

A Report on

**CREDIT CARD FRAUD DETECTION**

Under the Guidance (Mentor) of

Aishwarya Saxena

By- CH. Sai Teja

Machine Learning- Python

College: MVSR Engineering College, Hyderabad

Internship- Oct-Nov 2020



**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE NO.** |
| 1 | INTRODUCTION | 3 |
| 2 | OBJECTIVE | 3 |
| 3 | BACKGROUND | 3 |
| 4 | SOFTWARE AND HARDWARE  REQUIREMENTS | 4 |
| 5 | CODE & OUTPUT | 4-11 |
| 6 | FUTURE SCOPE | 11 |
| 7 | CONCLUSION | 12 |
| 8 | BIBLIOGRAPHY AND REFERENCE | 12 |

**INTRODUCTION:**

“Fraud detection is a set of activities that are taken to prevent money or property from being obtained through false pretenses.” Fraud can be committed in different ways and in many industries. The majority of detection methods combine a variety of fraud detection datasets to form a connected overview of both valid and non-valid payment data to make a decision. This decision must consider IP address, geolocation, device identification, “BIN” data, global latitude/longitude, historic transaction patterns, and the actual transaction information. In practice, this means that merchants and issuers deploy analytically based responses that use internal and external data to apply a set of business rules or analytical algorithms to detect fraud.

Credit card fraud is usually caused either by card owner’s negligence with his data or by a breach in a website’s security. Here are some examples:

* A consumer reveals his credit card number to unfamiliar individuals.
* A card is lost or stolen and someone else uses it.
* Mail is stolen from the intended recipient and used by criminals.
* Business employees copy cards or card numbers of its owner.
* Making counterfeit credit cards.

**OBJECTIVE:**

The main objective of Credit Card Fraud Detection is to minimize the fraudulent transactions for the merchant and to prevent losses for the issuing banks.

**BACKGROUND:**

Credit Card Fraud Detection with Machine Learning is a process of data investigation by developing a model that will provide the best results in revealing and preventing fraudulent transactions. There are a number of techniques and algorithms already have been introduced which are in use in detecting such frauds. Machine learning is also helping in this research. The decision trees and SVM algorithms are used to solve the problem in case of fraud. Also, we make use of libraries like NumPy, pandas, seaborn, matplotlib by using the tool jupyter Notebook. Machine Learning algorithms are employed for analyzing the authorized transactions and report the suspicious one.

**HARDWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Hardware Tools | Minimum Requirements |
| Processor | i5 or above |
| Hard Disk | 10GB |
| RAM | 8GB |
| Monitor | 17” colored |
| Mouse | Optical |
| Keyboard | 122 |

**SOFTWARE REQUIREMENTS:**

|  |  |
| --- | --- |
| Software Tools | Minimum Requirements |
| Platform | Windows, Linux (or) MacOS |
| Operating System | Windows, Linux (or) MacOS |
| Technology | Machine Learning-Python |
| Scripting Language | Python |
| IDE | PyCharm (and jupyter) |

**CODE:**

**Libraries Used:** NumPy, Pandas, Matplotlib, Seaborn, sklearn

**1. Importing libraries:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from matplotlib import gridspec

**2. Loading the Data:**

data = pd.read\_csv("creditcard.csv")

data.head()

