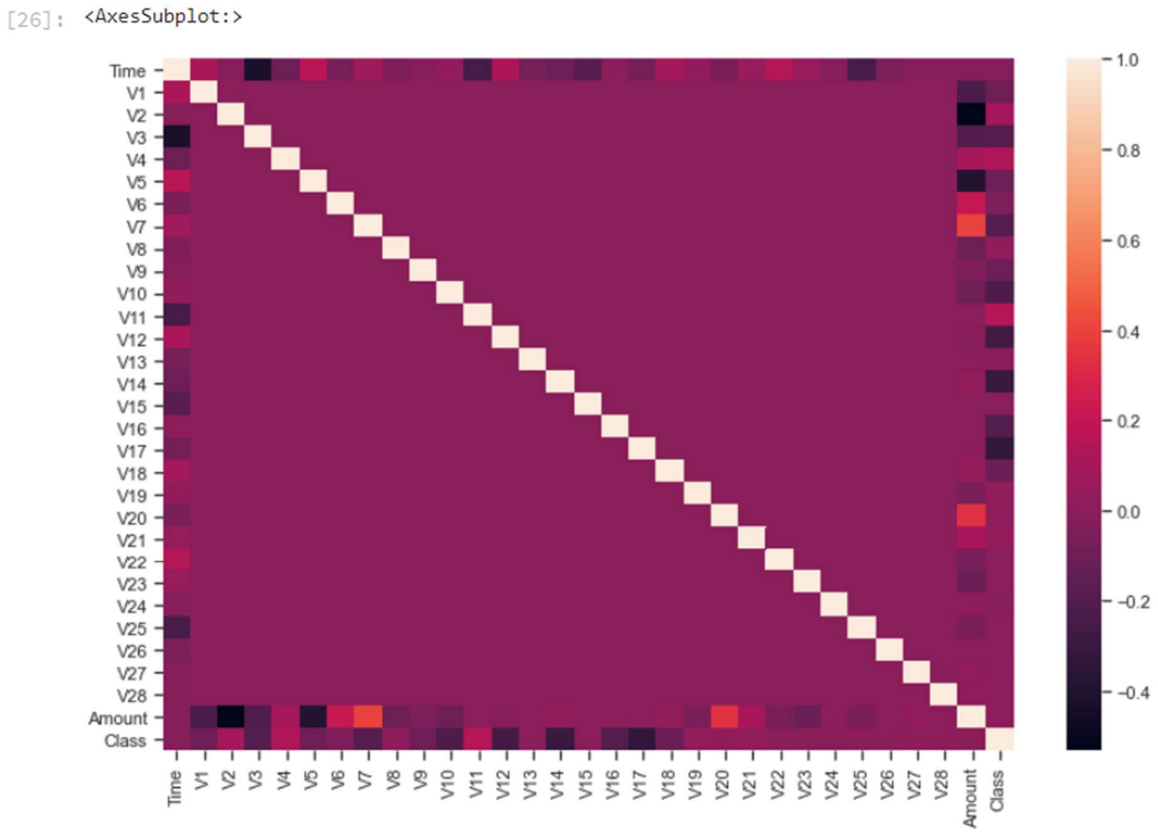
Transformed Pre-processed Dataset

Note. Transformed pre-processed dataset shows the type of data, number of valid, mismatched, and empty values.

Figure 9 shows the format of each field, and the spread of values of each field like the minimum, maximum and highest values of the records.

Figure 10

Correlation matrix



Note. The correlation matrix between all the features of the dataset.

Figure 10, is the correlation matrix between all the features of the dataset. It is obtained after performing a few transformations to the dataset after pre-processing to check the interrelation or dependency among all the variables.

Figure 11

Dimensionality Reduced Dataset

Dataset after the dimentionality reduction

[32]:

	PCA1	PCA2	Class
0	61.271382	1.319310	0
1	-85.661826	-1.043750	0
2	290.316696	0.810812	0
3	35.151659	0.928257	0
4	-18.360281	1.317406	0
...	...	...	...
284802	-87.586281	13.129519	0
284803	-63.560584	0.876867	0
284804	-20.470739	-1.970702	0
284805	-78.350638	0.408125	0
284806	128.652188	0.358662	0

284807 rows × 3 columns

Note. The number of features has been reduced using PCA

Figure 11 is the sample of the transformed dataset after lessening the number of features in the dataset, that has been done using the principal component analysis for easy processing, which will be very useful in processing the dataset in the following steps for training and testing the model. Till now the data is cleaned, formatted, smoothed, standardized, regularised, normalized, and transformed. Now it is ready for data preparation.

3.5 Data Preparation

In the data preparation step, we have split the transformed dataset into 2 sets for training and testing purposes. We have divided the whole dataset into an 80:20 ratio because we am

planning to use 80% of the data to train the model, and the remaining portion of the data we will use for testing the model. Also, a sample size of the dataset is taken for the Validation of the model.

Figure 12

Training Dataset

Training Dataset

	PCA1	PCA2
46038	-0.789572	0.550703
257265	-33.400502	0.682733
282877	-85.903044	1.715710
226150	-65.901692	-1.952008
278800	-32.860732	-1.921369
...	...	...
64434	-78.360618	1.052512
164469	295.523530	-1.734045
256083	-76.850243	-1.957442
217751	188.352639	0.834378
166836	-54.351664	-1.871630

182276 rows × 2 columns

Note. A training dataset consisting of PCA1 and PCA2 columns is obtained from PCA transformation.

Figure 13

Testing Dataset

Testing Dataset

	PCA1	PCA2
132514	-9.350206	1.263144
231874	538.885336	-1.858154
240972	-81.762904	8.562223
91983	52.651627	-1.251596
225669	1602.269868	6.091796
...	...	...
257165	-63.360406	0.076678
115726	240.873601	-1.268867
171040	-85.490328	-2.285120
8497	-70.501579	1.183766
193254	-87.361508	-1.986738

56962 rows × 2 columns

Note. A testing dataset consisting of PCA1 and PCA2 columns is obtained from PCA transformation.

Figure 14

Validation Dataset

Valid Dataset

	PCA1	PCA2
155469	2.089308	0.252500
281294	-87.571748	-1.992597
56434	-86.347268	1.416405
72882	2198.287515	0.233003
154497	-81.362197	-1.984247
...	...	...
130480	-87.350687	-1.129788
157649	-58.771801	-1.650168
166897	-10.640297	-1.739399
173301	-25.160438	1.405633
188984	-63.370286	0.865886

45569 rows × 2 columns

Note. Validation dataset consisting of PCA1 and PCA2 columns is obtained from PCA transformation

Now the training dataset has 182276 records which have shown in figure 12**,** the testing dataset has 56962 records which have shown in figure 13, and the validation dataset has 45569 records which have shown in figure 14. The training dataset will be used to train the models, whereas the test and valid dataset will be used for testing and validating the accuracy of the models.

3. 6 Data Statistics

This section summarizes the actions taken in the previous steps to prepare the data, transform the data, and pre-process the data. Primarily the raw dataset has a total of 31 columns and 284,807 records, the class field was in integer format, rest of all the fields were in float64 format which is also shown in the figure 15. Class is the target variable for detecting fraud in transactions, and rest of all the fields are the features of the model.

Figure 15

Raw Dataset Statistics

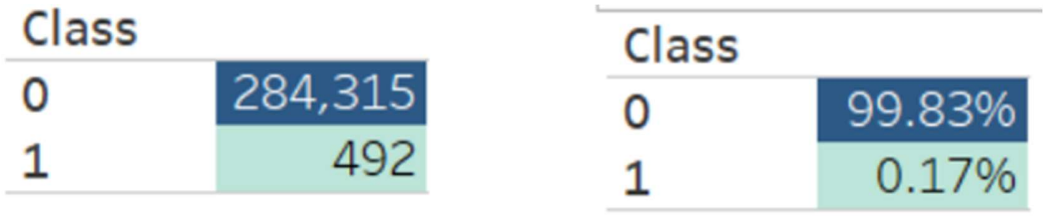
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Note. Dtype is the data type of the each field

Figure 15 shows the basic information of the raw dataset, like column name, number of Non-null records in each field, and Dtype is the data type of the fields. Which gives a quick overview of the raw dataset and its quality, that helps to analyze necessary actions should be taken to get a good quality of dataset.

