A PROJECT REPORT ON

CREDIT CARD EMBEZZLEMENT USING MACHINE LEARNING

Major project submitted in partial fulfilment to of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY
(2018 - 2022)

BY

D. Karthik	18241A1275
J. Ganesh	18241A1282
N. Sai Sujith	18241A1298
G. Lokesh Reddy	18241A1281
V. Sharath	18241A12B7

Under the Esteemed guidance of

Deepika Borgaonkar

Asst Prof, Dept of IT

DEPARTMENT OF INFORMATION TECHNOLOGY
GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY
(AUTONOMOUS)
HYDERABAD

CERTIFICATE

This is to certify that it is a bonafide record of Major Project work entitled **"CREDIT CARD EMBEZZLEMENT USING MACHINE LEARNING"** done by **D. Karthik (18241A1275), J. Ganesh (18241A1282), N. Sai Sujith (18241A1298), G. Lokesh Reddy (18241A1281)** and **V. Sharath (18241A12B7)** students of **B.Tech (IT)** in the Department of Information Technology, Gokaraju Rangaraju Institute of Engineering and Technology during the period 2018-2022 in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from GRIET, Hyderabad.

Deepika Borgaonkar
(Internal Project Guide)

Dr. N. Ganapathi Raju
(Head of the Department)

(Project External)

ACKNOWLEDGEMENT

We take immense pleasure in expressing gratitude to our Internal guide **Deepika Borgaonkar, Asst Prof,** Information Technology, GRIET. We express our sincere thanks for his encouragement, suggestions, and support, which provided the impetus and paved the way for the successful completion of the project work.

We wish to express our gratitude to **Dr. N. V. Ganapathi Raju,** our Project Co-coordinators **G. Vijendar Reddy** and **A. Sri Lakshmi** for their constant support during the project.

We express our sincere thanks to **Dr. Jandhyala N Murthy,** Director, GRIET**,** and **Dr. J. Praveen,** Principal, GRIET**,** for providing us a conducive environment for carrying through our academic schedules and project with ease.

We also take this opportunity to convey our sincere thanks to the teaching and non-teaching staff of GRIET College, Hyderabad.

Email: dvskarthik0824@gmail.com
Contact: 7981430282
Address: KPHB, Hyderabad.

Email: sharathmithun@gmail.com
Contact: 9381415799
Address: Mancherial, Hyderabad

Email: sujithnams@gmail.com
Contact: 8639009951
Address: Vivekananda Nagar, Hyderabad.

Email: jaini.ganesh@gmail.com
Contact: 6303001165
Address: Miyapur, Hyderabad.

Email: lokeshgundapureddy2001@gmail.com
Contact: 8374885876
Address: Deepthisri Nagar, Hyderabad.

DECLARATION

This is to certify that the project entitled "**CREDIT CARD EMBEZZLEMENT USING MACHINE LEARNING"** is a bonafide work done by us in partial fulfillment of the requirements for the award of the degree **BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY** from Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad.

We also declare that this project is a result of our own effort and has not been copied or imitated from any source. Citations from any websites, books, and paper publications are mentioned in the Bibliography.

This work was not submitted earlier at any other University or Institute for the award of any degree.

D. Karthik	***18241A1275***
N. Sai Sujith	***18241A1298***
J. Ganesh	***18241A1282***
V. Sharath	***18241A12B7***
G. Lokesh Reddy	***18241A1281***

TABLE OF CONTENTS

ABSTRACT

The main intention of this project is to build a predictive model which predicts the type of payment process done by the client as either a valid transaction or a fraudulent transaction. The data can be given to cyber security to further minimize the fraudulent actions on credit cards.

The project is implemented upon giving the model static training data of credit card transactions taken from 28 sectors with numerous ATM services and performing supervised learning Random Forest to make predictions and evaluate whether a certain transaction is valid or fraudulent.

The predicted range of payments is shown using a heatmap to find the accurate number of fraudulent payments and also show the rate of accuracy and precision using a confusion matrix through graphs

1. INTRODUCTION

1.1 Machine Learning:

Arthur Samuel, an American pioneer introduced the term Machine Learning in 1959, within the field of computer gaming and AI, and stated that “it gives computers the power to find out without being explicitly programmed”. ML is one of the most compelling technologies that one would ever have. As it is evident from the name, it gives the pc that creates it more almost like humans: the power to learn. Machine learning is widely getting used today, perhaps in more places than one would expect. Machine learning is a subset of Artificial Intelligence (AI) that provides systems the power to automatically learn and improve from experience without beginning to be explicitly programmed. Machine learning focuses on the development of computer programs which will access data and use it to find out from themselves.

Working of a Machine Learning Model

A Machine Learning framework gains from authentic information, constructs the forecast models, what's more, at whatever point it gets new information, predicts the result for it. The precision of the anticipated yield relies on how much information, as the enormous measure of information makes a difference to fabricate a superior model which predicts the result all the more precisely. Assume we have a complex issue, where we want to play out certain expectations, so rather than composing a code for it we simply need to take care of the information to conventional calculations, and with the assistance of these calculations, The machine constructs the rationale according to the information and predicts the result. AI has altered our perspective about the issue, Block outline of the AI calculation is as per the following.

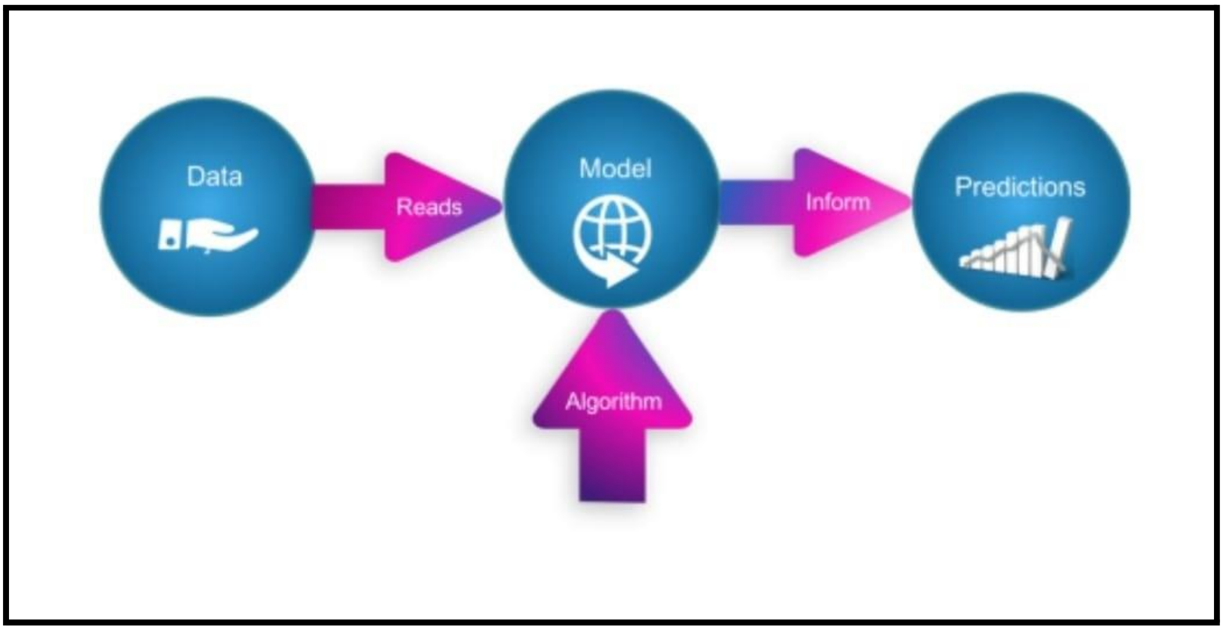


Fig 1 Machine Learning

Types of Machine Learning:

Machine learning is mainly classified into three categories. They are:

1. Supervised learning
2. Unsupervised learning
3. Reinforcement learning
4. Semi-supervised learning

1.Supervised learning: At the point when a calculation gains from model information and related target reactions that can comprise of numeric qualities or string names, like classes or tags, to later anticipate the right reaction when presented with new models goes under the class of Supervised learning.

2.Unsupervised learning: Though when a calculation gains from plain models with next to no related reaction, passing on to the calculation to decide the information designs all alone. This kind of calculation will in general rebuild the information into something else, for example, new highlights that might address a class or another series of un-connected values. They are very valuable in giving people experiences into the significance of information and new valuable contributions to directed AI calculations.

3.Reinforcement learning: Whenever you present the calculation with models that lack labels, as in unsupervised learning. Notwithstanding, you can go with a model with positive or negative criticism as per the arrangement the calculation proposes comes under the classification of Reinforcement learning, which is associated with applications for which the calculation should decide (so the item is prescriptive, not simply graphic, as in unsupervised learning), and the choices bear outcomes. In the human world, it is very much like advancing by experimentation.

4.Semi-supervised learning: Where a deficient training signal is given: a preparation set to prepare signal with some (frequently large numbers) of the objective results missing. There is an extraordinary instance of this rule known as Transduction where the whole arrangement of issue examples is known at learning time, then again, actually a piece of the objectives is absent.

Categorizing based on required Output:

1.Classification: Whenever inputs are isolated into at least two classes, and the student must produce a model that appoints concealed contributions to at least one (multi-label classification) of these classes.

2.Regression: This is likewise a supervised problem, A situation when the results are continuous instead of discrete.

3.Clustering: When a set of inputs is to be divided into many number of groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

1.2 EXISTING SYSTEM

According to the current situation regarding the technology which uses other supervised research algorithms to provide utmost accuracy possible using available technology with Machine Learning. The available accuracy of the static credit statistical data will be ranging about 75 to 85 percentage. We will be proposing a model that increases the algorithm result by about 12 to 14 percentage to help increase the modernised world in giving a better accuracy regarding the fraudulent information about the usage of the credit cards to the user and also stating the region where the data is most fraudulent. Most of the cases in the previous algorithm has less impact on the pre-processing of the data which results in the decrease of the accuracy levels of the raw data samples. This impacts the precision level of the model that gives the fraudulent data outputs.

