A Capstone Project Report on

PREDICTION OF FRADULENT TRANSACTIONS

Submitted by

RAGHURAJ S

ABSTRACT

The fraudulent transaction is a wide-ranging issue for financial institutions, involving theft and fraud committed using a payment gateway. In this report, we explore the application of machine learning models on real fraudulent transaction data. The models built are supervised fraud models that attempt to identify which transactions are most likely fraudulent. Here I have done two different types of iteration. In the first iteration, we discuss the processes of data exploration, data cleaning, checking outliers, balancing data, model algorithms, and results. Three different supervised models are explored and compared including logistic regression, random forest, decision tree, and GridSearchCV. The GridSearchCV model shows the best accuracy for logistic regression, which is known as the Hyperparameter tuning model. In the second iteration, I used MYSQL workbench to export the dataset and used POWERBI for visualization of the given dataset. A similar model development process can be performed in related business domains such as insurance and banking sectors, to avoid or detect fraudulent activity.

Dataset Link:

https://drive.google.com/uc?export=download&confirm=6gh6&id=1VNpyNkGxHdskfdTNRSjjyNa5qC9u0JyV

Source Link:

https://github.com/RaghurajSenthil/ML-Projects-on-Python/tree/main/CAPSTONE_PROJECT

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of my capstone project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

Further, I have fortunate to have Mr. PRASAD as my mentor. He has readily shared his immense knowledge in data analytics and guide me in a manner that the outcome resulted in enhancing my data skills.

I certify that the work done by me for conceptualizing and completing this project is original and authentic.

Date: July 10, 2022 Name: RAGHURAJ S

Certificate of Completion

I hereby certify that the project titled “Prediction of Fraudulent Transactions” was undertaken and completed the project (10th July, 2022).

Mentor : Mr. Prasad

Date : 10th July, 2022

Place : Karur

TABLE OF CONTENTS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

For a long time, fraudulent transactions and detectors have had a composite role. Fraudulent transactions are happening more frequently than ever before, principally in today's era of the Internet, and it is the cause of foremost financial losses. To overcome this problem in recent days, many are developing different types of classification models. Here I have attached two different types of models. Each model consists of three to four classification models.

In Method I, the dataset is imported directly from the internet. Basic EDA (Exploratory Data Analysis) can be taken place. And then the process of feature selection can be done. Feature selection includes three models namely Logistic Regression, Random Forest Classifier, and Decision Tree Classifier.

In Method II, the dataset is imported from the MYSQL workbench, where the interfacing of the MYSQL and Jupyter notebook takes place. Here feature selection also takes place and then GridSearchCV can be used for the hyperparameter tuning for the Logistic Regression model. And at last, Visualization of the dataset can be done using the POWER BI tool, which is also interfaced with the Jupyter notebook. Finally, a Model comparison can be done for both Models.

METHOD I

CHAPTER 2

DATA ANALYSIS AND PREPARATION

The Dataset is taken from the internet, which contains very large data. The number of rows and columns presents are totally of 6362620 rows and 11 columns.

Let's see the detailed view of the Dataset:

The name of the Columns present in the Dataset:

Step - maps a unit of time in the real world. In this case, 1 step is 1 hour. Total steps 744 (30 days simulation).

Type - CASH-IN, CASH-OUT, DEBIT, PAYMENT, and TRANSFER.

Amount - Amount of the transaction in local currency.

NameOrig - the customer who started the transaction.

OldbalanceOrg - initial balance before the transaction.

NewbalanceOrig - new balance after the transaction.

NameDest - the customer who is the recipient of the transaction.

OldbalanceDest - initial balance recipient before the transaction.

NewbalanceDest - new balance recipient after the transaction.

Is Fraud - This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset, the fraudulent behavior of the agents aims to profit by taking control of customers' accounts and trying to empty the funds by transferring them to another account and then cashing out of the system.

IsFlaggedFraud - The business model aims to control massive transfers from one account to another and flags illegal attempts.

2.1 UNDERSTAND THE DATASET

Importing the Dataset from Internet:

Read the data using read_csv() function from pandas

```
[ ]  path = r"https://drive.google.com/uc?export=download&confirm=6gh6&id=1VNpyNkGxHdskfdTNRSjjyNa5qC9u0JyV"
     df = pd.read_csv(path)
```

Here the dataset is present on internet, as the size of the dataset is very large, so it is imported directly from the internet using pandas.

The imported data has been viewed using head () and it shows the data that is present in the dataset. (i.e. Name of the columns and values present in the rows)

```
df.head()
```

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

Figure 2.1.1: Dataset

Data types:

The data types present in the Pandas data frames are mostly object, float and int64.It is important to know about the datatype of the data's that is present in the given dataset.

```
#Datatypes of the data present in the given Dataset
df.dtypes
```

```
step              int64
type             object
amount          float64
nameOrig         object
oldbalanceOrg   float64
newbalanceOrig  float64
nameDest         object
oldbalanceDest  float64
newbalanceDest  float64
isFraud           int64
isFlaggedFraud    int64
dtype: object
```

Figure 2.1.2: Datatypes

Statistical Summary:

Here we take a look at the summary of each attribute. This includes the count, mean, the min and max values as well as some percentiles for numeric variables.

```
[ ] df.describe(include='all')
```