Figure 16

Number and Percentage of Class Variable



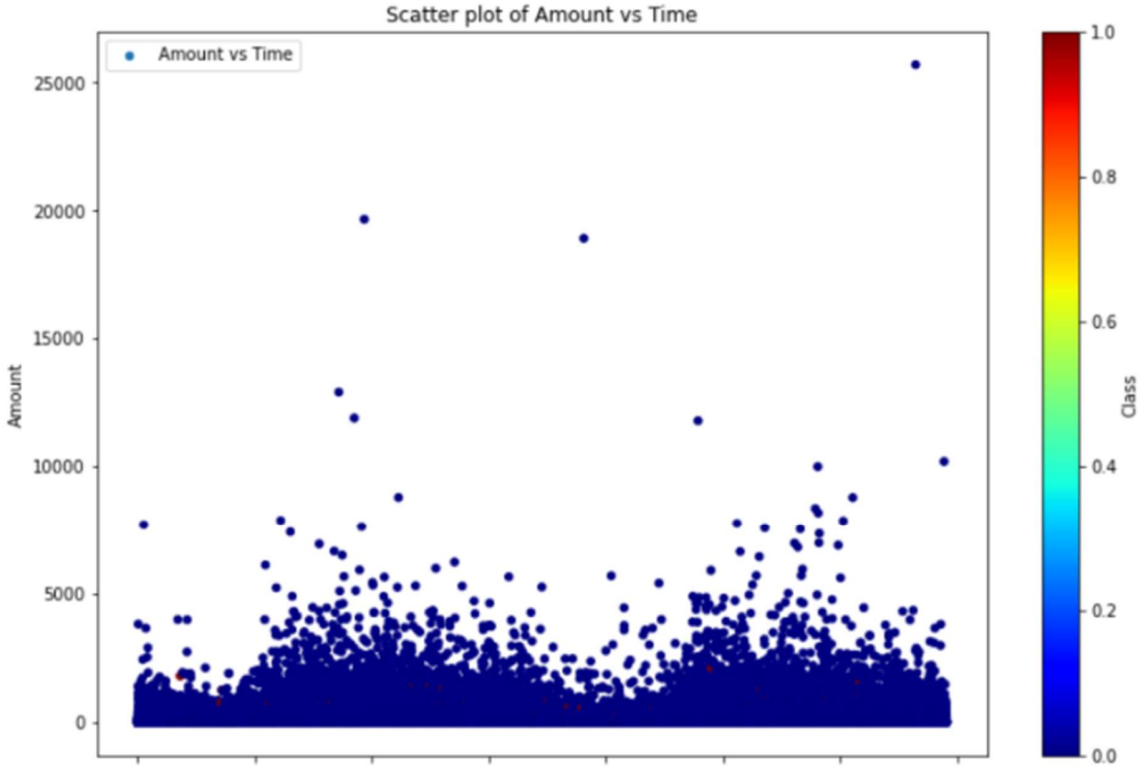
Note. "0" represents genuine transactions and "1" represents fraud transactions.

Figure 16 shows the number and percentage of fraud and legit transactions in the dataset. Which is useful to understand the ratio between good and bad transactions and how randomly a fraud transaction is taking place.

Scatterplot in the figure 17, provides a relationship between the Amount and Time features, the blue color represents good transactions whereas red color represents a fraud transaction. In the next stage, data pre-processing, we have cleaned the dataset by removing nulls and changing the type of data, this step was majorly focused on attaining the quality of the data, after preprocessing stage dataset has zero empty values in all the 31 fields, and we ensured that no field has mismatched values.

Figure 17

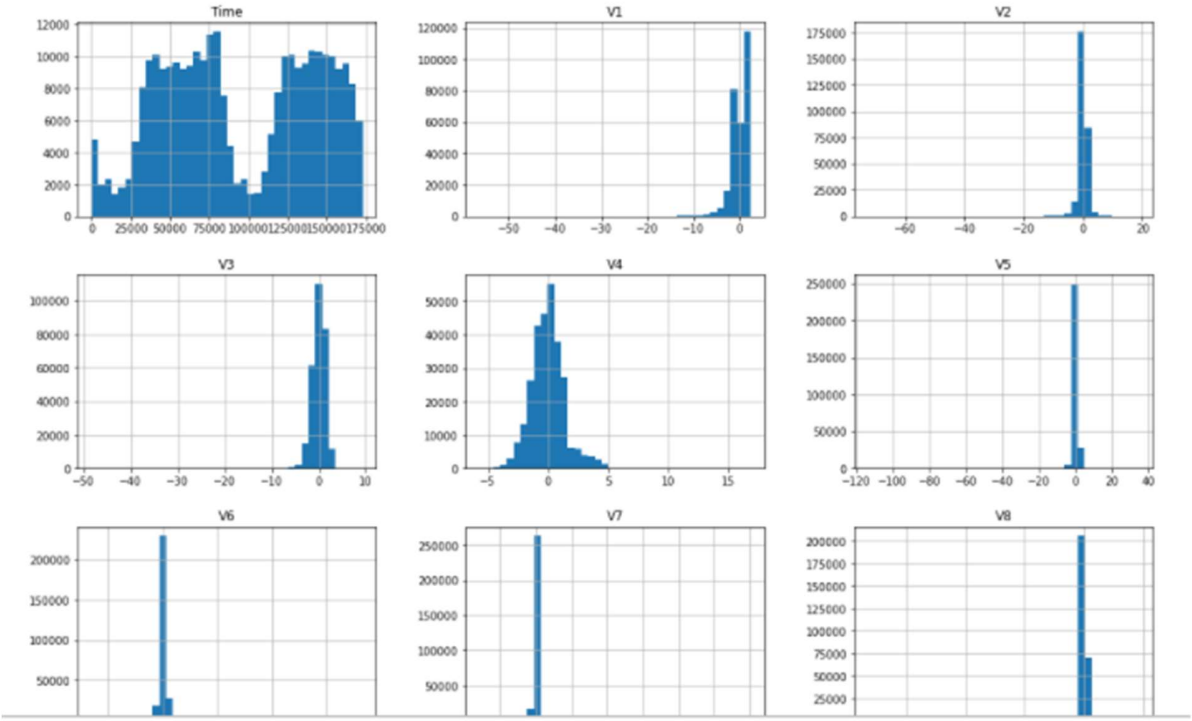
Scatterplot between Amount and Time



Note. Red color is used for fraud transactions and blue color is used for legit transactions.

Figure 18

Histograms of Some of the Features of the Dataset

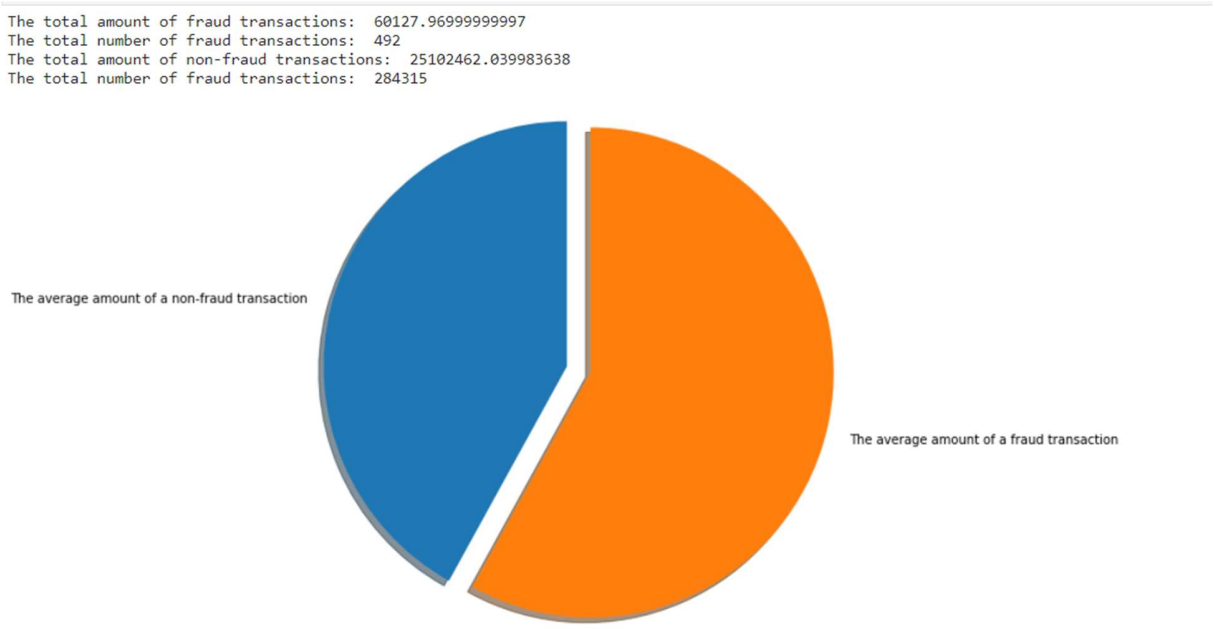


Note. Histogram provides details about some of the features of the dataset.

Figure 18 is the histograms of the features like Time, and from V1 to V8. Which is drawn using the python code on a Jupyter notebook. It helps for the visual analysis of the dataset by providing the maximum data lies between two points. It clearly indicates that most of the values of all the features from V1 to V8 lie around zero.

