Credit Card Fraud Detection using Machine Learning

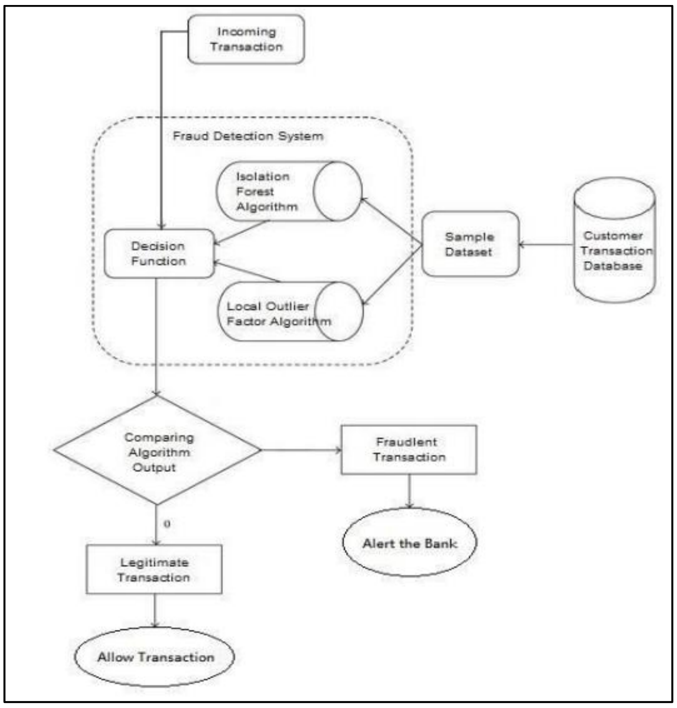
MACHINE LEARNING(CBS3006)

SUBMITTED BY:
SHIKHAR KUMAR(20BBS0215)
DEVYANSH YADAV(20BBS0155)
BHAVESH YADAV(20BBS0200)
CHIRAG SHARMA(20BBS202)

ABSTRACT

It's important for credit card companies to detect fraudulent credit card transactions and prevent customers from being charged for devices they no longer purchase. Such problems can be solved with data science, and its importance alongside machine learning cannot be overstated. The purpose of this task is to demonstrate the modeling of gadget usage information sets. This provides insight through credit card fraud detection. The problem of credit card fraud detection goes beyond credit card transactions to model the information of a person who is found to be fraudulent. This version is used to determine if new transactions are fraudulent. The goal here is to detect 100% of fraudulent transactions while minimizing false fraud classifications. Credit card fraud detection is a standard classification pattern. In addition to employing several anomaly detection algorithms such as local outlier factor rule sets and isolated forests for PCA-transformed credit card transaction information, this process should focus on information-wise analysis and preprocessing. .

Keywords— Credit card fraud, applications of machine learning, data science, isolation forest algorithm, local outlierfactor, automated fraud detection.



I. Introduction

"Fraud" in credit card transactions is the unauthorized and unwanted use of an account by a person other than the account holder. We are able to take the necessary precautions to prevent this abuse and to investigate such fraudulent behavior so that we can reduce it and protect against similar occurrences in the future. In other words, credit card fraud is when someone uses someone else's credit card for personal reasons and the card owner and the government are unaware of the fact that the card is being used. Fraud detection involves tracking the sports of our customer population so that we can assess, understand, or prevent fraudulent activity, intrusions, late payments, and other objectionable behavior. This is a perfectly applicable problem that requires a group eye as well as device mastering and information technology know-how to automatically solve this problem. This issue is particularly difficult from a mastering standpoint, as it is characterized by an imbalance of various factors and elegance. The types of legitimate transactions far outweigh the number of fraudulent transactions. Also, transaction styles often change the stat house over time. However, these are not simple challenges when implementing a truly global fraud detection system. In a real-world global example, a large stream of price requests is rapidly scanned by an automated device to determine which transactions to approve. Machine mastering algorithms are tasked with investigating all legitimate transactions and recording suspicious ones. These reviews are investigated through specialists who touch the cardholders to verify if the transaction became proper or fraudulent. The investigators offer a remark to the automatic device that is used to teach and replace theset of rules to ultimately enhance the fraud-detection overall performance over time.

Fraud detection strategies are constantly advanced to guard criminals in adapting to their fraudulent strategies. These frauds arelabeled as: Credit Card Frauds: Online and Offline

- Card Theft
- Account Bankruptcy
- Device Intrusion
- Application Fraud
- Counterfeit Card
- Telecommunication Fraud

Some of the currently used approaches to detection of such fraud are:

- Artificial Neural Network
- Fuzzy Logic
- Genetic Algorithm
- Logistic Regression
- Decision tree
- Support Vector Machines
- Bayesian Networks
- Hidden Markov Model
- K-Nearest Neighbour

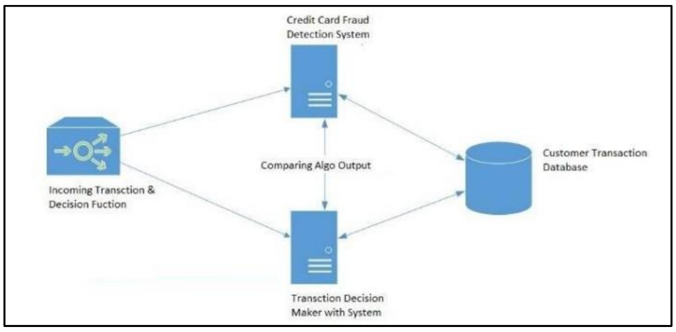
II. Literature review

Fraud because illegal or criminal deception is intended to result in financial or private gain. Any deliberate act contrary to law, regulation or concealment for the purpose of obtaining an illicit economic advantage. A large amount of literature on anomaly or fraud detection in this area has been published and is available to the public. A complete study conducted with the help of Clifton Phua and his friends found that the strategies employed in the field included information mining his application, automated fraud detection, and unwanted detection. I understand. In all other articles, Suman, his GJUS&T research fellow at Hisar HCE, Credit He provided strategies such as supervised and unsupervised learning for card fraud detection. These strategies and algorithms have achieved surprising success in some areas, but do not provide a persistent and consistent method of fraud detection. A similar research area was provided with the help of Wen-Fang YU and Na Wang. It used outlier mining, outlier detection mining, and distance summation algorithms to correctly predict fraudulent transactions in credit card transaction emulation tests. 1 Regular Industrial Bank. Outlier mining is an area of information mining primarily used in the financial and network sectors. It provides the ability to detect devices that may be unrelated to the main engine (i.e. non-genuine transactions). They took features of customer behavior and based primarily on the prices of those features, calculated this distance between the discovered price of that feature and its given price. The unconventional strategy that accompanies hybrid information mining/complex community class rule sets is primarily based on community reconstruction rule sets that contain increasing expressions of deviations from a single example to create a real card transaction information set. Illegal time can be understood. Allow referral authority. In general, medium-sized online transactions have proven to be greener. We are also focusing on development from a completely new aspect.