1.3 PROPOSED SYSTEM

Credit genuinely should card organizations can recognize deceitful Mastercard exchanges so clients are not charged for things that they didn't buy. Such issues can be handled with Data Science and its significance, alongside Machine Learning, couldn't possibly be more significant. This task expects to show the displaying of an informational collection utilizing AI with Credit Card Fraud Detection. The Credit Card Fraud Detection Problem incorporates displaying past Visa exchanges with the information of the ones that ended up being extortion. This model is then used to perceive regardless of whether another exchange is deceitful. Our goal here is to recognize 100 percent of the false exchanges while limiting the mistaken misrepresentation groupings. Credit Card Fraud Detection is an ordinary example of arrangement. In this interaction, we have zeroed in on examining and pre-handling informational indexes as well as the sending of different inconsistency recognition calculations, for example, Local Outlier Factor and Random Forest calculation on the PCA changed Credit Card Transaction information.

2. REQUIREMENT ENGINEERING

2.1 HARDWARE MODULE

A laptop or Desktop with minimum requirements of 4 GB RAM and an operating system like windows/ios/Linux. And also supports all virtual machines and all programming IDE’S. Some other specifications like A CPU that has an Intel processor or better and A Hard-disk drive (HDD) of 200 MB or more.

Fig 2 Hardware module

2.2 SOFTWARE MODULE

2.2.1 JUPYTER

Jupyter Notebook is open-source web software that lets you create and share documents with live code, equations, visualizations, and narrative text. Data cleansing and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and many other applications are all possible.

Fig 3 Jupyter

We have used jupyter IDE for our code execution. Jupyter IDE supports all the python libraries which are needed for our project.

3. LITERATURE OF SURVEY

Fingerprints and face recognition are examples of biometric approaches. Biometric Data mining is a type of knowledge discovery process in which biometric data is provided with the goal of identifying patterns. Obtaining information about a user's behavior over an extended period of time. It has been discovered that the user's behavior is centered rather than random. In the future, an application-based approach will be developed to provide accurate results while checking for false cards.

A Fraud Detection Method Using a Cost-Sensitive Decision Tree. For extortion recognition, an expense delicate choice tree approach was applied. The expense of misclassification is utilized, which is variable, similar to the needs of the misrepresentation, which differ as indicated by individual records. To stay away from this, another exhibition metric has been fostered that focuses on each false exchange in a significant manner and surveys the model's viability in limiting absolute monetary misfortune. The measurement utilized is the Saved Loss Rate (SLR), which is the level of potential monetary misfortune that is saved when the accessible usable constraints of the cards from which deceitful exchanges are led are included utilizing AI methods named oversampling and under sampling.

Data and Technique-Oriented Perspective on Credit Card Fraud Detection Techniques Variable Ink Watermarking, See Through Register, Latent Image, and Micro Lettering are some of the techniques used in this process. The amount of research being done in this sector is growing all the time, and various image processing techniques are being used to produce more precise results. Several software programs should be implemented. Only to a limited extent is it possible to identify.

Detection of Credit Card Fraud “The Hidden Markov Model” is used. The Hidden Markov Represent (HMM) is used to model the sequence of actions and can be used to detect fraud. It is programmed to mimic the cardholder's typical behaviour. The key benefit is that it does not require fraud signatures; it can detect fraud just looking at cardholder spending patterns. To some extent, this model can detect fraud transactions. It can handle massive volumes of data and is scalable. In this process, optimization is difficult. When a transaction is discovered to be fraudulent, an alarm is triggered, and a security form with a set of questions is displayed.

Computational Intelligence for Real-Time Credit Card Fraud Detection The key to detecting fraud is to develop a dynamic system that adapts to changing e-commerce trends.

Demerits are people who consume a lot of data. Moduli zing is a difficult task.
SOM aids in the detection of fraud to a large extent.

A New Machine Learning Algorithm for Detecting Credit Card Fraud The cortical algorithm, a unique machine learning algorithm, is employed in this. This research reveals a reliable method for detecting credit card fraud. For accurate detection, further strategies are required. The technology is effective in rejecting phone cards, according to test data.

Modified Fisher Discriminant Analysis for Detecting Credit Card Fraud Linear discriminant is a supervised learning method utilized in this approach. This solution appropriately labels transactions with large useable limits on the cards, preventing millions of dollars from being lost in real-world banking systems. It is a slow detecting process. The Linear Perceptron Discriminant function is utilized to solve all of the issues with the modified FDA.

3.1 Python:

Python is a programming language that can be understood by both humans and machines. Guido Van Rossum, a Dutch software developer, invented it in the early 1980s and released it in 1991 to help programmers write clear, logical code for general-purpose programming at small and big scales.

Python is an interpreted high-level programming language that can be used to create programs that are structured, procedural, functional, imperative, reflective, or object-oriented.

Python, a dynamically typed language, is gaining traction in the technology industry thanks to its simple programming syntax, code readability, English-like commands, and versatility in a well-organized structure. It is more efficient and straightforward to learn than other programming languages. As a result, Python is the greatest option for a wide range of jobs, from a simple web application to a whole operating system.

What is Python used for?

The most important advantage of Python is that it's a general-purpose language that can be applied in many varieties of fields. The most common domains where Python is applied are as follows:

1. Data science
2. Artificial intelligence and Machine learning
3. Web development
4. Software development
5. Scientific and numeric applications
6. Game development
7. Graphic design Applications
8. Operating systems
9. Desktop GUI

3.2 Jupyter Notebook:

The journal expands the control center based way to deal with intelligent registering in a subjectively new course, giving a web-based application reasonable to catching the entire calculation process: creating, recording, and executing code, as well as conveying the outcomes.

The IPython notebook merges two components:

1.Web application: a program based instrument for intuitive writing of archives that join informative text, math, calculations, and their rich media yield.

2.Notebook documents: a portrayal of all satisfied apparent in the web application, counting data sources and results of the calculations, illustrative text, arithmetic, pictures, and rich media portrayals of articles.

4. TECHNOLOGY

4.1 PYTHON LIBRARIES

The following libraries are used in our code

1. PANDAS
2. MATLPLOTLIB
3. SEABORN
4. SKLEARN

1. PANDAS

Pandas is a Python package that allows you to work with large data sets. It offers tools for data analysis, cleansing, exploration, and manipulation. Wes McKinney came up with the name "Pandas" in 2008, which refers to both "Panel Data" and "Python Data Analysis." Pandas make it possible to evaluate large amounts of data and provide conclusions based on statistical theory. Pandas can clean up and produce readable and useful data collections.

Fig 4 Pandas

Python is an open-source programming language. It's difficult to know which package is ideal for a certain task. For data science, there is one package that we definitely must learn: pandas. Although the strong machine learning and attractive visualization capabilities may have piqued your interest, you won't get very far if you don't know how to use Pandas.

Pandas have two basic data structures: **Series** and **Dataframes**. Dataframes are the most common way to store data, so handling them fast is perhaps the most critical skill set for data analysis.

The output format of pandas can be visualized using matplotlib and NumPy. pandas primarily have data structures and operations that can be done on numerical tables and temporal series. pandas have various methods that can be used to speed up the calculating process of analyzing the data.

2. MATPLOTLIB

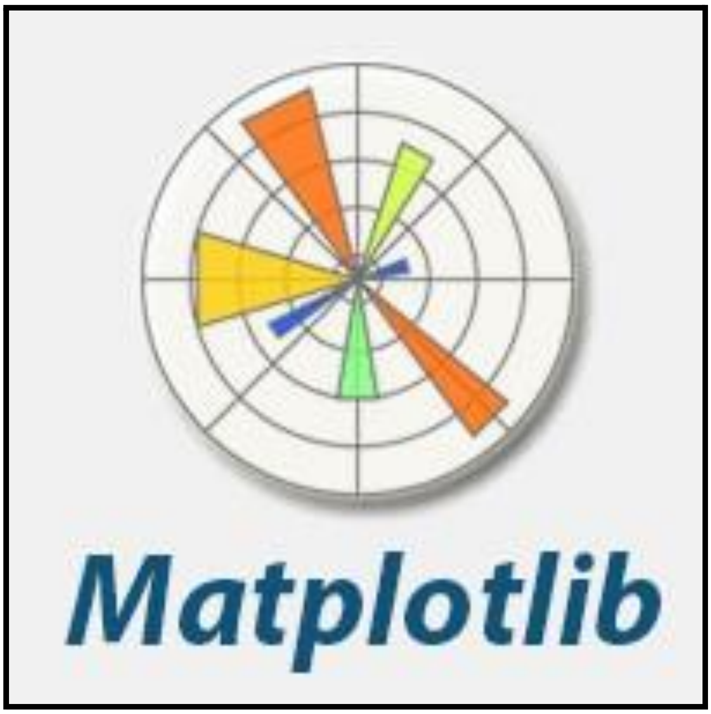


Fig 5 Matplotlib

Matplotlib is a fabulous Python representation bundle for 2D cluster graphs. Matplotlib is a multi-stage information representation bundle in light of NumPy clusters and planned to work with the SciPy stack in general. It was first presented in 2002 by John Tracker. One of the main benefits of perception is that it furnishes us with visual admittance to gigantic volumes of information in essentially justifiable illustrations. Matplotlib has an assortment of plots like line, bar, dissipate, histogram, etc.

Many of matplotlib's operations are contained in the pyplot submodule. It is used to plot the graph for the information provided by the user. We can add specific details to the plot by labelling the axes according to the user's requirements. We can also use matplotlib to evaluate trends by providing scatter plots to verify the crests and troughs in the trends.

3. SEABORN

Seaborn is a library in Python prevalently utilized for making statistical graphics. It gives a significant level point of interaction to drawing alluring and instructive measurable illustrations. Seaborn is an information representation library based on top of matplotlib and intently incorporated with pandas information structures in Python. It gives dataset-arranged APIs, so that we can switch between various visual portrayals for same factors for better comprehension of dataset.

