Credit Card Fraud Detection and Clustering of Fraudulent Cases

Bhumin Shah
Vinodh Kumar Selvam
Shweta Laxmikant Bangad
Venkatesh Naidu Chandrasekaran

TEAM Name: Mavericks

EXECUTIVE SUMMARY

Fraud cases in 2000's was constant and in 2008 fraud cases were increased by 21%. In 2007, UK faced a loss of £535 million in fraudulent cases of credit cards. It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

Fraud is one of the major ethical issues in the credit card industry. The main aims are to identify the different kinds of fraud, and then to use some techniques to understand the behavior of the credit card fraud. The sub-aim is to present, compare and analyze recently published findings in credit card fraud detection. This article defines common terms in credit card fraud and highlights key statistics and figures in this field. Depending on the type of fraud faced by banks or credit card companies, various measures can be adopted and implemented. The proposals made in this paper are likely to have beneficial attributes in terms of cost savings and time efficiency. The significance of the application of the techniques reviewed here is in the minimization of credit card fraud. Yet there are still ethical issues when genuine credit card transactions made by customers are misclassified as fraudulent.

The objective of our work is to utilize the available data about credit card transactions from the people of Europe and predict how many frauds have occurred in that period of time, further we can also analyze more from the data about the patterns of how people use credit cards. We can also analyze the trends of when there is likely to be a fraud transaction. And accordingly, we can take actions to prevent that from happening.

Our dataset contains transactions made by credit cards in September 2013 by European cardholders. There are total of 284807 records in our dataset. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. We will be doing a lot of analysis on this data set and find some interesting facts from it.

Contents

EXECUTIVE SUMMARY 2
1. ANALYTICS PLAN 5
1.1 Organization Background & Opportunity 5
1.2 Research Questions 5
1.3 Hypotheses 5
1.4 Data Definition 6
1.5 Measurements 6
1.6 Methodology and Computation Methods 7
1.6 Output Summary 7
1.7 Project Implementation 8
2. PROJECT PLAN 8
2.1 Project Management Approach 8
2.2 Project Deliverables & Milestones 9
2.2.1 Statement of Work 9
2.3 Project Requirements 10
2.4 Project Constraints 11
2.5 Project Assumptions 11
2.6 Project Risks 11
2.7 Communication Management 12
2.7.1 Stakeholder Register 12
2.7.2 Stakeholder Analysis 12
2.7.3 Communication Matrix 13
2.8 Team Roles & Responsibilities 13
2.9 Work Breakdown Structure (WBS) 15
2.10 Activity-On-Node Diagram 15
2.11 Project Loading & Leveling 16
2.12 Schedule Baseline 16
3.DATA QUALITY 17
3.1 Missing Data Cleansing & Imputation 17
3.2 Data Transformation and Preparation 17
3.3 Data Pre-processing techniques 18

Synthetic Minority Oversampling Technique 18
Random Over Sampling Examples (ROSE) 19
3.4 Exploratory Data Analysis 20
3.5 Quantitative Variables (Continuous Predictors) 23
3.5 Principal Component Analysis: 23
3.6 PCA Computation and Visualization: 23
3.7 Model Based EDA 26
3.7.1 Linear Regression 26
3.7.2 Logistic Regression 29
3.8. Variable Reduction 30
3.9. High Value variable 35
4. CLUSTERING/SEGMENTATION APPROACH 36
5. MODELLING 39
5.1 Using model on Original data set: 39
5.2 Using Decision Tree (Entropy) 41
5.3 Using Decision Tree (Gini) 44
5.4 Logistic Regression 47
5.5 Comparing Model Accuracies: 48
6. IMPLEMENTATION: 49
6.1 Methodology Overview: 49
6.1.1 Business Understanding: 50
6.1.2 Data Understanding: 50
6.1.3 Data Preparation: 51
6.1.4: Modelling: 51
6.1.5 Evaluation: 51
6.1.6 Deployment: 51
7. CONCLUSION AND FUTURE STEPS: 52
7.1. Conclusions: 52
7.2. Potential Improvements or Future Work: 52

1. ANALYTICS PLAN

1.1 Organization Background & Opportunity

Credit cards were first conceptualized in 1887 when Edward Bellamy used the words 'Credit Card' as a way to pay for purchases been made. Basic concept of credit cards was to provide credit or loan to customer in exchange for an interest in return. Several pieces of information are essential to use a credit card: the cardholder's name, the card number, the card's expiration date, and a verification number or code. This information is displayed on the card and may also be encoded on a magnetic strip. Debit cards also often include a personal identification number (PIN), which the cardholder can use to withdraw cash.

With use of technology credit cards have evolved and have better security features. However, the ease of access given by these cards has also increased the chances of fraud. Credit card fraud involves the theft or misuse of this information for financial gain. Ever since then credit cards have evolved and so has the fraudulent use of credit cards. Credit cards used over internet, departmental stores, gas stations, ATM and other places all have faced fraud.

In our analytical plan we will discuss how data was collected, some descriptive analysis about data, steps taken for preprocessing, model building, evaluation and end result.

1.2 Research Questions

The primary objective of this project is to study as many data elements as possible to identify key indicators that add to the predictive strength of the model in predicting if a transaction is fraudulent or not. We can think of some questions which we would like our dataset to answer.

1. Which transactions tend to be fraudulent?
2. What is the amount range of the fraudulent transactions?
3. How many fraudulent transactions have gone through on a similar account?
4. Do the fraudulent transactions follow a similar pattern?
5. What are the decisive factors and the key variables that affect the fraud?
6. Behavior of different variables in the fraudulent transactions?
7. Are there any patterns in the fraudulent accounts?

1.3 Hypotheses

Hypothesis does not restrict to finding if a transactional was fraudulent or not however the scope can be further broadened to understand the type of transactions that are most culprit or could be a set of users that are naïve to technology.

Hence next time before any fraudulent transaction we can predict the chances of that transaction is fraudulent. This can restrict a transaction in future that assumes with confidence level of more than 90% and add an extra layer of verification.
With data building and methods of fraud changing we might have to modify and build a new model with new data and attributes that help best predict the transaction.
We can expect the data to give us some insights on the similarities in fraudulent cases. Using these similarities we can predict the likeliness of fraud on that account. Using the previous history of the transactions we can compare if the customer has a similar transaction. There might be cases where the customer may not have made a purchase for a particulat amount or a product ever before which may be a possible flag for fraud. Such scenarios may not be identifiable in this dataset but it atleast would give an idea of the possibilities of developing advanced models that involve other constraints. We can identify certain variables that will predict the fraud. Some transaction amounts may be succeptible for fraud. There may be cases where certrain low amounts have been identified as fraud and later a higher amount of fraud on the same account. This may be due to the transaction not being identified as fraud by the customer or the bank.

1.4 Data Definition

Through intensive reach, we came across this source of data the technique used for evaluation. This data was found at Kaggle, original data source was by ULB Machine Learning Group.

https://www.kaggle.com/mlg-ulb/creditcardfraud/home

http://mlg.ulb.ac.be/

This dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

Since data is very secured original features have been replaced by key components of PCA. Just the 'Time' and 'Amount' feature have not been transformed. There are 31 Columns in the dataset which are named as V1,V2,V3 etc. and are the principal components. The attributes and their names have not been disclosed due to confidentiality issues.

Dataset snapshot:

Attributes/Features	31
No. of file (s)	1
Size of file(s)	66 MB
Records	284,807

1.5 Measurements

We will focus on the most impactful variables that can help determine factors that lead to fraud. Predictive accuracy of the model on the historical data will be used for evaluating success towards this objective. Our project can be deemed successful if we can potentially identify the factors that play a major part in

fraud. Identification of fraudulent transactions based on the data given by the factors we choose will determine the success of the model.

1.6 Methodology and Computation Methods

Preprocessing

Before beginning with model building we shall carry out the knowledge discovery steps of checking for any missing values, check for data quality, to merge these datasets, perform random sampling i.e. Data Integration, perform necessary transformations if required to bring the data on same level, eliminate and summarize the cleansed data.

For addressing the above, we have identified that we will perform binning by smoothening by bin medians to bring the range of values to the same level.

In case of outliers, we shall perform clustering so that the data points which are extremes are identified and are not part of the model building process.

We have identified that our datasets are scattered and requires a step of consolidation, we shall perform Data Integration as a step for carrying out this. Since our datasets has numeric values, we will have to perform Correlation Analysis to identify positively correlated, negatively correlated and independent datasets. Pearson correlation will be used for this purpose.

If we find that the range of dataset values are having high variance then we shall decide to perform data transformation to bring the range of values at same level.

Model Building

The dataset we have in hand needs certain steps of preprocessing, further we shall build various models of classification on this dataset, measure the accuracy and cost functions, test how the prediction accuracy works on various models and finally select the best model.

The algorithms which will be considered for model building will be as below:

a. Linear Regression
b. Random forest classifier
c. Support Vector Machine (SVM)
d. Logistic Regression
e. K Means Clustering

We shall perform steps of building a model on training datasets, run it on testing datasets, measure its accuracy on testing datasets, identify if there is an overfitting problem, take necessary steps for improving the models and re build the model.

1.6 Output Summary

We propose that this dataset be split into training and testing sets using Hold Out Evaluation technique. Based on the dataset we shall perform classification tasks.

Initially we shall set the percentage as 80-20 i.e. 80% training and 20% testing data. Our evaluation matrix would be comparing the accuracy based on the different techniques used. First level of comparison would be between accuracy at training and testing data. Comparing the training and testing dataset can arise classification problems of overfitting.

To eliminate the overfitting problem, we would check and eliminate margin of error. This would give us an optimal result having the best classification technique for the given data set.

a. Data Quality:

By the end of the project we would have cleansed and preprocessed data that can be used for any classification purpose. Our preprocessing technique will help reduce and select the attributes giving us the least margin of error and greater accuracy.

b. Prediction model:

Our prediction model would be based on different configurations applied to each classification algorithm such as Linear Regression, Random Forest, Logistic regression and SVM.

c. Accuracy comparison:

Accuracy could be best compared using the outcome of each classification technique derived from the predicted model and compare their accuracy derived by the testing dataset. Technique giving the best accuracy percentage with training dataset would be the best model.

d. Data visualization:

Wherever necessary, we shall present outputs in clearly understandable visualizations like Bar graphs and Histograms. Comparison of algorithm accuracies, comparison amongst prediction results on testing datasets shall be measured and presented to justify why we chose the final model.

1.7 Project Implementation

Our team will identify the most accurate model which can predict the likelihood of a fraudulent transaction. After the best model is selected we can apply that model to the new transaction data to identify the likelihood of fraud. This, probability can be used for every transaction added to the database. The goal is to finalize a solution which helps every bank or organization to correctly identify the fraudulent transactions.

2. PROJECT PLAN

2.1 Project Management Approach

Successful project completion depends on effectively meeting the project objectives which could be accomplished through thoughtful planning, organizing and communication within the team. The

Traditional Waterfall model approach would be followed to break down the project requirements into a distinct and sequential phase. The waterfall model allows to incorporate changes to the project plan based on the lessons learnt at the earlier stages of the project.

The typical waterfall model is comprised of four stages such as Project initiation, Planning, Execution, control and closing. Initiation phase involves the initial analysis, identifying the key objective and the scope of the project. Resource allocation and time scheduling will be carried out in planning phase. Data set pre-processing, cleansing, building model with better accuracy are the processes carried out in execution stage.

2.2 Project Deliverables & Milestones

A milestone is a marker in a project that signifies a better change or different stages in project development process. Milestones displays the key events and maps forward movements and prioritized as crucial component of a project plan. Below are the different milestones identified in our project. During the planning phase of the project life cycle, a end -to-end analytical solution will be developed with techniques and methods to accomplish the project objectives.

