Credit Card Fraud Detection

- INSTRUCTOR: Prof. Ashish Kumar Sahu

TEAM MEMBERS:

- Praneeth Gadipudi-19MIM10044
- Mada Sai SURYA-19MIM10095
- Kadiyala MEGHANATH-19MIM10097
- ABHISHEK SWAMY-19MIM10117

ZEROTH REVIEW

Topics

- Project Introduction
- Existing work with limitations
- Problem Statement
- Proposed work
- Methodology
- Real time usage
- Hardware & software requirements
- system architecture

Project Introduction

- Fraud' in credit card transactions is unauthorized and unwanted usage of an account by someone other than the owner of that account.
- Fraud detection involves monitoring the activities of populations of users in order to estimate, perceive or avoid objectionable Behavior , which consist of fraud, intrusion, and defaulting.
- Machine learning algorithms are employed to analyze all the authorized transactions and report the suspicious ones.

Existing work

- Research on Credit Card Fraud Detection Model Based on Distance Sum.

  “Wen-Fang YU, Na Wang”

Along with increasing credit cards and growing trade volume in China, credit card fraud rises sharply. How to enhance the detection and prevention of credit card fraud becomes the focus of risk control of banks. It proposes a credit card fraud detection model using outlier detection based on distance sum according to the infrequency and unconventionality of fraud in credit card transaction data, applying outlier mining into credit card fraud detection. Experiments show that this model is feasible and accurate in detecting credit card fraud.

Problem Statement

The high amount of losses due to fraud and the awareness of the relation between loss and the available limit has to be reduced. The fraud has to be deducted in real time and the number of false alert has to be minimized.

Using genetic algorithm the fraud is detected and the false alert is minimized and it produces an optimized result. The fraud is detected based on the customers behavior. A new classification problem which has a variable misclassification cost is introduced. Here the genetic algorithms is made where a set of interval valued parameters are optimized

Proposed Work

Methodology

- The approach that this paper proposes, uses the latest machine learning algorithms to detect anomalous activities, called outliers.
- First of all, we obtained our dataset, from a data analysis website which provides datasets. Inside this dataset, there are 31 columns out of which 28 are named as v1-v28 to protect sensitive data(like amount in account, credit card number).The other columns represent Time, Amount and Class. Time shows the time gap between the first transaction and the following one. Amount is the amount of money transacted.
- Class 0 represents a valid transaction and 1 represents a fraudulent one.

Contd.....

- We plot different graphs to check for inconsistencies in the dataset and to visually comprehend it.
- This data is fit into a model and the following outlier detection modules are applied on it:

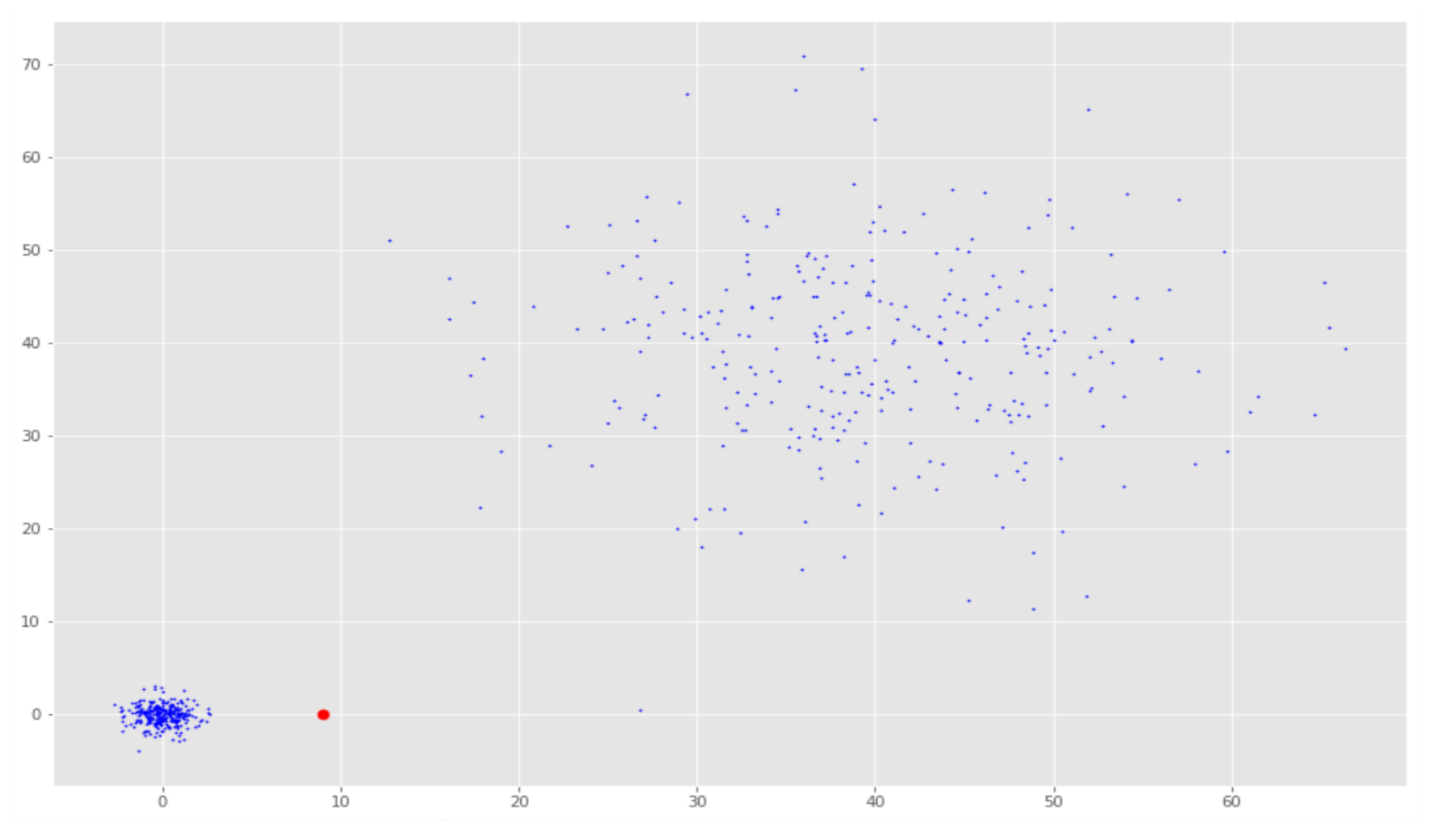
  A) Local Outlier Factor

  B) Isolation Forest Algorithm

Local Outlier Factor:

- It is an Unsupervised Outlier Detection algorithm.
- 'Local Outlier Factor' refers to the anomaly score of each sample. It measures the local deviation of the sample data with respect to its neighbours.
- More precisely, locality is given by k-nearest neighbours , whose distance is used to estimate the local data.