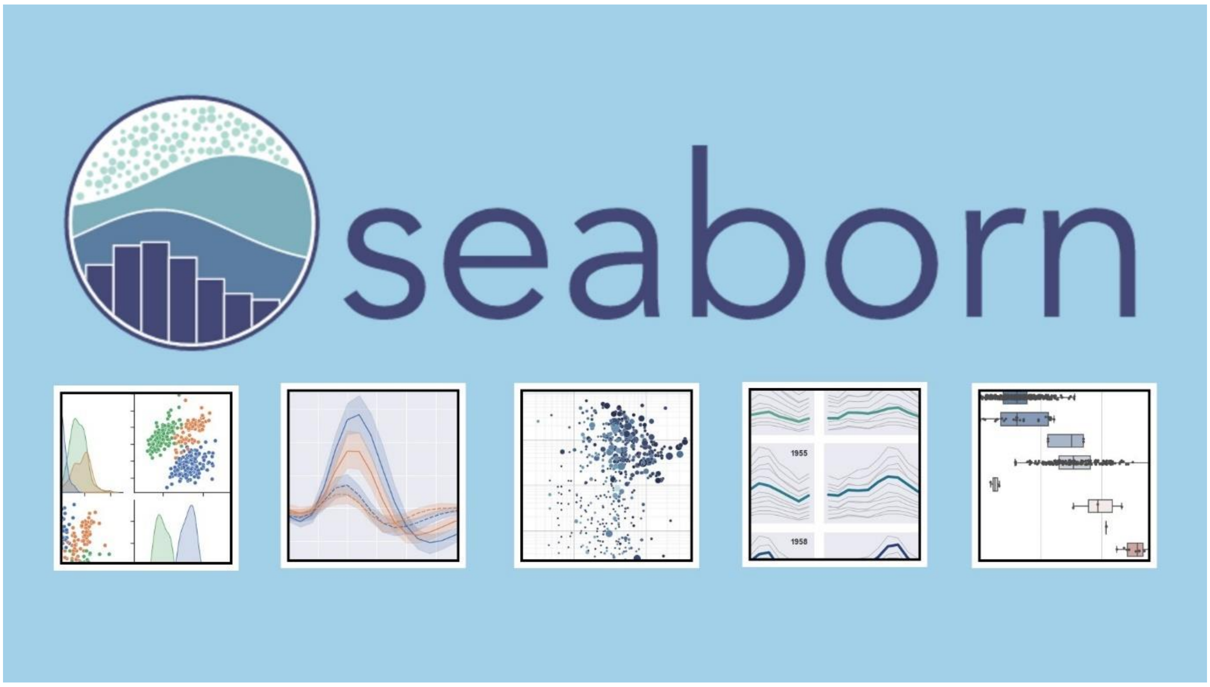


Fig 6 Seaborn

4. SKLEARN

This library is largely written in Python and is built upon NumPy, SciPy and Matplotlib sklearn is used majorly for supervised learning but subsequently also be used for clustering and cross-validation. Here we use metrics, ensemble and model-evaluation sub libraries to provide the necessary statistical results. The sklearn pre-processing package provides several common utility functions.

Fig 7 Sklearn

5. ALGORITHM

5.1 WHAT IS RANDOM FOREST?

Random Forest is a Machine Learning algorithm that has a place with Supervised Learning Method. It is utilized for both Classification and Regression. Random Forest is a classifier that contains various decision trees on different subsets of the dataset also, takes normal to further develop precision of that dataset. The "forest" it fabricates, is an gathering of decision trees, normally prepared with the "bagging" strategy. The general thought of the bagging strategy is that a blend of learning models that expands the outcome.

5.2 WHY RANDOM FOREST?

- It takes less training time when compared to other algorithms.
- It also predicts output with higher accuracy and even for large dataset it runs efficiently.
- The greater the number of decision trees that will leads to higher accuracy.

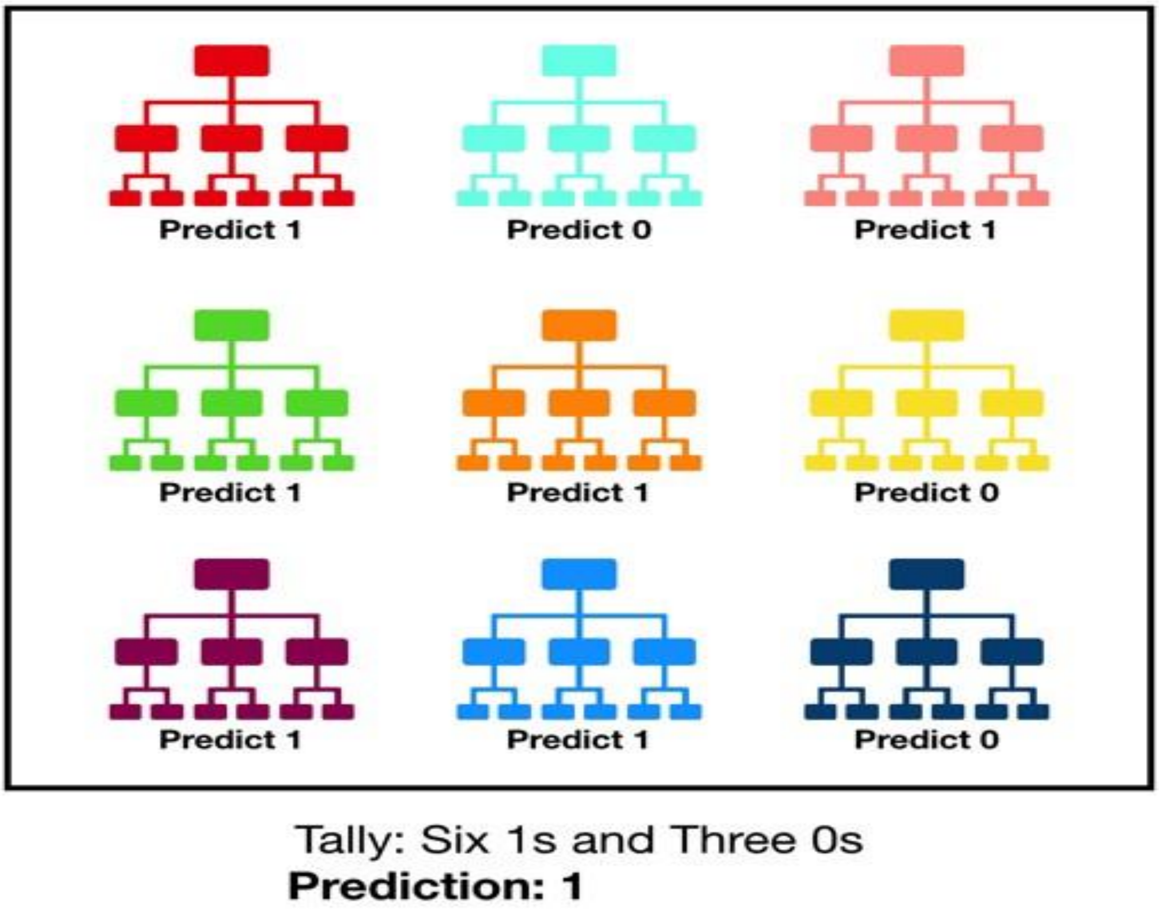


Fig 8

5.3 MODEL

- Training data
- Label the fraudity index
- Testing data split
- Perform Random Forest Algorithm
- Calculate Accuracy and Precision
- Print heatmaps between valid and fraud transactions