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
count	6.362620e+06	6362620	6.362620e+06	6362620	6.362620e+06	6.362620e+06	6362620	6.362620e+06	6.362620e+06	6.362620e+06	6.362620e+06
unique	NaN	5	NaN	6353307	NaN	NaN	2722362	NaN	NaN	NaN	NaN
top	NaN	CASH_OUT	NaN	C1902386530	NaN	NaN	C1286084959	NaN	NaN	NaN	NaN
freq	NaN	2237500	NaN	3	NaN	NaN	113	NaN	NaN	NaN	NaN
mean	2.433972e+02	NaN	1.798619e+05	NaN	8.338831e+05	8.551137e+05	NaN	1.100702e+06	1.224996e+06	1.290820e-03	2.514687e-06
std	1.423320e+02	NaN	6.038582e+05	NaN	2.888243e+06	2.924049e+06	NaN	3.399180e+06	3.674129e+06	3.590480e-02	1.585775e-03
min	1.000000e+00	NaN	0.000000e+00	NaN	0.000000e+00	0.000000e+00	NaN	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.560000e+02	NaN	1.338957e+04	NaN	0.000000e+00	0.000000e+00	NaN	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
50%	2.390000e+02	NaN	7.487194e+04	NaN	1.420800e+04	0.000000e+00	NaN	1.327057e+05	2.146614e+05	0.000000e+00	0.000000e+00
75%	3.350000e+02	NaN	2.087215e+05	NaN	1.073152e+05	1.442584e+05	NaN	9.430367e+05	1.111909e+06	0.000000e+00	0.000000e+00
max	7.430000e+02	NaN	9.244552e+07	NaN	5.958504e+07	4.958504e+07	NaN	3.560159e+08	3.561793e+08	1.000000e+00	1.000000e+00

Figure 2.1.3: Statistical Summary

The above statistical summary represents the values that is present in each of the rows and the columns. This includes the values like count, mean, maximum, minimum and percentile values for the numerical variables.

The numeric variables like the mean, median (50%), frequency, top and unique values, along with the standard deviation.

```
# data frame with categorical features
df.describe(include='object')
```

	type	nameOrig	nameDest
count	6362620	6362620	6362620
unique	5	6353307	2722362
top	CASH_OUT	C1902386530	C1286084959
freq	2237500	3	113

Checking Duplicates:

```
[ ]  df.duplicated().sum()

     0
```

Figure 2.1.4 Duplicates

Here, the dataset doesn't contain any duplicated values.

Checking Missing Values:

```
[ ]  # checking for missing values

     df.isna().sum().sort_values(ascending=False)

     step              0
     type              0
     amount            0
     nameOrig          0
     oldbalanceOrg     0
     newbalanceOrig    0
     nameDest          0
     oldbalanceDest    0
     newbalanceDest    0
     isFraud           0
     isFlaggedFraud    0
     dtype: int64
```

Figure 2.1.5: Missing values

Here, for finding the missing values, we used the isna().sum() function. The given dataset doesn't contain any missing values.

2.2 VISUALIZATION

Count plot:

The count plot () is the method is used to show the counts of observations in each categorical bin using bars. In the given dataset, the column ‘type’ is known as the categorical value. The count plot () is used for the categorical values only. The count plot of the given dataset can be plotted as follows,

```
# counting the number of transactions per type
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10,8))
sns.countplot(x="type", data=df,hue="isFraud" , palette="Set2")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f852ba64d90>
```

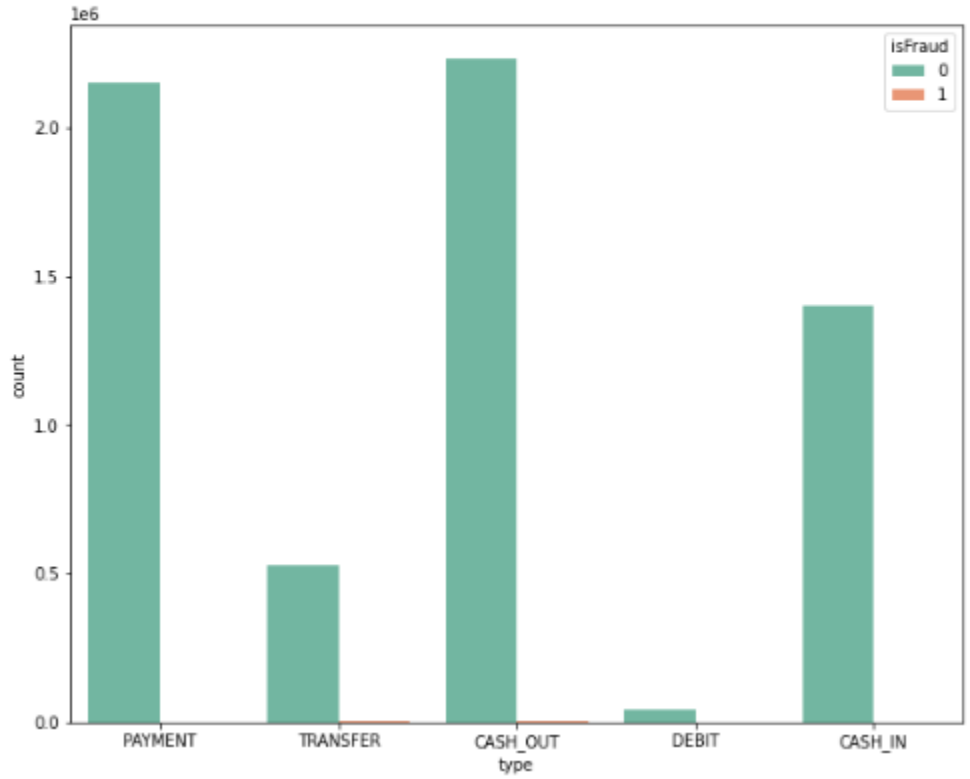


Figure 2.2.1: Count plot

Distplot:

Python Seaborn module contains various functions to plot the data and depict the data variations. The seaborn.distplot () function is used to plot the distplot. The distplot represents the univariate distribution of data i.e. data distribution of a variable against the density distribution.

```
#displot---- distplot represents the univariate distribution of data
positive_case = df[df['isFraud']==1]
sns.distplot(positive_case['amount'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f85151a8f90>
```

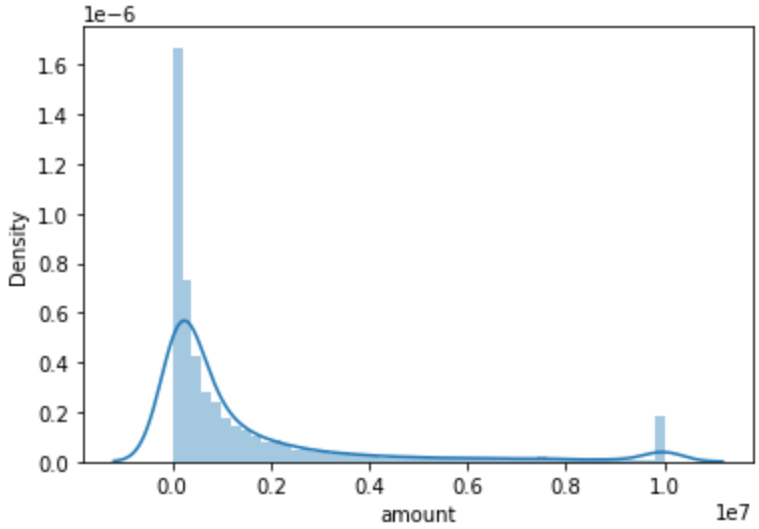


Figure 2.2.2: Distplot

The Distplot function plots the univariate data distribution of a variable against the density distribution.