Figure 19

Pie Chart for Amount of Fraud and Non-fraud Transactions



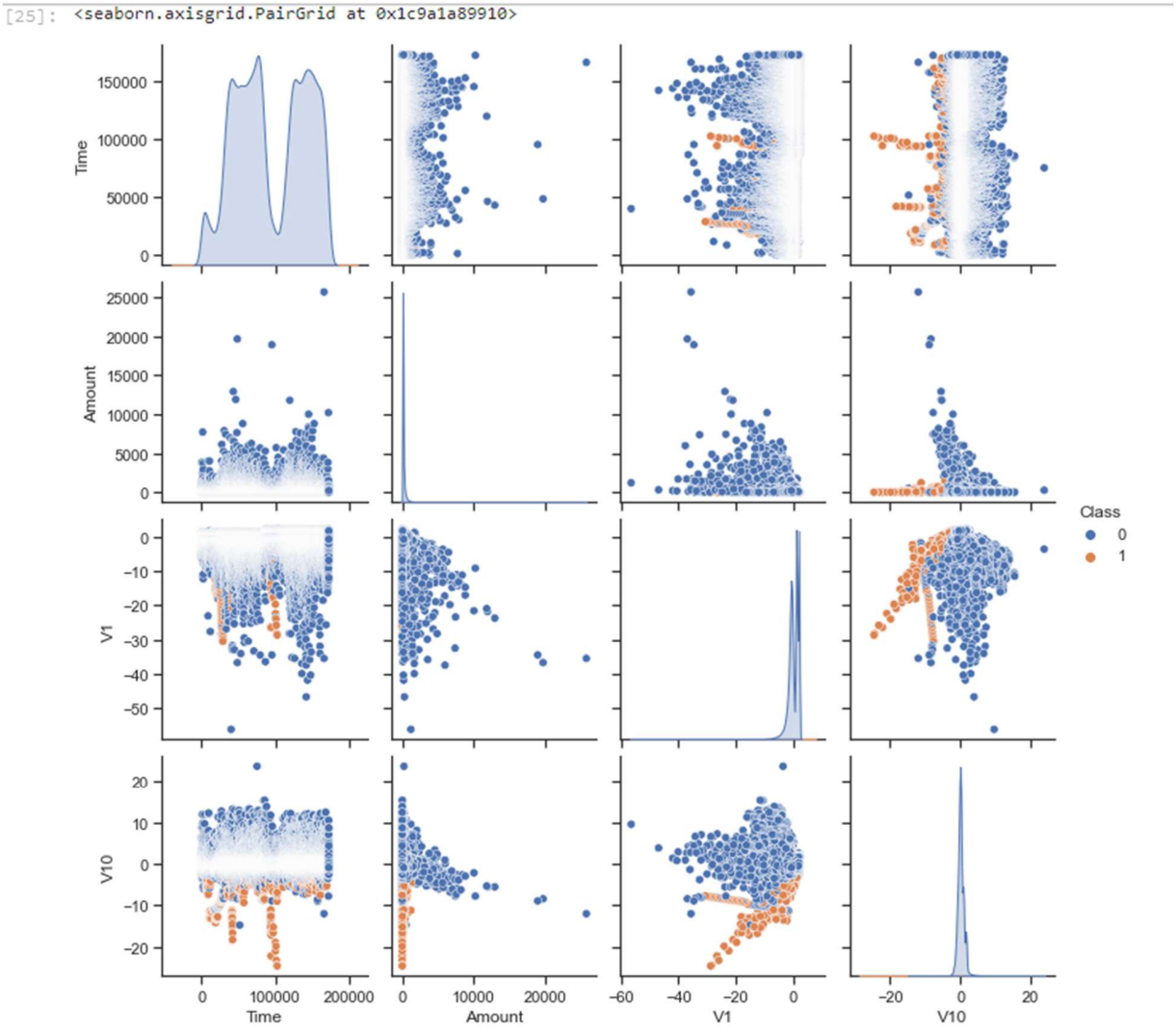
Note. Orange color shows average amount of a fraud transaction, blue color shows the average amount of a non fraud transaction.

Figure 19 is a pie chart drawn to analyze the amount of genuine and fraud transactions in the dataset. It clearly shows that the average amount of fraud transactions is much more than the average amount of the non fraud transactions even though the number of fraud transactions are far less than the non fraud transactions.

The third stage is data transformation, In this stage, we have formatted all the fields of the dataset, checked for duplicate data, dimensionality reduction is done by minimizing the features by replacing them with PCA transformation, checking for correlation between the feature of the dataset to tackle with over fitting models with the help of correlation matrix and scatterplot matrix. In the Fourth stage, data preparation, we have divided 20% of data for testing the model, and the rest of data is for validating the model, and training the model.

Figure 20

Scatter Plot Matrix between the Features

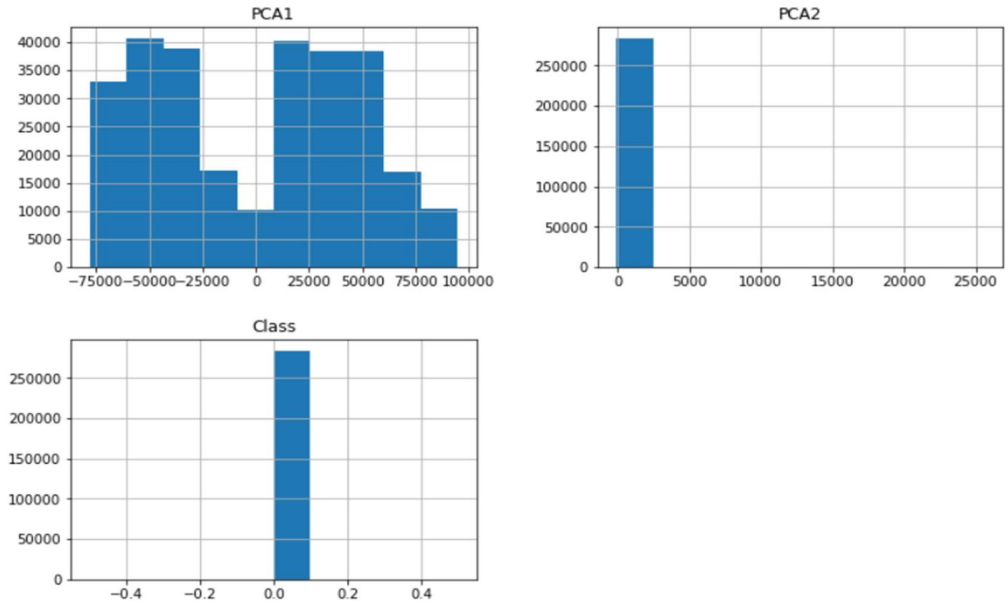


Note. scatter plot matrix drawn to check the correlation between the selected features

Figure 20, shows the strong correlation between the features of the dataset which is drawn using the Jupyter notebook. The scatter plot matrix in figure 20, represents the relationship between the Time, Amount, V1, and V10 features of the dataset.

Figure 21

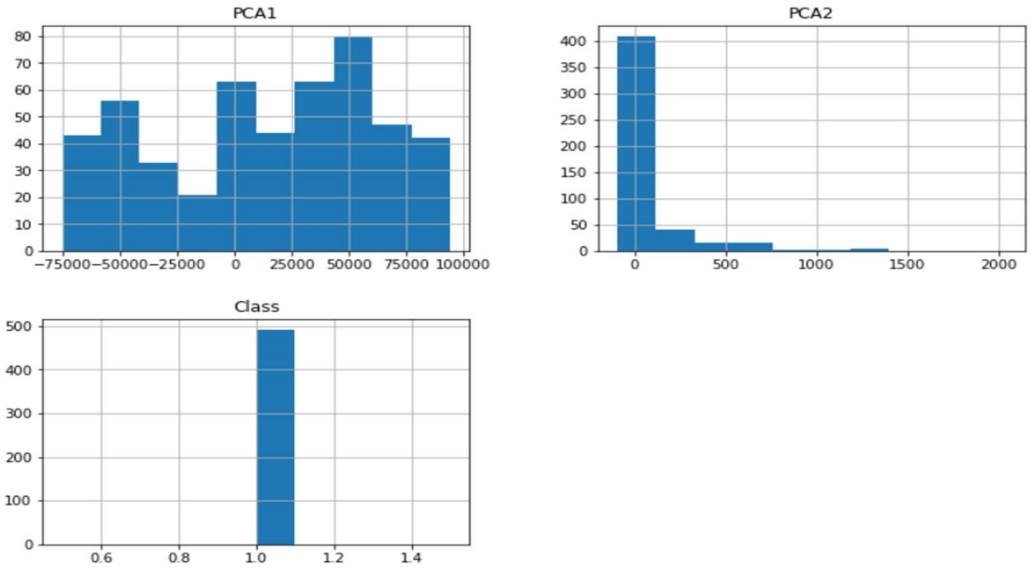
Histogram of Transformed Data for Class Zero Variables



Note. Class zero shows genuine transactions

Figure 22

Histogram of Transformed Data for Class One Variables



Note. Class One shows fraudulent transactions

Figure 21 and figure 22 are the histograms of the transformed dataset, between the PCA transformed features PCA1, PCA2, and the variable Class. It provides the range of values each feature has. Both the figures 21 and 22 show the values of PCA1 are equally spread on the negative and positive side of the zero, PCA2 are slightly more spread towards positive values, whereas the Class histogram represents quantity and type of values whether zero or one. All of the above-provided visualizations we have done under cleaning, pre-processing, transformation, and preparation are to better understanding of the dataset. Those are helpful in feature selection, and will be helpful in an accurate model building by removing the outliers.

Model Development

4.1 Model Proposals

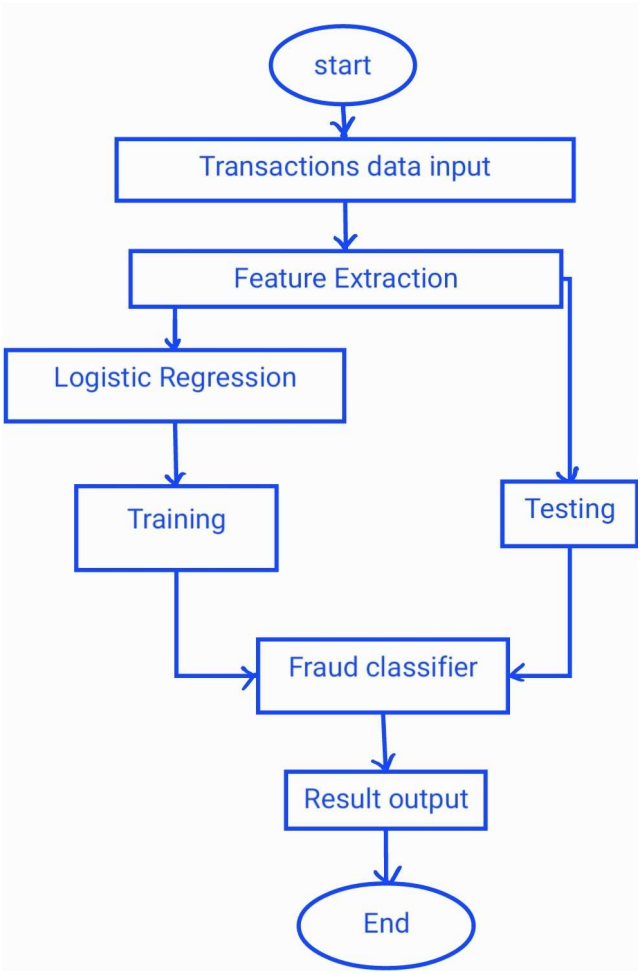
Machine Learning provides a great solution in detecting anomalies in data. Machine learning algorithms learn from the historical data and are used to predict the current anomalies in the data. Building machine learning models by training them using the already recorded fraudulent credit card transactions dataset will help in detecting the current frauds in the transactions. A large amount of transactional data can be used and divided for training and testing the models. In this project, we am using machine learning algorithms like Logistic Regression, Support Vector Machine (SVM), Artificial Neural Network(ANN), Decision Tree, and Random Forest. This section will focus on building, training, testing, and validating these models. For that, we have been using a dataset consisting of 284,807 recorded transactions and 31 variables. Out of these, the Time variable is not used because it does not hold any importance in determining the class of transaction. In the remaining 30 variables, Class is a dependent variable, and the rest are the input variables for the model. All the models have used the same features and the targeted output variable is also the same for all the models. These models are trained using the training dataset and are able to classify a transaction whether it is a fraudulent or genuine one that is the targeted problem for these models. The proposed models and their algorithms are provided in this section.