Milestone	Project Lifecycle Stage	Due Date
Project Plan	Initiating	10-07-2018
Solution Development & Implementation	Planning	10-27-2018
Data Analysis: Exploratory Data Analysis	Executing	11-04-2018
Model Building & Evaluation	Executing & Control	11-25-2018
Deploy Solution, Deliver Implementation Report	Closing	12-02-2018

Table 1: *Summary Milestone Schedule*

2.2.1 Statement of Work

Introduction

Fraud is one of the major ethical issues in the credit card industry. We want to build a model which will help us predict the available data about credit card transactions from the people of Europe and predict how many frauds has occurred in that period, further we can also analyze more from the data about the patterns of how people there use credit cards. Second, to use some clustering techniques to understand the trends of when there is likely to be a fraud transaction. And accordingly, we can present to take any actions to prevent that from happening.

Scope of Work

The scope of work for our project of Credit Card Fraud Detection includes we plan on meeting regularly and working on the CRISP-DM phases of Business Evaluation, Data Understanding, Data preparation, modelling, evaluation and deployment. After fixing our objective, we plan on collecting data which contains transactions made by credit cards in September 2013 by European cardholders and iteratively build a model post loads of data cleaning and transformation. During the course of our project, we started off with submitting our project proposal to our professor. Post green signal from him we will start working our Data understanding, preparation, modelling, evaluation and finally deployment. The entire summary of these phases would be presented to the professor & then we would start working on implementation phases. The project deliverables & milestones have already mentioned in work requirements.

Period of Performance

We started with our project on 10/07/2018. The detailed information about our product time table is given in project deliverables or milestones column. If any changes are needed or any extension is required, we would be seeking professor's permission.

Place of Performance

We will be working on a project in college's library mostly. If it doesn't work out, then we will gather at our colleague's apartment. Additionally, we would be gathering twice to discuss about project's development.

Work Requirement

We have divided our project in some tasks which will help us complete our project successfully. First we start with kickoff here we will discuss about project proposal. It will be presented to the professor for approval. Next would be the design phase, in this phase, we would be discussing about which algorithms to use in our project. Also, how can we improve the prediction accuracy & precision using those algorithms. Also, we would be working on the pattern or design of our project. Next would be implementation Phase, in this phase, we would be working on coding part of project. We would be implementing the algorithms that was decided in project proposal. Next, training phase, where we will be training our model to predict the predictions. Then we will be using that model on test data to identify the accuracy of the predictions. And last would be the project Hands off where once the project is complete & we are satisfied with our results, then we will be providing a document of the same.

Schedule/ Milestones

The acceptance of deliverables will be presented to the professor. We will make sure that professor is aware about the deliverables.

2.3 Project Requirements

The project requires that all deliverables and expectations were finished at the predictable time. To be completed on time, the team needed to fully understand the objectives in addition to acquiring the appropriate and expected data required to accomplish it. For this reason, the group need a full understanding of the business application in order to choose the data necessary. The team likewise required access to the appropriate tools required to execute the project. At long last, the project required that the team members have the time to dedicate to this project to ensure its success.

Priority Matrix			
	Cost	**Performance**	**Time**
Accept	☒		
Enhance		☒	
Constrain			☒

Table 2: Priority Matrix

The project is majorly depending on the Time. Hence, we will try to execute all the phases in each timeline. At the same time, we will try to improve the scope of our project.

2.4 Project Constraints

Due to data quality and missing value issues in the dataset, it becomes a tedious process to normalize the data. There are plans to perform clustering in case of outliers and data integration to consolidate the scattered dataset. There are only 492 frauds in a total of 284,807 observations, this resulted in only 0.172% fraud cases. So, this dataset is justified by the low number of fraudulent transactions.

2.5 Project Assumptions

There were a couple of assumptions that had to be made in order to guarantee the project's success. To begin with, the team assumed they would have access to the data and tools required to meet the objective. Further, it was presumed that all the team members had space plan available to work with the team as necessary such that the final product meets their needs. Finally, it was expected that the data collected will be adequate to space plan results predictive of the desired target.

2.6 Project Risks

There are various risks that may arise at different stages of the project. It is always a good idea to discuss these terms in the initial stages and look for solutions. Should these risks occur we will have a ready solution to follow.

1. Missing data or data required is not available : This situation might occur early in the process but since our data has been given in a particular format with not complete information of the variables due to confidentiality. The domain knowledge as to what is the variable saying or the context of the variable will be missing. We will have to research few more data sources to learn more about the missing pieces in our dataset.
2. Usage of incorrect model : We might encounter this scenario at various stages. We will have to use and try out a number of models before we select the correct model. We will have to test the models individually.
3. Unable to meet project deadlines : We will not be able to meet the deadlines of the project if some of the initial deadlines have not been met. Based on the research the timelines may vary. This can be overcome by having a time period during every phase to review the previous stages once again before starting off with the next phase.

   To overcome these above risks involved we met the requirements of the phase early to go back and review the previous drafts and phases in the project. The team worked and collaborated on various aspects to support and overcome the risk factors involved.

2.7 Communication Management

It is important to have a communication management plan for the project to communicate the requirements and how the information will be distributed. The communication management plan outlines what, when and how information will be communicated and distributed, and who is responsible for the communicating information, etc.

Define Roles: Project Sponsor is the one who is responsible for the funding of the project, Project Manager owns most of the resources and oversees the project and Key stakeholders includes all the individuals and organizations.

2.7.1 Stakeholder Register

Project: Credit Card Fraud Detection and Clustering of Fraudulent cases			**Stake Holder Register**		**Date: 3/25/2018**
Name	**Designation**	**Department**	**Contact**	**Internal**	**Influence**
Shweta Bangad	Project Manager	ITM	312-722-9650	Internal	High
Venkatesh Naidu	Program Manager	ITM	331-307-9250	Internal	High
Vinodh Kumar Selvam	Research Analyst	ITM	312-647-6048	Internal	Moderate
Bhumin Shah	Tech Lead	ITM	773-664-6056	Internal	Moderate
Robert Henkins	Professor	ITM	123-456-7890	External	High

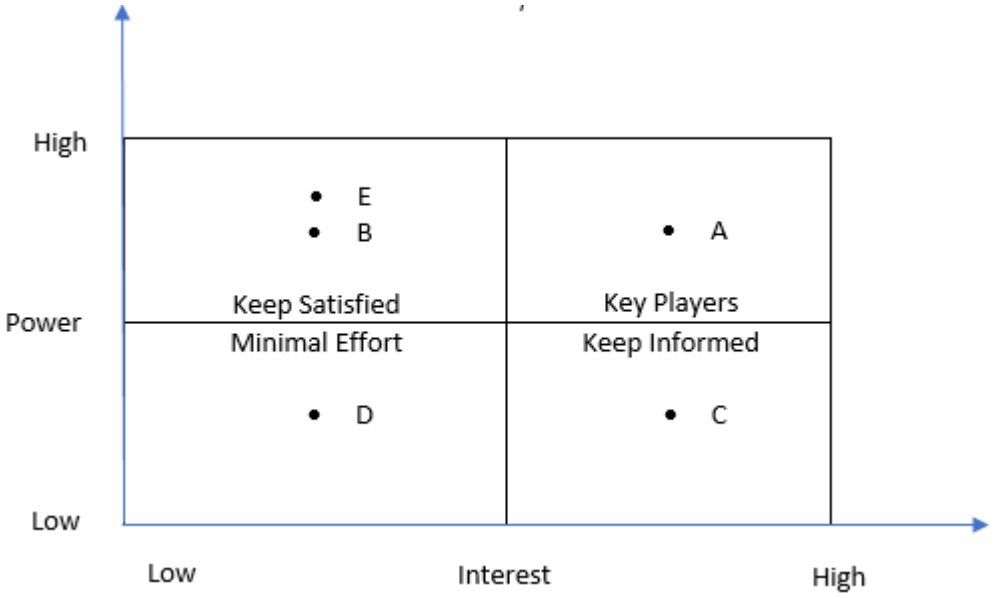
***Table 3**: Stakeholder Register*

2.7.2 Stakeholder Analysis

This portion depicts how the team will inspect its identified stakeholders. It gives the information about how stakeholders are assembled & what kind of contingent they will have depending on their influence, power & their involvement in the project

Key	**Organization**	**Name**	**Power**	**Interest**
A	ITM	Shweta Bangad	5	5
B	ITM	Venkatesh Naidu	4	3
C	ITM	Vinodh Kumar Selvam	3	4
D	ITM	Bhumin Shah	3	2
E	ITM	Robert Henkins	5	5

***Table 4**: Stakeholder Analysis*



- Stake holder D will require minimal efforts as they reside in lower left corner of the matrix.
- Stake holder E & B is in upper left corner. Therefore, they need to be keep satisfied by clearing their doubts & questions etc.
- Stake holder C needs to be keep informed as he resides in lower right corner of the matrix.
- Stake holder A is key player. She will be responsible for all the development & implementation requirements.

2.7.3 Communication Matrix

Information	**Receiver**	**Timing of Communication**	**Method of Communication**	**Sender**
Team Status Meeting	Project Team	Weekly	Call	Project Team
Team collaboration	Project Team	Daily	Email, Google Drive	Project Team
Team Status Reports	Project Team, Professor	Weekly	Blackboard	Project Team
Research Question Alignment	Project Team	At Project Start	Email	Project Team
Final Analysis	Professor, Project Team	At Project End	Email, hardcopy	Project Team

Table 5: *Communication Matrix*

2.8 Team Roles & Responsibilities

Team Member	Roles	Responsibility
Shweta Bangad	Project Lead & Tester	Design the project layout. Communicate the project distribution Information. Manage the project completion on time & test the models as needed and take care of the documentation
Venkatesh Naidu	Developer	Preprocess the Data. Develop the project model on training data set in R. If the model doesn't perform well on test data, then rebuild the model.
Vinodh Kumar Selvam	Analyst & Tester	Test the model on the test data set which is developed by training data. Find out the results & the model's performance.
Bhumin Shah	Developer & Analyst	Analyze the model accuracy, precision & recall using different algorithms. Identify which model performed better.

Table 6: Roles & Responsibilities

Team consisted of four team members: Shweta Bangad, Venkatesh Naidu, Vinodh Kumar Selvam, Bhumin Shah. Shweta will be primary communicator for the project distribution information. Bhumin and Vinodh will assist with Logistic Regression analysis while Venkatesh and Shweta will investigate classification techniques and accordingly Bhumin and Vinodh will work on it later. Shweta will take care of the overall documentation of the project. Once done each team member will have a look at all the model & techniques used.

2.9 Work Breakdown Structure (WBS)

Work Breakdown Structure (WBS) portions the venture work into consistent undertaking gatherings and tracks a task deliverable. In this each period of the model is further spilt into littler subsections to more readily pursue the stream. For Instance, in EDA we made recurrence tables for clear cut factors and additionally plotted histograms for representation of how every factor affect each other and in addition the reaction variable.