An attempt was made to improve the interaction of warnings and comments when fraudulent transactions occur. In the event of a fraudulent transaction, a warning will be sent to authorized machines, and a comment may be sent denying the ongoing transaction. It has proven itself right by identifying fraudulent transactions and minimizing the scope of false alerts. Nevertheless, it has been observed using class problems with variable misclassification costs

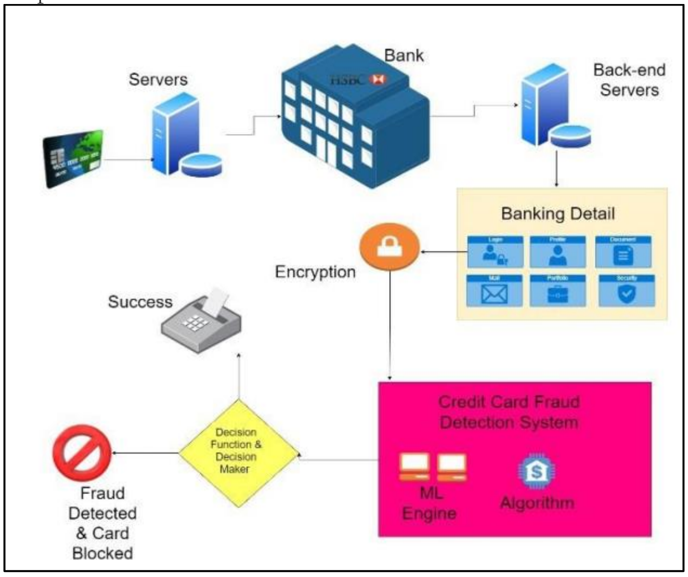
I. METHODOLOGY

The technique that this paper proposes, makes use of the modern day system mastering algorithms to stumble on anomalous activities, known as outliers. The fundamental hard structure diagram may be represented with the subsequent figure:



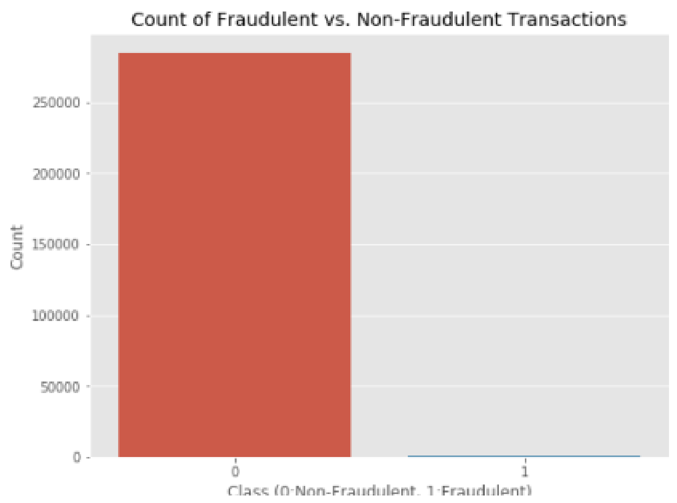
Credit Card Fraud Detection System
Incoming Transction & Decision Fuction
Comparing Algo Output
Customer Transaction Database
Transction Decision Maker with System

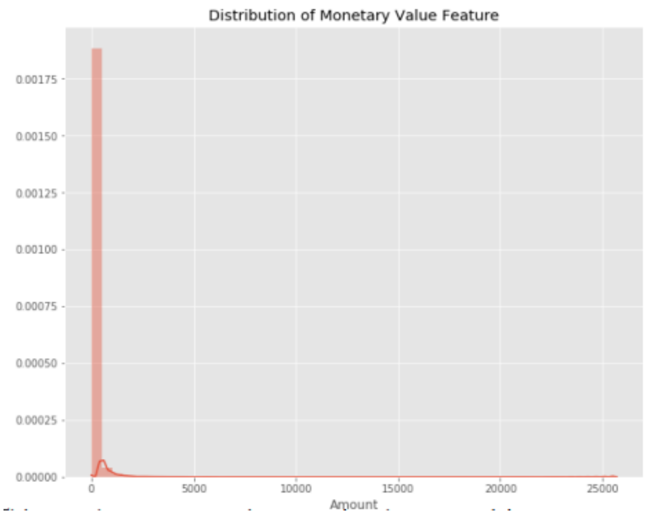
When checked out in element on a bigger scale at the side of actual existence elements, the entire structure diagram may be representedas follows:



Servers
Bank
Back-end Servers
Banking Detail
Encryption
Success
Decision Function & Decision Maker
Fraud Detected & Card Blocked
Credit Card Fraud Detection System
ML Engine
Algorithm

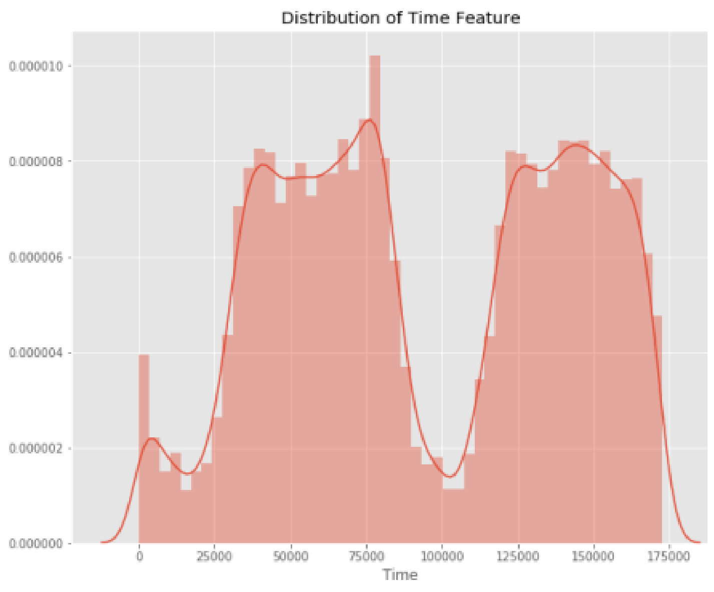
First of all, we received our dataset from Kaggle, a facts evaluation internet site which gives datasets. Inside this dataset, there are 31 columns out of which 28 are named as v1-v28 to guard touchy facts. The different columns constitute Time, Amount and Class. Time indicates the time hole amongthe primary transaction and the subsequent one. Amount is thequantity of cash transacted. Class zero represents a legitimate transaction and 1 represents a fraudulent one. We plot extraordinary graphs to test for inconsistencies withinside the dataset and to visually recognize it