The first model used for this project is Logistic regression. It is one of the statistical analysis methods which is useful in predicting binary classes. It is also one of the effective models in credit card fraud detection. We considered this model because of its efficiency in classification problems. It considers a linear relationship among its variables. The better the relationship among the variables gives the better model. According to Ramesh (2017), It helps to

guess the chances of an event happening by adjusting the data into a log function. Jain et al. (2019) explained that this model will be helpful while a transaction is actually taking place in real-time by analyzing the variables of the transaction and determining whether that transaction should occur or not. The flow chart of the Logistic regression is provided in Figure 23.

Figure 23

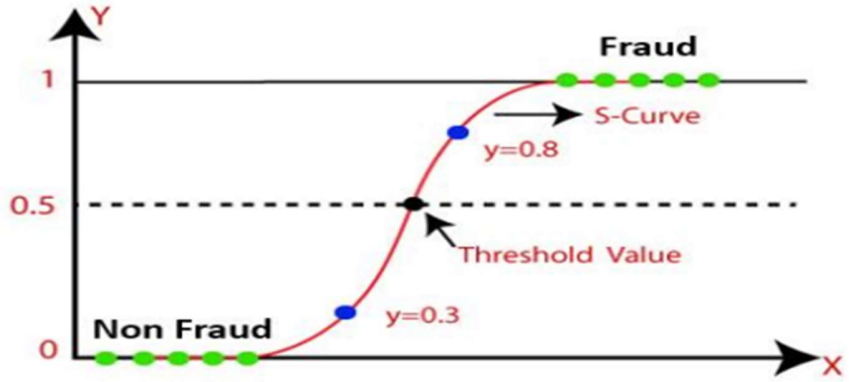
Flow chart of Logistic Regression



Note: Square boxes represents the different important stages of the model

Figure 24

Logistic Regression Classification



Note: Zero represents a genuine transaction and one represents a fraud transaction adapted from "Fraud detection in credit cards using logistic regression" by Alenzi & O, 2020, *International Journal of Advanced Computer Science and Applications*, *11*(12) (https://doi.org/10.14569/ijacsa.2020.0111265). Copyright 2021 by International Journal of Advanced Computer Science and Applications.

Figure 25

Logistic Function

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

Note: Logistic function also called sigmoid function adapted from *Python logistic regression with Sklearn & Scikit* by Navlani, 2019, DataCamp Community, (https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python), Copyright 2019 by DataCamp Community.

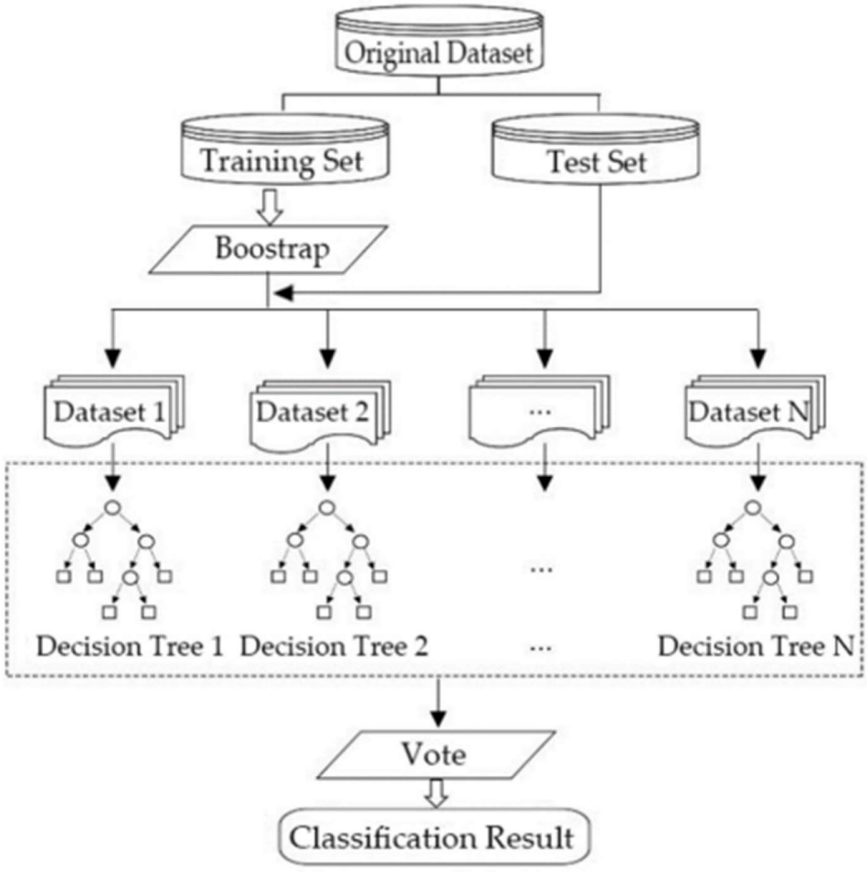
Logistic function is also called as a sigmoid function, it is calculated using the formula

provided in the figure 25. Basically. It provides an ‘S’ shape curve that corresponds to any given real values and can be transformed to values in the range of zero and one. If the curve turns towards the positive infinity, predict y will become a fraud transaction and if the curve turns towards negative infinity side, predict y will be determined as a genuine transaction. If the output of the logistic function is greater than 0.5 then we can determine such transactions as “1” or fraud, and whenever it goes lower than 0.5 we can determine such transactions as “0” or genuine (Navlani, 2019). Figure 24 is the visual representation of the logistic regression classification. In which the “1” represents the fraud transaction and “0” represents a genuine transaction. The “0.5” is a threshold value that is used to divide the fraud class and non fraud class (Alenzi & O, 2020). From the figure 24, whenever a transaction goes above the 0.8 limit the logistic function predicts a transaction as a fraud transaction and if a transaction goes below 0.3 limit then the logistic function predicts it as legit transaction.

The second model we used is Random forest. It is also used for fraud detection. It is also called a random decision forest. To solve classification and regression problems Random Forest technique is useful. That is why it is one of our choices. It is a summation of many decision trees and in which every individual tree makes a class prediction (Sadineni, 2020). For this model the best part is selecting features is not required, It can be trained very easily and quickly. All the variables except the Time variable are used as features for this model. The targeted variable is Class.

Figure 26

Algorithm of the Random Forest



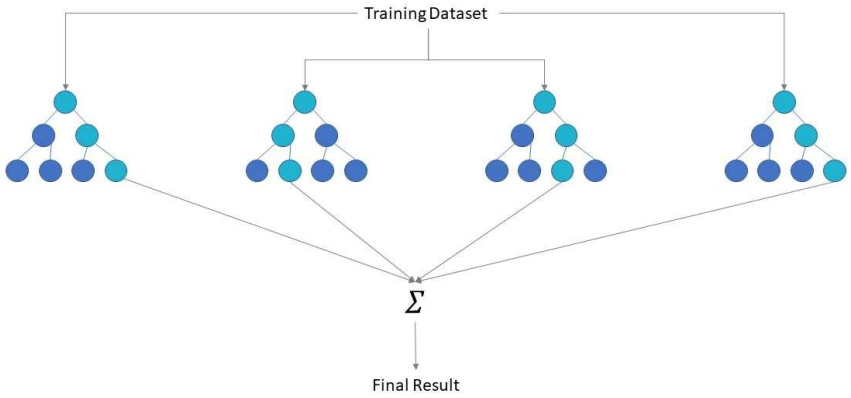
Note: adapted from "Pre-evacuation time estimation based emergency evacuation simulation in urban residential communities" by Chen et al., 2019, *International Journal of Environmental Research and Public Health*, *16*(23), 4599, (https://doi.org/10.3390/ijerph16234599). Copyright 2019 by International Journal of Environmental Research and Public Health.

Figure 26 is the Architecture of the Random Forest algorithm, as shown in that, primarily the dataset will be divided. A portion of dataset for training and remaining portion is for testing purpose, the training set data will be transferred to the each decision trees like each transaction of

the dataset will be goes under a decision tree and there the class of the transaction is predicted, like this there will be ‘N’ number of transactions goes under ‘N’ number of decision tree and class of each transaction is predicted. The average of all the results of the decision tree will be the final result of the random forest (Chen et al., 2019).

Figure 27

Diagram of Random Forest Classifier



Note: Adapted from *What is Random Forest?,* by IBM Cloud Education. (n.d.), 2021. (https://www.ibm.com/cloud/learn/random-forest). Copyright 2021 by IBM Cloud Education.