5.4 RANDOM FOREST PROCESS FLOW

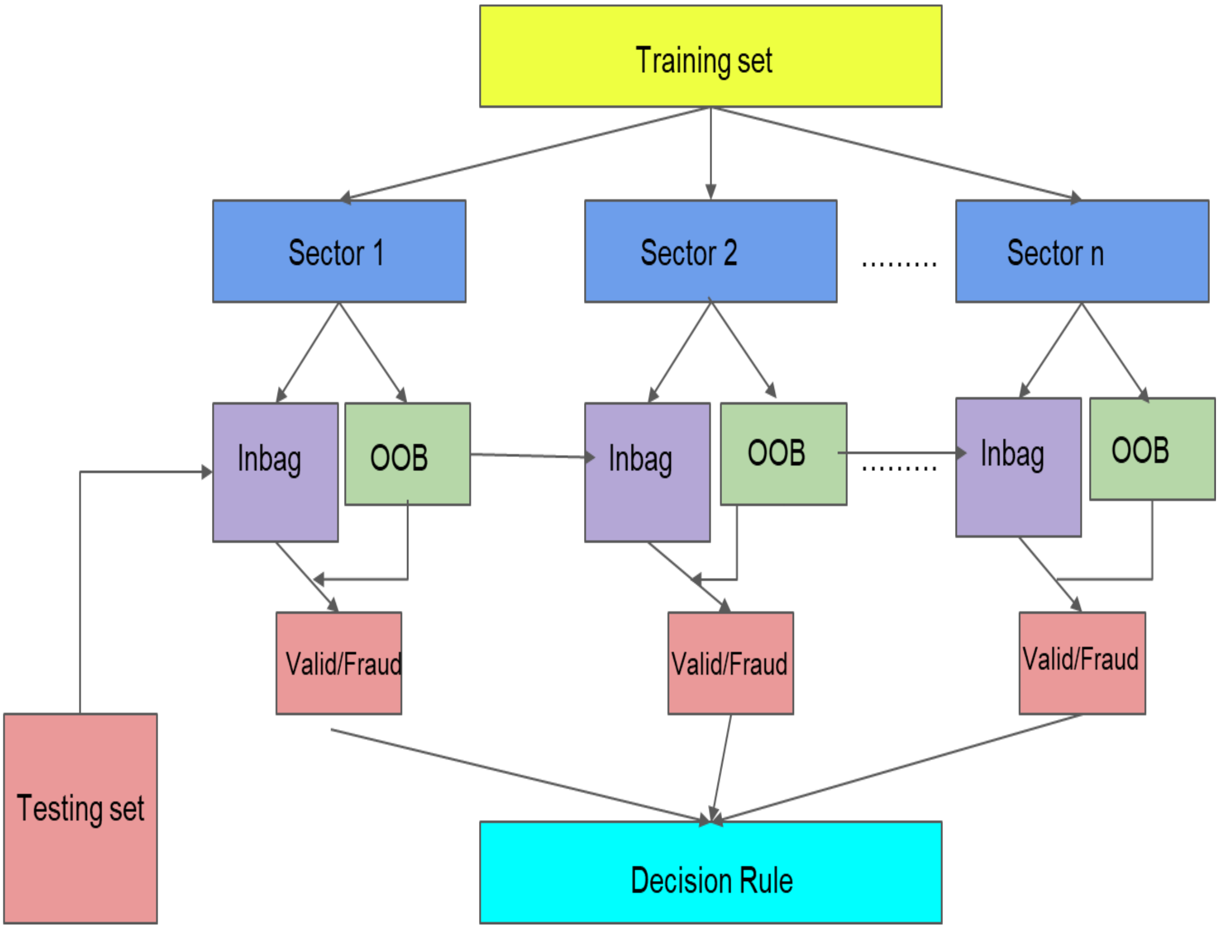
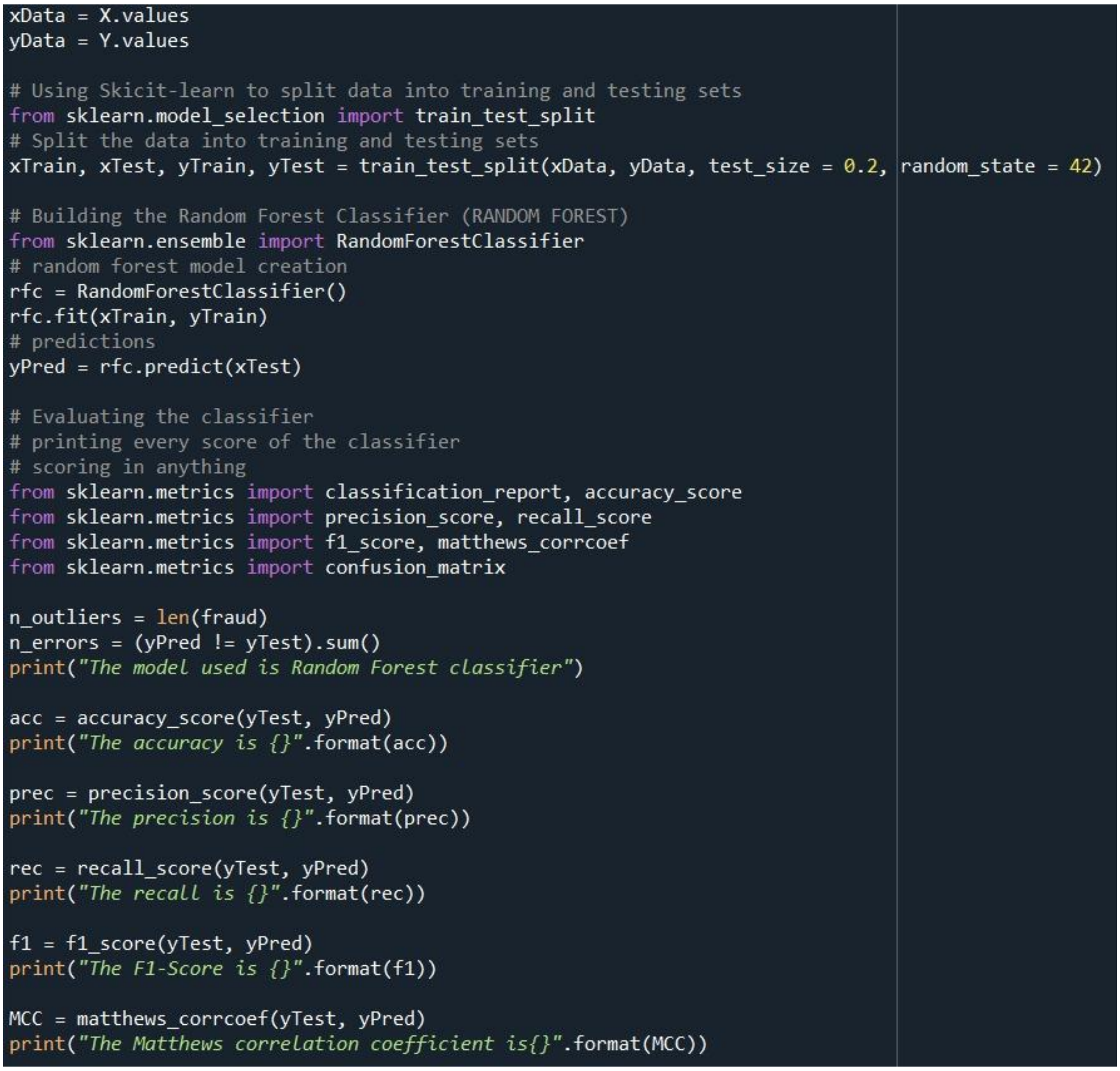


Fig 9 Random forest process flow

5.5 RANDOM FOREST ALGORITHM FLOW

- Segregate and allot the training and testing data size
- Assign Random Forest Classifier variable
- Allot prediction variable for the algorithm
- Import metric evaluations into the workspace
- Display accuracy and precision reports to the user



```python
xData = X.values
yData = Y.values

# Using Skicit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
xTrain, xTest, yTrain, yTest = train_test_split(xData, yData, test_size = 0.2, random_state = 42)

# Building the Random Forest Classifier (RANDOM FOREST)
from sklearn.ensemble import RandomForestClassifier
# random forest model creation
rfc = RandomForestClassifier()
rfc.fit(xTrain, yTrain)
# predictions
yPred = rfc.predict(xTest)

# Evaluating the classifier
# printing every score of the classifier
# scoring in anything
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix

n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Random Forest classifier")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))

rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is{}".format(MCC))
```

Fig 10 Random Forest Algorithm

6. ARCHITECTURE

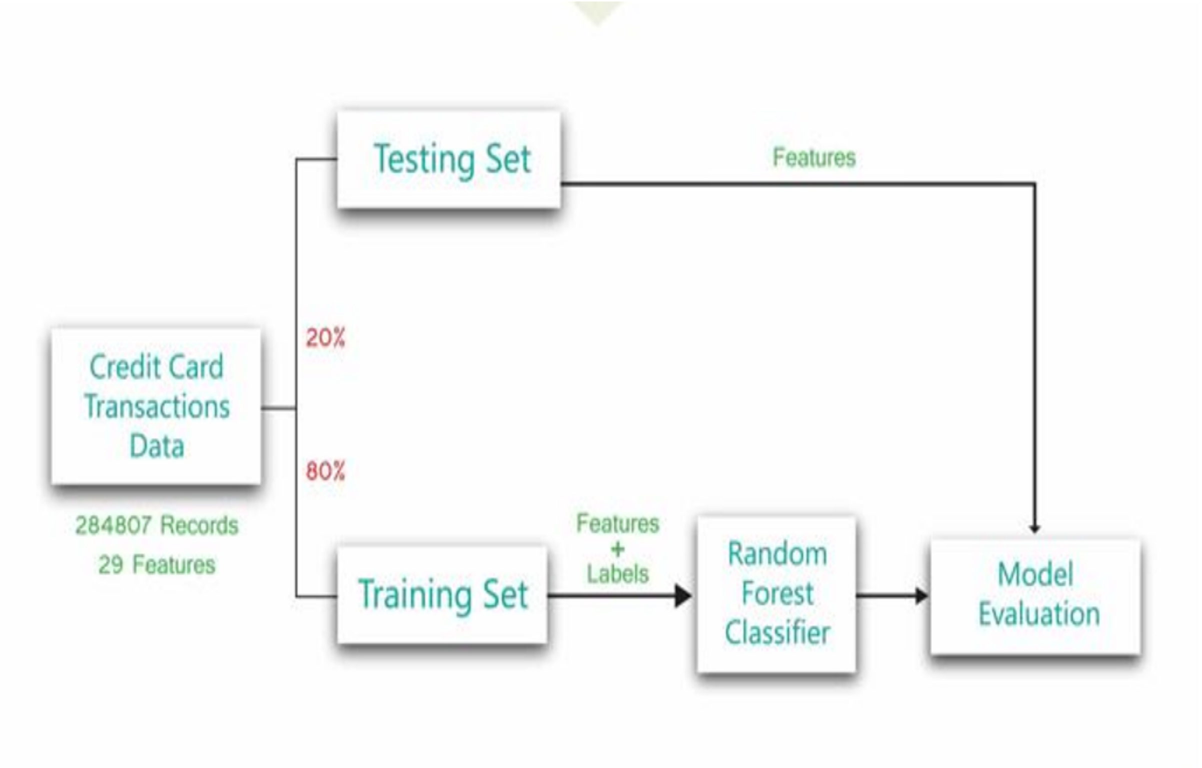


Fig 11 Architecture

Here, we contain a large number of credit card transactions data which will be splitting into training data set and testing data set. The training data comprises of features and labels and this dataset will be passing through the random forest algorithm which splits into many number of decision trees and then it evaluates the model based on the decision trees and also learns how to predict for the test dataset and when a testing set is supplied to the model it evaluates the model based on the testing set.

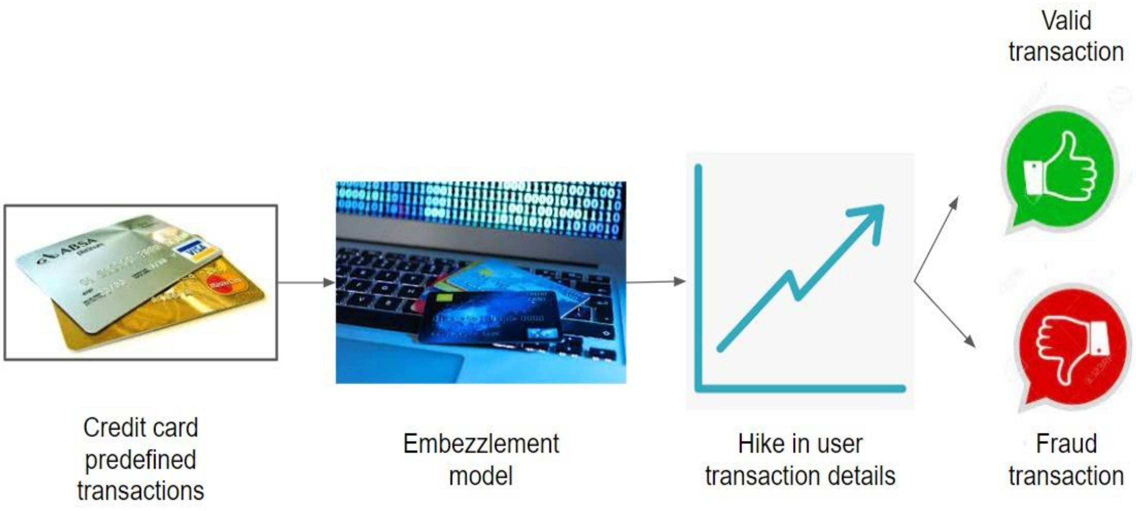

Valid transaction
Credit card predefined transactions
Embezzlement model
Hike in user transaction details
Fraud transaction

Fig 12 Algorithm Flow

There are many number of credit card predefined transactions coming from numerous sources and thus we will be passing all these transactions through the model which consists of the algorithm beneath it and this algorithm makes the decision trees and takes average of all the tree values and thus identifies whether the given transaction is valid or a fraudulent transaction.