Histogram:

A histogram is basically used to represent data provided in a form of some groups. It is accurate method for the graphical representation of numerical data distribution. It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

```
#Histogram
import matplotlib.pyplot as plt
fig=df.hist(figsize=(10,10),color='lightblue')
plt.show()
```

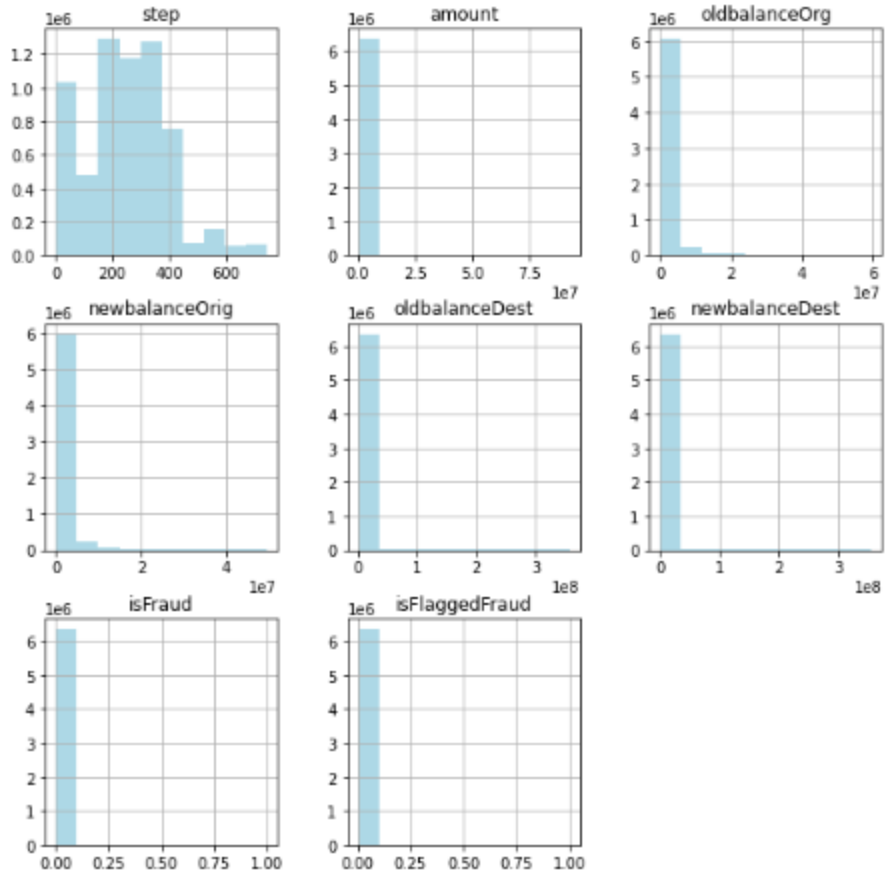


Figure2.2.3: Histogram

Heat map:

Heat map is defined as a graphical representation of data using colors to visualize the value of the matrix. In this, to represent more common values or higher activities brighter colors basically reddish colors are used and to represent less common or activity values, darker colors are preferred. Heat map is also defined by the name of the shading matrix. Heat maps in Seaborn can be plotted by using the seaborn.heatmap () function.

```
#Heatmap
f,ax = plt.subplots(figsize=(10, 10))
sns.heatmap(corr, annot=True, linewidths=.5, fmt= '.1f',ax=ax)
plt.show()
```