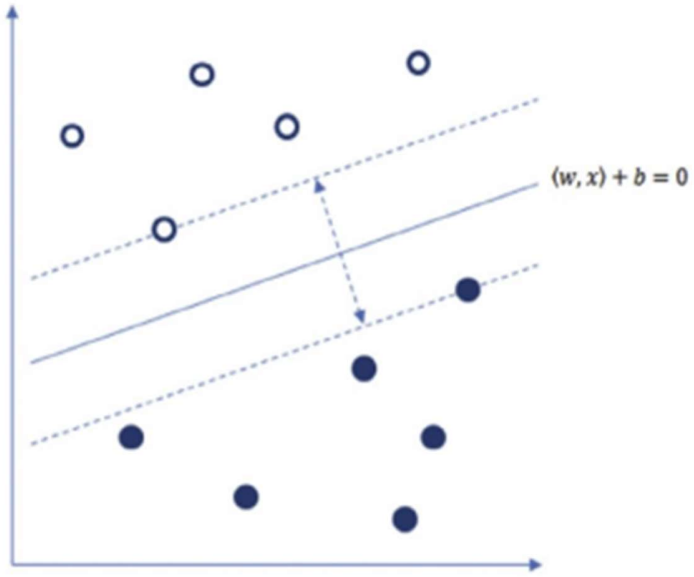
Figure 27 is the typical architecture of the random forest it consisting of a number of decision trees. The credit card transactions training data feed into the model, each transaction passed through the node and then each node divides into more trees so each transaction is analyzed by many trees. In which every tree predicts the class of the transaction. At last, a class with the highest votes is taken for prediction. This method is used because of its simplicity and it is used in the selection of features.

The Third method used in this project is the Support Vector Machine (SVM), which is one of the powerful supervised classification algorithms commonly used as a fraud detection model. In this technique, the data points of each transaction are transformed into vectors in high

dimensional space and hyperplanes are drawn to divide the space into various classes with separate behavioral characteristics (Rtayli & Enneya, 2020, p. 944). This method provides a solution to complex non-linear problems like credit card fraud detection using linear classification by leveraging kernel functions. (Popat & Chaudhary, 2018, p. 1123) Depending on the dataset and classification goal we can apply kernels like Gaussian radial basis function, polynomial function. The separation of classes by the SVM model is shown in figure 28.

Figure 28

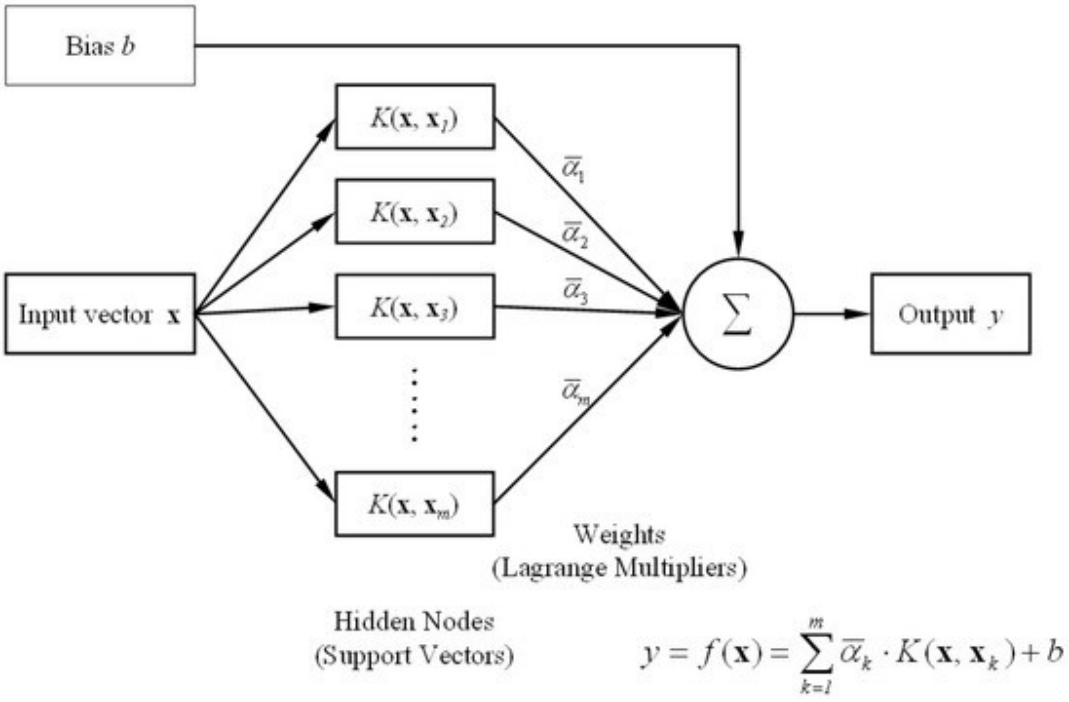
The separation between Support Vector and Hyperplane



Note: Adapted from "A survey on credit card fraud detection using machine learning", by Popat & Chaudhary, 2018, *International Conference on Trends in Electronics and Informatics,* p. 1123. (https://doi.org/10.1109/icoei.2018.8553963). Copyright 2021 by International Conference on Trends in Electronics and Informatics.

Figure 29

Architecture of support vector machine



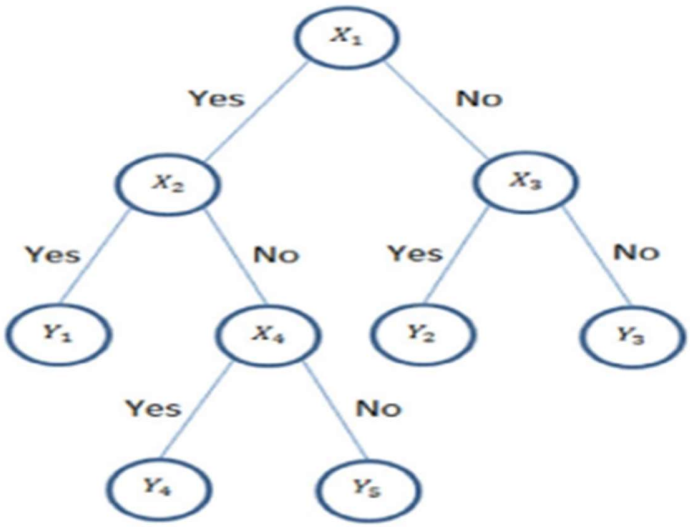
Note: Architecture Support vector machine algorithm adapted from "A review on hybrid empirical mode decomposition models for wind speed and wind power prediction" by Bokde et al., 2019, *Energies*, *12*(2), 254, (https://doi.org/10.3390/en12020254). Copyright 2019 by Energies.

Figure 29 is the architecture diagram of the SVM. Majorly there are three layers in this architecture : Input layer, hidden layer also known as support vectors, and output output layer. Hidden node layer is the most important layer among all. It consists of support vectors K(x, x1) where (x) is the input signal vector and (x1) is the support vector. The output (y) of the support vector machine can be calculated using the formula from figure 29 which is the summation of the outputs of all the hidden layer support vectors and a bias (b) obtained during the training phase (Bokde et al., 2019).

Another model used in this project is a Decision tree. It is used for the classification and prediction of data. In this technique, the entire transactional data which is referred to as the root node is initially split into two or more homogenous nodes (Popat & Chaudhary, 2018). Primarily each root node analyzes a transaction it then divides into sub-nodes. We then compare the value of the root attributes of the transaction with the record's attribute transaction and classify them accordingly. Later these nodes are further split into sub-nodes based on the decision using the above logic. Eventually, we arrive at a stage where further splitting is not possible and this node is called a terminal node. The algorithm of the decision tree is provided in Figure 30.

Figure 30

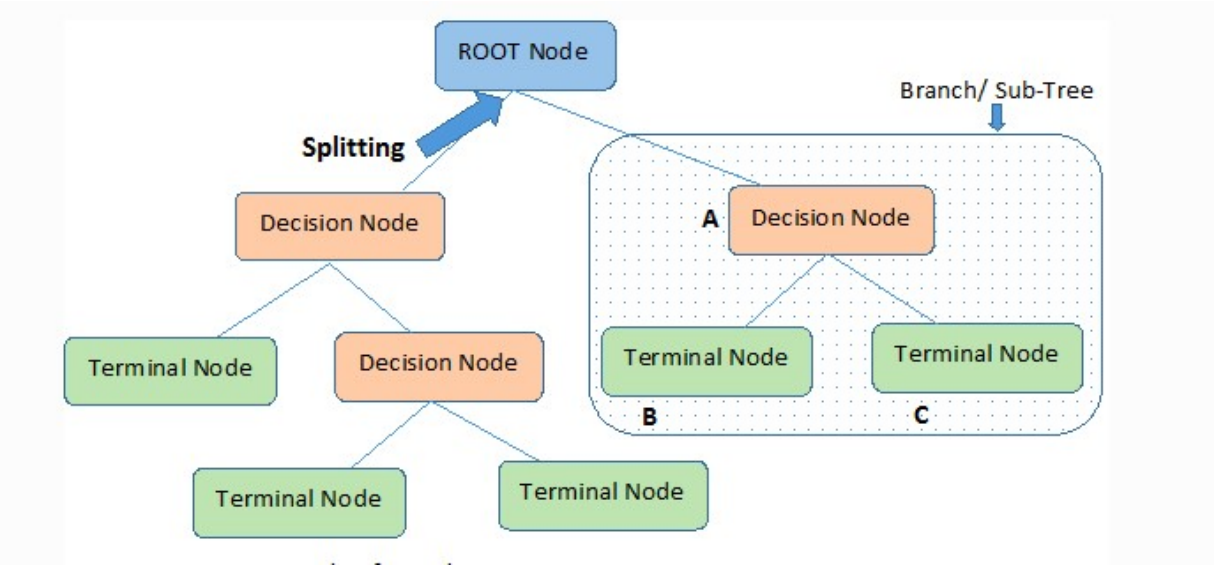
Binary Decision Tree



Note: Adapted from "A survey on credit card fraud detection using machine learning", by Popat & Chaudhary, 2018, *International Conference on Trends in Electronics and Informatics,* p. 1123. (https://doi.org/10.1109/icoei.2018.8553963). Copyright 2021 by International Conference on Trends in Electronics and Informatics.