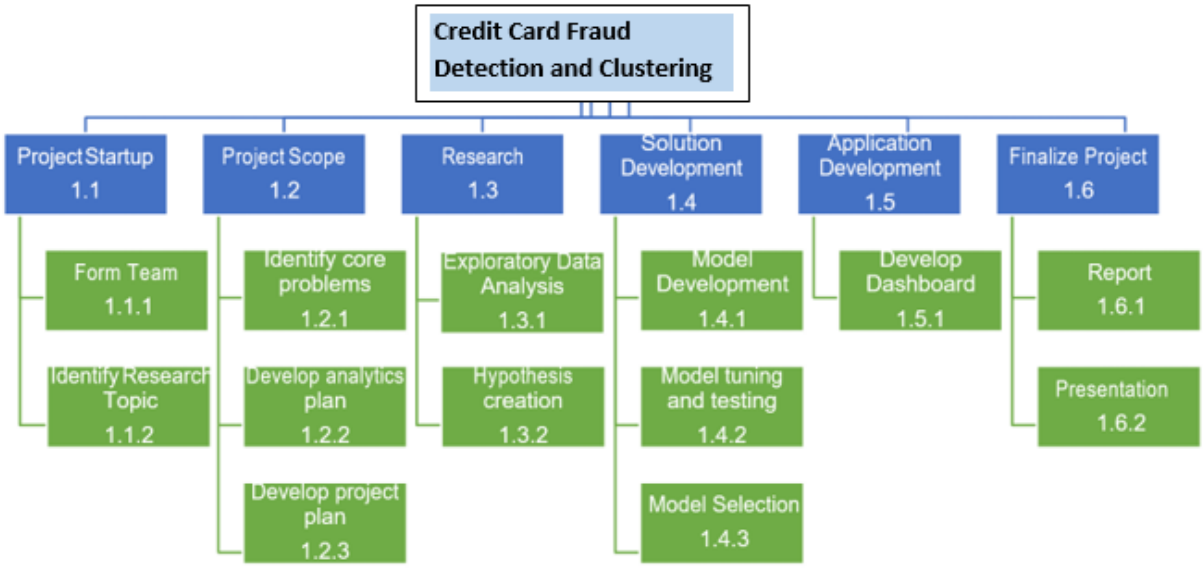


Figure 1: *Work Breakdown Structure*

2.10 Activity-On-Node Diagram

An activity-on-node diagram is designed to show which activities must be completed in order for other activities to commence. This is referred to as “finish-to-start” precedence – meaning one activity must be finished before the next one can start. By looking at the below diagram we can say that to start around by first looking at the data set and then we would be doing the data clean ups and visualizations. Then we will be working on Model Analysis, Classifier Selection and then perform analysis based on it and that’s the end.

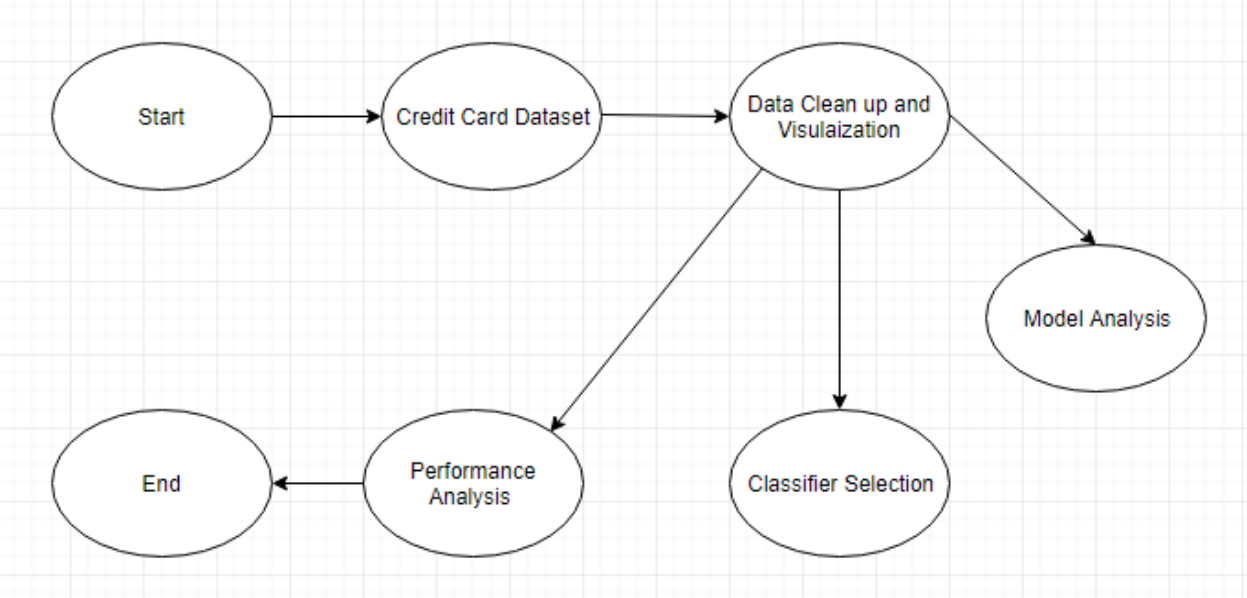


***Figure 2**: Activity-On-Node Diagram*

2.11 Project Loading & Leveling

According to schedule, there will be several instances where we will have to put more efforts due to shorten timelines. This will include 25th November, 2018 which is deadline of the project.

2.12 Schedule Baseline

A schedule baseline is the original approved project schedule, which is agreed by project stakeholders before the project starts. It does not change. It is a fixed measure which is used as a planning yard mark against which the progress on the actual project schedule can be measured. Schedule baseline would look same as the Summary Milestone Schedule.

When a project starts, the baseline schedule is the same as the actual schedule. But as circumstances eventuate, the actual schedule may deviate from the schedule baseline due to factors such as unforeseen risks eventuating, changes to project scope, and other changes outside the control of the Project Manager. The actual schedule responds dynamically to those factors affecting the project. The gap between it and the schedule baseline provides a measure of how much the project is ahead of or behind the originally planned schedule. If the variation is substantial, remedial action is needed. We were on time and did not need to change any timelines.

2.12.1 Summary budget and Cost

For the current project we did not require a lot of budget, but for future we can develop a UI where if we put the transaction id we can identify if it is fraud transaction or not, for that we would require a budget of around $10,000. We would need to find an investor to sponsor us for the project.

3.DATA QUALITY

3.1 Missing Data Cleansing & Imputation

There is no missing data apart from one missing value in the time column. This missing value is not of great significance as we will be removing the time column. The data cleaning steps were checked using R and Tableau Prep. No major missing elements were discovered. We can conclude that the data is clean. We have removed the time column because there is not a clear information regarding how they have been derived. The times appear to be just an addition of seconds between the first and the last transaction. We can see this removal as a process of normalization.

3.2 Data Transformation and Preparation

We can explore the distribution of the dataset.

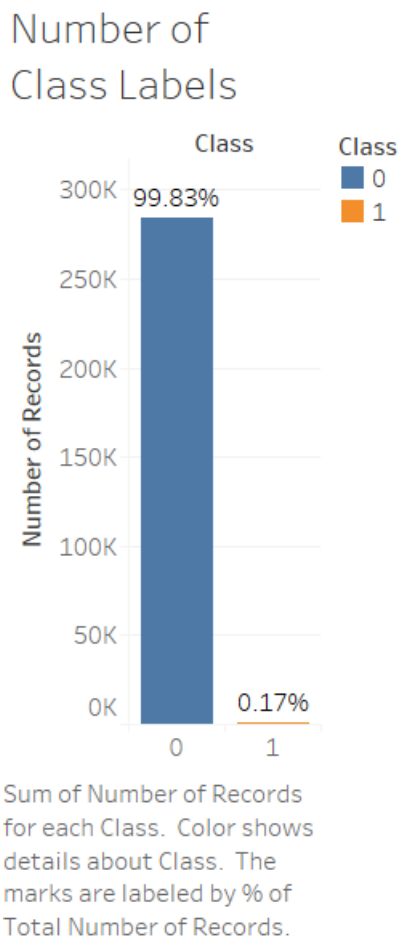


Figure 3: No of Class Labels

The dataset is not balanced. There is a clear inclination towards the number of records with class 0 as compared to the ones with class 1. Even if we use a null classifier we will be able to obtain 99% accuracy in predictions in this dataset

There is more variation in the transactions that are not fraudulent. The fraudulent transactions have a transaction amount that is pretty low.

Mean and median of class values

Class	Median Amount	Mean	Number of Records
0	22.00	88.291	284,315
1	9.25	122.211	492

Class: 0, 1

Median Amount, Mean and Number of Records broken down by Class. Color shows details about Class.

1. On observing this we can see that the fraudulent transactions have greater mean than the fraudulent ones. This would come to picture if we use the predictive models.

The data has already been transformed using PCA. There is not much of transformation that we will have to do on the dataset. The components that we have obtained after the preprocessing by PCA is used in the dataset.

3.3 Data Pre-processing techniques

Credit Card Fraud Detection dataset doesn't have any invalid or Null values, But the dataset is highly imbalanced with only 492 frauds in a total of 284,807 observations, this resulted in only 0.172% fraud cases. The Class variable in the dataset contains values of 1 and 0, where the '1' represents fraud transaction and '0' represents valid transaction. Since we have got very a smaller number of fraud transaction, building a model without any sampling will get biased prediction towards the majority class (Class '0'), in a real-time scenario, this prediction will result in bank predicting everything as fraud (or) not identifying the fraud transaction at all. We have tried implementing two techniques that can help balance the ratio of fraud vs non-fraud cases. Following are the techniques:

1. Synthetic Minority Oversampling Technique (SMOTE)

2. Random Over Sampling Examples (ROSE)

Synthetic Minority Oversampling Technique

We have tried using synthetic minority oversampling technique (SMOTE) in our model to solve the imbalanced dataset by generating artificial data. The artificial data is generated based on minority observation(class'1') to shift the classifier learning bias towards minority class. In the later stage of the

project, we will be implementing different sampling technique to figure out the model with better accuracy.

```
>
> mydata=read.csv('creditcard.csv')
> table(mydata$class)

     0      1
284315    492
>
```

SMOTE:

```
> mydata$class <- factor(ifelse(mydata$class == "2","1","0"))
> newdata <- SMOTE(mydata$class~.,mydata,perc.over = 4000,perc.under=100)
> table(newdata$class)

    0     1
19680 20172
>
```

Random Over Sampling Examples (ROSE)

Creates a sample of synthetic data by enlarging the features space of minority and majority class examples. Operationally, the new examples are drawn from a conditional kernel density estimate of the two classes. In our dataset the minority class are the fraudulent cases and majority are normal transactions. We will try creating sampling those ROSE technique and check conditions.

```
>
> mydata=read.csv('creditcard.csv')
> table(mydata$class)

     0      1
284315    492
>
```

As mentioned earlier we already know that fraud cases (Class = 1) are 492 compared to normal transactions (Class =0) are 284315.

```
> data.rose <- ROSE(Class ~ ., data = train, seed = 1)$data
> table(data.rose$Class)

     0      1
106697 106908
```

ROSE technique has generated synthetic data and now has almost equal data for classification purpose.

3.4 Exploratory Data Analysis

Exploratory data analysis is one of the main steps in our project that will help us to summarize the main characteristics with the help of tableau as a powerful visualization tool along with R visualization packages like ggplot2.

We will look at the overall picture of the data initially and later dig into the variables in the later sections. The following visualizations give us an idea of the variables that may be significant in our analysis.

A recap of the previous stages shows the number of fraudulent cases and the legit cases.

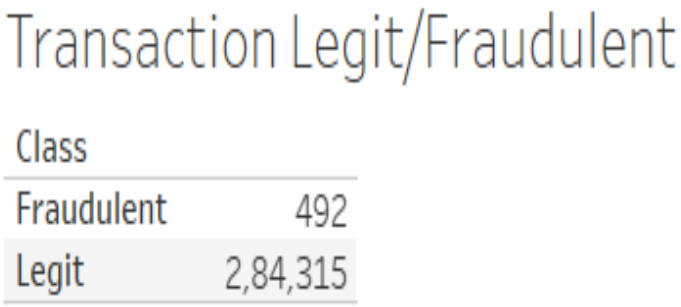


Figure 4: Amount per Class

In the above images we have an idea of the range of the classes. Below we can clearly see how the amount is spread for all the fraudulent cases.