**Output:**

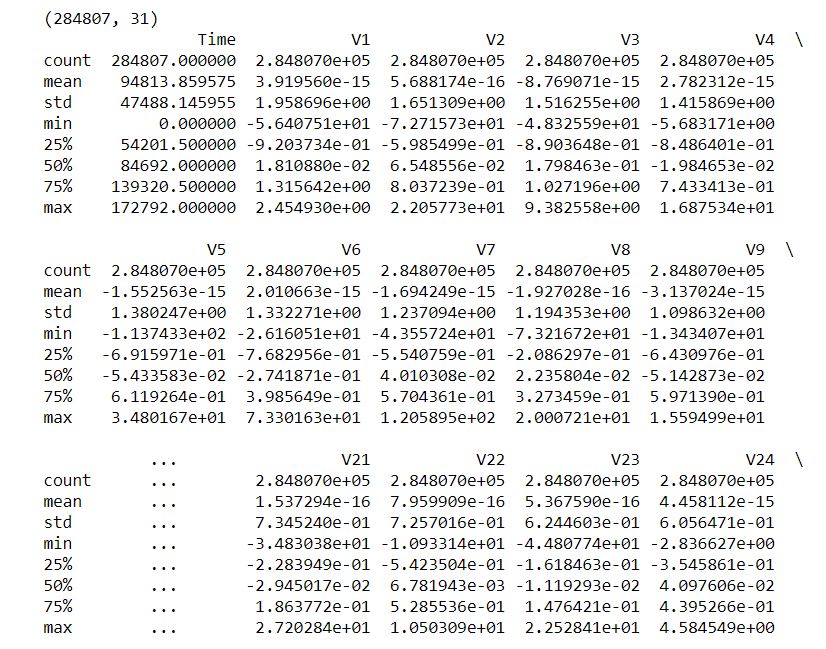


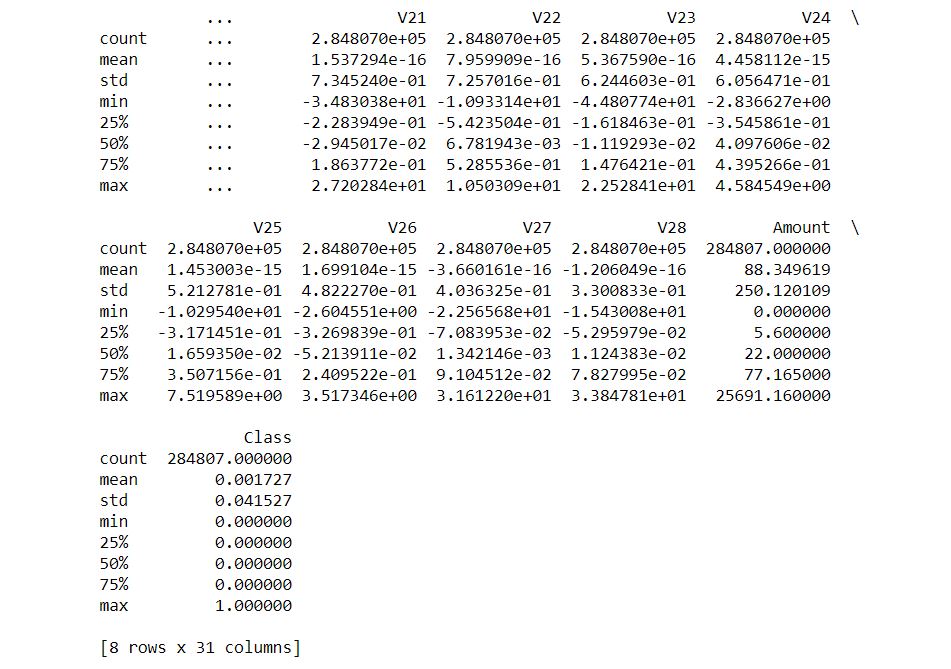
**3. Describing the Data**

print(data.shape)

print(data.describe())

**Output:**





**4. Determine number of fraud cases in dataset**

fraud = data[data['Class'] == 1]

valid = data[data['Class'] == 0]

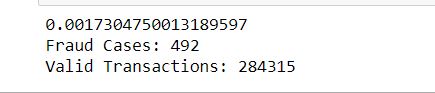
outlierFraction = len(fraud)/float(len(valid))

print(outlierFraction)

print('Fraud Cases: {}'.format(len(data[data['Class'] == 1])))

print('Valid Transactions: {}'.format(len(data[data['Class'] == 0])))