Graph of Local Outlier Factor



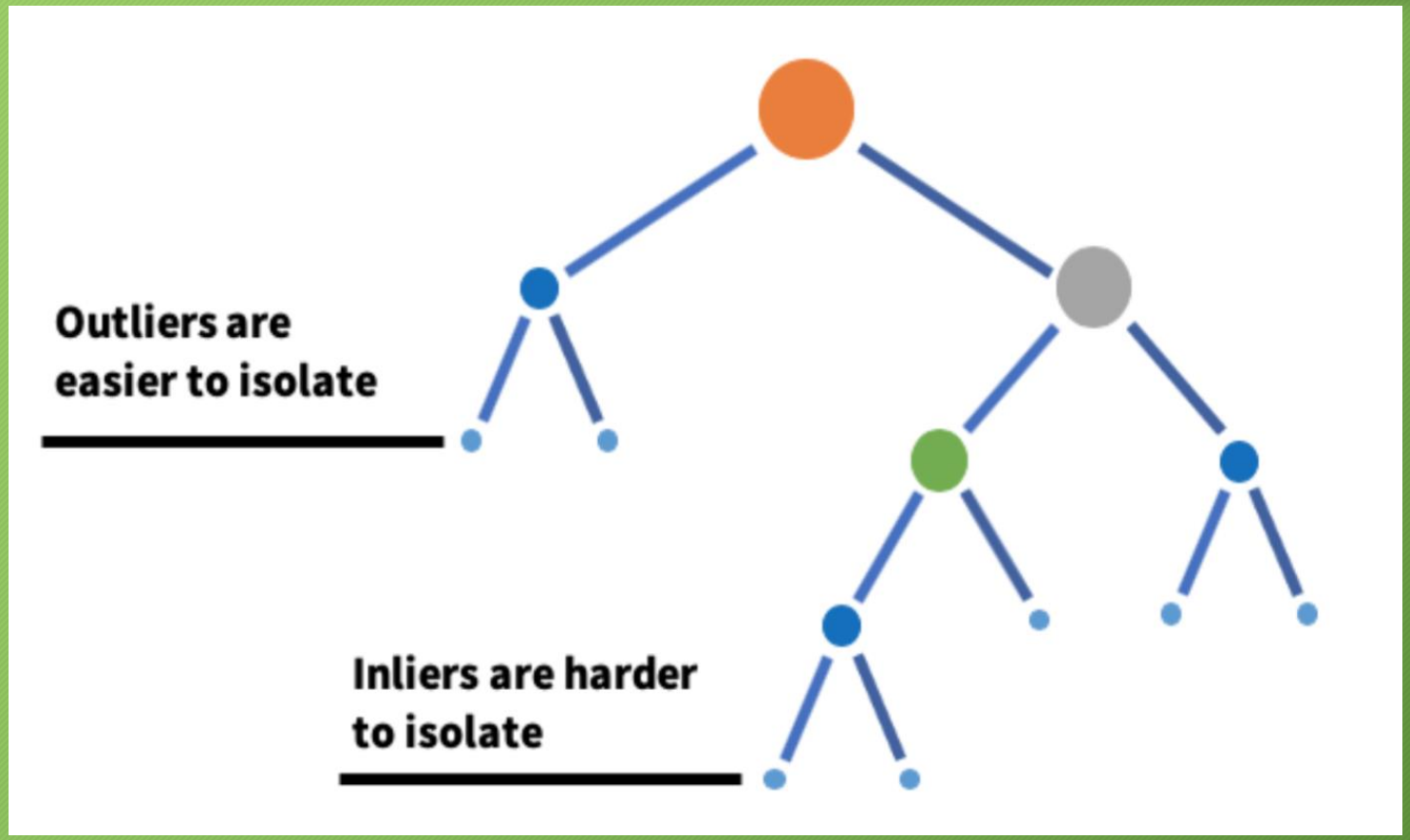
Isolation Forest Algorithm

One of the newest techniques to detect anomalies is called Isolation Forests. The algorithm is based on the fact that anomalies are data points that are few and different. As a result of these properties, anomalies are susceptible to a mechanism called isolation.

This method is highly useful and is fundamentally different from all existing methods. It introduces the use of isolation as a more effective and efficient means to detect anomalies than the commonly used basic distance and density measures. Moreover, this method is an algorithm with a low linear time complexity and a small memory requirement. It builds a good performing model with a small number of trees using small sub-samples of fixed size, regardless of the size of a data set.

Typical machine learning methods tend to work better when the patterns they try to learn are balanced, meaning the same amount of good and bad behaviors are present in the dataset.

Isolation Forest Algorithm



Real Time Usage

The number of online transactions has grown in large quantities and online credit card transactions holds a huge share of these transactions. Therefore, banks and financial institutions offer credit card fraud detection applications much value and demand.

Hardware & software requirements

Hardware:

- Processor - Intel
- RAM - 4 Gb
- Hard Disk - 260 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse

