**TITLE:** CodTech IT Solutions Internship - Task Documentation: “To-DO LIST” Using CSS, HTML, JAVASCRIPT.

**INTERN INFORMATION:**

**Name:** Sunkara Vamakeswari

**ID:** C0D6720

**INTRODUCTION**

**Task-1 CREDIT CARD FRAUD DETECTION**

**Introduction**

In today's digital age, with the widespread use of credit cards for financial transactions, the threat of fraudulent activities looms large. Credit card fraud can lead to significant financial losses for both individuals and businesses. Therefore, implementing robust fraud detection mechanisms is crucial to safeguarding financial assets and maintaining trust in the financial system.

This documentation outlines the implementation of a credit card fraud detection system using machine learning techniques. The goal of this project is to develop a model capable of accurately identifying fraudulent credit card transactions while minimizing false positives.

**Implementation**

The implementation of the credit card fraud detection system involves several key steps:

**1. Data Collection :** The first step is to gather a dataset containing historical credit card transactions. This dataset typically includes various features such as transaction amount, timestamp, merchant information, and whether the transaction was fraudulent or genuine.

**2. Data Preprocessing :** Before training the machine learning model, the raw transaction data needs to be preprocessed and normalized. This includes handling missing values, encoding categorical variables, and scaling numerical features to ensure uniformity and compatibility with the chosen machine learning algorithms.

**3. Handling Class Imbalance** : Credit card fraud datasets often suffer from class imbalance, where the number of genuine transactions far outweighs the number of fraudulent ones. To address this issue, techniques such as oversampling (e.g., SMOTE) or undersampling can be employed to balance the classes and prevent the model from being biased towards the majority class.

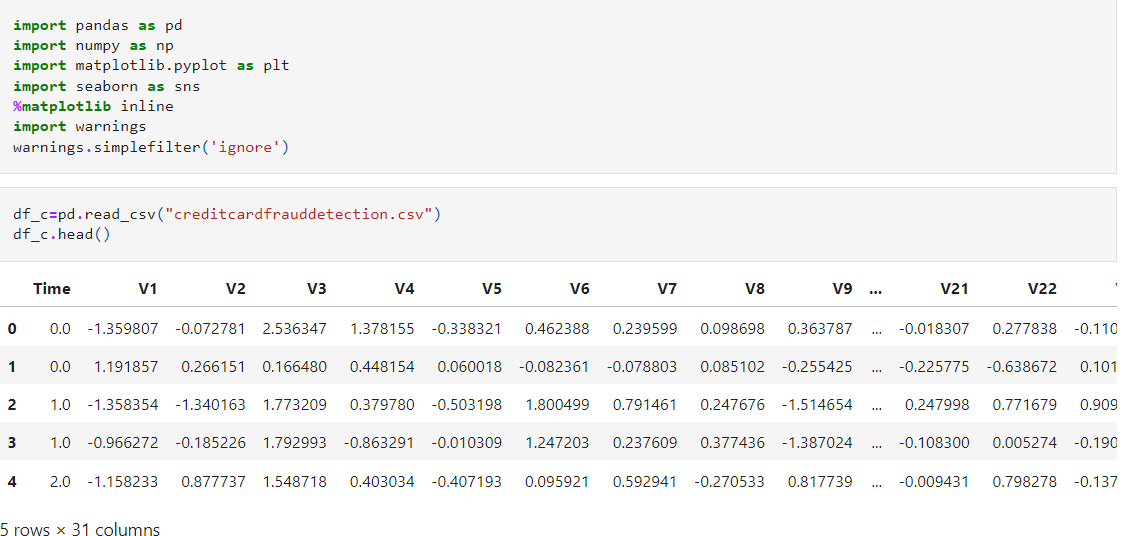
**4. Model Training :** Once the data is preprocessed, it is split into training and testing sets. A classification algorithm, such as logistic regression, random forests, or support vector machines (SVM), is then trained on the training data to learn patterns and relationships between features and the target variable (fraudulent or genuine transaction).

**5. Model Evaluation :** The trained model's performance is evaluated using various metrics such as precision, recall, F1-score, and accuracy. These metrics provide insights into the model's ability to correctly identify fraudulent transactions while minimizing false positives.

**6. Optimization and Fine-Tuning :** Depending on the model's performance, further optimization and fine-tuning may be necessary. This could involve adjusting hyperparameters, exploring different feature engineering techniques, or experimenting with different machine learning algorithms to improve the model's accuracy and robustness.

**7. Deployment and Monitoring :** Once the model is deemed satisfactory, it can be deployed into production environments where it can continuously monitor incoming transactions in real-time. Regular monitoring and updates are essential to ensure the model remains effective against evolving fraud patterns and techniques.

