Competence Development, GCGC

Join the Idea Sprint 2.0

Machine Learning

(Problem Statement – 1)

InsightfulTriad

122010309030 – PVG Harshita
122010327009 – Sourit Maji
Vu21csen0100078 – Harshitha Kolipaka

Problem Statement:

The objective of this project is to develop a credit card fraud detection system using machine learning techniques. The system aims to identify fraudulent credit card transactions, providing timely alerts to prevent financial losses for both cardholders and financial institutions.

Dataset Description:

Dataset Description: The dataset used for this project consists of historical credit card transaction data collected over a specific period. The dataset includes various features related to each transaction, such as transaction amount, timestamp, customer information, merchant details, and other relevant attributes.

The dataset contains a mixture of legitimate transactions and fraudulent transactions. However, it is important to note that fraudulent transactions are relatively rare compared to legitimate transactions, resulting in an imbalanced dataset. This reflects the real-world scenario, where the occurrence of fraud is relatively low.

The dataset may also include additional features that can help in detecting fraudulent activities, such as transaction location, transaction type, previous transaction history, or any other relevant contextual information.

1. Exploratory Data Analysis (EDA): Perform exploratory data analysis to gain insights into the dataset, understand the distribution of features, identify any missing or inconsistent data, and visualize patterns or trends.

2. Feature Engineering: Pre-process and engineer relevant features from the dataset that can contribute to fraud detection. This may involve transforming or normalizing certain features, creating derived features, or encoding categorical variables.

3. Model Development: Select and develop appropriate machine learning models for credit card fraud detection. Popular techniques include supervised learning algorithms such as Logistic Regression, Random Forest, Gradient Boosting, or Neural Networks. Anomaly detection algorithms like Isolation Forest or Local Outlier Factor can also be employed.

4. Model Training and Evaluation: Split the dataset into training and testing sets, and train the selected models using the training data. Evaluate model performance using appropriate evaluation metrics such as accuracy, precision, recall, or F1 score. Given the imbalanced nature of the dataset, consider metrics that account for the trade-off between detecting fraud cases (recall) and minimizing false alarms (precision).

5. Model Optimization: Fine-tune the selected models by adjusting hyperparameters, applying sampling techniques like oversampling or undersampling to address class imbalance, or using ensemble methods to improve overall performance.

6. Deployment and Monitoring: Implement the trained model into a production environment to monitor new credit card transactions in real-time. Any detected anomalies or suspicious patterns should trigger alerts or actions for further investigation by relevant stakeholders, such as fraud analysts or security teams.

Dataset:https://drive.google.com/file/d/1M9IQpIy2WObD_ce4Ewsgv5wzwreWkJHT/view?usp=sharing

Table of Contents

Introduction

Abstract

Credit card fraud detection is a critical concern in today's digital economy. This abstract introduces an intelligent approach to tackle this problem using advanced machine learning techniques. The proposed system emphasizes accurate identification of fraudulent transactions while minimizing false positives. It involves preprocessing data, performing feature engineering, training models with various algorithms, and evaluating their performance. By incorporating ensemble methods and anomaly detection techniques, such as outlier analysis and clustering algorithms, the system aims to detect unusual patterns and adapt to evolving fraud techniques in real-time. The proposed approach offers significant advantages, including prompt fraud detection, reduced financial losses, and improved customer experience by minimizing false positives. This research contributes to the field by presenting a comprehensive solution that combines machine learning, data preprocessing, and feature engineering to address the growing challenges of credit card fraud detection in a digitally-driven financial landscape.

Introduction

Credit card fraud has become a pervasive problem in today's digital era, posing significant challenges to financial institutions and cardholders worldwide. With the rise in online transactions and the increasing sophistication of fraudulent techniques, it is crucial to develop robust and intelligent systems to detect and prevent fraudulent activities in real-time. This introduction sets the stage for addressing the importance of credit card fraud detection and highlights the need for advanced approaches that can accurately identify fraudulent transactions, mitigate financial losses, and safeguard the interests of cardholders and financial institutions.