**Output:**

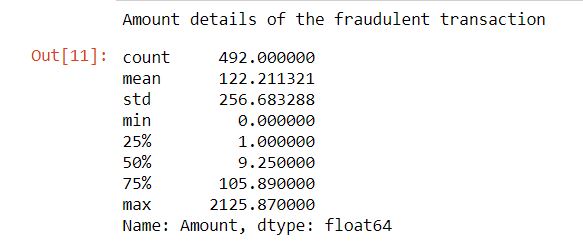


#Print the amount details for Fraudulent Transaction

print("Amount details of the fraudulent transaction")

fraud.Amount.describe()

**Output:**

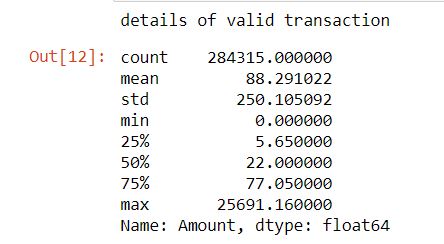


#Print the amount details for Normal Transaction

print("details of valid transaction")

valid.Amount.describe()

**Output:**



**5. Plotting the correlation Matrix:**

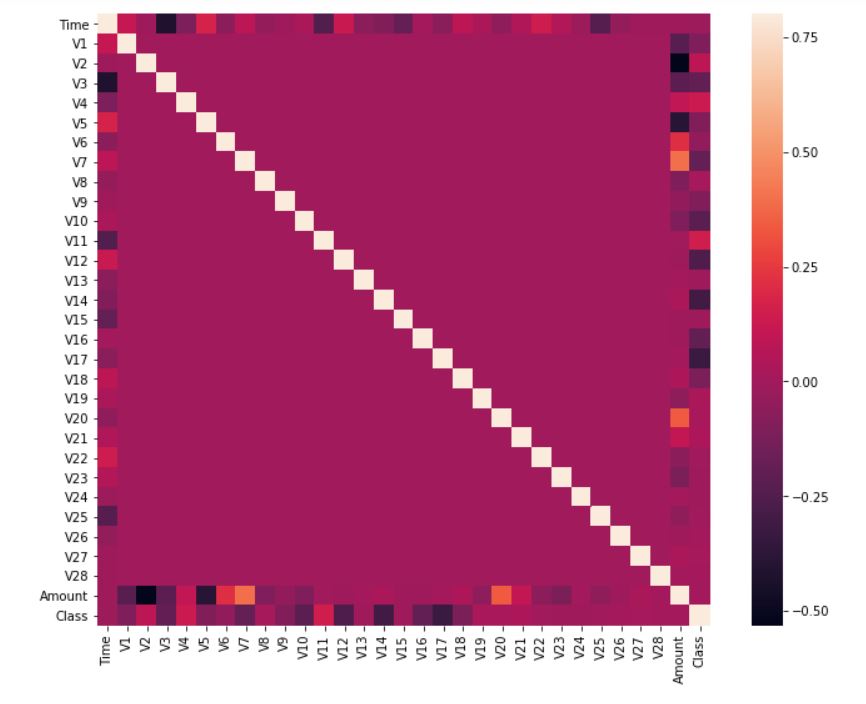
corrmat = data.corr()

fig = plt.figure(figsize = (12, 9))

sns.heatmap(corrmat, vmax = .8, square = True)

plt.show()

**Output:**



**6. Separating X and Y values**

X = data.drop(['Class'], axis = 1)

Y = data["Class"]

print(X.shape)

print(Y.shape)

xData = X.values

yData = Y.values

**Output:**

(284807, 30)

(284807,)

**7. Training and Testing Data Bifurcation**

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)

**8. Building a Random Forest Model using skicit learn**

from sklearn.ensemble import RandomForestClassifier

# random forest model creation

rfc = RandomForestClassifier()

rfc.fit(xTrain, yTrain)

# predictions

yPred = rfc.predict(xTest)

**9. Building all kinds of evaluating parameters**

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))