6.1 Use Case Diagram

A use case diagram in the Unified Modeling Language (UML) is a kind of social graph characterized by actors and made from a Use-case investigation. Its motivation is to present a graphical outline of the usefulness given by a framework as far as actors, their objectives (addressed as a use case), and any conditions between those cases. The primary motivation behind a use case diagram is to show framework capacities are performed for which actor. Jobs of the actors inside the framework are frequently portrayed.

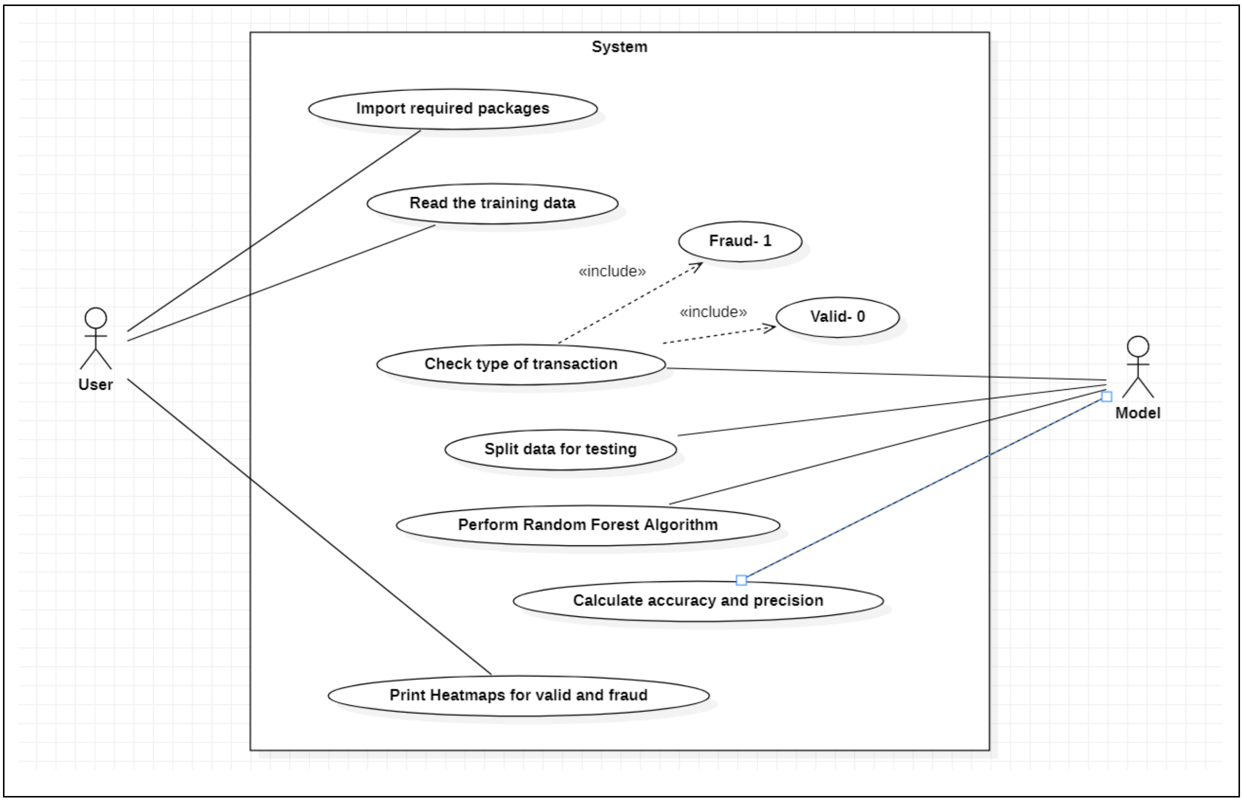

System
Import required packages
Read the training data
Fraud- 1
«include»
«include»
Valid- 0
User
Check type of transaction
Model
Split data for testing
Perform Random Forest Algorithm
Calculate accuracy and precision
Print Heatmaps for valid and fraud

Fig 13 Use case diagram

6.2 Activity Diagram

An Activity Diagram is a type of behavioral diagram and it represents the behavior of a system. The basic components of an Activity diagram are Action, Decision Node, Control flow, Start node, and end node. An Activity diagram shows us the control flow from the starting point to the finishing point showing the various decision paths that exist while the activity is being executed. Activity diagrams are useful for business modeling where they're used for detailing the processes involved in business activities.

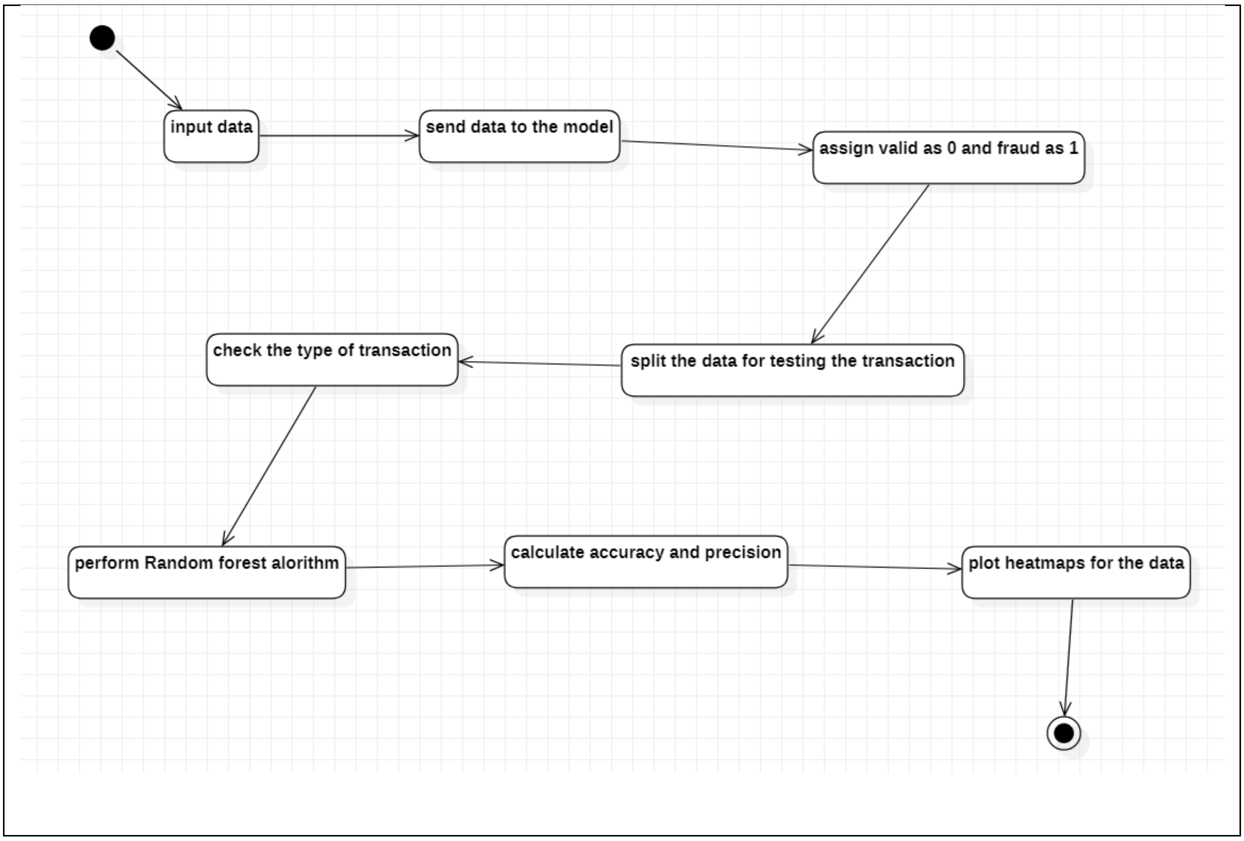

input data
send data to the model
assign valid as 0 and fraud as 1
split the data for testing the transaction
check the type of transaction
perform Random forest alorithm
calculate accuracy and precision
plot heatmaps for the data

Fig 14 Activity diagram

6.3 Class Diagram

In Software Engineering, a class diagram in the Unified Modeling Language (UML) is a sort of static construction diagram that depicts the design of a framework by appearing the framework their classes, their attributes, operations (or techniques), and the connections among the classes makes sense of which class contains data.