Figure 31

Decision tree algorithm



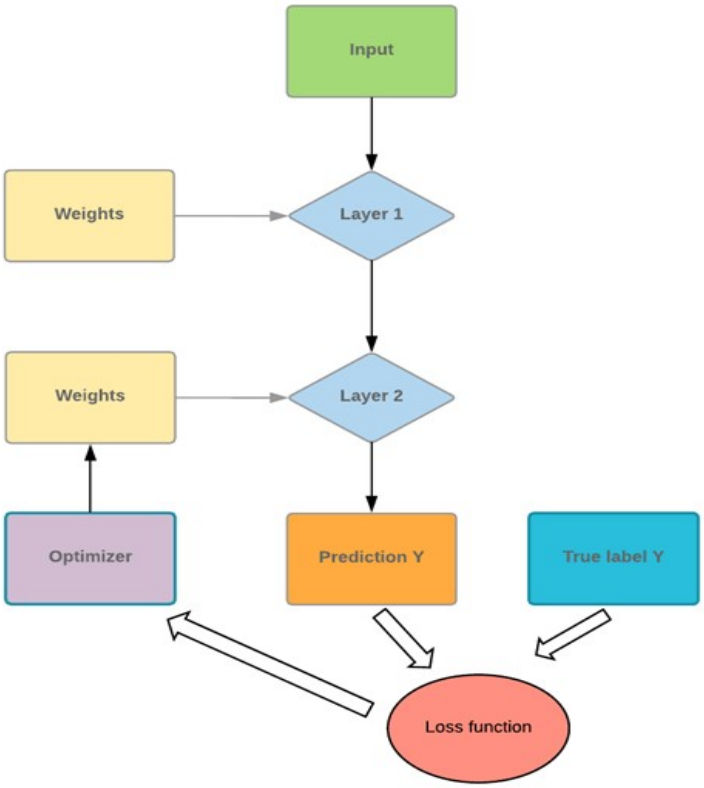
Note: A is parent node of terminal node B and C, adapted from *Decision tree algorithm, explained,* by Chauhan, 2020, KDnuggets, (https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html). Copyright 2020 by KDnuggets.

Figure 31 represents the algorithm and working procedure of a Decision tree algorithm, the Credit card transactions dataset will be inputted at the Root Node, it will be further splitted into two further Decision Nodes where a transaction of the dataset will be analyzed if it is a fraud or genuine one, like that the analysis will go until no further splitting is possible and it is the out of the final classification of the algorithm (Chauhan, 2020).

Artificial Neural Network (ANN) is also used for this project. ANN is one of the accurate models that have been used for fraud and anomalies detection. It stimulates the set of complex neurons which pretend and make decisions like a human brain so the computer starts learning things and makes decisions like a human (Sadineni, 2020, p. 661).

Figure 32

Algorithm for ANN

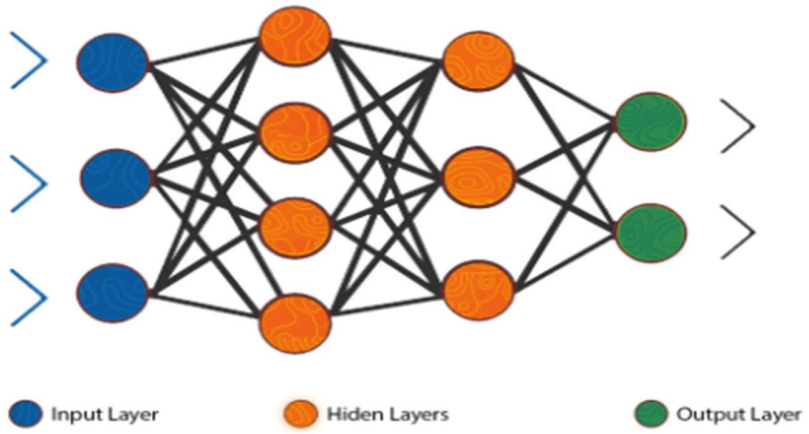


Note: Schematic flow algorithm of ANN model adapted from *Artificial Neural Network tutorial with tensorflow ann examples,* by Johnson, 2021, (https://www.guru99.com/artificial-neural-network-tutorial.html). Copyright 2021 by Guru99.

Figure 32 represents the algorithm of the ANN model, Layer 1 and Layer 2 in the rhombus shaped boxes are the multiple layers of the ANN, basically there are three layers, which are input, output and the hidden layer, all the analysis is done in these three layers whenever a transactional data provided through the inputs layer. The performance of the learning phase can be calculated using the loss function. An optimizer is used to improve the performance of the model (Johnson, 2021).

Figure 33

Artificial Neural Network



Note: ANN model's architecture consists of different layers, Adapted from *Ann vs CNN vs RNN: Types of neural networks*, by Pai, 2020, Analytics Vidhya, (https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/). Copyright 2020 by Analytics Vidhya.

An Artificial Neural Network (ANN) is one of the effective models for transactional data. That is why it is used for this project. In this model, the transactional data ingested from the input layer then it transfers each transaction to the next layer which is under the hidden layer, which also bridges to each node of the first layer in the hidden layer, and each node in that layer bridges with the nodes of the input layer and also all the nodes in the next layer of the hidden layer. Like that there can be multiple hidden layers connected to each other and the last hidden layer bridges with the output layer (Pai, 2020).

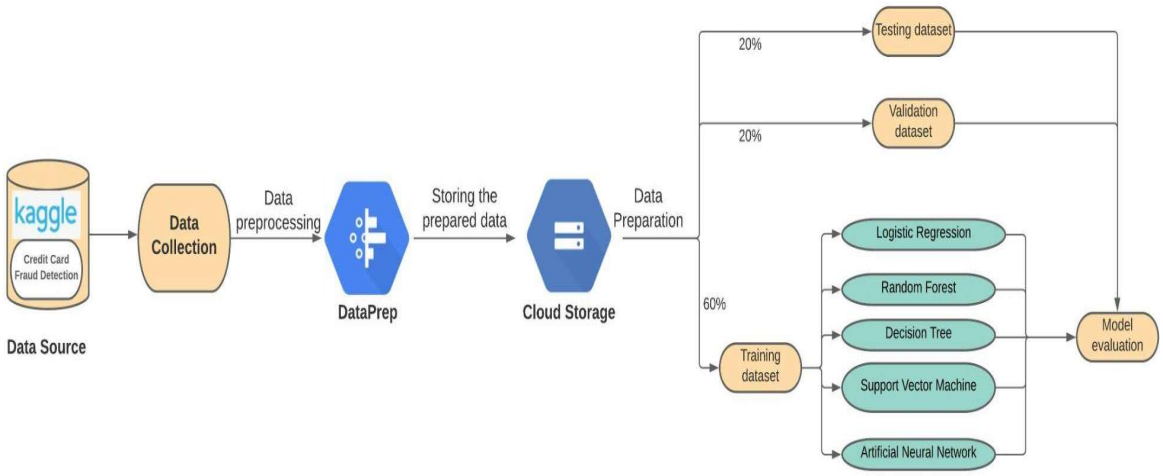
4.2 Model Supports

After choosing the models we have to consider the required resources which supports for building each model, and also design an architecture for data flow and model development. So this section is majorly focused on the tools, platforms, software, and environments required to

develop each selected model and architecture of the models. Figure 34, shows the dataflow architecture and the process of the credit card fraud detection model. It starts with the data collection from the Kaggle and ends with the model evaluation. At the first stage, the data is collected from the kaggle then it was cleaned and preprocessed using the Dataprep, after that the cleaned and preprocessed data is securely stored in the cloud storage, after that the stored data is used for preparation for model building by splitting the data into 60:20:20 for training, testing, and validation. The training dataset is used to train the selected models, the testing and validation datasets are used to test and validate these models' performance in terms of accuracy and precision.

Figure 34

The dataflow architecture of credit card fraud detection models



Note: Tensorflow, Keras, Scikit-learn, and other machine learning frameworks and software used at model part.

Table 2, provides the information about the required environment, software, and platform required to build, and operate each proposed model.

Table 2

Machine learning models used and their required tools and software

Models	Platform	Software	Environment
Logistic Regression	Google Cloud Platform	Vertex AI, Tensorflow, Keras, Scikit-learn, Python 3.9	Anaconda, Google DataProc, pyspark, Gcloud CLI
Random Forest	Google Cloud Platform	Vertex AI, Tensorflow, Keras, Scikit-learn, Python 3.9	Anaconda, Google DataProc, pyspark, Gcloud CLI
Decision Tree	Google Cloud Platform	Vertex AI, Tensorflow, Keras, Scikit-learn, Python 3.9	Anaconda, Google DataProc, pyspark, Gcloud CLI
Support Vector Machine (SVM)	Google Cloud Platform	Vertex AI, Tensorflow, Keras, Scikit-learn, Python 3.9	Anaconda, Google DataProc, pyspark, Gcloud CLI
Artificial Neural Network (ANN)	Google Cloud Platform	Vertex AI, Tensorflow, Keras, Scikit-learn, Python 3.9	Anaconda, Google DataProc, pyspark, Gcloud CLI

Note: Google Cloud Platform provides all cloud computing services of Google. Gcloud CLI is a Gcloud command-line tool.