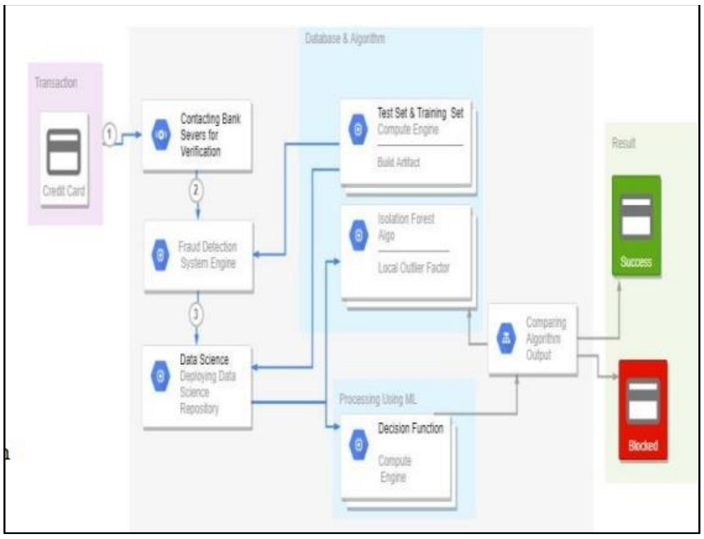




This graph shows that the number of fraudulent transactions is much lower than the legitimate ones.

This graph represents the quantity that turned into transacted. A majority of transactions are incredibly small and handiest a handful of them come near the most transacted amount. After checking this dataset, we plot a histogram for each column. This is performed to get a graphical rep. of the dataset which may be used to affirm that there aren't anyt any lacking any values withinside the dataset. This is performed to make sure that we don’t require any lacking fee imputation and the device getting to know algorithms can method the dataset smoothly.





This graph indicates the instances at which transactions had been accomplished inside days. It may be visible that the least wide variety of transactions had been made at some stage in night time time and maximum at some stage in the days.

Importing all the libraries:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
```

Loading and reading the Dataset:

```
df = pd.read_csv("ml creditcard.csv")
df
```

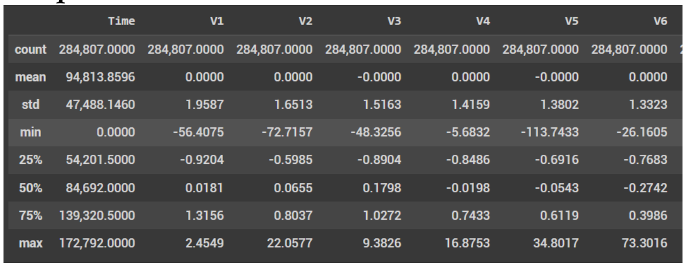
Identifying Class:

```
df.Class.unique()
```

Describing the Data:

```
df.describe()
```

Output:

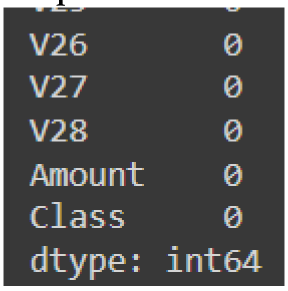


	Time	V1	V2	V3	V4	V5	V6
count	284,807.0000	284,807.0000	284,807.0000	284,807.0000	284,807.0000	284,807.0000	284,807.0000
mean	94,813.8596	0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
std	47,488.1460	1.9587	1.6513	1.5163	1.4159	1.3802	1.3323
min	0.0000	-56.4075	-72.7157	-48.3256	-5.6832	-113.7433	-26.1605
25%	54,201.5000	-0.9204	-0.5985	-0.8904	-0.8486	-0.6916	-0.7683
50%	84,692.0000	0.0181	0.0655	0.1798	-0.0198	-0.0543	-0.2742
75%	139,320.5000	1.3156	0.8037	1.0272	0.7433	0.6119	0.3986
max	172,792.0000	2.4549	22.0577	9.3826	16.8753	34.8017	73.3016

Identifying any NaN values:

```
df.isnull().sum()
```

Output: No null values detected



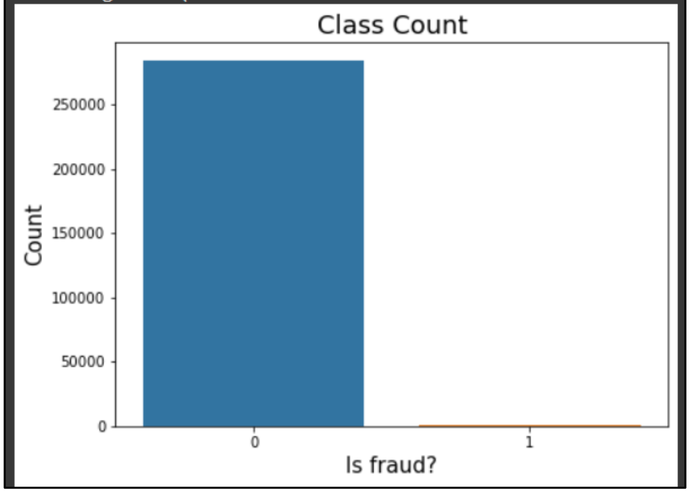
```
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Determining Fraud data:

```
#Lets start looking the difference by No
rmal and Fraud transactions
print("Distribuition of Normal(0) and Fr
auds(1): ")
print(df["Class"].value_counts())

plt.figure(figsize=(7,5))
sns.countplot(df['Class'])
plt.title("Class Count", fontsize=18)
plt.xlabel("Is fraud?", fontsize=15)
plt.ylabel("Count", fontsize=15)
plt.show()
```