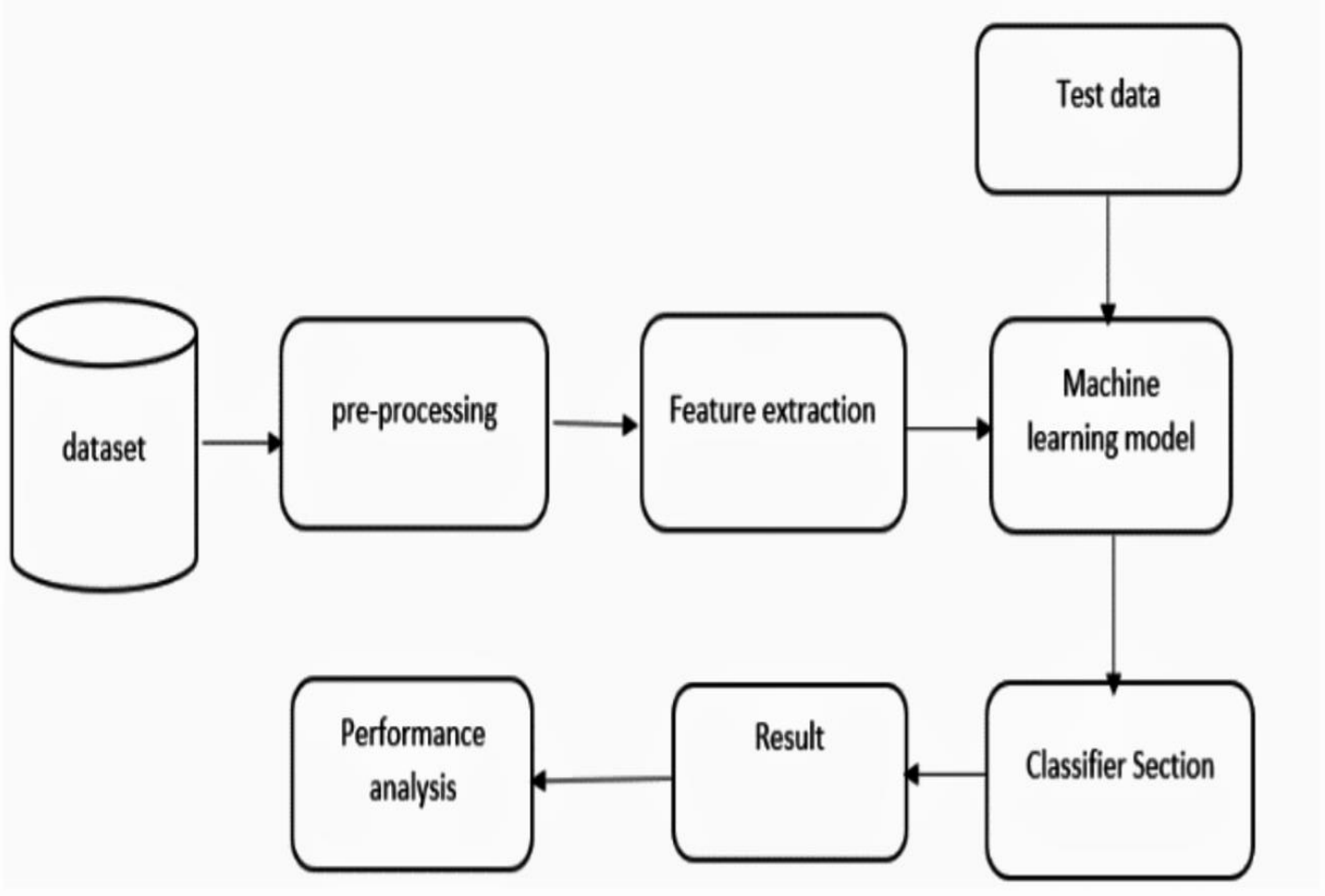
Software:

Python Anaconda

OS - Windows 7, 8 and 10 (32 and 64 bit)

System Architecture

- First the credit card dataset is taken from the source and cleaning and validation is performed on the dataset which includes removal of redundancy, filling empty spaces in columns, converting necessary variable into factors or classes then data is divided into 2 part, one is training dataset and another one is test dataset. Now the original sample is randomly partitioned into teat and train dataset.


Test data
dataset
pre-processing
Feature extraction
Machine learning model
Performance analysis
Result
Classifier Section

Bibliography

Websites

1.http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html

2.http://www.kxcad.net/cae_MATLAB/toolbox/gads/f6691.html

3.http://python.sun.com/developer/onlineTraining/Programming//front.python.html

4.http://www.faqs.org/patents/app/20100094765

Books

1.Pressman, Roger S. Software engineering

2.The Elements of Statistical Learning: Data Mining, Inference, and Prediction

3.Pattern Recognition and Machine Learning

4.Python with Machine Learning

First Review

Topics

- Literature review
- Module description
- Module workflow explanation (First two modules)
- Implementation and coding.
- Demo video.
- Snapshot of project.

Literature Review

- Fraud act as the unlawful or criminal deception intended to result in financial or personal benefit. It is a deliberate act that is against the law, rule or policy with an aim to attain unauthorized financial benefit.
- Multiple Supervised and Semi-Supervised machine learning techniques are used for fraud detection , but the aim is to overcome three main challenges with card frauds related dataset i.e., strong class imbalance, the inclusion of labelled and unlabeled samples, and to increase the ability to process a large number of transactions.

- A similar research domain was presented by Wen-Fang YU and Na Wang where they used Outlier mining, Outlier detection mining and Distance sum algorithms to accurately predict fraudulent transaction in an emulation experiment of credit card transaction data set of one certain commercial bank. Outlier mining is a field of data mining which is basically used in monetary and internet fields. It deals with detecting objects that are detached from the main system i.e. the transactions that aren't genuine.
- There have also been efforts to progress from a completely new aspect. Attempts have been made to improve the alert feedback interaction in case of fraudulent transaction. In case of fraudulent transaction, the authorized system would be alerted, and a feedback would be sent to deny the ongoing transaction.