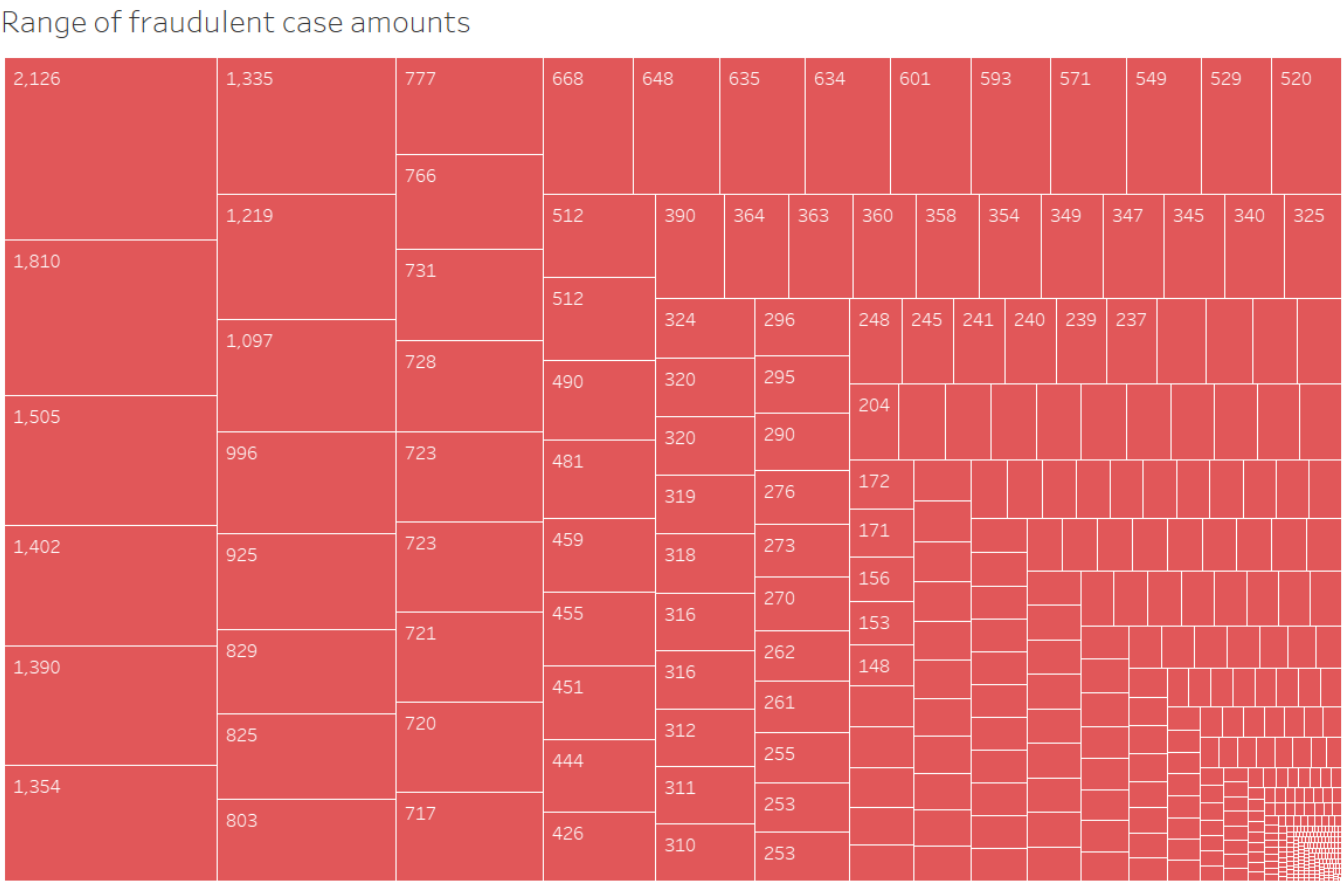


Figure 5 : Range of Fraudulent Case Amounts

The graph given above shows the transactions for a range of amounts. There are a lot of transactions that are with a lower amount. Another observation is that the transactions with a large amount does not result in fraud. In fact, there have been no transactions above 2400 resulting in fraud. Also, this may be because of the security involved with the larger amounts.

Using this image, we can try to find patterns in the fraudulent cases. The amount seems to be uniformly spread between $0 to $2126. This information is valuable and adds to the domain knowledge. Also, there are a number of cases with 0 transaction amount which can be interpreted as an attempted fraud by the fraudster or an attempt to break through the system. A successful breach would result in a larger fraudulent transaction in the future attempts.

From the image we can see that as the sizes of the boxes in the image is large it signifies the higher amount. The image shows that there are more fraudulent transactions which have a lower value in terms of amount.

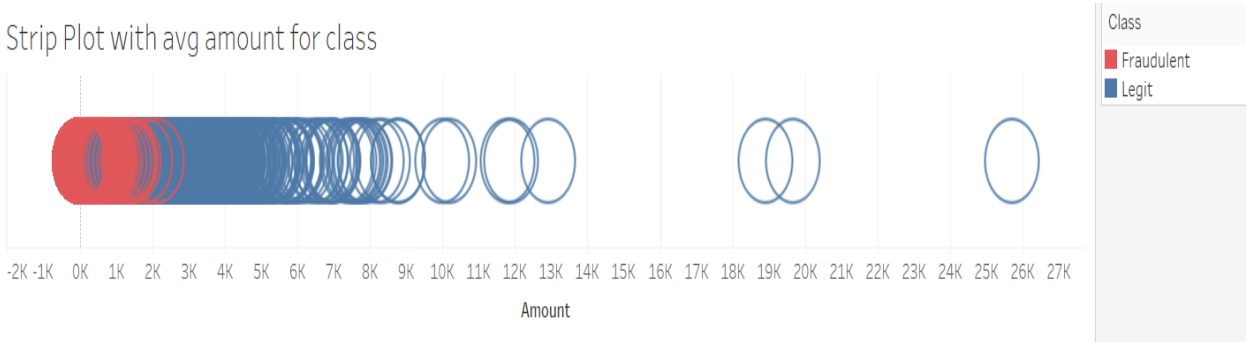


Figure 7 : Strip plot with average amount for class

We can see the average amounts of the classes and also observe the extreme values. From this image it is clear that the fraudulent transactions are focused towards the lower amounts. We can also see that the extremely high amounts are not fraudulent.

3.5 Quantitative Variables (Continuous Predictors)

From the analysis of the Initial set, it could be inferred that there are 29 quantitative variables, building a model with 29 variables would return poor accuracy, hence we would be using principle component analysis to identify the few important variables.

3.5 Principal Component Analysis:

Principal Component Analysis is a technique that could summarize and visualize the information in a dataset. It's used to extract the significant variables from the dataset, which has more influence on the dependent variable and to reduce the dimensionality in the data by neglecting the noise and redundancy in the dataset. Finding the highly correlated variables could also be possible with Principal component Analysis.

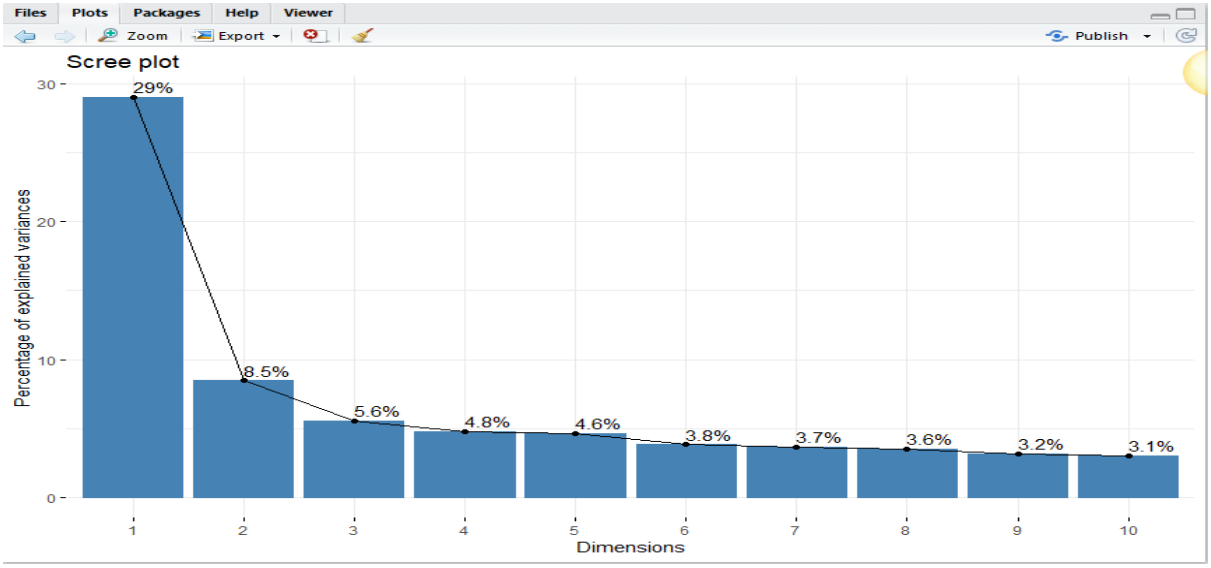
3.6 PCA Computation and Visualization:

We will be using prcomp() which is a built-in function in R to compute the PCA. R implementation of PCA to compute principal component analysis.

```
> library(factoextra)
> # Processing the Dataset
> pcadata1<-data.rose[,1:29]
> res.pca <- prcomp(pcadata1, scale = FALSE)
> print(res.pca)
Standard deviations (1, .., p=29):
 [1] 316.0303576  15.4558544   6.9658624   6.0370529   4.8416902   4.4205634   4.1049460   3.8733774   3.6260945
[10]   3.0636851   3.0162508   2.8377150   2.5198420   2.3445136   1.9754062   1.9022398   1.8402740   1.7859896
[19]   1.4111687   1.3426201   1.2962537   1.2697924   1.2329502   1.1368185   1.0957463   0.7679443   0.6697220
[28]   0.5728355   0.5392164

Rotation (n x k) = (29 x 29):
                PC1           PC2           PC3           PC4           PC5           PC6           PC7           PC8
V1    -5.526624e-04  0.3371964819  0.2564319298 -0.4252928353  0.2318948868 -0.633822654  0.328948506 -0.1464257367
V2    -2.446945e-03 -0.2115072982 -0.1182759580  0.0513276090  0.1043205027  0.040116093 -0.022546591 -0.0167670319
V3    -4.451092e-05  0.4128233126  0.0778871763 -0.2794488727 -0.6244157938  0.016110178 -0.474272145  0.2111714439
V4     1.211005e-04 -0.1699703165  0.0526329174 -0.1036826506  0.1482468064 -0.007174866 -0.115796588  0.1128314996
V5    -1.398831e-03  0.2509251167  0.0662233697 -0.2262397193  0.1728947808  0.024214395 -0.253257284  0.0049135619
V6     8.521989e-04  0.0503488508 -0.1196813097 -0.0335073303 -0.0528924548  0.002600947  0.053785714 -0.0183980261
V7     2.209796e-03  0.3829541502  0.3460717870 -0.0449128360  0.2376405004  0.638194128  0.361912998  0.3203241271
V8     2.583840e-04 -0.0435372552  0.7413690843  0.5731067718 -0.0816241314 -0.217215341 -0.165458670 -0.0109469555
V9     1.575654e-04  0.1264657685  0.0026186708  0.0474062875 -0.0230877208  0.039953727  0.001900210 -0.1510799694
V10    1.465780e-04  0.2884749323  0.0126323625  0.1410578885 -0.0514784632  0.110334666  0.046171183 -0.4607733348
V11   -1.673802e-04 -0.1423168435  0.1111820064 -0.1365437213  0.1071726812  0.029783743 -0.114408629  0.0115073827
V12    2.663037e-04  0.2633216465 -0.1816294564  0.2606402466 -0.1258986803 -0.015742078  0.174175352 -0.1475599829
V13    2.105788e-05 -0.0049664188  0.0320825462  0.0205624520 -0.0131156129 -0.003133145  0.013712436 -0.0100815058
V14    5.003826e-04  0.2240858996 -0.2622393977  0.3117897604 -0.2970836426 -0.127876877  0.348289514  0.0475038983
V15    1.870246e-04  0.0089645001  0.0296097464 -0.0006015844 -0.0102319248  0.017031347 -0.007394244 -0.0281942532
```