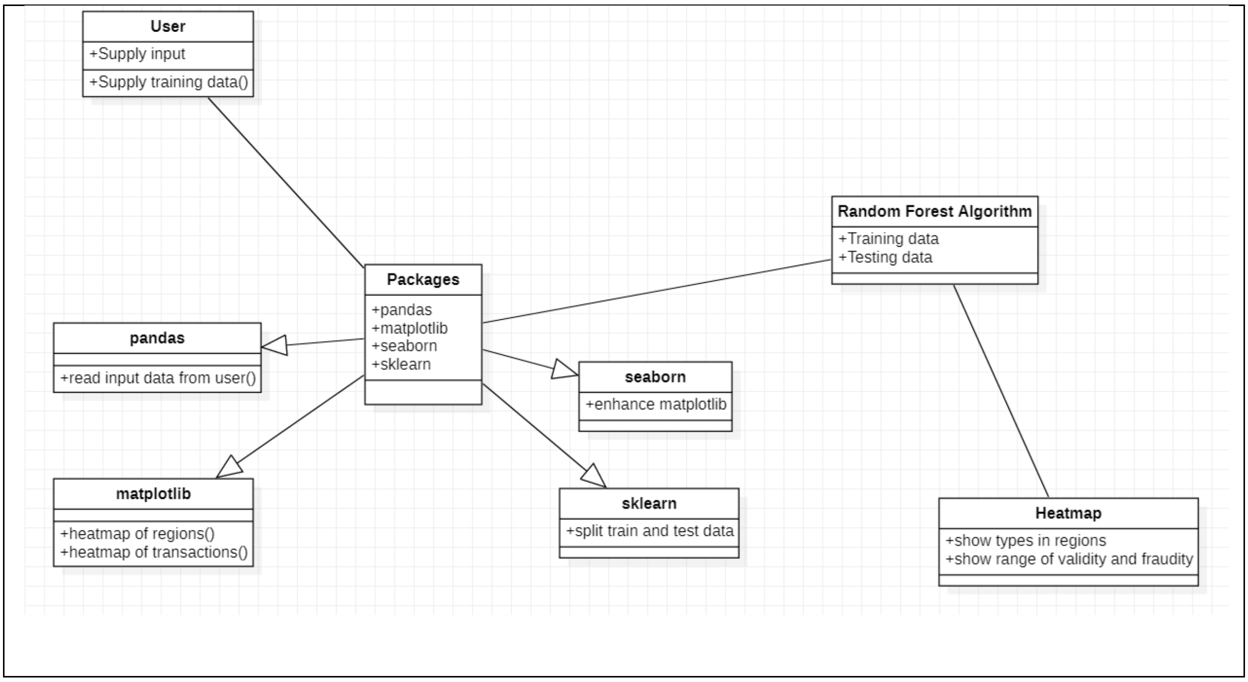


Fig 15 Class diagram

6.4 Sequence Diagram

A sequence diagram in UML is a kind of interaction diagram which shows how each process of the system operates with one another and in what order. It is constructed as a message sequence chart. A sequence diagram shows the interaction of objects through messages which are arranged sequentially. It represents the objects involved and the sequence of messages exchanged between the objects needed to carry out the functionality. Sequence diagrams are sometimes called event diagrams or timing diagrams.

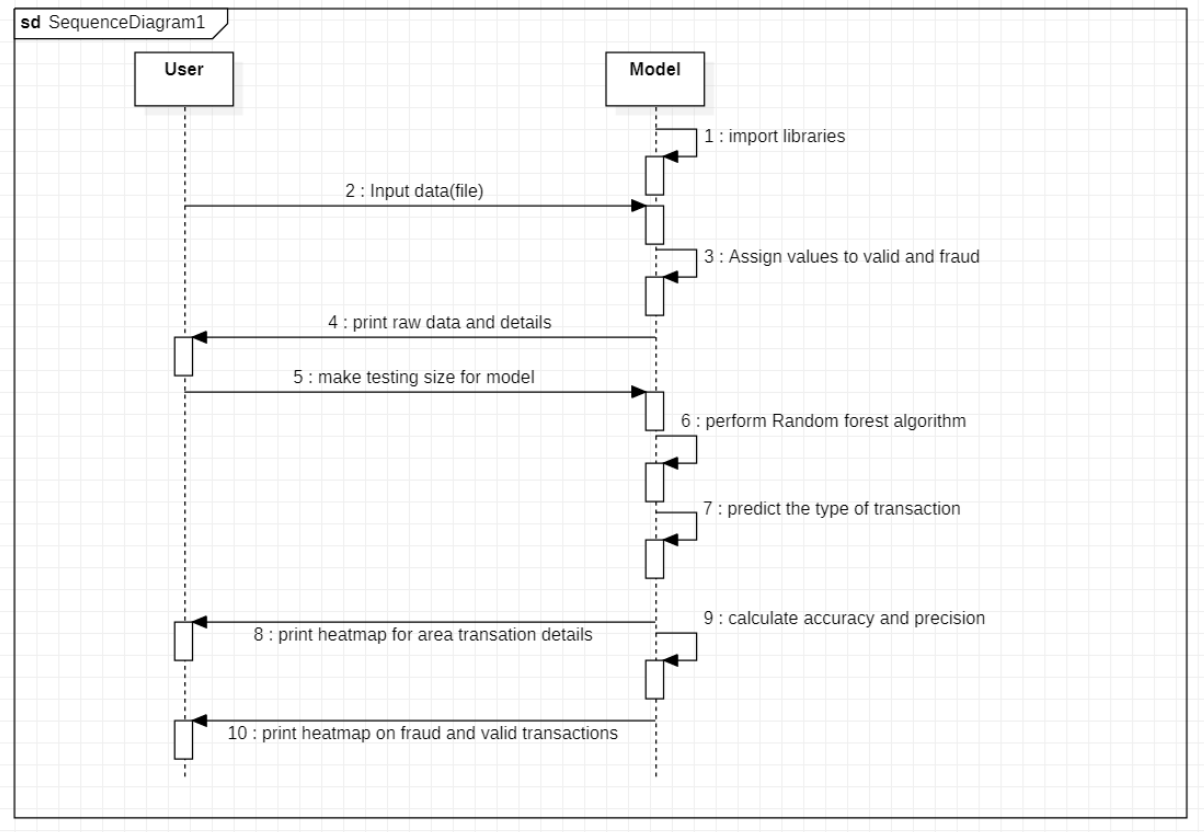
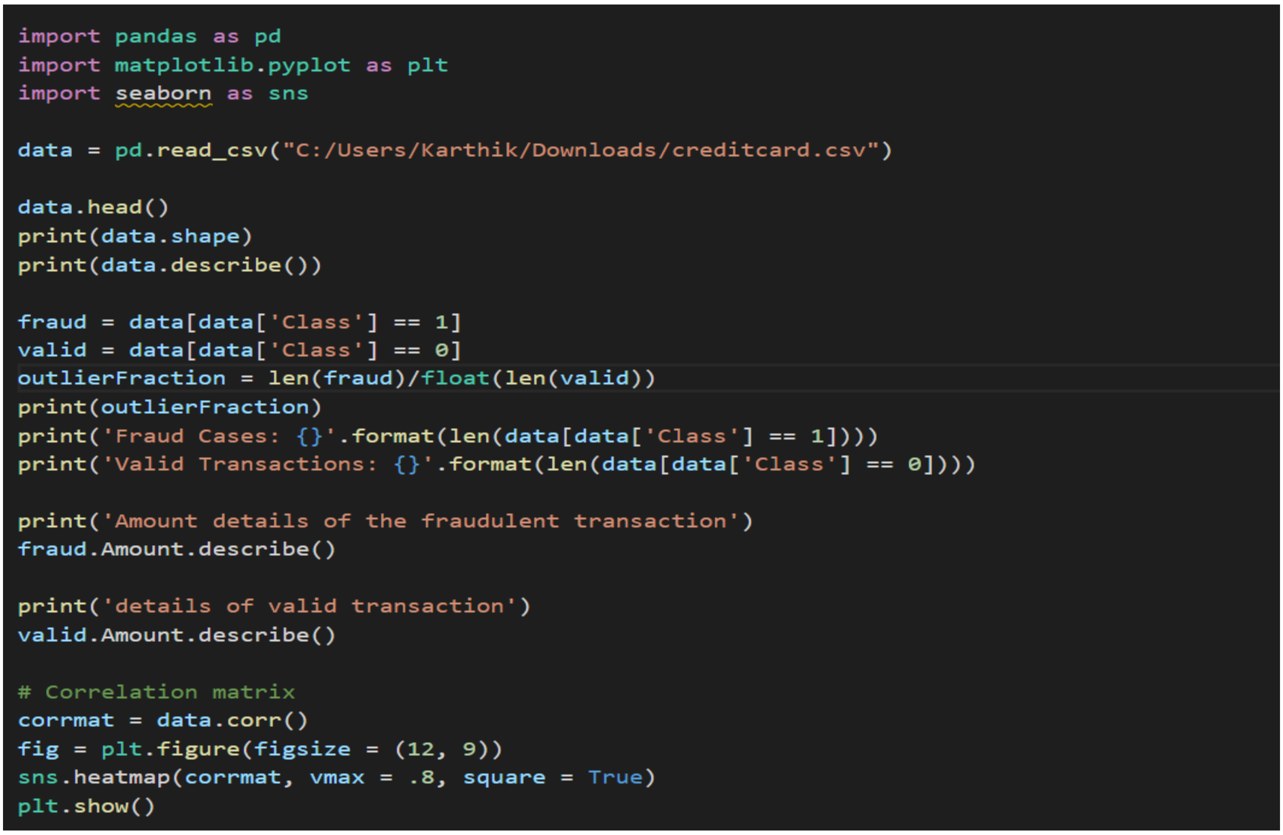

sd SequenceDiagram1
User
Model
1 : import libraries
2 : Input data(file)
3 : Assign values to valid and fraud
4 : print raw data and details
5 : make testing size for model
6 : perform Random forest algorithm
7 : predict the type of transaction
9 : calculate accuracy and precision
8 : print heatmap for area transation details
10 : print heatmap on fraud and valid transactions