Output:

```
Distribuition of Normal(0) and Frauds(1):
0    284315
1       492
Name: Class, dtype: int64
C:\Users\Devyansh Yadav\anaconda3\lib\site-packages\seaborn\_c
  warnings.warn(
```

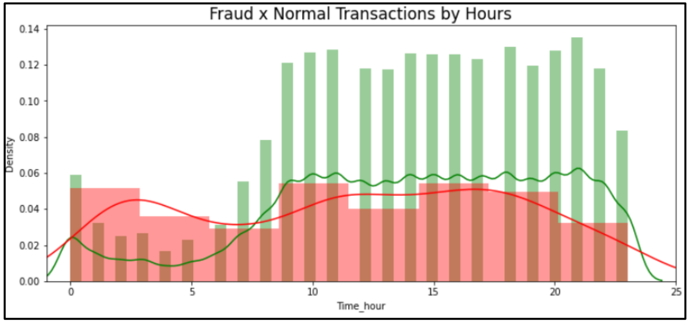


Time Features and some Feature Engineering

```
timedelta = pd.to_timedelta(df['Time'],
unit='s')
df['Time_min'] = (timedelta.dt.component
s.minutes).astype(int)
df['Time_hour'] = (timedelta.dt.componen
ts.hours).astype(int)
```

```
#Exploring the distribuition by Class ty
pes throught hours and minutes
plt.figure(figsize=(12,5))
sns.distplot(df[df['Class'] == 0]["Time_
hour"],color='g')
sns.distplot(df[df['Class'] == 1]["Time_
hour"],color='r')
plt.title('Fraud x Normal Transactions b
y Hours', fontsize=17)
plt.xlim([-1,25])
plt.show()
```

Output



Print the amount details for Fraudulent Transaction and Normal Transaction:

```
#To clearly the data of frauds and no fr
auds
df_fraud = df_credit[df_credit['Class']
== 1]
df_normal = df_credit[df_credit['Class']
 == 0]

print("Fraud transaction statistics")
print(df_fraud["Amount"].describe())
print("\nNormal transaction statistics")
print(df_normal["Amount"].describe())
```

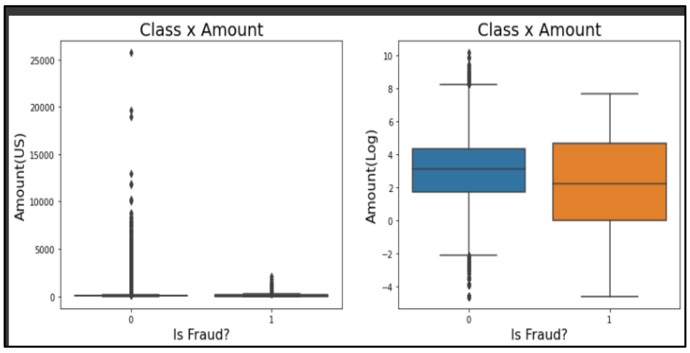
Output:

```
Fraud transaction statistics
count      492.0000
mean       122.2113
std        256.6833
min          0.0000
25%          1.0000
50%          9.2500
75%        105.8900
max      2,125.8700
Name: Amount, dtype: float64

Normal transaction statistics
count    284,315.0000
mean          88.2910
std          250.1051
min            0.0000
25%            5.6500
50%           22.0000
75%           77.0500
max       25,691.1600
Name: Amount, dtype: float64
```

As we are able to honestly observe from this, the common Money transaction for the fraudulent ones is more. This makes this trouble vital to deal with.

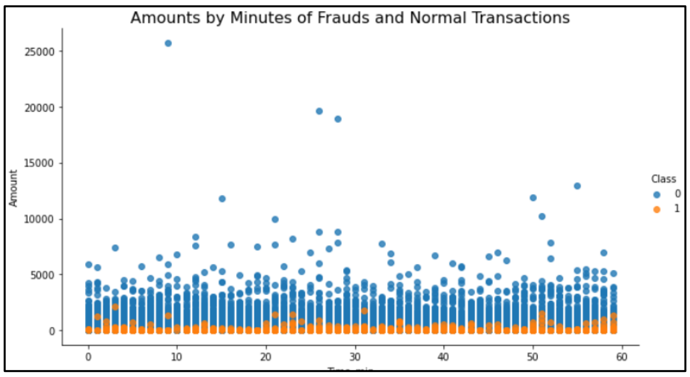
Feature engineering to a better visualization of the values

```
df_credit['Amount_log'] = np.log(df_cred
it.Amount + 0.01)
```

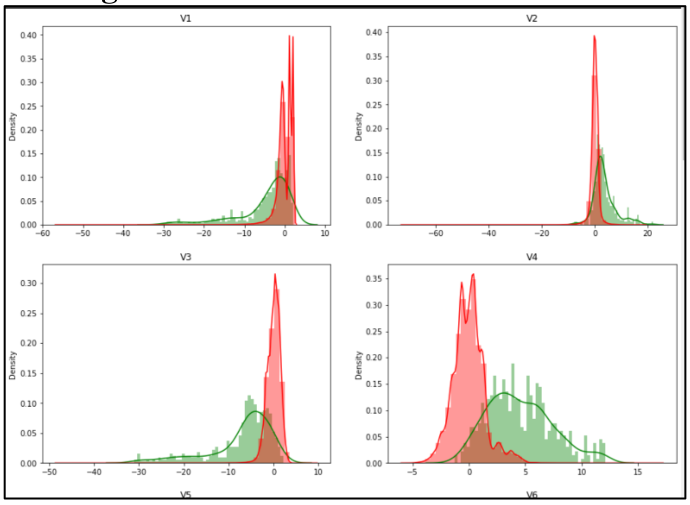


Looking the Amount and time distribution of FRAUD transactions

```
ax = sns.lmplot(y="Amount", x="Time_min"
, fit_reg=False,aspect=1.8,data=df_credi
t, hue='Class')
plt.title("Amounts by Minutes of Frauds
and Normal Transactions",fontsize=16)
plt.show()
```



Looking the V's features



Feature Selection:

```
#We will select the variables where frau
d class have a interesting behavior and
might can help us predict

df_credit = df_credit[["Time_hour","Time
_min","V2","V3","V4","V9","V10","V11","V
12","V14","V16","V17","V18","V19","V27",
"Amount","Class"]]
```

