**TITLE:** CodTech IT Solutions Internship - Task Documentation: “CREDIT CARD FRAUD DETECTION” Using Python programming

**INTERN INFORMATION:**

**Name:** Dibjyoti Hota

**ID:** ICOD6251

**INTRODUCTION**

In today's digital age, the widespread use of credit cards has revolutionized the way we conduct financial transactions. However, this convenience comes with inherent risks, chief among them being credit card fraud. Fraudulent transactions can lead to significant financial losses for both cardholders and financial institutions, making the detection and prevention of such activities paramount.

The aim of this project is to leverage machine learning techniques to develop a robust credit card fraud detection system. By analyzing historical transaction data, our objective is to build a model capable of accurately identifying fraudulent transactions in real-time, thereby minimizing financial losses and ensuring the security of credit card users.

**Implementation**

* Python Programming: Used python to build the credit card fraudulent detection
* Numpy/pandas: use of numpy and pandas modules to store data and to calculate mathematical problems
* Responsive Design: Implement responsive design principles to ensure optimal viewing experience across desktop and mobile devices.
* User Interface Components: Utilize UI libraries for designing interactive and visually appealing components.

**CODE EXPLAINATION**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from matplotlib import gridspec

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

print(data.shape)

print(data.describe())

# 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])))

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()

# 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

* First loading of data takes place

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

* Understanding the Data

print(data.shape)

* Describing the Data

print(data.shape)

print(data.describe())

* Imbalance in the Data

Time to explain the data we are dealing with

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 the amount details for Fraudulent Transaction

print("Amount details of the fraudulent transaction")

fraud.Amount.describe()

* Print the amount details for Normal Transaction

print("details of valid transaction")

valid.Amount.describe()

* Plotting the Correlation Matrix

corrmat = data.corr()

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

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

plt.show()

* Seperating the X and the Y values

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

Y = data["Class"]

print(X.shape)

print(Y.shape)

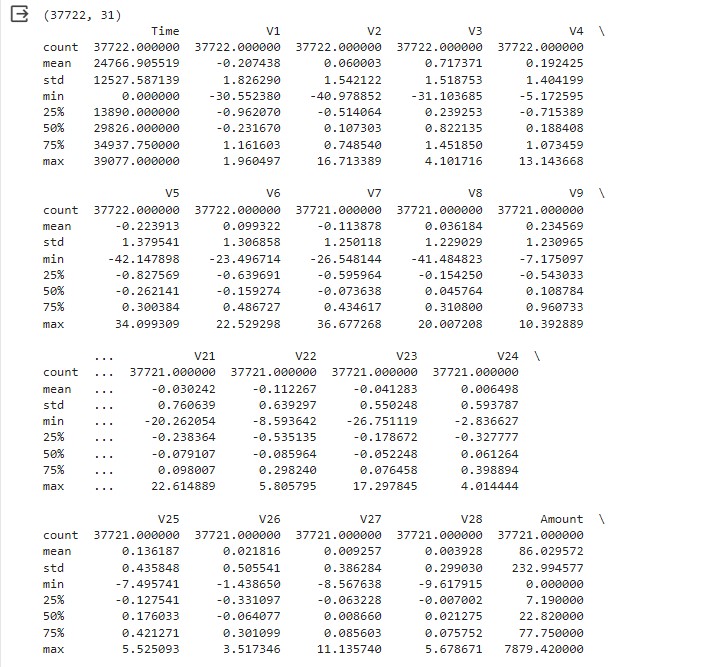
# getting just the values for the sake of processing

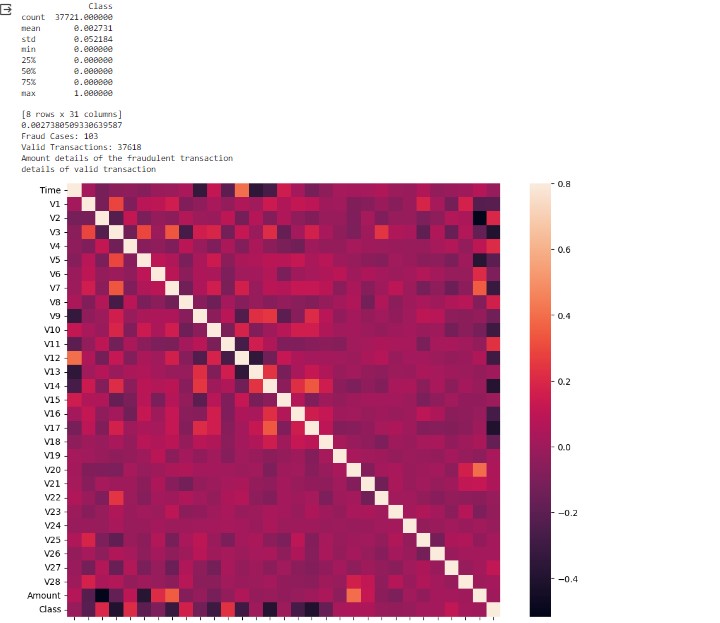
# (its a numpy array with no columns)

xData = X.values

yData = Y.values

* OUTPUTS





**USAGE**

**Fraud Detection:** Users can use this machine learning model to detect frauds

**Observation:** Users can know all sorts of types of transaction done from the credit card

**Easiness:** Easiness in sorting the data and the economy by knowing the time and date of transaction easy tracking of transactions.

**CONCLUSION**

In conclusion, the Credit card fraud Detection model that identify fraudulent credit card transactions. . Preprocess and normalize the transaction data, handle class imbalance issues, and split the dataset into train in and testing sets. We trained a classification algorithm, such as fraudulent or genuine. We evaluated the model’s performance using metrics like precision, Recall, and F1-score, and consider techniques like oversamling or under sampling for improving results.