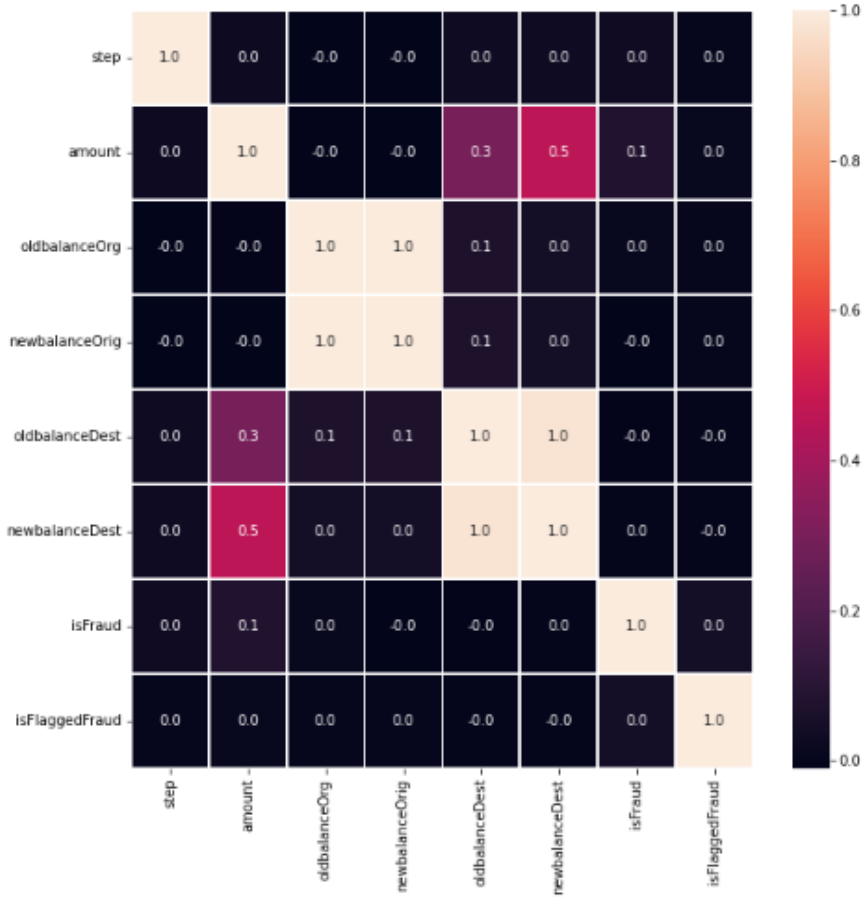


Figure 2.2.4: Heat map

2.3 CHECKING OUTLIERS

The outliers are the extreme values within the dataset. That means the outlier data points vary greatly from the expected values—either being much larger or significantly smaller.

```
# plot a boxplot to visualize the outliers in all the numeric variables
df.boxplot()

# set plot label
# set text size using 'fontsize'
plt.title('Distribution of all Numeric Variables', fontsize = 15)

# xticks() returns the x-axis ticks
# 'rotation = vertical' rotates the x-axis labels vertically
plt.xticks(rotation = 'vertical', fontsize = 15)

# display the plot
plt.show()
```

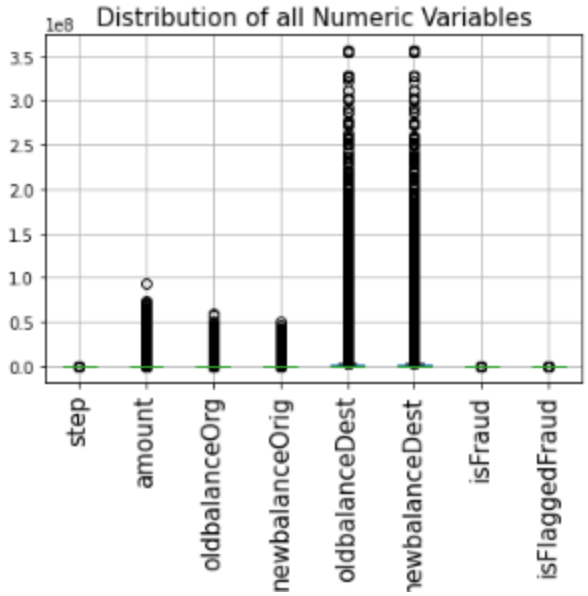


Figure 2.3.1: Outliers

2.4 BALANCING DATA:

SMOTE is an oversampling technique where the synthetic samples are generated for the minority class. This algorithm helps to overcome the overfitting problem posed by random oversampling.

SMOTE (Synthetic Minority Oversampling Technique) for balancing data

```
 #splitting features and the target variable
# consider all the columns except 'target' using 'iloc'
df_features = df.iloc[:, df.columns != 'isFraud']
# consider the target variable
df_target = df.iloc[:,df.columns == 'isFraud']
```

```
[ ]  # get counts of 0's and 1's in the 'target' variable using 'value_counts()'
     # store the values in 'class_frequency'
     class_frequency = df_target.isFraud.value_counts()
     class_frequency
```

```
0    6354407
1       8213
Name: isFraud, dtype: int64
```

Using SMOTE we can tweak the model to reduce false negatives, at the cost of increasing false positives. The result of using SMOTE is generally an increase in recall, at the cost of lower precision.

The comparison between the dataset before and after SMOTE analysis will be shown as follows,

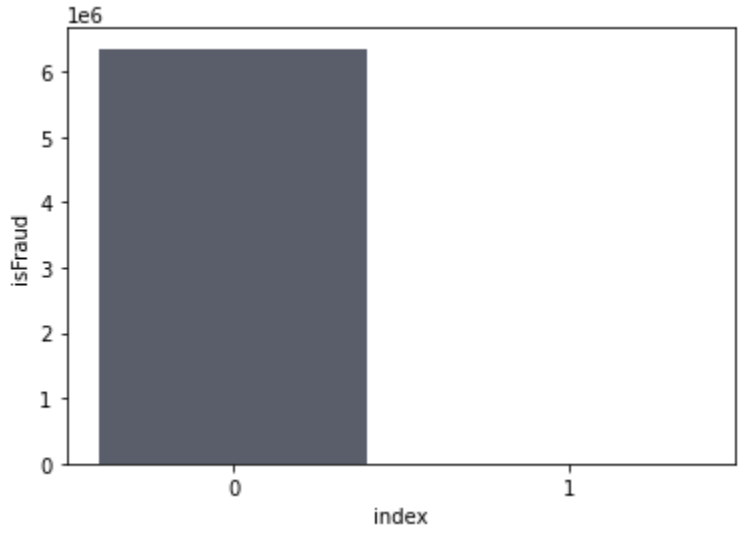


Figure 2.4.1: Before SMOTE analysis

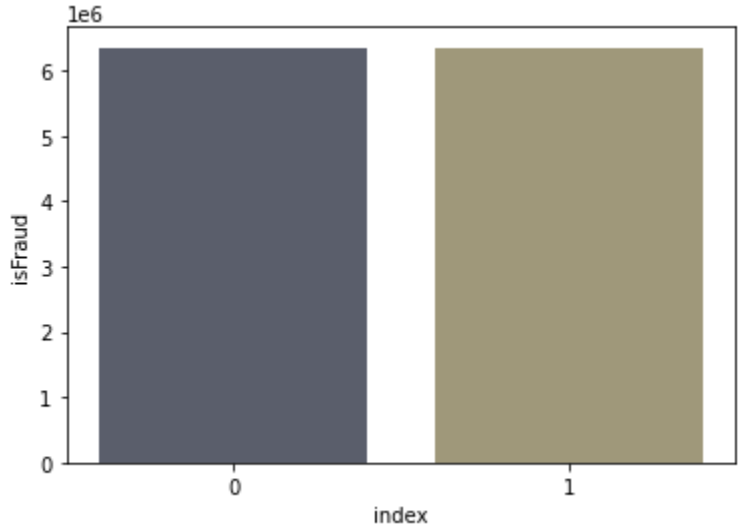


Figure 2.4.2: After SMOTE analysis

CHAPTER 3

FEATURE SELECTION

Feature Selection is the method of reducing the input variable to a model by using only relevant data and getting rid of noise in data. It is the process of automatically choosing relevant features for the machine learning model based on the type of problem we are trying to solve.