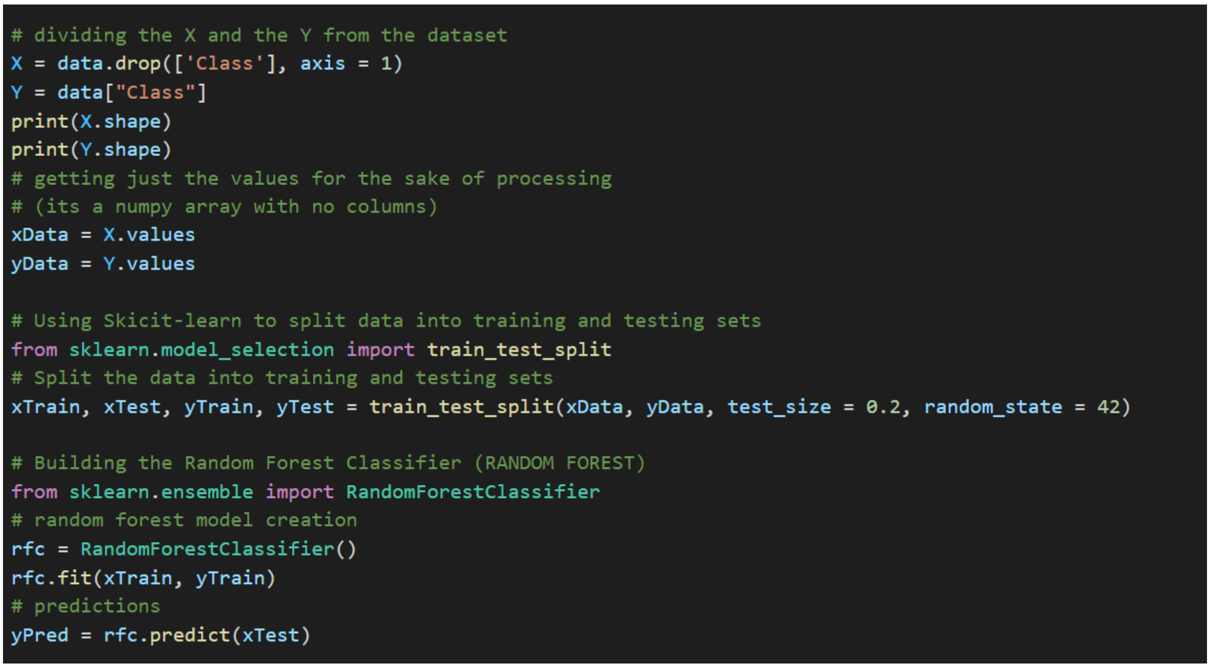
Fig 16 Sequence diagram

7. IMPLEMENTATION AND CODING



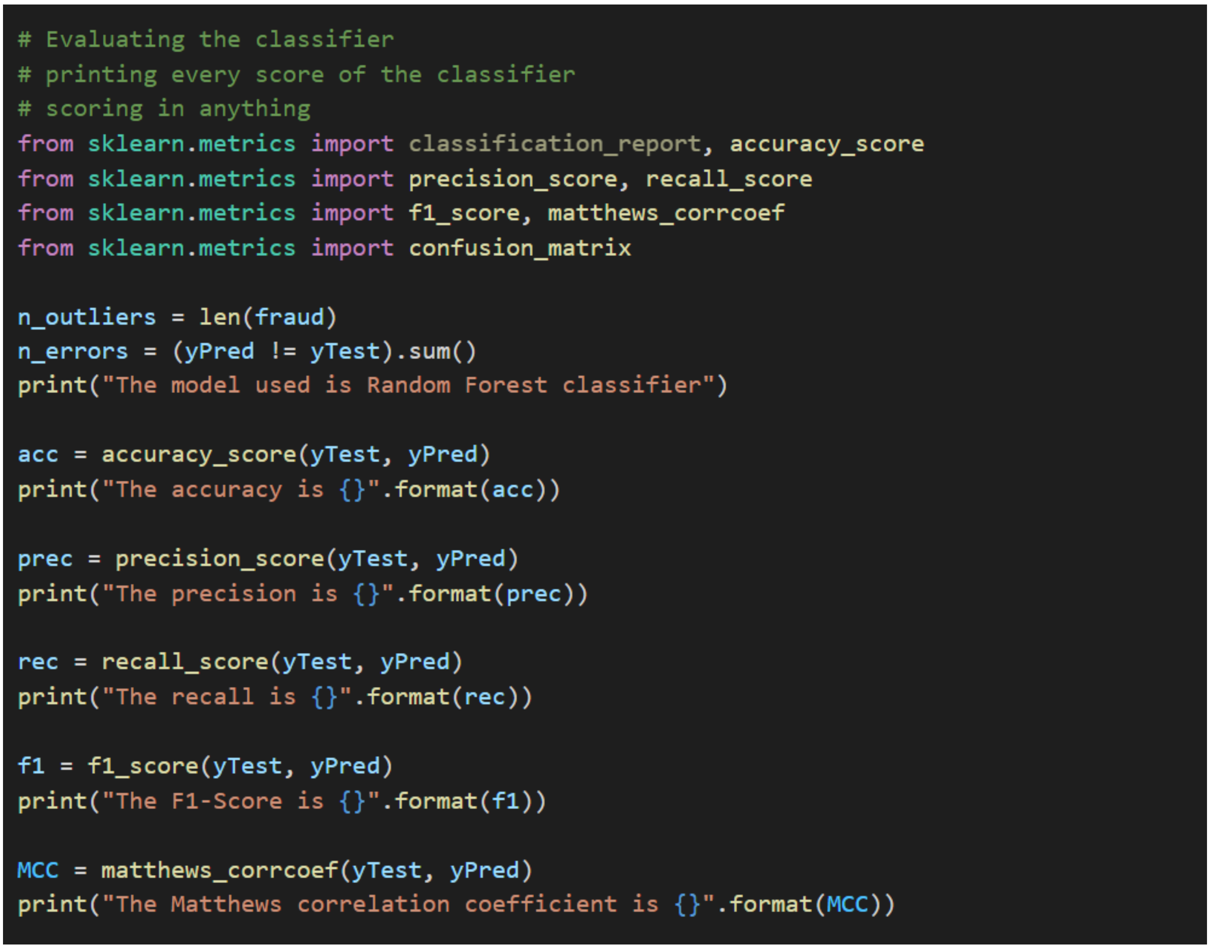
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv("C:/Users/Karthik/Downloads/creditcard.csv")

data.head()
print(data.shape)
print(data.describe())

fraud = data[data['Class'] == 1]
valid = data[data['Class'] == 0]
outlierFraction = len(fraud)/float(len(valid))
print(outlierFraction)
print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))
print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))

print('Amount details of the fraudulent transaction')
fraud.Amount.describe()

print('details of valid transaction')
valid.Amount.describe()

# Correlation matrix
corrmat = data.corr()
fig = plt.figure(figsize = (12, 9))
sns.heatmap(corrmat, vmax = .8, square = True)
plt.show()
```

Fig 17 Model code 1



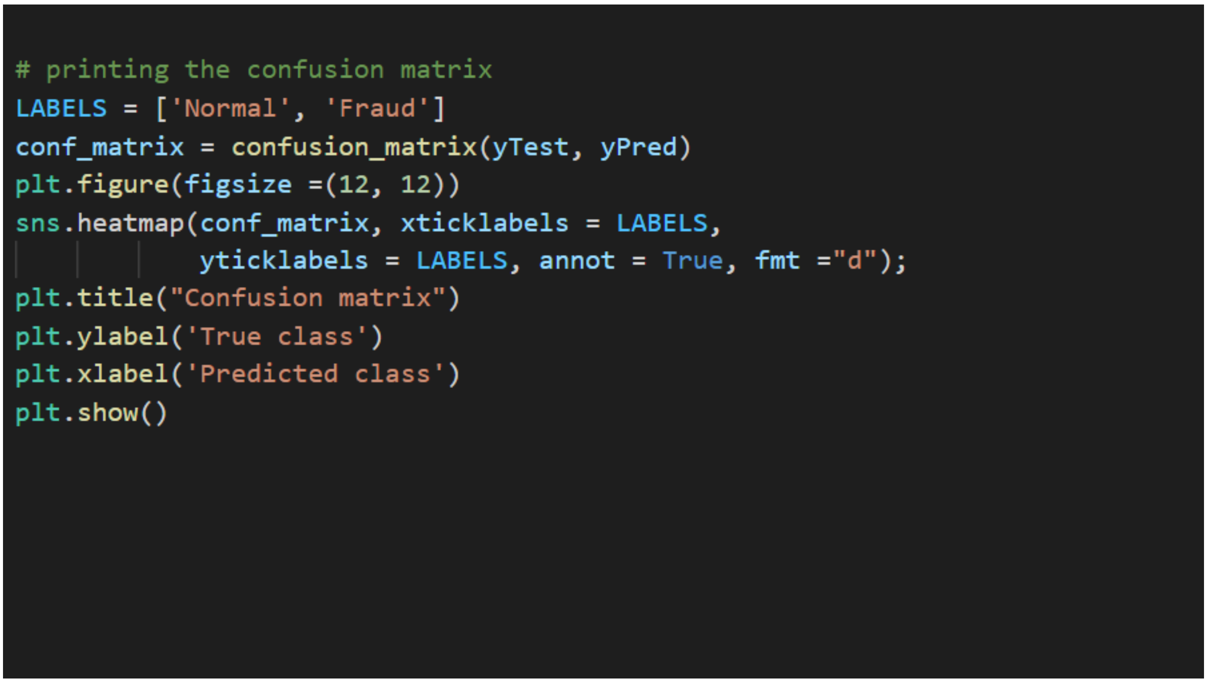
```
# dividing the X and the Y from the dataset
X = data.drop(['Class'], axis = 1)
Y = data["Class"]
print(X.shape)
print(Y.shape)
# getting just the values for the sake of processing
# (its a numpy array with no columns)
xData = X.values
yData = Y.values

# Using Skicit-learn to split data into training and testing sets
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
xTrain, xTest, yTrain, yTest = train_test_split(xData, yData, test_size = 0.2, random_state = 42)

# Building the Random Forest Classifier (RANDOM FOREST)
from sklearn.ensemble import RandomForestClassifier
# random forest model creation
rfc = RandomForestClassifier()
rfc.fit(xTrain, yTrain)
# predictions
yPred = rfc.predict(xTest)
```

Fig 18 Model code2



```python
# Evaluating the classifier
# printing every score of the classifier
# scoring in anything
from sklearn.metrics import classification_report, accuracy_score
from sklearn.metrics import precision_score, recall_score
from sklearn.metrics import f1_score, matthews_corrcoef
from sklearn.metrics import confusion_matrix

n_outliers = len(fraud)
n_errors = (yPred != yTest).sum()
print("The model used is Random Forest classifier")

acc = accuracy_score(yTest, yPred)
print("The accuracy is {}".format(acc))

prec = precision_score(yTest, yPred)
print("The precision is {}".format(prec))

rec = recall_score(yTest, yPred)
print("The recall is {}".format(rec))

f1 = f1_score(yTest, yPred)
print("The F1-Score is {}".format(f1))