4.3 Model Comparison and Justification

This section majorly focuses on the reason for choosing each model, the strengths of each model backed to selecting a particular model, like implementation, flexibility and the accuracy of each model have been discussed. Tables 3, 4, and 5 provide information about the strengths, disadvantages, and reasons for choosing each model.

Table 3

Model Comparison and Justification of Logistic Regression and Random Forest

Technique	Advantages	Disadvantages	Reason for Selection
Logistic Regression	This technique is easy to understand, explain, implement, and train the model.	This method has shown less predictive scores with the used credit card dataset. This model has different response and calculator times for different features of the dataset.	The dataset used for this project is highly imbalanced, it is suitable for transactional data. It has given high accuracy compared to other models.
Random Forest	It accurately predicts the classifications. It has given high accuracy and precision.	It requires a lot of time to decide a transaction's class as it is a composition of many sub decision trees so it is a complex process to process each transaction. Transactional data usually consists of billions of records so it needs a lot of computational power and time.	Feature selection is easy, suitable for transactional data, easy to understand, Given high accuracy compared to other models.

Note: The provided advantages and disadvantages of the model are with respect to credit card frauds detection.

Table 4

Model Comparison and Justification of Decision Tree and Support Vector Machine (SVM)

Technique	Advantages	Disadvantages	Reason for Selection
Decision Tree	This model is one of the easiest models to understand and is easily explainable. It is adaptable for classification problems. It can work on linear and non-linear datasets and is very flexible.	The Complex algorithm of this model makes it complex. The very small amount of change in data can alter the whole structure of its tree algorithm. Each credit card transaction is checked one by one so it takes time and splitting criteria for data is also complex.	It is one of the effective methods in making the decision. Easiest and most promising model among all the models.
Support Vector Machine (SVM)	This method is useful for detecting non-linear classification-related problems. This technique helps in detecting a transaction while it's actually taking place.	It is complex to understand and explain. Sometimes it fails in detecting even after a transaction takes place. It has low precision and a high false-positive rate.	In this project we are working on a highly imbalanced dataset it is one of the recommended models for the imbalanced model.

Note: The provided advantages and disadvantages of the model are with respect to credit card frauds detection.

Table 5

Model Comparison and Justification of Artificial Neural Network

Technique	Advantages	Disadvantages	Reason for Selection
Artificial Neural Network (ANN)	High accuracy and precision which is commendable. This method is highly suitable for classification problem solving and highly recommended when working with imbalanced datasets.	This model needs very high computational power to execute. It is complex to understand and difficult in future selection. This technique can not detect transactions while the transaction is going on.	This is the highest accurate model among all the models we have considered at the beginning.

Note: The provided advantages and disadvantages of the model are with respect to credit card frauds detection.

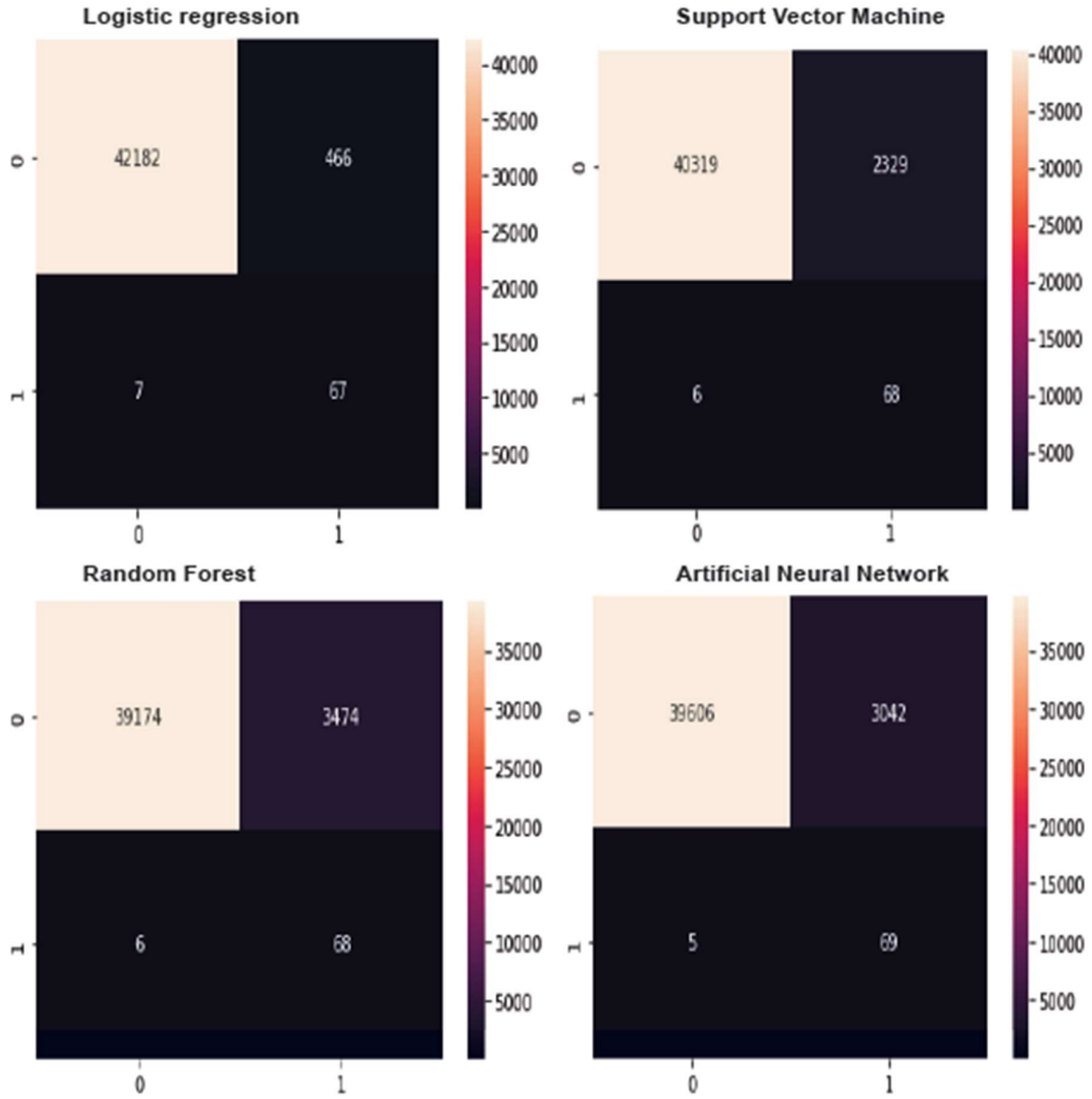
4.4 Model Evaluation Methods

The final stage of the model building is Model Evaluation, In this section, we will provide the different methods and metrics used to evaluate and test the model's performance. This comes after training the model. Evaluating the model using the training and the validation datasets is done in this section. The training dataset is 20 percent of the whole dataset, and for validation, we have taken 20 percent of the whole dataset. These two datasets were used to test and validate the proposed models using the confusion matrix, Receiver Operating Characteristics curve (ROC). The evaluation metrics used for this project are accuracy, precision, and False-

positive rate. A confusion matrix is drawn for each model and presented in Figure 35. After training the models with the prepared dataset, now evaluate their results using the ROC. The confusion matrix is provided in Figure 35 for each model proposed and trained. True Negatives are presented on the Top Left square box, those are the number of correct classifications of Non Fraudulent Detected classes. False Negatives are presented on the Top Right Square box, those are the number of incorrect classifications of the Non Fraudulent Detected classes. False Positives are presented on the Bottom Left Square box, those are the number of incorrect classifications of the Fraudulent Detected classes. Whereas, The True Positives are presented on the Bottom Right Square box those are the number of correct classifications of the Fraudulent Detected classes.

Figure 35

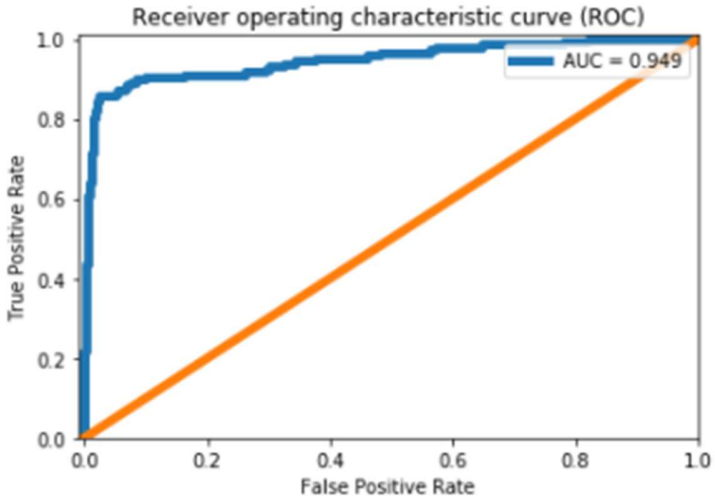
Confusion matrices of proposed models



Note: Confusion matrix used to calculate the different evaluation metrics

Figure 36

ROC curve



Note: ROC curve is drawn between the true positive and false positive rates

Figure 36, is the ROC curve plotted between the false positive rate and the true positive rate on a different set of values it shows Area Under the Curve (AUC) used to differentiate among the classes. The blue line should be at the top left corner as close as possible, here because of the highly imbalanced dataset that gap is more. The AUC reflects the model's ability while forecasting a higher score for the purpose of favourable examples in contrast to unfavourable examples. We can use the AUC metric to gain a sense of your model's prediction accuracy without setting a threshold because it has nothing to do with the cut-off score (Ramesh, R., 2017).

The evaluation metrics used to calculate the performance of the model are Accuracy, Precision, and ROC. Accuracy is the most efficient way of calculating the performance of the model, It is the proportion between the correctly identified transactions to the total number of identified transactions (Srivastava, 2020).