In this project we used three different models such as Logistic Regression, Random Forest Classifier and Decision Tree Classifier.

3.1 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

The given fraudulent dataset gives the following output as the accuracy score, precision, recall and f1 score for Logistic Regression,

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.935495	0.965678	0.903116	0.935481	0.93335

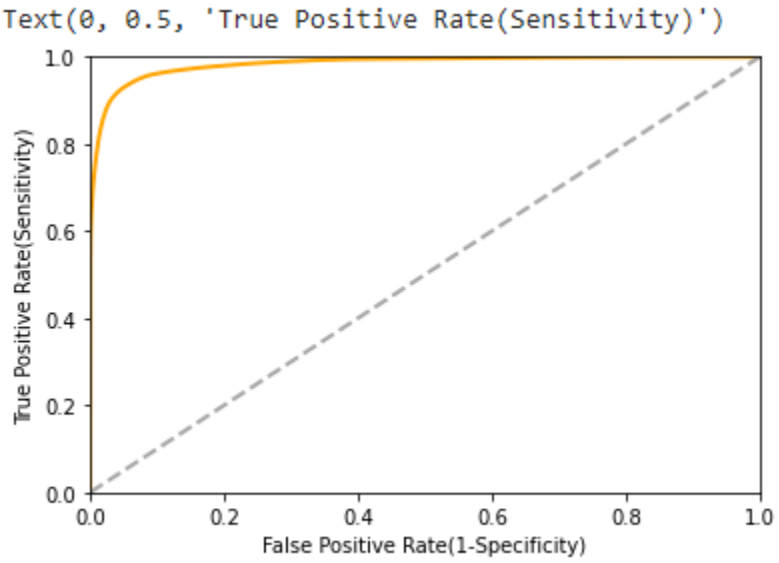


Figure 3.1.1: LogisticRegression

Area under the curve is **0.9836535330633818**

3.2 Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The given fraudulent dataset gives the following output as the accuracy score, precision, recall and f1 score for Random Forest Classifier,

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.935495	0.965678	0.903116	0.935481	0.933350
1	RandomForestClassifier	0.999598	0.999267	0.999931	0.999598	0.999599

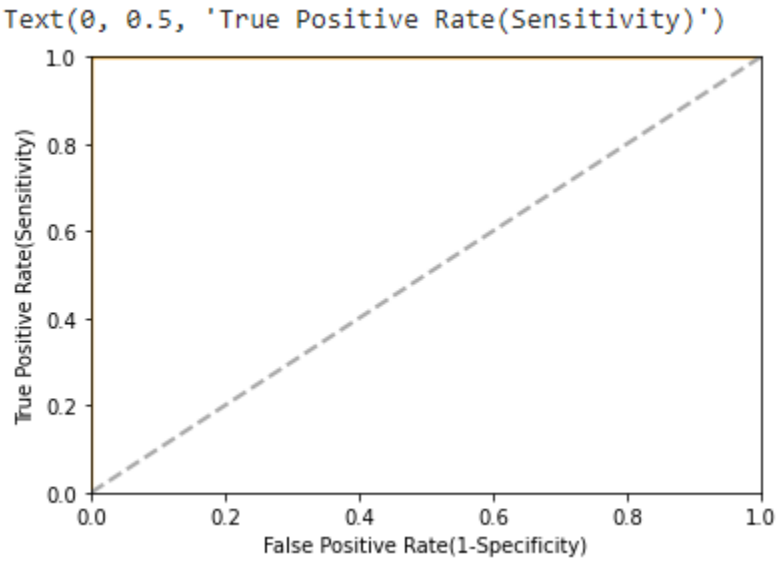


Figure 3.2.1: RandomForestClassifier

Area under the curve is **0.9999524140816728**

3.3 Decision Tree Classifier

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.935495	0.965678	0.903116	0.935481	0.933350
1	RandomForestClassifier	0.999598	0.999267	0.999931	0.999598	0.999599
2	DecisionTreeClassifier	0.999595	0.999392	0.999800	0.999596	0.999596

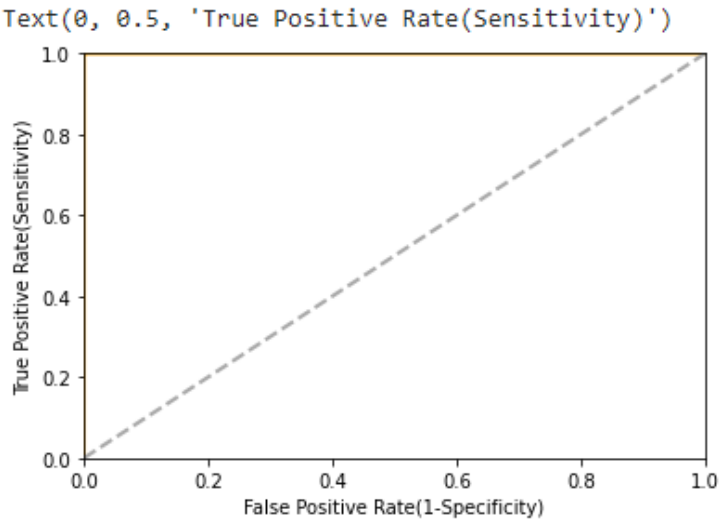


Figure 3.3.1: DecisionTreeClassifier

Area under the curve is **0.9999524140816728**

CHAPTER 4

MODEL COMPARISION

By comparing all the three models, we conclude that the accuracy score for the Logistic Regression is low when compared with other two models, the Random Forest Classifier and the Decision Tree Classifier are the two models which gives the better accuracy score than the Logistic Regression.

As a result, the comparison between the three models can be shown as follows,

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.935495	0.965678	0.903116	0.935481	0.933350
1	RandomForestClassifier	0.999598	0.999267	0.999931	0.999598	0.999599
2	DecisionTreeClassifier	0.999595	0.999392	0.999800	0.999596	0.999596

From the above results table, we conclude that **the accuracy score for the given dataset is more ,when using RandomForestClassifier and DecisionTreeClasifier** classification models.