Screen plot to determining the number of principal components



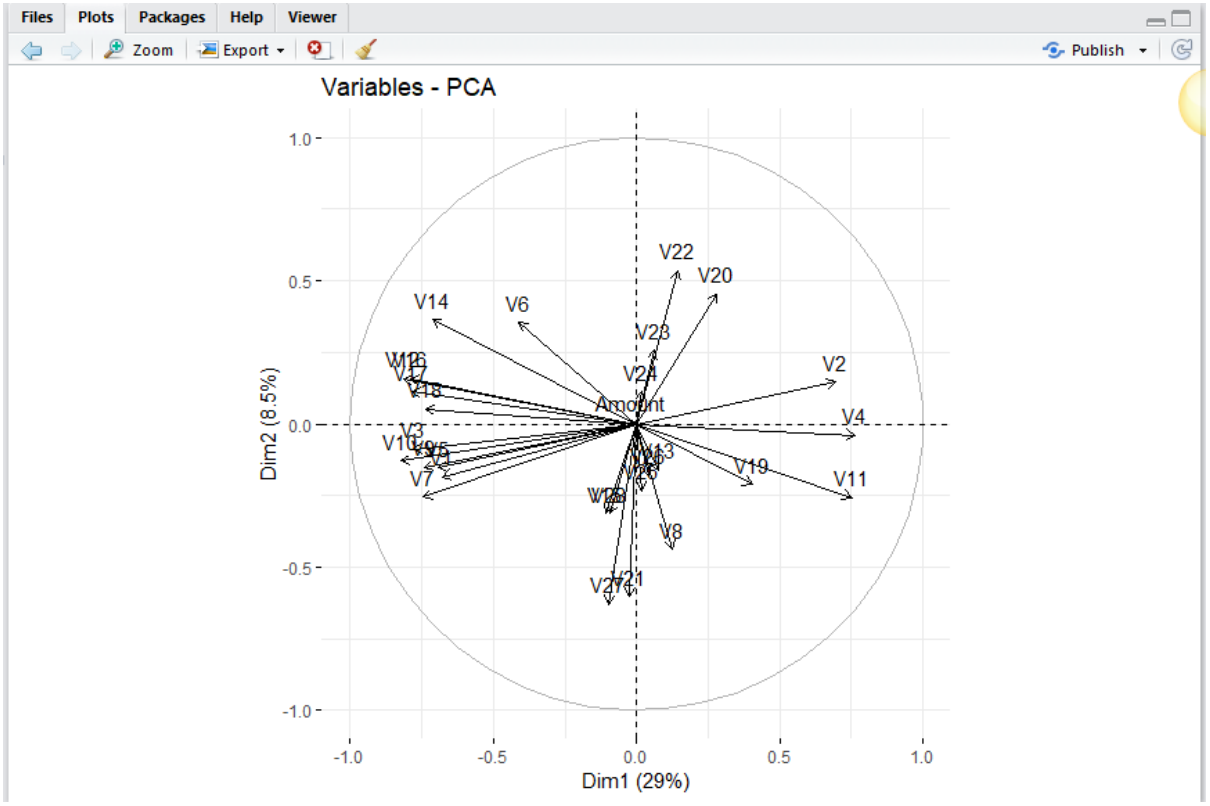
From the above scree plot, it could be inferred that the first eight principal components explain the 76% of the variation. which can be accepted.

To get results from PCA:

Correlation Circle:

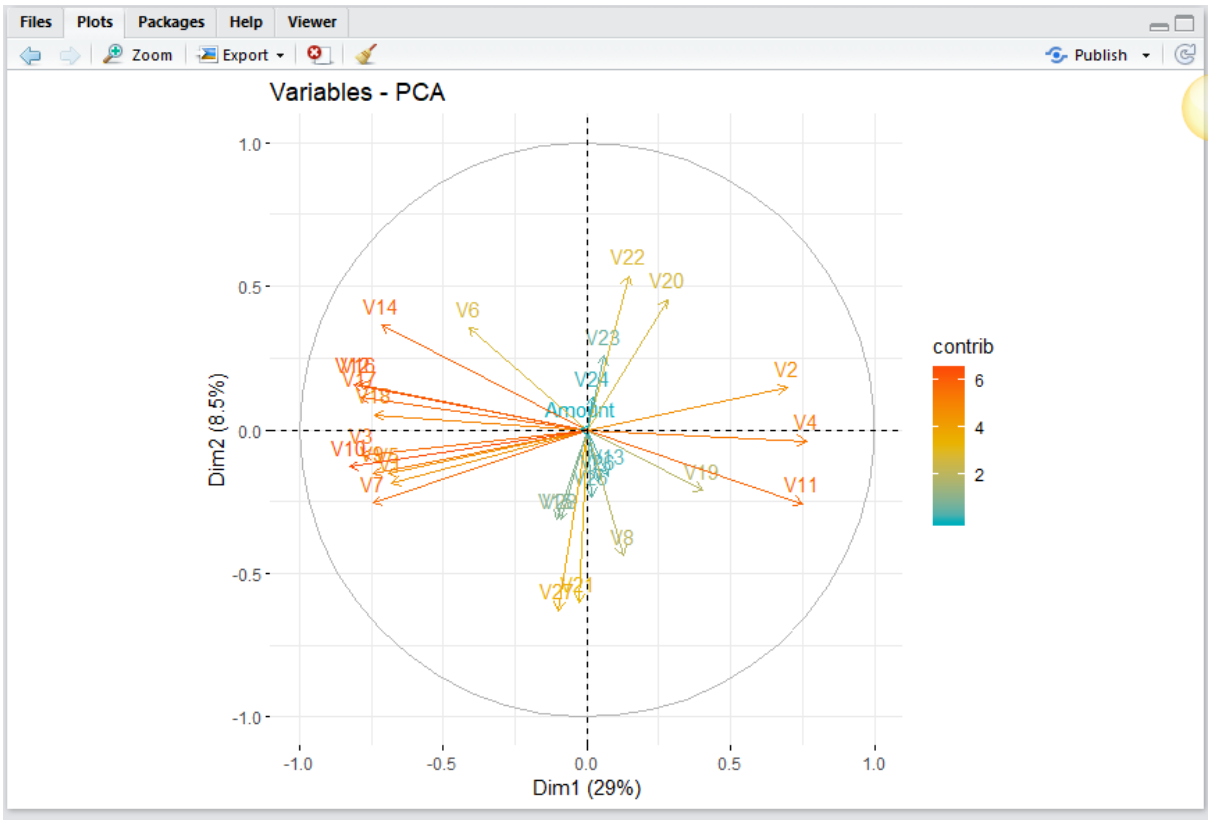
The plot below describes about the relationship between all the variables and could be referred as variable correlation plot, which infers Positively correlated variables are clustered together and Negatively

correlated variables are situated on inverse sides of the plot inception (opposed quadrants).



Contributions of variables to PCs

The Factors that are related with PC1 (i.e., Dim.1) and PC2 (i.e., Dim.2) are the most vital in clarifying the variability in the data set. Factors that don't related with any PC or correlated with the last dimensions will be removed to improve the overall analysis.



3.7 Model Based EDA

With model-based EDA we would be able to identify attributes that are important within that model. In this approach we have decided to build linear regression model and logistic regression with all attributes. Firstly, we have spilt data into training and testing and then performed model building.

3.7.1 Linear Regression

We started building our linear regression model using the “lm” function. We can see the results in the image below:

```
> fit_check = lm(Class1~V11+V12+V13+V14+V15+V16+V17+V18+V19+V120+V121+V122+V123+V124+V125+V126+V127+V128,
data=data.check)
> summary(fit_check)

Call:
lm(formula = Class1 ~ V11 + V12 + V13 + V14 + V15 + V16 + V17 +
    V18 + V19 + V120 + V121 + V122 + V123 + V124 + V125 + V126 +
    V127 + V128, data = data.check)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3490 -0.2582 -0.1748  0.2608  1.3649

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.2481065  0.0008073  307.342  < 2e-16 ***
V11          0.0273858  0.0003233   84.697  < 2e-16 ***
V12         -0.0152495  0.0002077  -73.439  < 2e-16 ***
V13         -0.0245397  0.0005137  -47.772  < 2e-16 ***
V14         -0.0374510  0.0001980 -189.174  < 2e-16 ***
V15         -0.0030568  0.0005556   -5.501 3.77e-08 ***
V16         -0.0071827  0.0002582  -27.823  < 2e-16 ***
V17         -0.0024100  0.0001450  -16.621  < 2e-16 ***
V18          0.0048900  0.0003229   15.144  < 2e-16 ***
V19         -0.0029109  0.0004816   -6.044 1.50e-09 ***
V120         0.0171963  0.0005212   32.993  < 2e-16 ***
V121         0.0038940  0.0002256   17.261  < 2e-16 ***
V122         0.0029562  0.0005255    5.625 1.86e-08 ***
V123        -0.0052106  0.0004449  -11.711  < 2e-16 ***
V124        -0.0122597  0.0009799  -12.511  < 2e-16 ***
V125         0.0010539  0.0008044    1.310     0.19
V126        -0.0351239  0.0011407  -30.791  < 2e-16 ***
V127        -0.0068110  0.0005535  -12.306  < 2e-16 ***
V128         0.0229710  0.0012009   19.128  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3512 on 284788 degrees of freedom
Multiple R-squared:  0.5066,    Adjusted R-squared:  0.5066
F-statistic: 1.625e+04 on 18 and 284788 DF,  p-value: < 2.2e-16
```

For any model, to check for the importance of predictor value, we conduct individual parameter test. This test runs on the following hypothesis.

Null hypothesis	H_0= The field is an important predictor variable
Alternate hypothesis	H_A= The field is NOT an important predictor variable

If the Pr(|z|) value in the output is greater than 0.05, we will go ahead and reject the null hypothesis and accept the alternate hypothesis. In model "fit_check" we see V125 is greater than 0.05. Hence, V125 fails in the individual parameter test and we can reject the null hypothesis. Removing V125 from our existing model and build a new model

```
> fit_check1 = lm(Class1~V11+V12+V13+V14+V15+V16+V17+V18+V19+V120+V121+V122+V123+V124+V126+V127+V128, dat
a=data.check)
> summary(fit_check1)

Call:
lm(formula = Class1 ~ V11 + V12 + V13 + V14 + V15 + V16 + V17 +
    V18 + V19 + V120 + V121 + V122 + V123 + V124 + V126 + V127 +
    V128, data = data.check)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3494 -0.2582 -0.1748  0.2609  1.3653

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.2481103  0.0008073  307.348  < 2e-16 ***
V11          0.0273863  0.0003233   84.698  < 2e-16 ***
V12         -0.0152410  0.0002075  -73.434  < 2e-16 ***
V13         -0.0245240  0.0005135  -47.754  < 2e-16 ***
V14         -0.0374676  0.0001976 -189.648  < 2e-16 ***
V15         -0.0030629  0.0005556   -5.513 3.54e-08 ***
V16         -0.0071726  0.0002580  -27.796  < 2e-16 ***
V17         -0.0024135  0.0001450  -16.647  < 2e-16 ***
V18          0.0048905  0.0003229   15.146  < 2e-16 ***
V19         -0.0029679  0.0004796   -6.188 6.11e-10 ***
V120         0.0172320  0.0005205   33.107  < 2e-16 ***
V121         0.0038971  0.0002256   17.275  < 2e-16 ***
V122         0.0028910  0.0005232    5.526 3.28e-08 ***
V123        -0.0051764  0.0004442  -11.654  < 2e-16 ***
V124        -0.0122695  0.0009799  -12.522  < 2e-16 ***
V126        -0.0350786  0.0011402  -30.765  < 2e-16 ***
V127        -0.0067472  0.0005513  -12.238  < 2e-16 ***
V128         0.0230615  0.0011989   19.235  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3512 on 284789 degrees of freedom
Multiple R-squared:  0.5066,    Adjusted R-squared:  0.5066
F-statistic: 1.72e+04 on 17 and 284789 DF,  p-value: < 2.2e-16
```

Based on the results above we see that our model has passed individual parameter test and all the attributes P value is less than 0.05.