Separating X and Y values:

```
X = df_credit.drop(["Class"], axis=1).va
lues
y = df_credit["Class"].values
```

Training and Testing Data Bifurcation:

We will divide the dataset into two groups. One for training the
model and the other for testing our trained model's performance.

```
from sklearn.model_selection import trai
n_test_split
```

```
X_train, X_test, y_train, y_test = train
_test_split(X, y, random_state=2, test_s
ize=0.20)
```

Building a Random Forest Model using skicit learn:

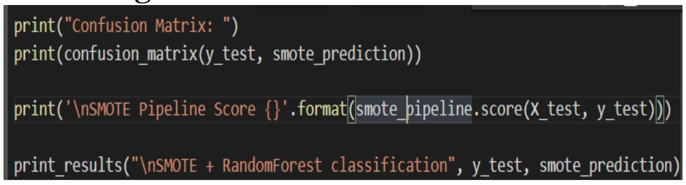
```
from sklearn.ensemble import RandomFores
tClassifier
from sklearn.pipeline import make_pipeli
ne
```

```
classifier = RandomForestClassifier
```

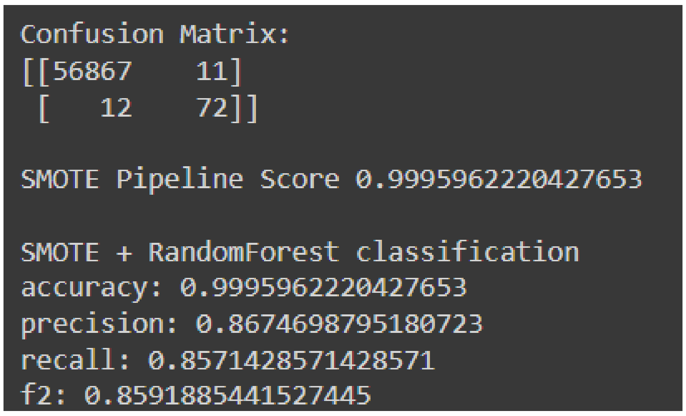
```
# build model with SMOTE imblearn
smote_pipeline = make_pipeline_imb(SMOTE
(random_state=4), \
                                   class
ifier(random_state=42))
```

```
smote_model = smote_pipeline.fit(X_train
, y_train)
```

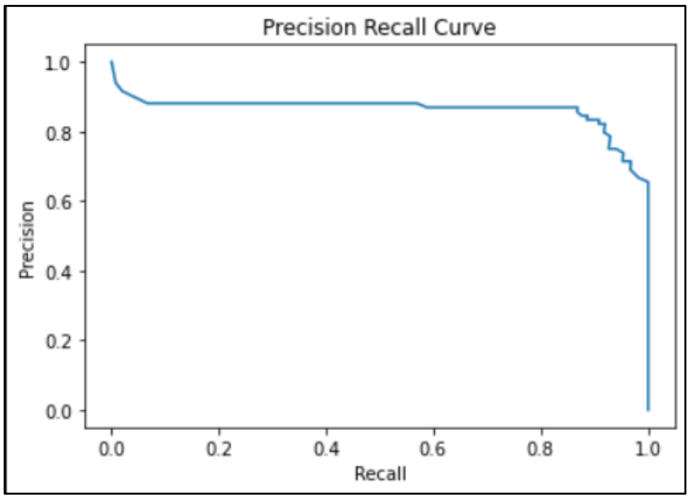
Evaluating the model SMOTE + Random Forest



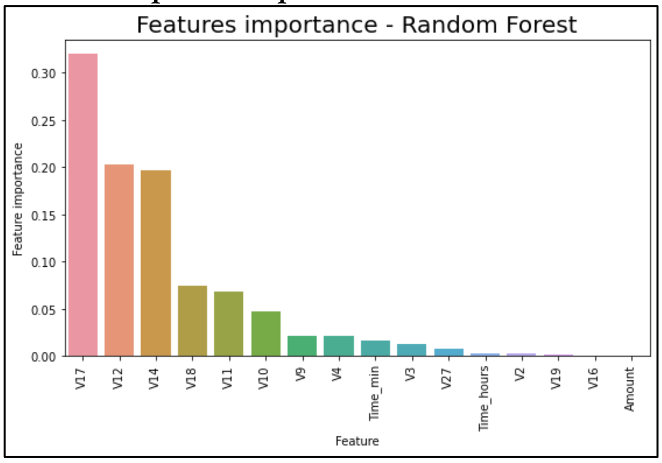
```
print("Confusion Matrix: ")
print(confusion_matrix(y_test, smote_prediction))

print('\nSMOTE Pipeline Score {}'.format(smote_pipeline.score(X_test, y_test)))

print_results("\nSMOTE + RandomForest classification", y_test, smote_prediction)
```