Figure 4.1: Model Comparison

The Random Forest Classifier and the Decision Tree Classifier gives the better Accuracy score, Precision score, Recall value and f1-score.

METHOD II

In method II, the Jupyter notebook is used for the interfacing of MYSQL workbench and the POWER BI tool.

This Method is done where the visualization of the given dataset can be made very clearly and in the efficient manner.

In method II, after importing the dataset from MYSQL workbench, the size of the dataset decreases so only some small amount data only used. Totally, 22556 rows and 11 columns only used here.

CHAPTER 5

IMPORT PACKAGES FOR MYSQL INTERFACE

Here, for the interfacing of the MYSQL in the Jupyter notebook, it requires some of the packages and the libraries which is used to fetch the dataset from MYSQL workbench.

```
1 #Pakages required for MYSQL Interfacing
2 !pip install imblearn
3 !pip install mysql
```

These are the packages which is used to interface the MYSQL workbench with the Jupyter notebook.

CHAPTER 6

FETCHING DATASET FROM MYSQL WORKBENCH

While importing the dataset from MYSQL to Jupyter it is required to install some of the packages and the libraries.

```
!pip install mysql-connector-python
```

```
import mysql.connector as mysql
```

```
db=mysql.connect(host="localhost",user="root",password="Raghu@123",database="Capstone", auth_plugin='mysql_native_password')
```

```
print(db)
```

```
myc=db.cursor()
myc=db.cursor(buffered=True, dictionary=True)
```

```
myc.execute('SELECT * FROM capstone.`fraud(1)`')
```

Figure 6.1: MYSQL Library

These are the required libraries for importing the dataset from the MYSQL workbench, where the dataset is already stored in the localhost of the MYSQL workbench, we need to fetch the dataset.

After importing the dataset from the MYSQL workbench, the dataset that is present in the workbench can be shown as follows,

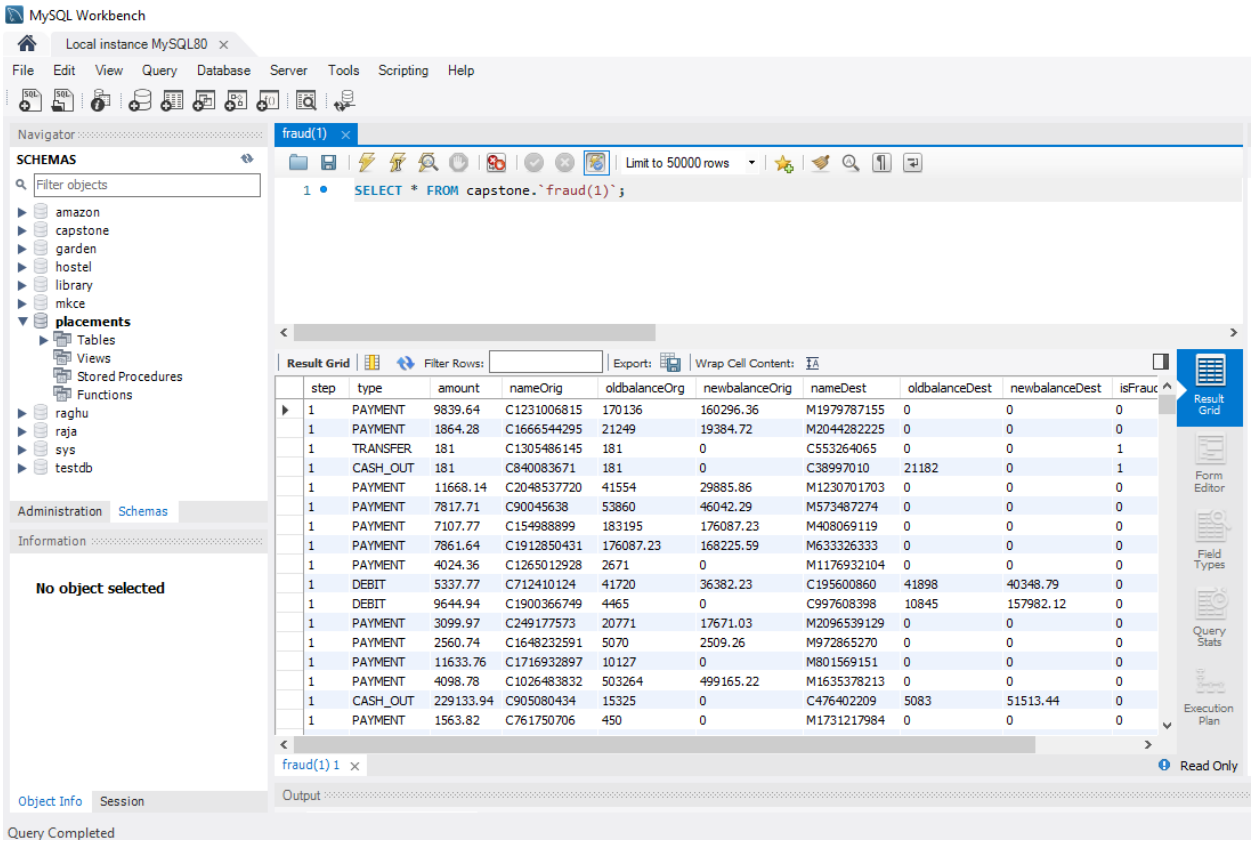


Figure 6.2: Dataset in MYSQL

The above image is the dataset in the MYSQL workbench which is imported to Jupyter notebook. As we need to prepare the data for training and testing.

The size of the dataset can be small when compared to the method I, because while importing dataset from the MYSQL workbench, it is possible to obtain only 22556 rows and 11 columns. Here the column remains same.

CHAPTER 7

FEATURE ENGINEERING

7.1 Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.

Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

The given fraudulent dataset gives the following output as the accuracy score, precision, recall and f1 score for Logistic Regression,

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.863992	0.852335	0.881392	0.864033	0.86662

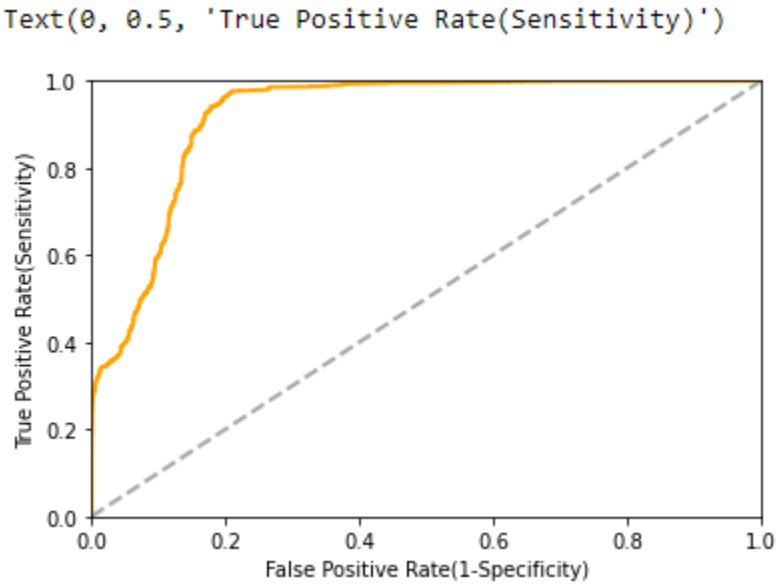


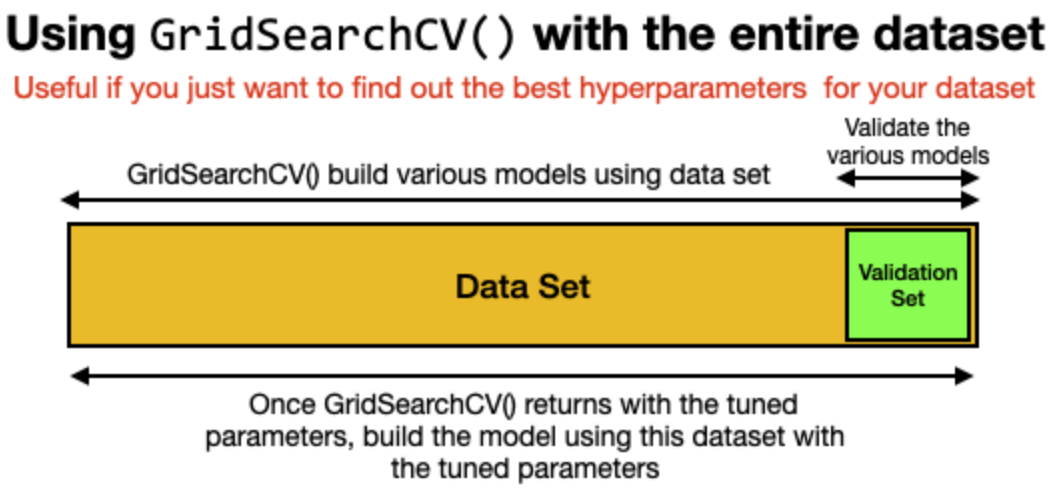
Figure 7.1.1: LogisticRegression

Area under the curve is **0.920592511473097**

7.2 Hyperparameter tuning using Gridsearchcv

GridSearchCV is a function that is in sklearn's model selection package. It allows you to specify the different values for each hyperparameter and try out all the possible combinations when fitting your model. It does the training and testing using cross validation of your dataset — hence the acronym "CV" in GridSearchCV.

The end result of GridSearchCV is a set of hyperparameters that best fit your data according to the scoring metric that you want your model to optimize on.



After using GridSearchCV in Logistic Regression, it shows the tuned hyperparameters, and it helps to improve the accuracy score of the given Logistic Regression.

```
tuned hpyerparameters :(best parameters)  {'C': 1000.0, 'penalty': 'l2'}
accuracy : 0.9279188585355588
```

Here, the usage of the GridSearchCV is only to improve the accuracy score of he given dataset when the Logistic Regression is used. Therefore, the accuracy score increased when compared to the Logistic Regression.

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.863992	0.852335	0.881392	0.864033	0.86662
1	GridSearchCV	-	-	-	0.927919	-

Figure 7.2.1: GridSearchCV

7.3 Random Forest Classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.863992	0.852335	0.881392	0.864033	0.86662
1	GridSearchCV	-	-	-	0.927919	-
2	RandomForestClassifier	0.998843	0.99858	0.999112	0.998843	0.998846

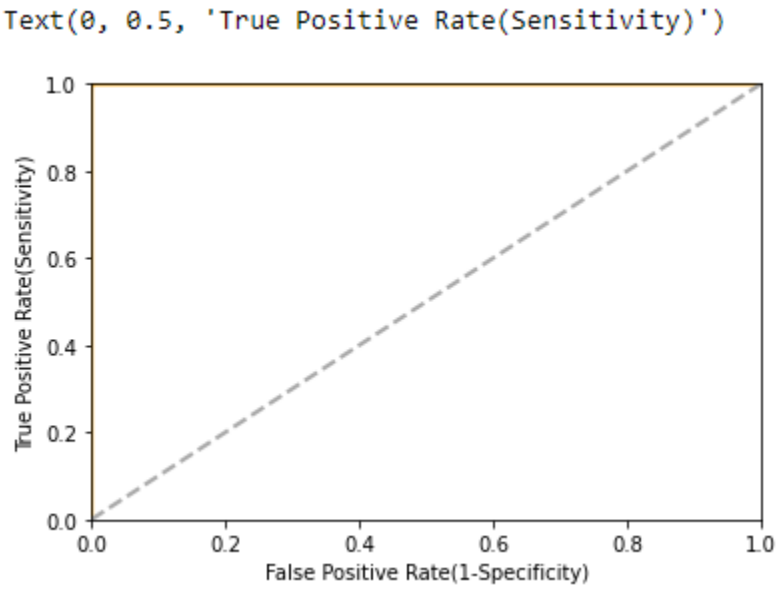


Figure 7.3.1: RandomForestClassifier

Area under the curve is **0.999904823717689**

7.4 Decision Tree Classifier

Decision Tree is a supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.863992	0.852335	0.881392	0.864033	0.86662
1	GridSearchCV	-	-	-	0.927919	-
2	RandomForestClassifier	0.998843	0.99858	0.999112	0.998843	0.998846
3	DecisionTreeClassifier	0.996614	0.994694	0.99858	0.996619	0.996633