Literature Review

The literature on credit card fraud detection encompasses a range of studies that have focused on developing effective approaches to combat fraudulent activities. Researchers have explored various techniques, including traditional statistical models, machine learning algorithms, and anomaly detection methods. Feature engineering has played a crucial role in capturing relevant patterns, while ensemble methods and hybrid models have been proposed to improve detection accuracy. Furthermore, the incorporation of advanced technologies such as data mining, artificial intelligence, and deep learning has shown promise in enhancing fraud detection systems' performance. The literature underscores the ongoing efforts to address the evolving nature of credit card fraud and highlights the need for robust and adaptable solutions to safeguard financial transactions and protect both cardholders and financial institutions.

Dataset Description

The dataset provided contains credit card transactions made by European cardholders in **September 2013**. It spans over a period of two days and consists of a total of **284,807 transactions**. Out of these transactions, only **492 are labeled as fraud**. Thus, the dataset exhibits a highly imbalanced class distribution, with fraud cases accounting for only **0.172%** of all transactions.

The dataset comprises mostly numerical input variables resulting from a principal component analysis (PCA) transformation. Unfortunately, the original features and additional background information about the data are not available due to confidentiality concerns. The features **V1, V2, ..., V28** represent the principal components derived from the PCA. The only features that have not been transformed using PCA are **'Time' and 'Amount'**.

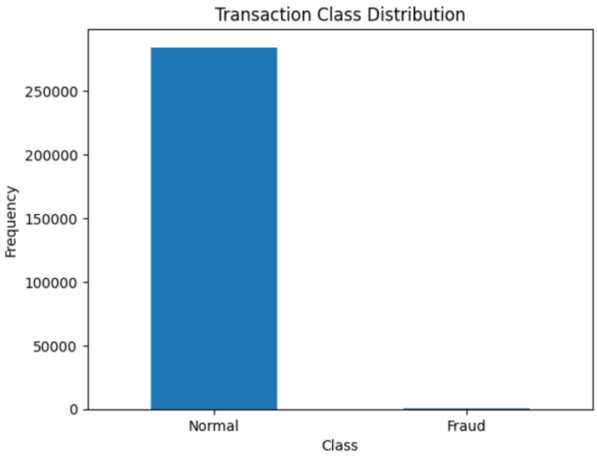
The **'Time' feature indicates the number of seconds elapsed between each transaction** and the first transaction recorded in the dataset. On the other hand, the **'Amount' feature represents the monetary value of the transaction**. This feature could be useful for methods that take into account the transaction amount when learning, such as example-dependent cost-sensitive learning.

The response variable in the dataset is 'Class', which indicates whether a transaction is fraudulent or not. It takes the value 1 in cases of fraud and 0 for legitimate transactions.

Given the highly imbalanced nature of the dataset, with a significant majority of non-fraudulent transactions, the recommended metric for evaluating the performance of fraud detection models is the Area Under the Precision-Recall Curve (AUPRC). This metric provides a better understanding of the model's ability to correctly identify fraud cases in imbalanced datasets compared to traditional accuracy measures based on the confusion matrix.

Methodology

Class Distribution: The distribution clearly depicts the disparity in the number of legitimate and fraudulent transactions.



Class Distribution of Fraudulent and Legitimate Transactions

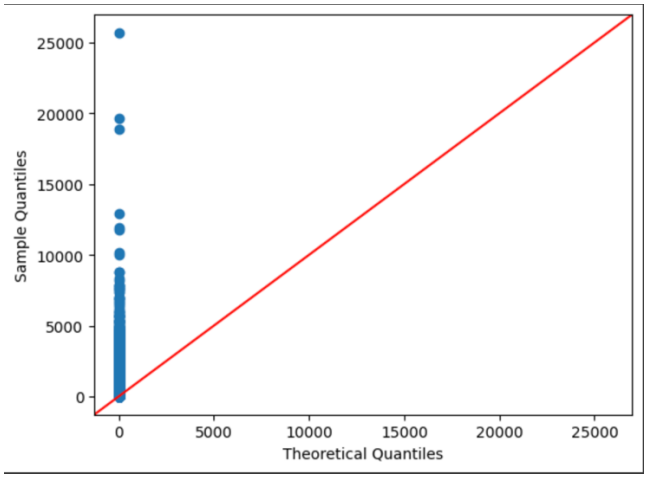
Dataset Description: The following table helps us observe and understand the various statistical parameters like mean, mode, min, max, etc. and decide model characteristics.