**CODE EXPLAINATION**

****

**Importing Libraries:**

**import pandas as pd:** Imports the pandas library, which is used for data manipulation and analysis.

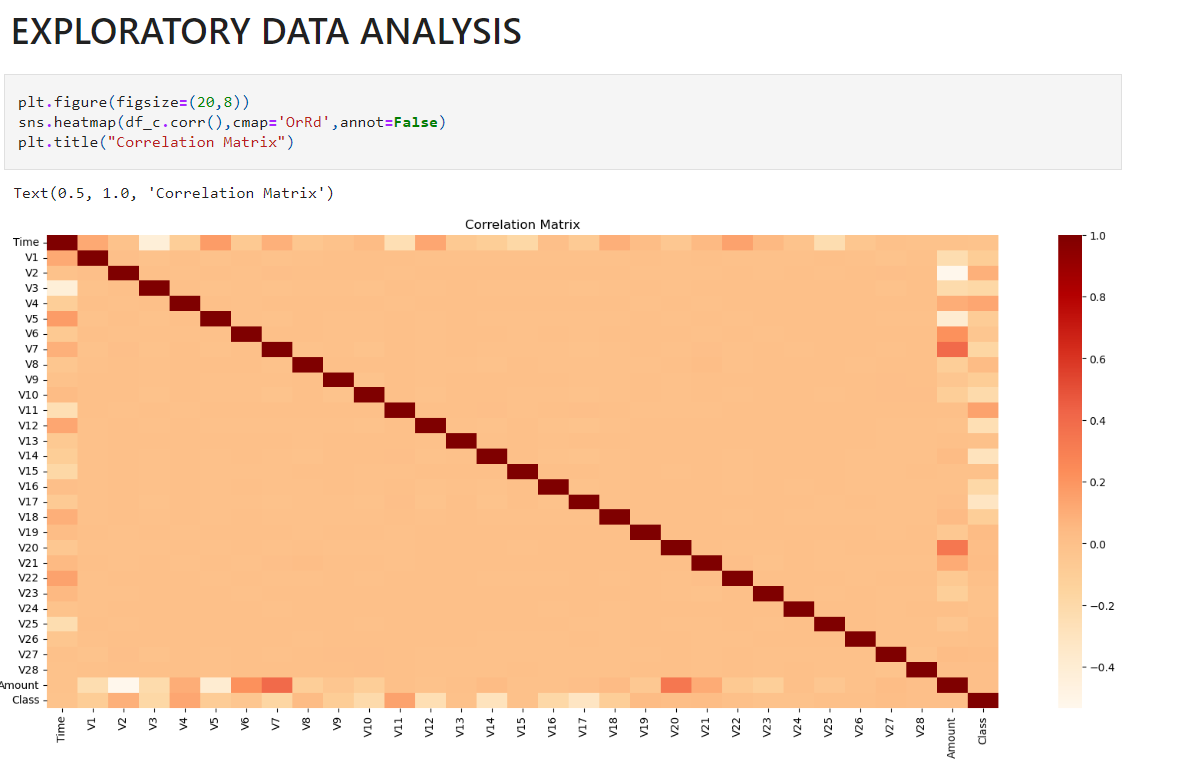
**import numpy as np:** Imports the numpy library, which provides support for arrays, matrices, and mathematical functions.

**import matplotlib.pyplot as plt:** Imports the matplotlib library, specifically the pyplot module, for data visualization.

**import seaborn as sns:** Imports the seaborn library, which is built on top of matplotlib and provides a high-level interface for drawing attractive and informative statistical graphics.

**%matplotlib inline:** A magic command used in Jupyter Notebook to display matplotlib plots inline within the notebook.

**import warnings:** Imports the warnings module, which is used to handle warnings in Python code.



fx,(ax1,ax2)**=**plt**.**subplots(2,1,sharex**=True**)

fx**.**suptitle('Amount per transaction by class')

ax1**.**hist(fraud**.**Amount,bins**=**50)

ax1**.**set\_title('Fraud')

ax2**.**hist(normal**.**Amount,bins**=**50)

ax2**.**set\_title('Normal')

plt**.**xlabel('Amount')

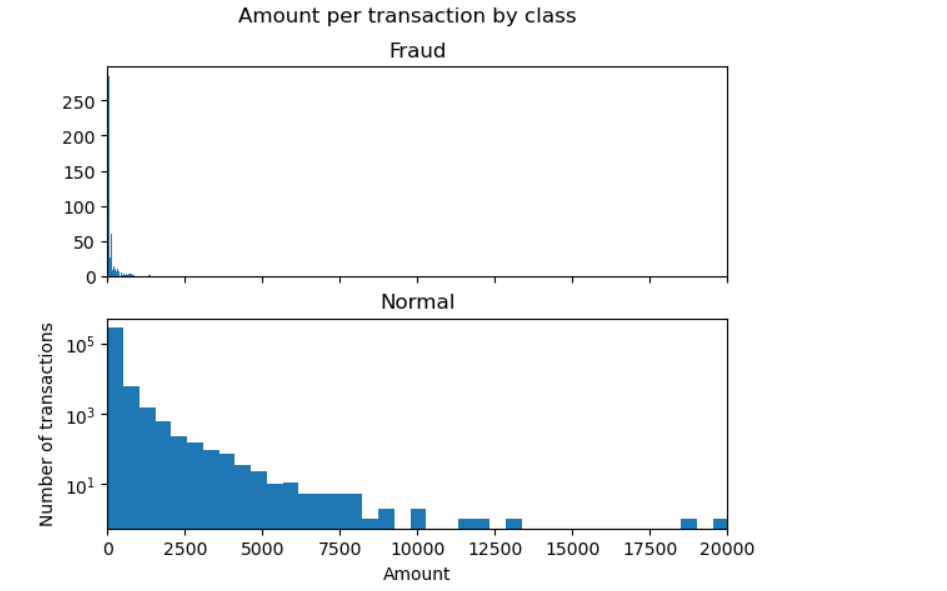
plt**.**ylabel('Number of transactions')

plt**.**xlim((0,20000))

plt**.**yscale('log')

plt**.**show()

#for fraud transaction transaction withrespect to amount is very small amount

****

**CONCLUSION**

While we couldn’t reach out goal of 100% accuracy in fraud detection, we did end up creating a system that can, with enough time and data, get very close to that goal. In this paper, Machine learning technique like Logistic regression, Decision Tree, SVM and Random forest were used to detect the fraud in credit card system. Sensitivity, Specificity, accuracy and error rate are used to evaluate the performance for the proposed system.By comparing all the methods, found that decision tree is better than the logistic regression and random forest classifier.