MCC = matthews_corrcoef(yTest, yPred)
print("The Matthews correlation coefficient is {}".format(MCC))
```

Fig 19 Model code 3

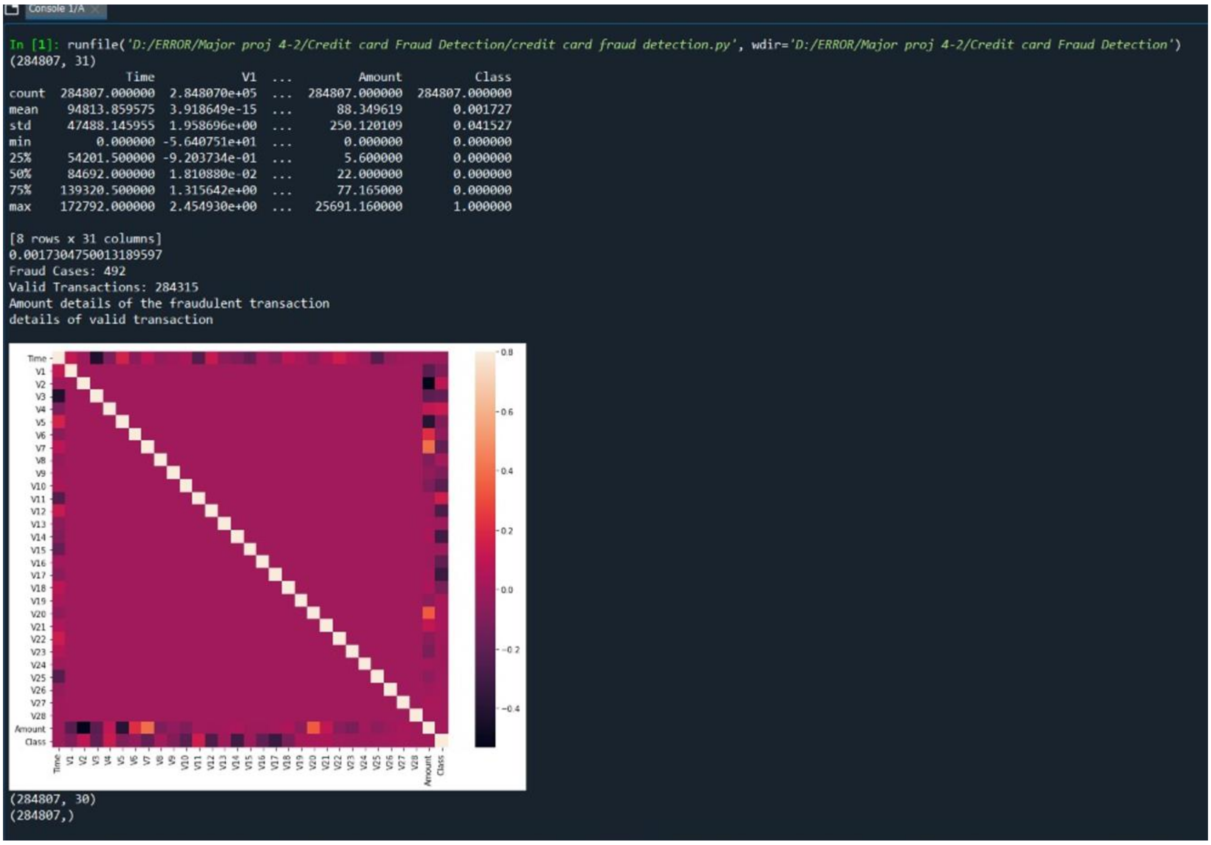


```python
# printing the confusion matrix
LABELS = ['Normal', 'Fraud']
conf_matrix = confusion_matrix(yTest, yPred)
plt.figure(figsize =(12, 12))
sns.heatmap(conf_matrix, xticklabels = LABELS,
            yticklabels = LABELS, annot = True, fmt ="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
```

Fig 20 Model code4

8. RESULTS

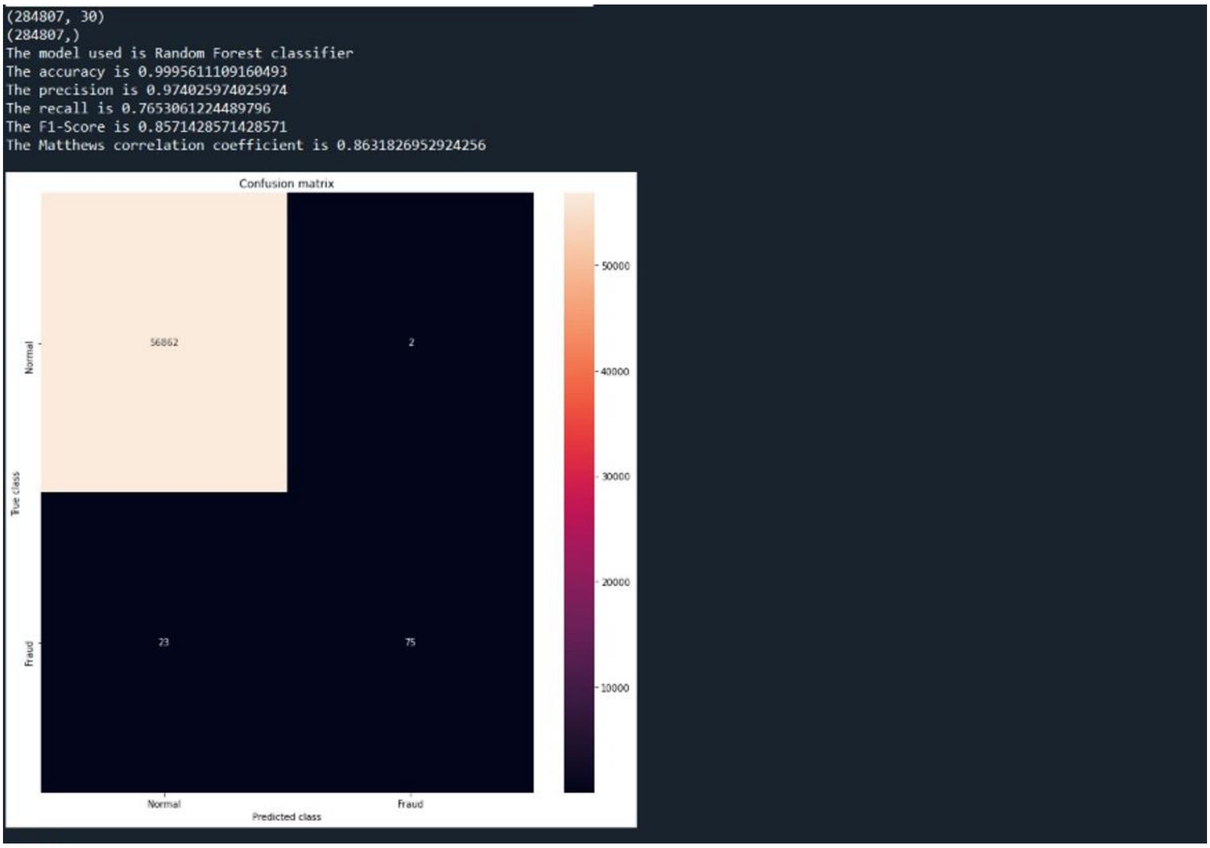
Below screenshots are the output when we execute the code in the system.



```
In [1]: runfile('D:/ERROR/Major proj 4-2/Credit card Fraud Detection/credit card fraud detection.py', wdir='D:/ERROR/Major proj 4-2/Credit card Fraud Detection')
(284807, 31)
                Time            V1  ...         Amount          Class
count  284807.000000  2.848070e+05  ...  284807.000000  284807.000000
mean    94813.859575  3.918649e-15  ...      88.349619       0.001727
std     47488.145955  1.958696e+00  ...     250.120109       0.041527
min         0.000000 -5.640751e+01  ...       0.000000       0.000000
25%     54201.500000 -9.203734e-01  ...       5.600000       0.000000
50%     84692.000000  1.810880e-02  ...      22.000000       0.000000
75%    139320.500000  1.315642e+00  ...      77.165000       0.000000
max    172792.000000  2.454930e+00  ...   25691.160000       1.000000

[8 rows x 31 columns]
0.0017304750013189597
Fraud Cases: 492
Valid Transactions: 284315
Amount details of the fraudulent transaction
details of valid transaction
```

```
(284807, 30)
(284807,)
```

Fig 21 Result 1



```
(284807, 30)
(284807,)
The model used is Random Forest classifier
The accuracy is 0.9995611109160493
The precision is 0.974025974025974
The recall is 0.7653061224489796
The F1-Score is 0.8571428571428571
The Matthews correlation coefficient is 0.8631826952924256
```

Fig 22 Result 2

9. CONCLUSION

Conclusion

The current project can forecast and predict the fraudulent transaction of a credit card for up to an accuracy of 95% to 99%. It also indicates the total number of fraud cases and valid transactions with the help of Heatmap. The precision of these transactions also lies between 95% to 99%. The F-1 Score and Matthews correlation co-efficient are also close to +1 which indicates that the model is efficient even for the large data sets.

10. BIBLIOGRAPHY

1. Prof. Kenneth Aguilar, prof. Cesar Ponce,2017 Biometric Approach
2. Y. Sahin, S. Bulkan, and E. Duman,2013 A Cost-Sensitive Decision Tree Approach for Fraud Detection
3. A. Srivastava, A. Kundu, S. Sural, and A. Majumdar 2009. Credit Card Fraud Detection Techniques: Data and Technique Oriented Perspective
4. A. Srivastava, A. Kundu, S. Sural, and A. Majumda 2008 Credit Card Fraud Detection Using Hidden Markov Model
5. J. T. Quah and M. Sriganesh Real Time Credit Card Fraud Detection using Computational Intelligence
6. N. S. Halvaiee and M. K. Akbari, A Novel Machine Learning Algorithm to credit card fraud detection
7. S. Panigrahi, A. Kundu, S. Sural, and A. K. Majumdar Credit Card Fraud Detection: A Fusion Approach using Dempster–Shafer Theory and Bayesian Learning
8. N. Mahmoudi and E. Duman Detecting Credit Card Fraud by Modified Fisher Discriminant Analysis

REFERENCES

Paper 1

https://www.researchgate.net/publication/336800562_Credit_Card_Fraud_Detection_using_Machine_Learning_and_Data_Science

Paper 2

https://www.sciencedirect.com/science/article/abs/pii/S0957417413003072

Paper 3

https://ieeexplore.ieee.org/abstract/document/8123782

Paper 4

https://arxiv.org/pdf/1611.06439

Paper 5

https://ieeexplore.ieee.org/iel5/8858/4447479/04358713.pdf

Paper 6

https://www.sciencedirect.com/science/article/pii/S0957417407003995