3.7.2 Logistic Regression

We started building our Logistic regression model using the “glm” function. We can see the results in the image below:

```
> fit_check_log = glm(Class1~V11+V12+V13+V14+V15+V16+V17+V18+V19+V120+V121+V122+V123+V124+V125+V126+V127+
V128, family="binomial", data=data.check)
> summary(fit_check_log)

Call:
glm(formula = Class1 ~ V11 + V12 + V13 + V14 + V15 + V16 + V17 +
    V18 + V19 + V120 + V121 + V122 + V123 + V124 + V125 + V126 +
    V127 + V128, family = "binomial", data = data.check)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.4629  -0.5356   0.0000   0.0962   4.4797

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -1.840637   0.007589 -242.538  < 2e-16 ***
V11          0.348612   0.003956   88.122  < 2e-16 ***
V12         -0.290246   0.002904  -99.964  < 2e-16 ***
V13         -0.161048   0.005190  -31.028  < 2e-16 ***
V14         -0.482447   0.003104 -155.453  < 2e-16 ***
V15         -0.046782   0.005578   -8.387  < 2e-16 ***
V16         -0.149422   0.003253  -45.933  < 2e-16 ***
V17         -0.052628   0.001921  -27.393  < 2e-16 ***
V18         -0.001423   0.003955   -0.360  0.71895
V19         -0.052736   0.005281   -9.986  < 2e-16 ***
V120         0.089536   0.005621   15.930  < 2e-16 ***
V121         0.042492   0.003069   13.848  < 2e-16 ***
V122         0.093361   0.006039   15.459  < 2e-16 ***
V123        -0.076699   0.005361  -14.306  < 2e-16 ***
V124        -0.071038   0.009273   -7.661 1.84e-14 ***
V125        -0.024034   0.009015   -2.666  0.00768 **
V126        -0.129594   0.011134  -11.639  < 2e-16 ***
V127         0.129510   0.007844   16.510  < 2e-16 ***
V128         0.240974   0.014103   17.087  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 394825  on 284806  degrees of freedom
Residual deviance: 157489  on 284788  degrees of freedom
AIC: 157527

Number of Fisher Scoring iterations: 8
```

Here we see that V18 has P value which is greater than 0.05 and fails the individual parameter test. We will re-create the mode removing V18.

```
> fit_check_log1 = glm(Class1~V11+V12+V13+V14+V15+V16+V17+V19+V120+V121+V122+V123+V124+V125+V126+V127+V12
8, family="binomial", data=data.check)
> summary(fit_check_log1)

Call:
glm(formula = Class1 ~ V11 + V12 + V13 + V14 + V15 + V16 + V17 +
    V19 + V120 + V121 + V122 + V123 + V124 + V125 + V126 + V127 +
    V128, family = "binomial", data = data.check)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-5.4628  -0.5356   0.0000   0.0963   4.4797

Coefficients:
             Estimate Std. Error  z value Pr(>|z|)
(Intercept) -1.840699   0.007587 -242.604  < 2e-16 ***
V11          0.348667   0.003953   88.203  < 2e-16 ***
V12         -0.290328   0.002895 -100.298  < 2e-16 ***
V13         -0.161011   0.005189  -31.027  < 2e-16 ***
V14         -0.482429   0.003103 -155.475  < 2e-16 ***
V15         -0.046772   0.005578   -8.385  < 2e-16 ***
V16         -0.149608   0.003212  -46.583  < 2e-16 ***
V17         -0.052769   0.001881  -28.056  < 2e-16 ***
V19         -0.052628   0.005272   -9.982  < 2e-16 ***
V120         0.089570   0.005620   15.939  < 2e-16 ***
V121         0.042471   0.003067   13.846  < 2e-16 ***
V122         0.093438   0.006035   15.482  < 2e-16 ***
V123        -0.076732   0.005360  -14.315  < 2e-16 ***
V124        -0.071003   0.009272   -7.657  1.9e-14 ***
V125        -0.024067   0.009015   -2.670  0.00759 **
V126        -0.129537   0.011133  -11.636  < 2e-16 ***
V127         0.129420   0.007839   16.509  < 2e-16 ***
V128         0.240908   0.014101   17.084  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 394825  on 284806  degrees of freedom
Residual deviance: 157489  on 284789  degrees of freedom
AIC: 157525

Number of Fisher Scoring iterations: 8
```

After removing V18 we see all the attributes are within 0.05 and which accepts Null hypothesis.

3.8. Variable Reduction

Variable reduction is one of the most important process to build a best model without impacting the accuracy and prediction. It enhances the efficiency of the model by reducing the run time and could remove the multi-collinearity problems between the independent variables as well. The credit card fraud detection has 29 variables including 1 dependent variable. We have already identified the

correlation between the variables and contribution of the variables using principal component analysis, in addition to that we will be implementing Backward elimination method to reduce the variable based on the P-value.

Implementation of backward elimination in R. Backward elimination starts with the full model and eliminates one variable at the time until a reasonable candidate regression model is found. It typically uses a criterion based on the goodness-of-fit F-test.

```
> data.rose <- ROSE(Class ~ ., data = train, seed = 1)$data
> Backelimin1= lm(Class~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+
> summary(Backelimin1)

Call:
lm(formula = Class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 +
    V9 + V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 +
    V19 + V20 + V21 + V22 + V23 + V24 + V25 + V26 + V27 + V28,
    data = data.rose)

Residuals:
    Min      1Q  Median      3Q     Max
-1.3840 -0.2329 -0.1393  0.2257  1.3009

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.2177065  0.0008908  244.397  < 2e-16 ***
V1          -0.0023361  0.0001467  -15.926  < 2e-16 ***
V2           0.0026291  0.0002221   11.840  < 2e-16 ***
V3          -0.0030306  0.0001486  -20.391  < 2e-16 ***
V4           0.0347181  0.0003013  115.215  < 2e-16 ***
V5           0.0032978  0.0001926   17.121  < 2e-16 ***
V6          -0.0089793  0.0004006  -22.414  < 2e-16 ***
V7           0.0031163  0.0001492   20.883  < 2e-16 ***
V8          -0.0038485  0.0001331  -28.917  < 2e-16 ***
V9          -0.0076909  0.0003700  -20.786  < 2e-16 ***
V10         -0.0057990  0.0002177  -26.642  < 2e-16 ***
V11          0.0173526  0.0003616   47.993  < 2e-16 ***
V12         -0.0082387  0.0002343  -35.163  < 2e-16 ***
V13         -0.0222422  0.0005611  -39.637  < 2e-16 ***
V14         -0.0270638  0.0002278 -118.796  < 2e-16 ***
V15         -0.0005025  0.0006010   -0.836 0.403101
V16         -0.0030718  0.0002874  -10.688  < 2e-16 ***
V17         -0.0013817  0.0001618   -8.538  < 2e-16 ***
V18          0.0076810  0.0003651   21.038  < 2e-16 ***
V19         -0.0002669  0.0005330   -0.501 0.616472
V20          0.0054210  0.0005645    9.603  < 2e-16 ***
V21          0.0082031  0.0002454   33.434  < 2e-16 ***
V22          0.0031177  0.0005660    5.509 3.62e-08 ***
V23         -0.0166801  0.0005543  -30.091  < 2e-16 ***
V24         -0.0125168  0.0010627  -11.778  < 2e-16 ***
V25         -0.0034075  0.0009207   -3.701 0.000215 ***
V26         -0.0312814  0.0012380  -25.268  < 2e-16 ***
V27          0.0036758  0.0006084    6.042 1.53e-09 ***
V28          0.0391424  0.0013121   29.833  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3291 on 213576 degrees of freedom
Multiple R-squared:  0.5667,    Adjusted R-squared:  0.5667
F-statistic:  9977 on 28 and 213576 DF,  p-value: < 2.2e-16
```

Considering the P-value, the variable with highest P-value more than confidence level would be removed manually and the model would be built again. From the first model, variable V22 will be removed based on the P-value

```
> Backelimin2 = lm(Class~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V15+V16+V17+V18+V19+V2
> summary(Backelimin2)

Call:
lm(formula = Class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 +
    V9 + V10 + V11 + V12 + V13 + V14 + V15 + V16 + V17 + V18 +
    V19 + V20 + V21 + V23 + V24 + V25 + V26 + V27 + V28, data = data.rose)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4305 -0.2329 -0.1394  0.2257  1.2959

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)
(Intercept)  2.177e-01  8.908e-04  244.385  < 2e-16 ***
V1          -2.294e-03  1.465e-04  -15.658  < 2e-16 ***
V2           2.566e-03  2.218e-04   11.571  < 2e-16 ***
V3          -3.007e-03  1.486e-04  -20.243  < 2e-16 ***
V4           3.475e-02  3.013e-04  115.338  < 2e-16 ***
V5           3.326e-03  1.926e-04   17.270  < 2e-16 ***
V6          -8.930e-03  4.005e-04  -22.296  < 2e-16 ***
V7           3.125e-03  1.492e-04   20.938  < 2e-16 ***
V8          -3.838e-03  1.331e-04  -28.836  < 2e-16 ***
V9          -7.813e-03  3.694e-04  -21.153  < 2e-16 ***
V10         -5.864e-03  2.174e-04  -26.979  < 2e-16 ***
V11          1.735e-02  3.616e-04   47.982  < 2e-16 ***
V12         -8.287e-03  2.341e-04  -35.392  < 2e-16 ***
V13         -2.227e-02  5.612e-04  -39.681  < 2e-16 ***
V14         -2.698e-02  2.273e-04 -118.683  < 2e-16 ***
V15         -5.363e-04  6.010e-04   -0.892    0.372
V16         -3.126e-03  2.873e-04  -10.883  < 2e-16 ***
V17         -1.411e-03  1.618e-04   -8.724  < 2e-16 ***
V18          7.644e-03  3.651e-04   20.939  < 2e-16 ***
V19         -9.438e-05  5.321e-04   -0.177    0.859
V20          5.818e-03  5.599e-04   10.391  < 2e-16 ***
V21          7.723e-03  2.294e-04   33.670  < 2e-16 ***
V23         -1.665e-02  5.543e-04  -30.030  < 2e-16 ***
V24         -1.247e-02  1.063e-03  -11.734  < 2e-16 ***
V25         -3.894e-03  9.165e-04   -4.248 2.15e-05 ***
V26         -3.114e-02  1.238e-03  -25.159  < 2e-16 ***
V27          3.244e-03  6.034e-04    5.377 7.59e-08 ***
V28          3.868e-02  1.309e-03   29.537  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3292 on 213577 degrees of freedom
Multiple R-squared:  0.5667,    Adjusted R-squared:  0.5666
F-statistic: 1.034e+04 on 27 and 213577 DF,  p-value: < 2.2e-16
```