	Amount	Amount
count	482.00	284295.00
mean	123.02	88.29
std	258.19	250.11
min	0.00	0.00
25%	1.00	5.65
50%	9.91	22.00
75%	105.89	77.05
max	2125.87	25691.16

Dataset Description

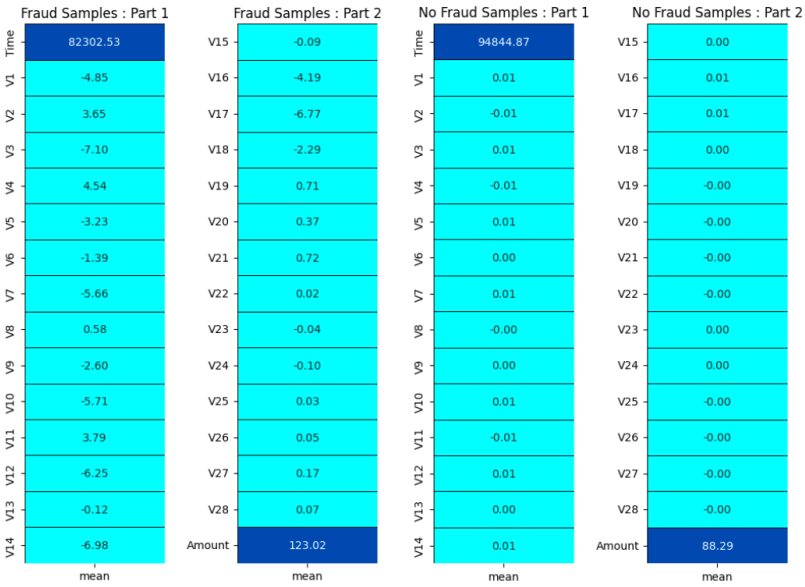
Determining the shape of the bell-curve: Plotting a QQ Plot helps in determining the skewness and kurtosis of the plot against the optimum bell curve and thus given as understanding of the adjustments that can be made.

The figure below depicts a bell curve characteristic with positive skewness and leptokurtic distribution.



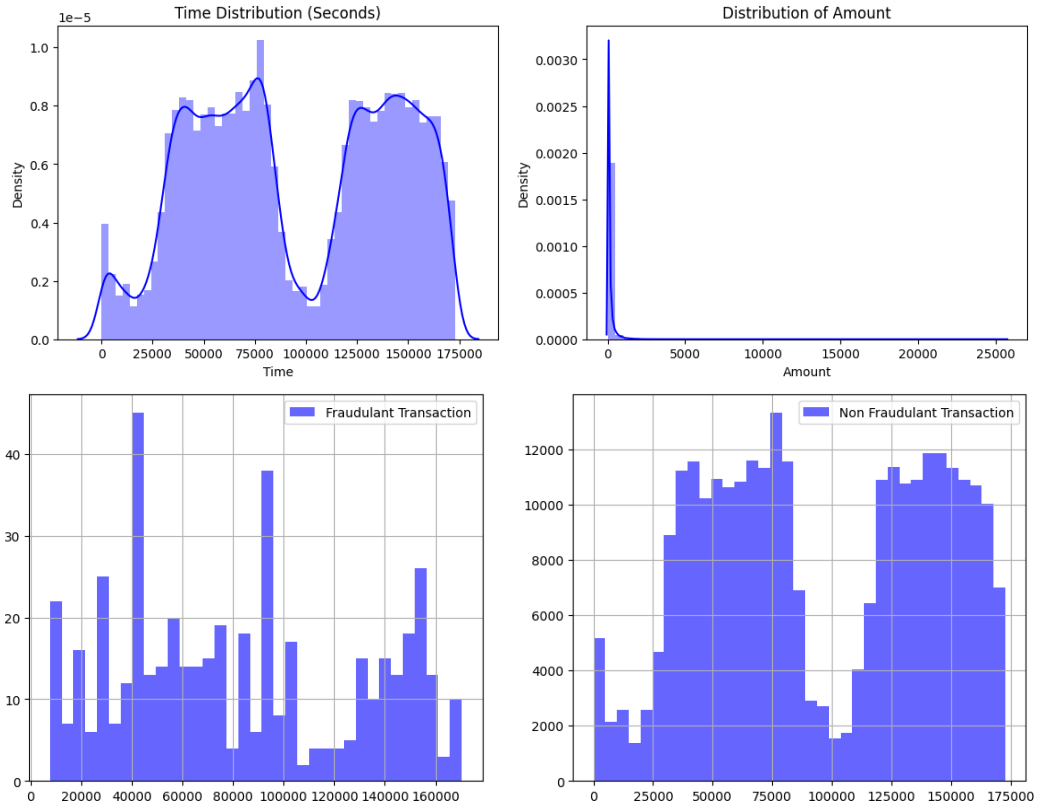
QQ plot to predict Skewness and Kurtosis