Figure 37

Accuracy of models

Confusion Matrix		Target			
		Positive	Negative		
Model	Positive	a	b	*Positive Predictive Value*	a/(a+b)
	Negative	c	d	*Negative Predictive Value*	d/(c+d)
		Sensitivity	*Specificity*	**Accuracy** = (a+d)/(a+b+c+d)	
		a/(a+c)	d/(b+d)		

Note: Adapted from *Evaluation metrics machine learning,* by Srivastava, 2020, Analytics Vidhya, (https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/). Copyright 2020 by Analytics Vidhya.

The second metric used to test and evaluate the proposed models is precision. It is the ratio of correctly detected positive fraud transactions to the summation of correctly detected positive fraud transactions and wrongly detected positive fraud transactions. It is used to calculate the effectiveness of the model (Shung, 2020).

Figure 38

Precision Formula

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Note: Adapted from *Accuracy, precision, recall or F1?, by* Shung, 2020, Medium, (https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9). Copyright 2020 by Medium.

Figure 38 provides the formula to calculate the precision of the models. The precision score is calculated for all proposed models using this method. The values to calculate precision for models are taken from the confusion matrix in figure 35.

The next evaluation metric used in this project is Receiver Operator Characteristic (ROC). ROC curve is a binary classification matrix. It evaluates it's issue metric. This is a probability curve that represents the values by plotting the True Positive Rate (TPR) versue the False Positive Rate (FPR) at multiple permissible levels, so it does it by separating the signal from the noise (Bhandari, 2020). So the values of TPR and FPR are obtained using the following formulas provided in figures 39 and figure 40 respectively. Using these values the ROC curve has been plotted in figure 35 the confusion matrix.

Figure 39

True Positive Rate or sensitivity

$$Sensitivity = \frac{TP}{TP + FN}$$

Note: True positive rate also called as sensitivity, Adopted from *AUC-Roc Curve in machine learning clearly explained, by* Bhandari, 2020, Analytics Vidhya, (https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/). Copyright 2020 by Analytics Vidhya 2020.

Figure 40

False Positive Rate (FPR)

$$FPR = \frac{FP}{TN + FP}$$

Note: Adopted from *AUC-Roc Curve in machine learning clearly explained, by* Bhandari, 2020, Analytics Vidhya, (https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/). Copyright 2020 by Analytics Vidhya 2020.

4.5 Model Validation and Evaluation

This section focuses and compares the results obtained from the different metrics used for model evaluation and comparison of overall results of all the models.

Table 6

Accuracy, Precision, and False-positive rate scores

Machine Learning Model	Accuracy	Precision	False-positive rate
Logistic Regression	95.75%	86.56%	2.7%
Random Forest	99.3%	98.78%	1.2%
Decision Tree	98.1%	98.65%	2.2%
Support Vector Machine (SVM)	96.25%	88.55%	4.4%
Artificial Neural Network (ANN)	99.8%	99.76%	0.11%

Note: all the values have been calculated using the training dataset

Table 6 provides information about the metrics used to evaluate and test the models and their scores. Logistic regression has given the accuracy of 95.75% with 86.56% of precision and a 2.7% of the false-positive rate. Though the accuracy reached more than 95% the false positive rate is high and precision is also low as compared to other models. The random forest has given the second-highest accuracy of 99.3% among all the models we have used with 98.78% of precision and a 1.2% of false-positive rate. With high accuracy and precision, this model promises the best results to use for detecting fraudulent credit card transactions. The decision tree model has given the accuracy of 98.1% with 98.65% of precision and a 2.2% of the false-

positive rate. It has given moderate results with a high false-positive rate so this model is limited to use for credit card fraud detection. The Support Vector Machine has given an accuracy of 96.25% with 88.55% of precision and a 4.4% of the false-positive rate. The accuracy of this model reached more than 96% but the false positive rate is high and precision is also low as compared to other models. ANN has given the highest accuracy of 99.8% among all the models we have used with 99.76% of precision and a 0.11% of false-positive rate. With the highest accuracy and precision, this model promises the best solution for detecting fraudulent credit card transactions.

There are many ways to detect fraud transactions, but Machine Learning is most sought among all because a model can be easily built and re-model. And it is proved in this project. All of the models that we have used gave good results. Among those, ANN model has given the highest fraud detection rate. ANN model is recommended to use for credit card fraud detection. Which is expected to help banks, financial institutions limit losses and resources putting in these frauds. Also helpful to merchants, and customers by limiting fraud and its detection time so the banks can recover their money easily.

References

Alenzi, H. Z., & O, N. (2020). Fraud detection in credit cards using logistic regression. *International Journal of Advanced Computer Science and Applications*, *11*(12). https://doi.org/10.14569/ijacsa.2020.0111265

Best, R. de. (2021, July 9). *Visa, MasterCard, UnionPay Transaction Volume 2020*. Statista. Retrieved October 23, 2021, from https://www.statista.com/statistics/261327/number-of-per-card-credit-card-transactions-worldwide-by-brand-as-of-2011/.

Bhandari, A. (2020, July 20). *AUC-Roc Curve in machine learning clearly explained*. Analytics Vidhya. Retrieved December 9, 2021, from https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/.

Bokde, N., Feijóo, A., Villanueva, D., & Kulat, K. (2019). A review on hybrid empirical mode decomposition models for wind speed and wind power prediction. *Energies*, *12*(2), 254. https://doi.org/10.3390/en12020254

BrainStation (Ed.). (2018, September 7). *Machine learning 101: Supervised, unsupervised, Reinforcement & Beyond*. Medium. Retrieved December 8, 2021, from https://towardsdatascience.com/machine-learning-101-supervised-unsupervised-reinforcement-beyond-f18e722069bc.

Chauhan, N. S. (2020, January). *Decision tree algorithm, explained*. KDnuggets. Retrieved December 10, 2021, from https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html.

Chen, Yu, Wen, Zhang, Yin, Wu, & Yao. (2019). Pre-evacuation time estimation based emergency evacuation simulation in urban residential communities. *International Journal*

of Environmental Research and Public Health, *16*(23), 4599. https://doi.org/10.3390/ijerph16234599

Dornadula, V. N., & Geetha, S. (2019). Credit card fraud detection using machine learning algorithms. *Procedia Computer Science*, *165*, 631–641. https://doi.org/10.1016/j.procs.2020.01.057

IBM Cloud Education. (n.d.). *What is Random Forest?* IBM. Retrieved November 20, 2021, from https://www.ibm.com/cloud/learn/random-forest.

Jain, Y., Tiwari, N., Dubey, S., & Jain, S. (2019, January). *A comparative analysis of various credit card fraud detection techniques*. *2019 International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878,* Volume-7 Issue-5S2, *79,* 402-407. https://www.ijrte.org/download/volume-7-issue-5s2/

Johnson, D. (2021, October 7). *Artificial Neural Network tutorial with tensorflow ann examples*. Guru99. Retrieved December 10, 2021, from https://www.guru99.com/artificial-neural-network-tutorial.html.

Machine Learning Group - ULB (Université Libre de Bruxelles). (2018). *Credit card fraud detection* (Version 3) [Anonymized credit card transactions labeled as fraudulent or genuine] [Data set]. Kaggle. https://www.kaggle.com/mlg-ulb/creditcardfraud

Maniraj, S. P., Saini, A., Ahmed, S., & Sarkar, S. D. (2019, September). *(PDF) credit card fraud detection using machine learning and data science. 2019 International Journal of Engineering Research & Technology (IJERT) volume 08, issue 09.* http://dx.doi.org/10.17577/IJERTV8IS090031

Marr, B. (2018, September 24). *What are artificial neural networks - a simple explanation for absolutely anyone*. Forbes. Retrieved December 8, 2021, from

https://www.forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/?sh=254ed1631245.

Navlani, A. (2019, December 16). *Python logistic regression with Sklearn & Scikit*. DataCamp Community. Retrieved December 10, 2021, from https://www.datacamp.com/community/tutorials/understanding-logistic-regression-python.

Pai, A. (2020, October 19). *Ann vs CNN vs RNN: Types of neural networks*. Analytics Vidhya. Retrieved November 21, 2021, from https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/.

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)*. https://doi.org/10.1109/icoei.2018.8553963

Ramesh, R. (2017). Predictive analytics for banking user data using AWS machine learning cloud service. *2017 2nd International Conference on Computing and Communications Technologies (ICCCT)*. https://doi.org/10.1109/iccct2.2017.7972282

Rtayli, N., & Enneya, N. (2020). Selection features and support vector machine for credit card risk identification. *Procedia Manufacturing*, *46*, 941–948. https://doi.org/10.1016/j.promfg.2020.05.012

Sadgali, I., Sael, N., & Benabbou, F. (2019). Fraud detection in credit card transaction using machine learning techniques. *2019 1st International Conference on Smart Systems and Data Science (ICSSD)*. https://doi.org/10.1109/icssd47982.2019.9002674

Sadineni, P. K. (2020). Detection of fraudulent transactions in credit card using machine learning algorithms. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. https://doi.org/10.1109/i-smac49090.2020.9243545

Shung, K. P. (2020, April 10). *Accuracy, precision, recall or F1?* Medium. Retrieved November 20, 2021, from https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9.

Srivastava, T. (2020, August 5). *Evaluation metrics machine learning*. Analytics Vidhya. Retrieved November 20, 2021, from https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/.

Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S., & Kuruwitaarachchi, N. (2019). Real-time credit card fraud detection using machine learning. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. https://doi.org/10.1109/confluence.2019.8776942

Volpi, G. F. (2020, October 1). *The most gentle introduction to principal component analysis*. Medium. Retrieved December 8, 2021, from https://towardsdatascience.com/the-most-gentle-introduction-to-principal-component-analysis-9ffae371e93b.