From the second model, variable V27 will be removing based on the P-value. On further iteration, we can reduce 4 variables (V22, V27, V19, V15) by using Backward elimination which doesn’t have much significance on dependent variable.

```
> Backelimin5 = lm(Class~V1+V2+V3+V4+V5+V6+V7+V8+V9+V10+V11+V12+V13+V14+V16+V17+
> summary(Backelimin5)

Call:
lm(formula = Class ~ V1 + V2 + V3 + V4 + V5 + V6 + V7 + V8 +
    V9 + V10 + V11 + V12 + V13 + V14 + V16 + V17 + V18 + V20 +
    V21 + V23 + V24 + V25 + V26 + V28, data = data.rose)

Residuals:
    Min      1Q  Median      3Q     Max
-1.4501 -0.2330 -0.1396  0.2258  1.2954

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept)  0.2177426  0.0008909  244.414  < 2e-16 ***
V1          -0.0022659  0.0001464  -15.478  < 2e-16 ***
V2           0.0025194  0.0002210   11.401  < 2e-16 ***
V3          -0.0029998  0.0001485  -20.205  < 2e-16 ***
V4           0.0347665  0.0003009  115.541  < 2e-16 ***
V5           0.0033627  0.0001921   17.501  < 2e-16 ***
V6          -0.0089570  0.0004003  -22.378  < 2e-16 ***
V7           0.0031733  0.0001487   21.336  < 2e-16 ***
V8          -0.0037554  0.0001312  -28.617  < 2e-16 ***
V9          -0.0077372  0.0003691  -20.962  < 2e-16 ***
V10         -0.0057563  0.0002161  -26.639  < 2e-16 ***
V11          0.0175645  0.0003593   48.887  < 2e-16 ***
V12         -0.0082825  0.0002341  -35.384  < 2e-16 ***
V13         -0.0221807  0.0005599  -39.618  < 2e-16 ***
V14         -0.0270679  0.0002265 -119.514  < 2e-16 ***
V16         -0.0031350  0.0002827  -11.089  < 2e-16 ***
V17         -0.0014197  0.0001606   -8.839  < 2e-16 ***
V18          0.0076564  0.0003638   21.046  < 2e-16 ***
V20          0.0059377  0.0005595   10.613  < 2e-16 ***
V21          0.0079235  0.0002256   35.127  < 2e-16 ***
V23         -0.0168131  0.0005532  -30.395  < 2e-16 ***
V24         -0.0130390  0.0010573  -12.333  < 2e-16 ***
V25         -0.0033885  0.0009067   -3.737 0.000186 ***
V26         -0.0307791  0.0012347  -24.928  < 2e-16 ***
V28          0.0391789  0.0013050   30.021  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3292 on 213580 degrees of freedom
Multiple R-squared:  0.5666,    Adjusted R-squared:  0.5665
F-statistic: 1.163e+04 on 24 and 213580 DF,  p-value: < 2.2e-16

>
```

All the variables in the above model has P-value less than level of significance and passes the individual parameter test as well. The Adjusted R-square value is 0.5665, which explains 56.56% variation in dependent variable.

3.9. High Value variable

Decision Tree

Decision Tree is basically used for both classification or regression problems. In our case, a classification decision tree was built with the help of DecisionTreeClassifier() function. We split the data using the ratio of 70:30 in training and testing.

We performed DecisionTree with 2 methods: Gini Index and Entropy

Results Using Gini Index:

Predicted values:

[0. 0. 0. ... 0. 0. 0.]

Confusion Matrix: [[85281 20]

[34 108]]

Accuracy : 99.9367999719111

Report : precision recall f1-score support

0.0 1.00 1.00 1.00 85301

1.0 0.84 0.76 0.80 142

avg / total 1.00 1.00 1.00 85443

```
Results Using Gini Index:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:  [[85281    20]
 [   34   108]]
Accuracy :  99.9367999719111
Report :              precision    recall  f1-score   support

        0.0       1.00      1.00      1.00     85301
        1.0       0.84      0.76      0.80       142

avg / total       1.00      1.00      1.00     85443
```

Through the Gini Index we see that we have 20 features that best made the decision tree with 99.93% accuracy.

Results Using Entropy:

Predicted values:

[0. 0. 0. ... 0. 0. 0.]

Confusion Matrix: [[85273 28]

[34 108]]

Accuracy : 99.92743700478681

Report : precision recall f1-score support

0.0 1.00 1.00 1.00 85301

1.0 0.79 0.76 0.78 142

avg / total 1.00 1.00 1.00 85443


```
Results Using Entropy:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:  [[85273    28]
 [   34   108]]
Accuracy :  99.92743700478681
Report :              precision    recall  f1-score   support

        0.0       1.00      1.00      1.00     85301
        1.0       0.79      0.76      0.78       142

avg / total       1.00      1.00      1.00     85443
```


Using Entropy we see that out of the 29 features 28 were used for making the Decision tree with an accuracy of 99.93%.

4. CLUSTERING/SEGMENTATION APPROACH

Clustering on credit card fraud detection dataset has been implemented to divide the data points into different clusters so that the data points in the same clusters are more like other data points in the same clusters than those in other clusters. The point is to segregate groups with comparable traits and assign into clusters. The Significant variables identified from Principal Component Analysis will be used for cluster analysis.

KMeans Clustering implementation:

```
# Splitting the dataset into the Training set and Test set
ros = RandomOverSampler ()
X_ros, Y_ros = ros.fit_sample (X, y)

X_train, X_test, y_train, y_test = train_test_split(X_ros, Y_ros, test_size

# Using the elbow method to find the optimal number of clusters
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(X_train)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 42)
y_kmeans = kmeans.fit_predict(X_train)

plt.scatter(X_train[y_kmeans == 0, 9], X_train[y_kmeans == 0, 28], s = 100,
plt.scatter(X_train[y_kmeans == 1, 9], X_train[y_kmeans == 1, 28], s = 100,
"plt.scatter(X_train[y_kmeans == 2, 9], X_train[y_kmeans == 2, 28], s = 100,
plt.scatter(kmeans.cluster_centers_[:, 9], kmeans.cluster_centers_[:, 28], s
plt.title('Credit Card Fraud Dection')
plt.xlabel('Amount (k$)')
plt.ylabel('V9')
plt.legend()
plt.show()
```

Using the elbow method to find the optimal number of clusters. The Optimal number of clusters is 2.

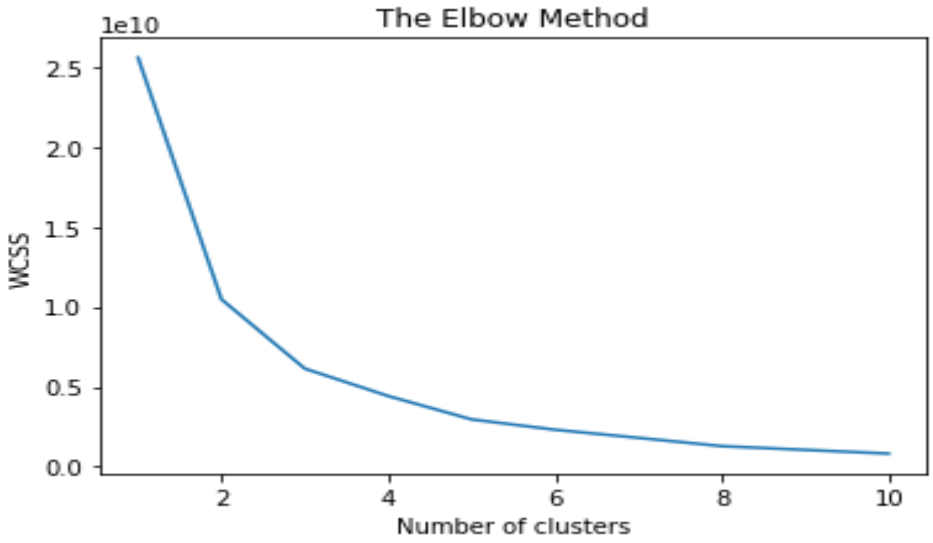


Figure 8: The Elbow Method

Cluster Visualization

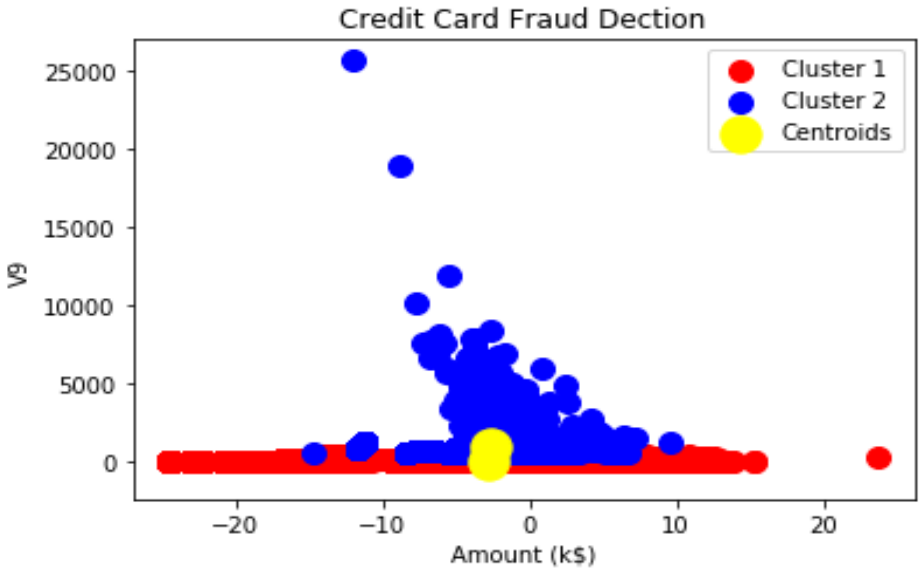


Figure 9: Scatter Plot

From the clusters analysis, we couldn't get much insights from the segregated data, since there is no business information regarding the variable "V9" available . On assuming the variable "V9" as different transactions occurred would predict one cluster as Normal Transaction and other as Fraud Transaction.

5. MODELLING

We will start working on different models and explore the accuracy of each model with a goal of choosing the best model. We will compare the confusion matrix and the accuracy of all the models.