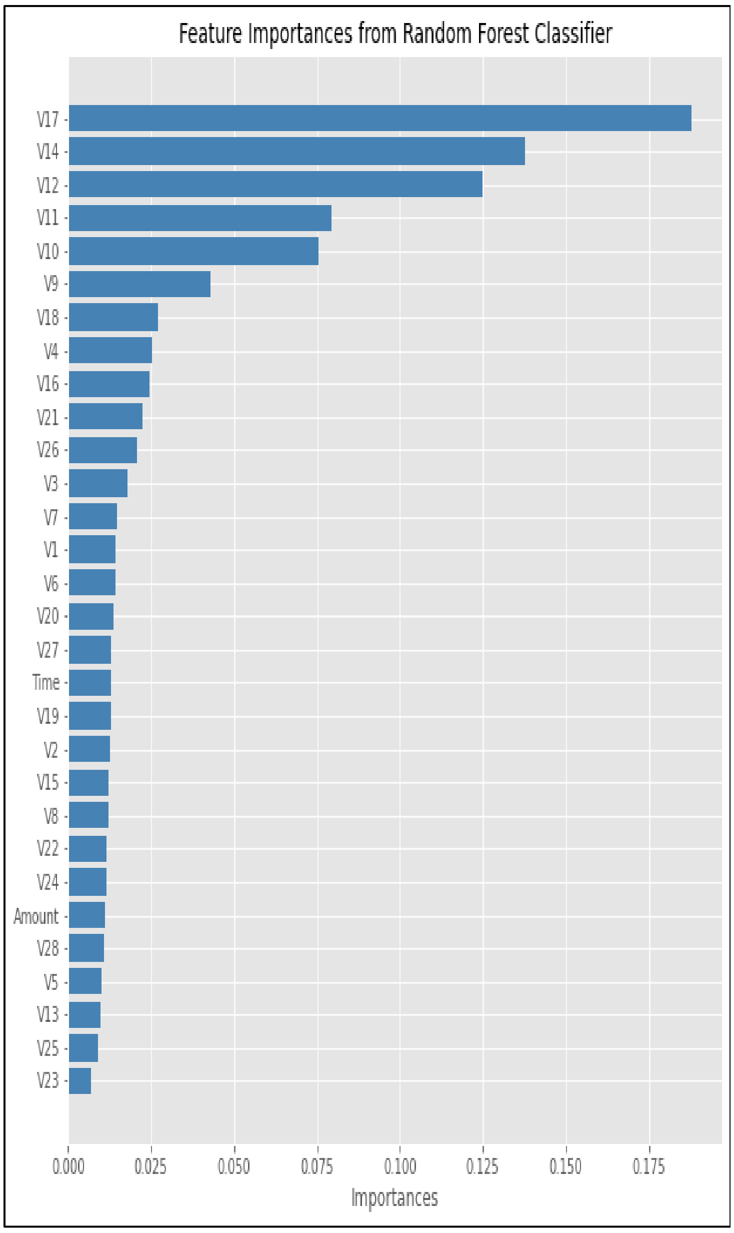
```
Confusion Matrix:
[[56867    11]
 [   12    72]]

SMOTE Pipeline Score 0.9995962220427653

SMOTE + RandomForest classification
accuracy: 0.9995962220427653
precision: 0.8674698795180723
recall: 0.8571428571428571
f2: 0.8591885441527445
```

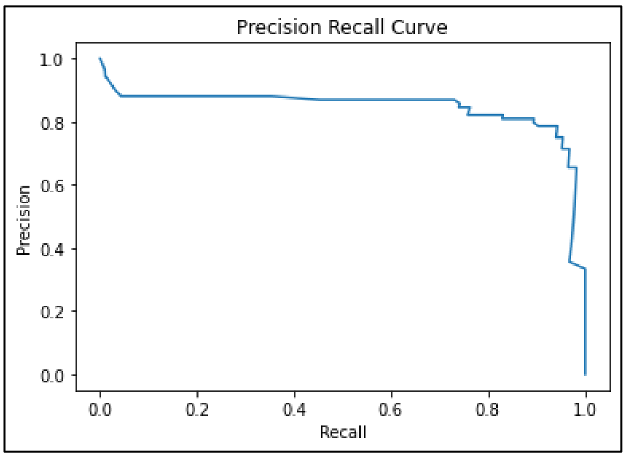


Feature importance plot

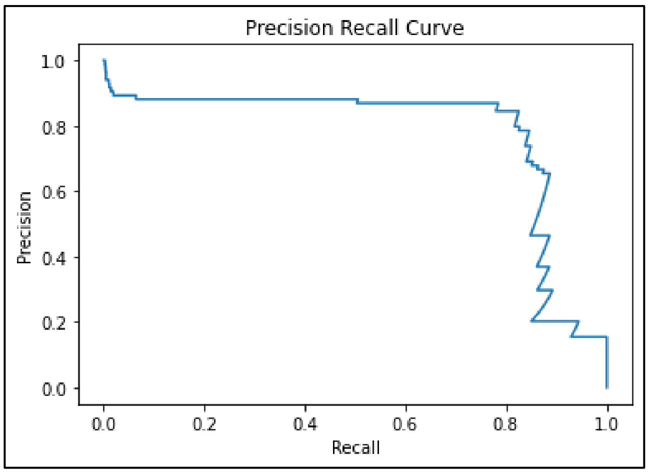


ROC CURVE - Random Forest





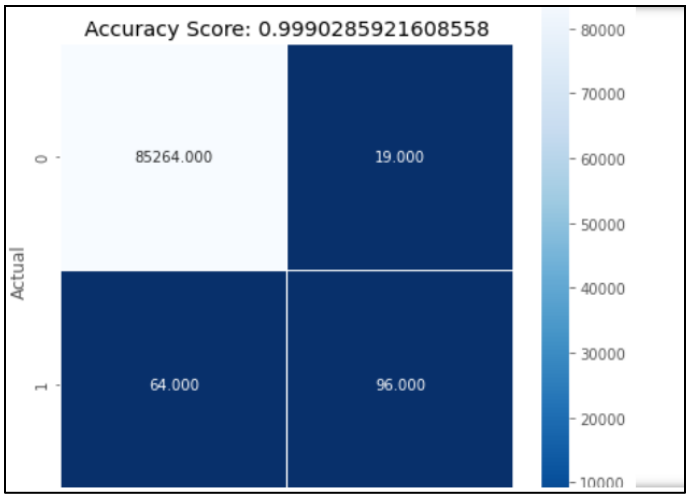
Precision Recall Curve of Logistic Regression



LOCAL OUTLIER FACTOR

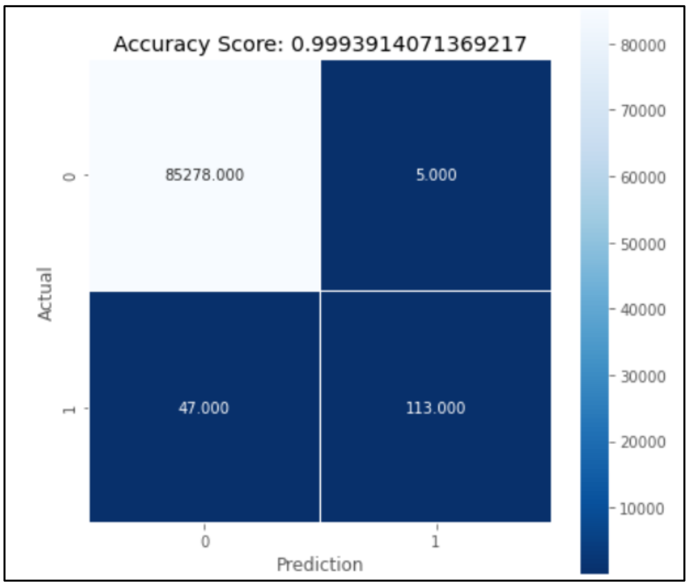
```
from sklearn.ensemble import
IsolationForest

# FIT THE MODEL

model=IsolationForest(n_estimators=1000,
max_samples='auto',
contamination=float(0.1),max_features=1.0)
model.fit(df[['Amount']])
```