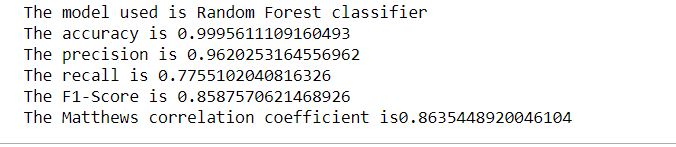
f1 = f1\_score(yTest, yPred)

print("The F1-Score is {}".format(f1))

MCC = matthews\_corrcoef(yTest, yPred)

print("The Matthews correlation coefficient is{}".format(MCC))

**Output:**



**10. Visualizing 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");

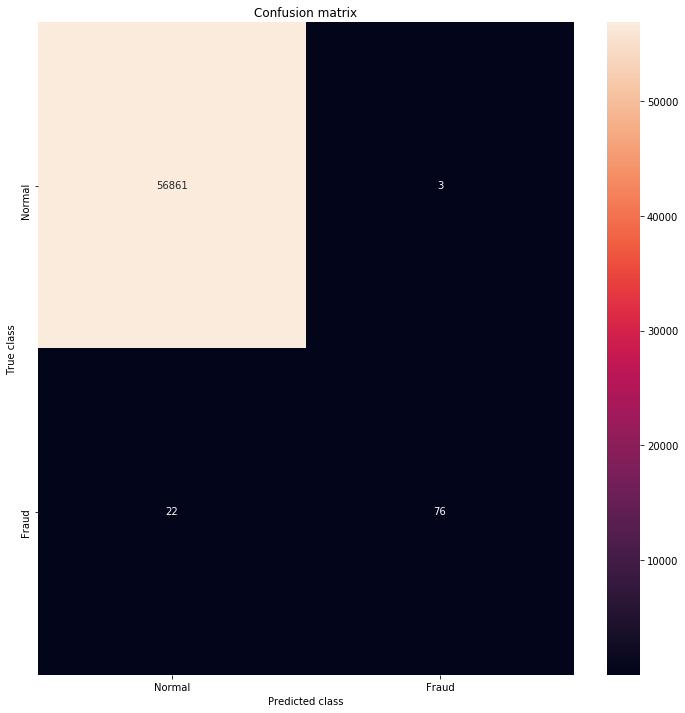
plt.title("Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

**Output:**



**FUTURE SCOPE:**

The work can be enhanced by using hybrid algorithms and artificial intelligence system. The security can be enhanced on the level on which the credit card fraud activity cannot happen even if the fraud person knows the card number and secret pin.

**CONCLUSION:**

The credit card or debit card fraud activities are increasing day by day in past years. There are many incidents are happened in presently where because of lack of knowledge the credit card users are sharing their personal details, card details and one-time password to an unknown fake call. The credit card fraud activity may happen in many ways and if any of the algorithm discussed in this study is applied by the website or business fraud detection, the probability of fraud may be minimized. There are many anti-fraud methods or applications are available to prevent the business loss. This research gives contribution towards the detecting such illegal activities using machine learning algorithms.

**REFERENCES AND BIBLIOGRAPHY:**

1. Machine Learning Methods for Analysis of Fraud Credit Card Transaction - (<https://www.ijeat.org/wp-content/uploads/papers/v8i6S/F11640886S19.pdf> )
2. Linda Delamaire (UK), Hussein Abdou (UK), John Pointon (UK), “Credit card fraud and detection techniques: a review”, Banks and Bank Systems, Volume 4, Issue 2, 2009.
3. Khyati Chaudhary, Jyoti Yadav, Bhawna Mallick, “A review of Fraud Detection Techniques: Credit Card”, International Journal of Computer Applications (0975 – 8887) Volume 45– No.1, May 2012.
4. Credit Card Fraud Detection Solutions- (<https://spd.group/machine-learning/credit-card-fraud-detection> )
5. Dataset- (<https://www.kaggle.com/mlg-ulb/creditcardfraud> )
6. ML | Credit Card Fraud Detection (<https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/>)