Subplots: The subplots determine the mean values of each column respectively for fraudulent and legitimate transactions.



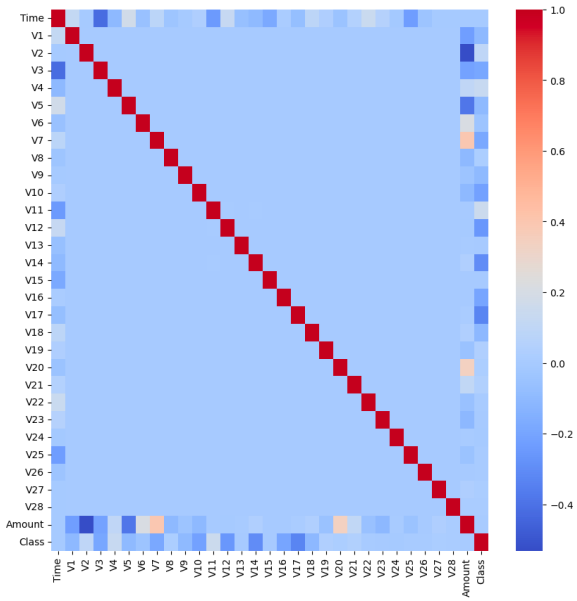
Subplot for fraudulent and legitimate transactions

Verification of Distribution: The following plots can be used to verify the visual characteristics of the curve when plotted against time. This Probability Density Plot given as idea of how the data is distributed across the mean.