Module Description

- **Data Preprocessing** - the collection and manipulation of items of data to produce meaningful information
- **Scoring Rule** - Percentage of fraud in transaction
- **Classification of Alerts** -that will train and update the data based on feedback and delayed samples.
- **Ranking of Alert** -rank each alert based on correctness of security question.
- **Performance Analysis**

Data Preprocessing

- Formatting: The data which is been selected may not be in a suitable format. The data may be in a file format and we may like it in relational database or vice versa.
- Cleaning: Removal or fixing of missing data is called as cleaning. The dataset may contain record which may be incomplete or it may have null values. Such records need to remove.
- Sampling: As number of frauds in dataset is less than overall transaction, class distribution is unbalanced in credit card transaction. Hence sampling method is used to solve this issue.

Scoring Rule

Percentage of fraud in transaction is called as score. This module assigns score by matching recent transaction pattern with the past transaction pattern of cardholder. If score is greater then the transaction is considered as suspicious and further proceeding is stopped. Otherwise it is moved to next module.

Implementation

This project is implemented using two Algorithms:

- Local Outlier Factor : By comparing the local values of a sample to that of its neighbors, one can identify samples that are substantially lower than their neighbors. These values are quite amanous and they are considered as outliers.
- Isolation Forest Algorithm : Recursive partitioning can be represented by a tree, the number of splits required to isolate a sample is equivalent to the path length root node to terminating node.

Coding

- We chose Python Language to code this Project, as it is easy and very powerful.
- We used Jupyter Notebooks in Anaconda to code this Project because **Jupyter Notebooks** allows for cell by cell execution of code blocks which **some** programmers find advantageous because it allows for convenient testing of blocks of code.
- We used different packages present in Python especially NumPy, MatPlotLib and Pandas.

Demo Video

Snapshot of the Project

In [1]:
```
import pandas as pd
import sklearn
import scipy
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import classification_report,accuracy_score
from sklearn.ensemble import IsolationForest
from sklearn.neighbors import LocalOutlierFactor
from sklearn.svm import OneClassSVM
from pylab import rcParams
rcParams['figure.figsize'] = 14, 8
RANDOM_SEED = 42
LABELS = ["Normal", "Fraud"]
```

In [2]:
```
data = pd.read_csv('creditcard.csv',sep=',')
data.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.1671
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206

Testing

In [30]:
```
n_outliers = len(Fraud)
for i, (clf_name,clf) in enumerate(classifiers.items()):
    #Fit the data and tag outliers
    if clf_name == "Local Outlier Factor":
        y_pred = clf.fit_predict(X)
        scores_prediction = clf.negative_outlier_factor_
    elif clf_name == "Support Vector Machine":
        clf.fit(X)
        y_pred = clf.predict(X)
    else:
        clf.fit(X)
        scores_prediction = clf.decision_function(X)
        y_pred = clf.predict(X)
    #Reshape the prediction values to 0 for Valid transactions , 1 for Fraud transactions
    y_pred[y_pred == 1] = 0
    y_pred[y_pred == -1] = 1
    n_errors = (y_pred != Y).sum()
    # Run Classification Metrics
    print("{}: {}".format(clf_name,n_errors))
    print("Accuracy Score :")
    print(accuracy_score(Y,y_pred))
    print("Classification Report :")
    print(classification_report(Y,y_pred))
```

Result and Conclusion

```
Isolation Forest: 73
Accuracy Score :
0.9974368877497279
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.26      0.27      0.26        49

    accuracy                           1.00     28481
   macro avg       0.63      0.63      0.63     28481
weighted avg       1.00      1.00      1.00     28481

Local Outlier Factor: 97
Accuracy Score :
0.9965942207085425
Classification Report :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     28432
           1       0.02      0.02      0.02        49

    accuracy                           1.00     28481
   macro avg       0.51      0.51      0.51     28481
weighted avg       1.00      1.00      1.00     28481
```

Conclusion

- We are getting Fraud cases as : 73 by using Isolation Forest Algorithm and its Accuracy score as : 0.997436877497279.
- We are getting Fraud cases as : 97 by using Local Outlier Factor Algorithm and its Accuracy Score as : 0.9965942207085425.
- Since we have better Accuracy Score for Isolation Forest Algorithm, we can conclude **73** as Fraud Cases.