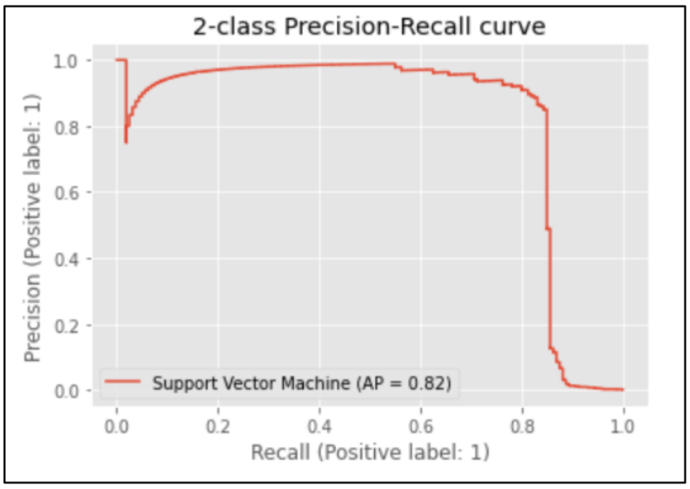
Logistic Regression Classifier:



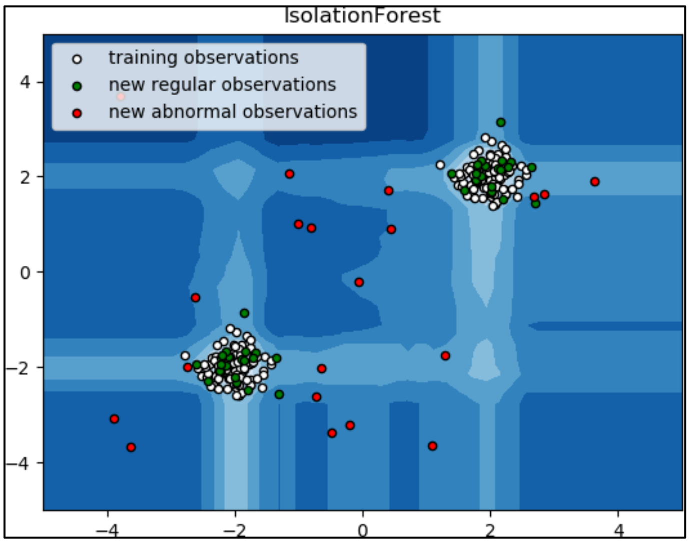
Support Vector Machine (SVM)

```
from sklearn.svm import SVC

svc = SVC(kernel='rbf', C = 1, gamma='scale')
svc.fit(train_x, train_y)

pred_y_svc = svc.predict(test_x)
```

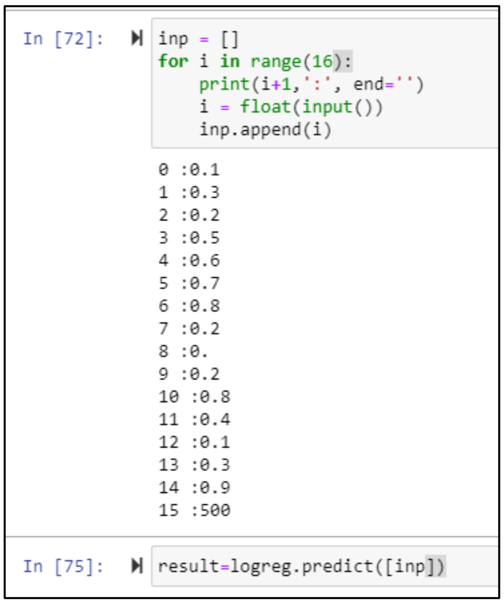




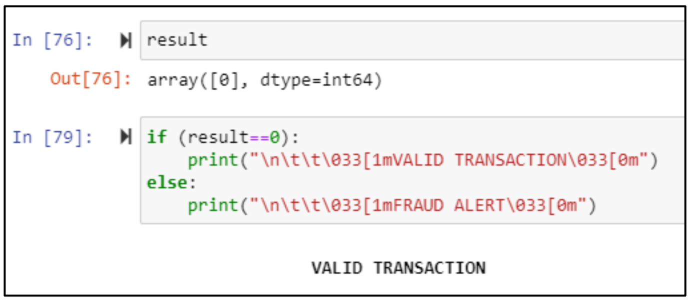
On plotting the results of Isolation Forest algorithm, we get the figure:



Implementation:



```
In [72]:  inp = []
          for i in range(16):
              print(i+1,':', end='')
              i = float(input())
              inp.append(i)

          0 :0.1
          1 :0.3
          2 :0.2
          3 :0.5
          4 :0.6
          5 :0.7
          6 :0.8
          7 :0.2
          8 :0.
          9 :0.2
          10 :0.8
          11 :0.4
          12 :0.1
          13 :0.3
          14 :0.9
          15 :500

In [75]:  result=logreg.predict([inp])
```



```
In [76]:  result

Out[76]:  array([0], dtype=int64)

In [79]:  if (result==0):
              print("\n\t\t\033[1mVALID TRANSACTION\033[0m")
          else:
              print("\n\t\t\033[1mFRAUD ALERT\033[0m")

                  VALID TRANSACTION
```

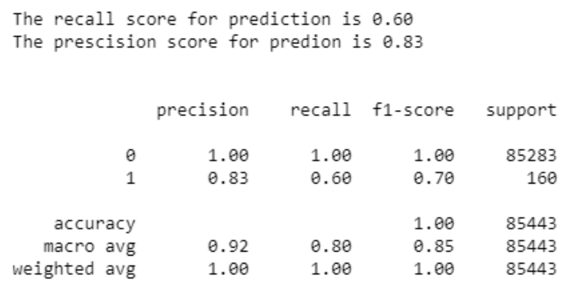
Result:

The code prints out the quantity of fake positives it detected and compares it with the real values. This is used to calculate the accuracy rating and precision of the algorithms. The fraction of records we used for quicker trying out is 10% of the complete dataset. The whole dataset is likewise used on the cease and each the consequences are printed. These consequences in conjunction with the category file for every set of rules is given with inside the output as follows, wherein magnificence zero manner the transaction became decided to be legitimate and 1 manner it became decided as a fraud transaction. This end result matched in opposition to the magnificence values to test for fake positives. Results whilst 10% of the dataset is used:

Dummy Classifier:

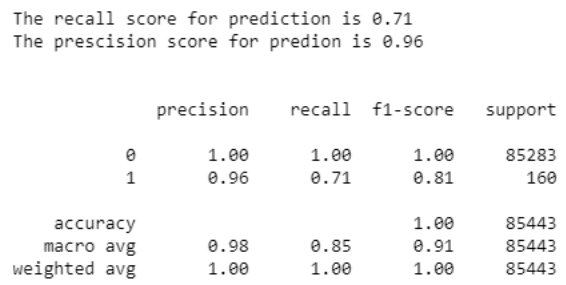
```
[[85131   152]
 [  159     1]]

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85283
           1       0.01      0.01      0.01       160

    accuracy                           1.00     85443
   macro avg       0.50      0.50      0.50     85443
weighted avg       1.00      1.00      1.00     85443
```

Logistic Regression Classifier:



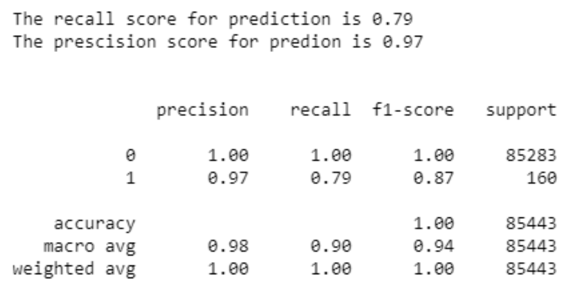
```
The recall score for prediction is 0.60
The prescision score for predion is 0.83

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85283
           1       0.83      0.60      0.70       160

    accuracy                           1.00     85443
   macro avg       0.92      0.80      0.85     85443
weighted avg       1.00      1.00      1.00     85443
```

Support Vector Machine (SVM)



```
The recall score for prediction is 0.71
The prescision score for predion is 0.96

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85283
           1       0.96      0.71      0.81       160

    accuracy                           1.00     85443
   macro avg       0.98      0.85      0.91     85443
weighted avg       1.00      1.00      1.00     85443
```

Random Forest Classifier:



```
The recall score for prediction is 0.79
The prescision score for predion is 0.97

              precision    recall  f1-score   support

           0       1.00      1.00      1.00     85283
           1       0.97      0.79      0.87       160

    accuracy                           1.00     85443
   macro avg       0.98      0.90      0.94     85443
weighted avg       1.00      1.00      1.00     85443
```

Conclusion:

In conclusion, a credit card fraud detection system was created using four different classifiers: Dummy, Logistic Regression, Support Vector Machine (SVM), and Random Forest. The system was trained and tested on a dataset consisting of both fraudulent and non-fraudulent transactions, including dead transactions.

The results showed that all four classifiers performed extremely well in detecting fraudulent transactions, with accuracy scores ranging from 0.999 to 0.9996. This high level of accuracy suggests that the system is able to accurately identify fraudulent transactions and prevent unauthorized charges to credit cards.

Moreover, the inclusion of dead transactions in the dataset helped to identify transactions that were not fraudulent but also not useful for analysis. The system was able to distinguish between these dead transactions and actual fraudulent transactions, which is an important aspect of credit card fraud detection.

Overall, this credit card fraud detection system is a valuable tool for financial institutions and consumers alike in preventing fraudulent activity and protecting against unauthorized charges. The high accuracy scores achieved by all four classifiers demonstrate the system's effectiveness and reliability in detecting fraudulent transactions.

Future Enhancement:

While we couldn't attain out intention of 100 accuracy in fraud detection, we did turn out to be growing a machine that can, with sufficient time and records, get very near that intention. As with the sort of assignment, there may be a few room for development here. The very nature of this assignment lets in for a couple of algorithms to be included collectively as modules and their outcomes may be blended to boom the accuracy of the very last result. This version can similarly be stepped forward with the addition of extra algorithms into it. However, the output of those algorithms desires to be withinside the identical layout because the others. Once that situation is satisfied, the modules are smooth to feature as completed withinside the code. This presents a brilliant diploma of modularity and flexibility to the assignment. More room for development may be determined withinside the dataset. As verified before, the precision of the algorithms will increase whilst the scale of dataset is increased. Hence, extra records will virtually make the version extra correct in detecting frauds and decrease the variety of fake positives. However, this calls for authentic guide from the banks themselves.

References:

[1] "Credit Card Fraud Based on Transaction Behaviour -by John Richard D. Kho, Larry A. Vea" published by Proc. of the 2017 IEEE Region 10 Conference (TENCON), Malaysia, November 5-8, 2017

[2] CLIFTON PHUA1, VICENT LEE1, KATE SMITH1 & ROSS GAYLER2 " A Comprehensive Survey of Data Mining-based Fraud Detection Research" published by School of Business Systems, Faculty of Information Technology, Monash University, Wellington Road, Clayton, Victoria 3800, Australia

[3] "Survey Paper on Credit Card Fraud by Sman" , Research Scholar, GJUS&T Hisar HCE, Sonepat published by International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 3, March 2014

[4] "Research on Credit Card Fraud Detection Model Based on Distance Sum – by Wen-Fang YU and Na Wag" published by 2009 International Joint Conference on Intelligence

[5] "Credit Card Fraud Detection through Parenclitic Network Analysis- By Massimiliano Zanin, Migu Romance, Regino Criado, and SantiagoMoral" published by Hindawi Complexity Volume 2018, Article ID 5764370, 9 pages

[6] "Credit Card Fraud Detection: A Realistic Modeling and a Novel Learning Strategy" published by IEEE TRANSACTIONS ON NEURAL NETWORKS AND SYSTEMS, VOL. 29, NO. 8, AUGUST 2018

[7] "Credit Card Fraud-by Ishu Trivedi, Monika, Mrigya, Mridushi" published by International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016

[8] David J.Wetson,Daid J.Hand,M Adams,Witrow and Piotr

Jusczak “Plastic Card Fraud Detection using Peer Group Analysis” Springer, Issue 2008.

Link:

https://drive.google.com/drive/folders/1J0DfP7R_wNCbh9QxdaOjwLOKncttE9qn?usp=sharing

Dataset:

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