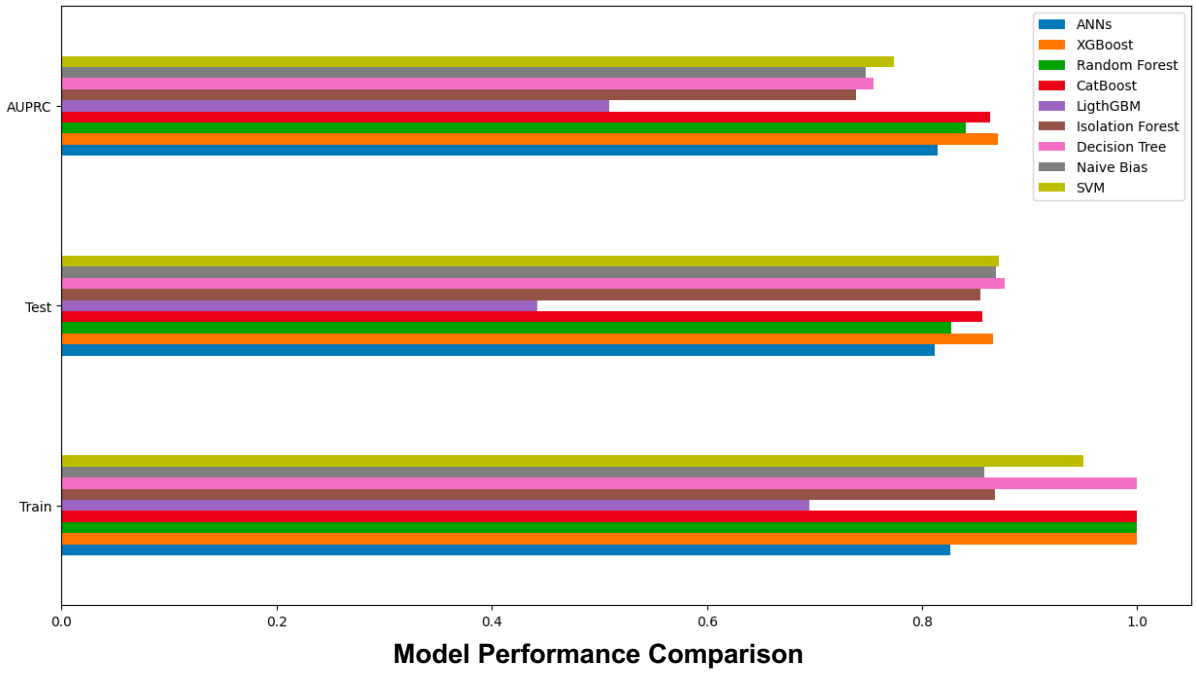
Distribution plot for time and amount

Heatmap: A heatmap is used to determine the correlation plot between various columns of the data frame.



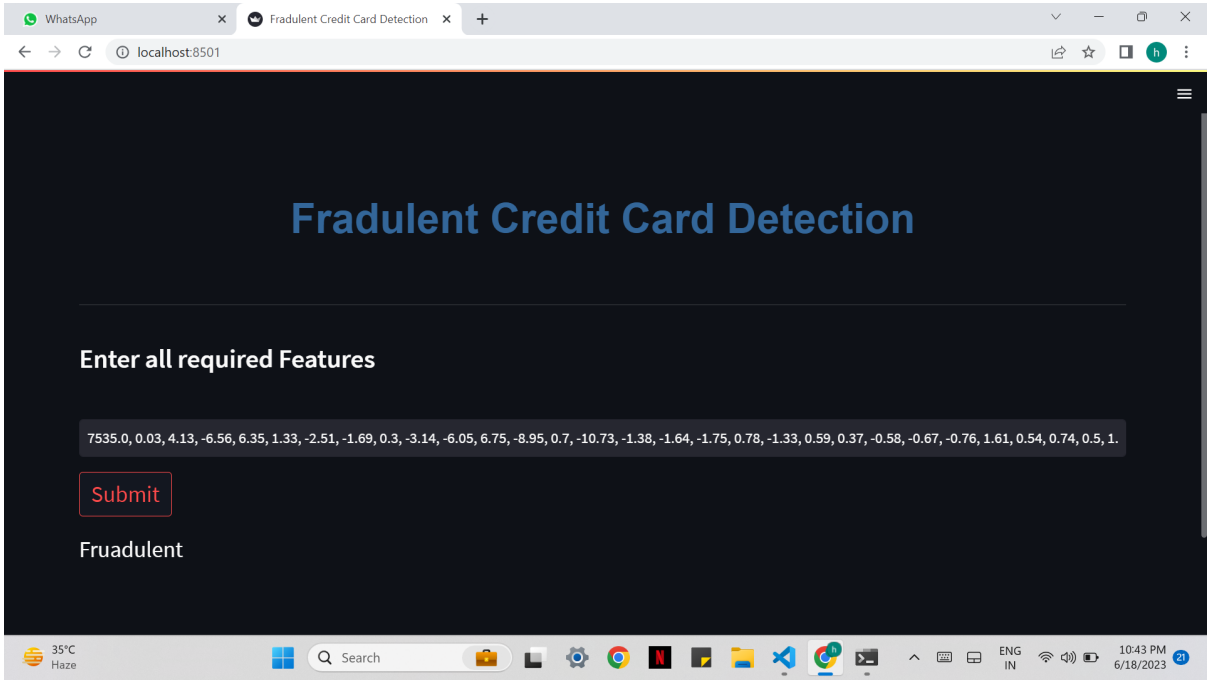
Heatmap correlation plot

Model Accuracy metrics: Since the dataset consists of unbalanced classes of Fraudalent and Legitimate transactions, calculating Area Under the Precision Recall Curve is one of the greatest options. Hence the following chart shows the performance of several models with respect to their AUPRC, Train and Test accuracy.