5.1 Using model on Original data set:

5.1.1 Entropy Model

```
Results Using Entropy on original dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85275    26]
 [   35   107]]
Accuracy :  99.92860737567734
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.80      0.75      0.78       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.90      0.88      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

5.1.2 Gini Index Model

```
Results Using Gini Index on original dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85281    20]
 [   35   107]]
Accuracy :  99.93562960102057
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.84      0.75      0.80       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.92      0.88      0.90     85443
weighted avg       1.00      1.00      1.00     85443
```

5.1.3 Logistic Regression Model

```
Results Using Logistic Regression on Original dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85277    24]
 [   50    92]]
Accuracy :  99.9133925541004
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.79      0.65      0.71       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.90      0.82      0.86     85443
weighted avg       1.00      1.00      1.00     85443
```

5.1.4 SVM Model

```
Implementing SVM on original data set
Confusion matrix:
 [[8527100    3000]
 [   3800   10400]]
Report:
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.78      0.73      0.75       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.89      0.87      0.88     85443
weighted avg       1.00      1.00      1.00     85443

Accuracy:  99.9204147794436
```

5.2 Using Decision Tree (Entropy)

```
Implementing Decision Tree (Entropy) on MaxAbs preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85272    29]
 [   38   104]]
Accuracy :  99.92158515033414
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.78      0.73      0.76       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.89      0.87      0.88     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Implementing Decision Tree (Entropy) on MinMax preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85277    24]
 [   75    67]]
Accuracy :  99.88413328183702
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.74      0.47      0.58       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.87      0.74      0.79     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Implementing Decision Tree (Entropy) on Normalizer preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85283    18]
 [   48    94]]
Accuracy :  99.92275552122467
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.84      0.66      0.74       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.92      0.83      0.87     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Implementing Decision Tree (Entropy) on Scaler preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85274    27]
 [   34   108]]
Accuracy :  99.92860737567734
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.80      0.76      0.78       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.90      0.88      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Results Using Entropy on ROSE data:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[84238  1063]
 [   31   111]]
Accuracy :  98.71961424575449
Report :
              precision    recall  f1-score   support

         0.0       1.00      0.99      0.99     85301
         1.0       0.09      0.78      0.17       142

   micro avg       0.99      0.99      0.99     85443
   macro avg       0.55      0.88      0.58     85443
weighted avg       1.00      0.99      0.99     85443
```

```
Results Using Entropy on SMOTE data:
Predicted values:
[1. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[83571  1730]
 [   28   114]]
Accuracy :  97.9424879744391
Report :
              precision    recall  f1-score   support

         0.0       1.00      0.98      0.99     85301
         1.0       0.06      0.80      0.11       142

   micro avg       0.98      0.98      0.98     85443
   macro avg       0.53      0.89      0.55     85443
weighted avg       1.00      0.98      0.99     85443
```

5.3 Using Decision Tree (Gini)

```
Implementing Decision Tree (GINI) on MaxAbs preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85275    26]
 [   35   107]]
Accuracy :  99.92860737567734
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.80      0.75      0.78       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.90      0.88      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Implementing Decision Tree (GINI) on MinMax preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85278    23]
 [   61    81]]
Accuracy :  99.90168884519505
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.78      0.57      0.66       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.89      0.79      0.83     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Implementing Decision Tree (GINI) on Normalizer preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85288    13]
 [   86    56]]
Accuracy :  99.88413328183702
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.81      0.39      0.53       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.91      0.70      0.77     85443
weighted avg       1.00      1.00      1.00     85443


Implementing Decision Tree (GINI) on Scaker preprocessed data
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85277    24]
 [   35   107]]
Accuracy :  99.93094811745841
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.82      0.75      0.78       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.91      0.88      0.89     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Results Using Gini Index on ROSE data:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[84635   666]
 [   27   115]]
Accuracy :  99.1889329728591
Report :
              precision    recall  f1-score   support

         0.0       1.00      0.99      1.00     85301
         1.0       0.15      0.81      0.25       142

   micro avg       0.99      0.99      0.99     85443
   macro avg       0.57      0.90      0.62     85443
weighted avg       1.00      0.99      0.99     85443
```

```
Results Using Gini Index on SMOTE data:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[84635   666]
 [   27   115]]
Accuracy :  99.1889329728591
Report :
              precision    recall  f1-score   support

         0.0       1.00      0.99      1.00     85301
         1.0       0.15      0.81      0.25       142

   micro avg       0.99      0.99      0.99     85443
   macro avg       0.57      0.90      0.62     85443
weighted avg       1.00      0.99      0.99     85443
```

5.4 Logistic Regression

```
Results Using Logistic Regression on MinMax preprocessed dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85284    17]
 [   80    62]]
Accuracy :  99.88647402361809
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.78      0.44      0.56       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.89      0.72      0.78     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Results Using Logistic Regression on MaxAbs preprocessed dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85283    18]
 [   60    82]]
Accuracy :  99.90871107053826
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.82      0.58      0.68       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.91      0.79      0.84     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Results Using Logistic Regression on Scaler preprocessed dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85277    24]
 [   50    92]]
Accuracy :  99.9133925541004
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.79      0.65      0.71       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.90      0.82      0.86     85443
weighted avg       1.00      1.00      1.00     85443
```

```
Results Using Logistic Regression on Normalizer preprocessed dataset:
Predicted values:
[0. 0. 0. ... 0. 0. 0.]
Confusion Matrix:
 [[85290    11]
 [   84    58]]
Accuracy :  99.88881476539916
Report :
              precision    recall  f1-score   support

         0.0       1.00      1.00      1.00     85301
         1.0       0.84      0.41      0.55       142

   micro avg       1.00      1.00      1.00     85443
   macro avg       0.92      0.70      0.77     85443
weighted avg       1.00      1.00      1.00     85443
```

5.5 Comparing Model Accuracies:

Below is the table which shows the Accuracy we are getting from different models

	MINMAX	MAXABS	SCALER	NORMALIZER
GINI	99.9016	99.9286	99.9309	99.8841
ENTROPY	99.8841	99.9215	99.9286	99.9227
Logistic	99.8864	99.9087	99.91339	99.8888

Table 7: Comparison table

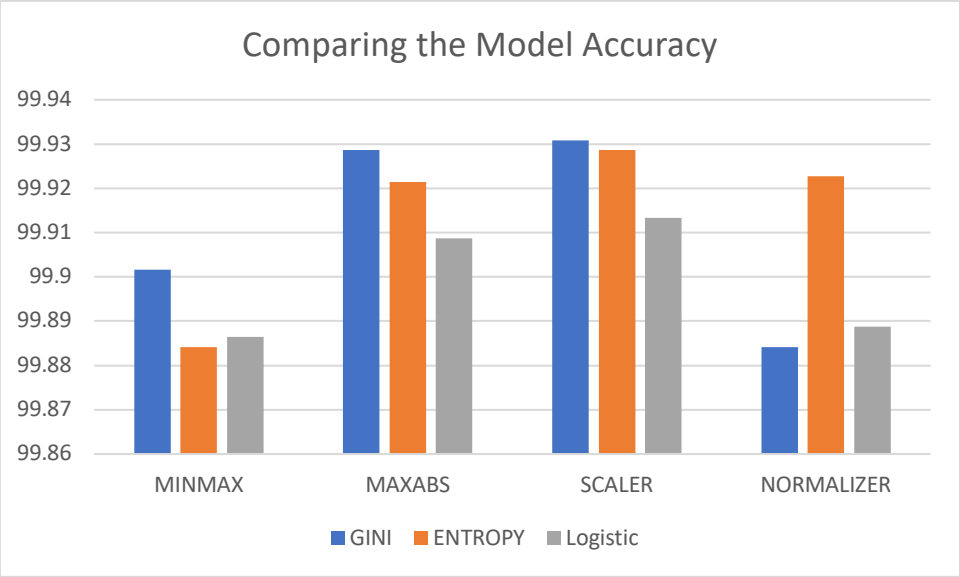


Figure 10.: Comparing the Model Accuracy

6. IMPLEMENTATION:

6.1 Methodology Overview:

CRISP-DM methodology or the Cross Industry Standard Process for Data mining is the methodology we have decided to implement in out project. A CRISP-DM goes through 6 phases however, the sequence of the phases moves back and forth as the project grows. The diagram below showcases the flow of a CRISP-DM methodology.

CRISP-DM procedure or the Cross Industry Standard Process for Data mining is the approach we have chosen to implement in our project. A CRISP-DM experiences 6 stages be that as it may, the succession of the stages move forward and backward as the venture develops. The outline beneath grandstands the stream of a CRISP-DM methodology.

***Figure 11**: CRISP-DM phase process flow*

The outcome of each phase in the diagram usually helps in determining which task should be performed next or which phase to move onto post completion of the current phase. The outermost loop references the cyclical flow of data mining process. A process of data mining does not end once the deployment of a solution has been done. The outcomes for the deployment might possibly give rise to new more business focused problems or questions. The process becomes more learning and beneficial through this cyclic process.

6.1.1 Business Understanding:

A CRISP-DM process usually begins with the phase of "Business Understanding" where in which we are required to understand the objectives and needs of the application from a business perspective. Once we have acquired this knowledge, we are required to transform this knowledge into a data definition and a rudimentary data mining plan to help us achieve the objective. In our case, our objective is to utilize the available data about credit card transactions from the people of Europe and predict how many frauds have occurred in that period of time. We can also analyze the trends of when there is likely to be a fraud transaction. And accordingly, we can take actions to prevent that from happening.

6.1.2 Data Understanding:

For this, we will need to move to the next phase of "Data Understanding" in which we collected data and proceeded with getting accustomed to the data obtained, figuring out the problems with our data, like if

there are any missing values or not and understand the response that we are determined to obtain. We further can understand the season change of transactions happening. These seasonal changes could be transactions happening during holiday season or during sales. Also, understanding chances of a transaction being a fraudulent by amount and locations from where transactions are been made.

6.1.3 Data Preparation:

Post this we move onto the phase of "Data Preparation". In this phase we will build the final dataset which will be fed to modelling tools. For Data preparation, we will be performing data cleansing, taking care of missing values and transforming data to get a normalized data set. Until a couple of iteration understanding the variance of data and best preprocessing steps could be best practice. From previous step of 'Understanding Data' we would get a trend and understanding of data better. After which systematic approach of using preprocessing techniques can be implemented. Like MaxAbs preprocessing technique best work on sparse data and MinMax to scale data as per maximum and minimum of data. Based on the data descriptiveness a preprocessing technique needs to be used.

6.1.4: Modelling:

From Data preparation we move onto "Modelling" in which various Data Modelling techniques are applied on the dataset and the parameters for those techniques will be calibrated to optimum values. Various Modelling techniques can be applied to the same problem, but some techniques might have special requirements on the type of data and hence, which requires to go back to the data preparation stage. Here, we have used various models like logistic, decision tree and support vector machines. These methods have shown good accuracy with the current sample dataset however when data increases, we truly would understand which out of these is a better model.

6.1.5 Evaluation:

Post modelling, we move to "Evaluation" in which we will build a model with the highest quality in terms of data analysis for making predictions. Before we finalize the model, it is essential that we evaluate the models and the steps to build it thoroughly to make sure that it helps us achieve our business requirement. With model selected and build, we can evaluate model on number of parameters like accuracy, confusion matrix, ROC curve. We can keep a threshold of 95% accuracy in cases to evaluate the model and incase threshold isn't met we can try with different model or change some preprocessing steps.

6.1.6 Deployment:

Finally, a model has been finalized through evaluation we can go ahead and move to the phase of "Deployment" which phase can help us make predictions, create reports of even do data analysis. Through our deployment we should be able to run the model on real time data to ensure that between the pending phase of transaction to final stage, our model should be able to decide if the transaction is normal or fraudulent. This is help create an extra layer of security for consumers and prevent monetory loss.

7. CONCLUSION AND FUTURE STEPS:

7.1. Conclusions:

Different classifications models have been built and have accuracy as the metric to compare these models. Multiple data preprocessing technique has been implemented to figure out the model with better accuracy and efficiency. We found the accuracy using confusion matrix and couldn't conclude the best fit model since accuracy for most of the models vary only by minute difference and have the average accuracy rate as 99.86, which aligns with the accuracy of most of the top kernels in Kaggle for Credit card fraud detection Dataset. The processing run-time for each model could be considered using a large data set to identify the model with better efficiency. Additional business Information regarding the 'V1-V28' variables should have paved way for multiple findings to detect fraud transaction and better insight from Cluster Analysis. Lack of Business Information for few variables is the major limitation of this Project.

7.2. Potential Improvements or Future Work:

1. Anomaly detection using Isolation Forest Algorithm to identify out-of-the-ordinary transactions.
2. Using the Time and amount Variables different patterns could be identified. The resulting patterns could identify the fraudulent transactions based on the timeline the transaction occurred.
3. Feedforward neural network could be implemented to detect fraudulent transactions.
4. In-depth analysis using Time could Identify short span multiple Transaction patterns for predicting fraud transaction could be developed.
5. This model been built for European market and with success, can be implemented for other countries too.
6. From a business scenario, this will add as a layer of security over each transaction and prevent any loss to consumer or banking institutes.
7. Further, model should be re-evaluated with any new market changes with respective to credit card changes. This will ensure that anomaly are detected and prevent any new fraudulent charges.
8. This can be a machine learning problem, where with every transaction the prediction for a transaction may get better and result in providing better accuracy.