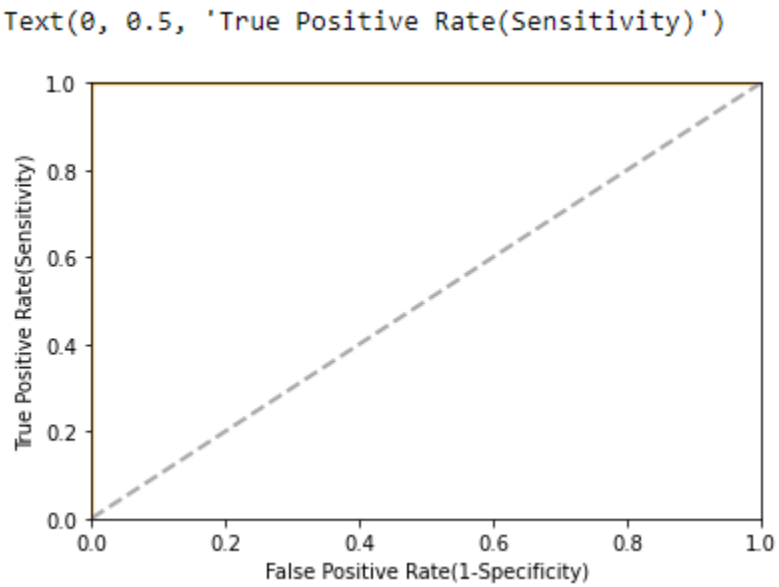


Figure 7.4.1: DecisionTreeClassifier

Area under the curve is **0.999904823717689**

CHAPTER 8

MODEL COMPARISON

By comparing all the four models, we conclude that the accuracy score for the Logistic Regression is low, in order to improve the accuracy score we have used the hyperparameter tuning method, the tuning method is known as the GridSearchCV.

After using the GridSearchCV tuning method we get the improved accuracy score, when compared with Logistic Regression. Let's see the comparison of the models that we done as follows,

	Model	AUC Score	Precision Score	Recall Score	Accuracy Score	f1-score
0	Logistic_Regression	0.863992	0.852335	0.881392	0.864033	0.86662
1	GridSearchCV	-	-	-	0.927919	-
2	RandomForestClassifier	0.998843	0.99858	0.999112	0.998843	0.998846
3	DecisionTreeClassifier	0.996614	0.994694	0.99858	0.996619	0.996633

From the above results table, we conclude that **the accuracy score for the given dataset is more ,when using RandomForestClassifier and DecisionTreeClasifier** classification models.

Figure 8.1: Model Comparision

Therefore, the model comparison between the four models that we have used are compared successfully.

CHAPTER 9

IMPORT PACKAGES FOR POWER BI INTERFACE

Power BI is an interactive data visualization software product. Power BI is a collection of software services, apps, and connectors that work together to turn unrelated sources of data into coherent, visually immersive, and interactive insights. Data may be input by reading directly from a database, webpage, or structured files such as spreadsheets, CSV, XML, and JSON.

The Packages that are used for interfacing the Power BI with the Jupyter notebook can be shown as,

```
pip install powerbiclient
```

```
#import libraries
from powerbiclient import Report, models
```

```
# Import the DeviceCodeLoginAuthentication class to authenticate against Power BI
from powerbiclient.authentication import DeviceCodeLoginAuthentication

# Initiate device authentication
device_auth = DeviceCodeLoginAuthentication()
```

Figure 9.1: Power BI Library

After successfully Authenticated, we need to open the following link for the visualization of the power BI, “To sign in, use a web browser to open the page **https://microsoft.com/devicelogin** and enter the code **HYUNPAJ4L** to authenticate”.

CHAPTER 10

POWER BI VISUALIZATION

The visualization of the Power BI report needs some codes, so the coding part of the Power BI includes the following codes,

```
group_id="c022f164-e173-4892-850c-e67976ca07a1"
report_id="a36e8346-254c-42ad-80de-ed5ce1486e39"
report = Report(group_id=group_id, report_id=report_id, auth=device_auth)

report
```

POWER BI VISUALIZATION

Figure 10.1: Power BI Report

CHAPTER 11

CONCLUSION

I have evaluated the fraudulent dataset in two different methods. The methods that are used for the prediction of fraudulent transactions are of the same kind only the difference is importing the dataset. In the method I, the dataset is directly taken from the internet and the taken dataset is a very large dataset. The handling of the large dataset is quite harder because I have faced many struggles while testing and training the dataset. After testing and training were done, the fitting of the dataset is very tough, because it takes more time for running the cell. In method II, while importing the dataset from the MYSQL workbench, the size of the dataset becomes small, and therefore here it is easy to test, train and fit the dataset for classification models. Also, I have interfaced the Power BI visualization with the Jupyter notebook. Further, I am working with the Py-spark tool for the handling of the larger dataset.

CHAPTER 12

REFERENCE

[1] Maruti Tech Labs: How Machine Learning facilitates fraud detection https://ww4aw.marutitech.com/machine-learning-fraud-detection.

[2] Diwakar Tripathia, Bhawana Nigamb, Damodar Reddy Edlaa (2017) "A Novel Web Fraud Detection Technique using Association Rule Mining ", 7th International Conference on Advances in Computing & Communications, ICACC-2017.

[3] Zohreh Darbandian, Alimohammad Latif , Sima Emadi "The discovery of the credit card transactions suspicious of fraud using supervised data-mining methods"(2016)International Journal Of Humanities And Cultural Studies ISSN 2356-5926

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

[5]https://www.kaggle.com/code/naveenkonam1985/fradulent-transactions-prediction.

[6]https://www.kaggle.com/code/chaurasiya123/fraudlent-transaction-prediction.