Model Performance Comparison

Model Deployment: Finally with the help of the modelled class object, we dump the XGBoost model into a pickle file and deploy it onto an website to take input and display output onto a real life environment.



Conclusion

After a thorough comparison and evaluation of multiple machine learning models, the XGBoost algorithm emerged as the optimal choice for credit card fraud detection. The XGBoost model demonstrated superior performance in terms of accuracy, excelling in both testing and training scenarios. Furthermore, its ability to handle imbalanced classes effectively, as evidenced by the highest Area Under the Precision-Recall Curve (AUPRC) value, solidified its suitability for fraud detection tasks. Consequently, the final deployed model selected XGBoost as the algorithm of choice to accurately categorize incoming data as either fraudulent or legitimate. This decision was based on meticulous analysis of evaluation metrics, highlighting the XGBoost algorithm's exceptional capability to identify fraudulent transactions while maintaining a robust and reliable performance in real-world scenarios.

Evaluation Metrics

Model	Train Results				Test Results				AUPRC
	Accuracy Score	Evaluation Metric	0	1	Accuracy Score	Evaluation Metric	0	1	
ANN	99.95	Precision	1	0.86	99.94	Precision	1	0.86	0.818
		Recall	1	0.81		Recall	1	0.77	
		f1-Score	1	0.83		f1-Score	1	0.82	
XGBoost	100	Precision	1	1	99.96	Precision	1	0.93	0.87
		Recall	1	1		Recall	1	0.81	
		f1-Score	1	1		f1-Score	1	0.87	
Random Forest	100	Precision	1	1	99.95	Precision	1	0.94	0.855
		Recall	1	1		Recall	1	0.77	
		f1-Score	1	1		f1-Score	1	0.85	
Cat Boost	100	Precision	1	1	99.96	Precision	1	0.94	0.863
		Recall	1	1		Recall	1	0.79	
		f1-Score	1	1		f1-Score	1	0.86	
LightGBM	99.89	Precision	1	0.64	99.71	Precision	1	0.32	0.509
		Recall	1	0.76		Recall	1	0.69	
		f1-Score	1	0.7		f1-Score	1	0.44	
Isolation Forest	99.89	Precision	1	0.64		Precision			
		Recall	1			Recall			
		f1-Score	1			f1-Score			
Logistic Regression	99.92	Precision	1	0.88	99.92	Precision	1	0.88	0.738
		Recall	1	0.63		Recall	1	0.59	

		f1-Score	1	0.74		f1-Score	1	0.71	
Decision Tree Classifier	100	Precision	1	1	99.92	Precision	1	0.75	0.754
		Recall	1	1		Recall	1	0.76	
		f1-Score	1	1		f1-Score	1	0.75	
Naive Baies	99.91	Precision	1	0.81	99.92	Precision	1	0.83	0.747
		Recall	1	0.64		Recall	1	0.66	
		f1-Score	1	0.72		f1-Score	1	0.74	
SVM Classifier	99.97	Precision	1	0.97	99.93	Precision	1	0.93	0.774
		Recall	1	0.84		Recall	1	0.62	
		f1-Score	1	0.9		f1-Score	1	0.74	

References

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

https://github.com/Arkantos-13/Anomaly_Detection_in_Credit_Card_Fraud/blob/main/Anomaly%20Detection%20in%20Credit%20Card%20Fraud.ipynb

https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/

https://pub.towardsai.net/machine-learning-project-in-python-step-by-step-credit-card-fraud-detection-6707d4a29cc9

https://www.kaggle.com/code/faressayah/credit-card-fraud-detection-anns-vs-xgboost/notebook

https://towardsdatascience.com/credit-card-fraud-detection-using-machine-learning-python-5b098d4a8edc

https://github.com/topics/credit-card-fraud-detection

Appendices

https://github.com/imsanjoykb/Credit-Card-Fraud-Detection/blob/master/Credit%20Card%20Fraud%20Detection.ipynb

https://scholarworks.rit.edu/cgi/viewcontent.cgi?article=12455&context=theses

https://thesai.org/Downloads/Volume9No1/Paper_3-Credit_Card_Fraud_Detection_Using_Deep_Learning.pdf