**Bhavana Tallapaka**

Date: - 14-08-2024

# Credit Card Fraud Detection

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, … V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependent cost-sensitive learning. Feature 'Class' is the response variable, and it takes value 1 in case of fraud and 0 otherwise.

Given the class imbalance ratio, we recommend measuring the accuracy using the Area Under the Precision-Recall Curve (AUPRC). Confusion matrix accuracy is not meaningful for unbalanced classification. (Reference By Learning Group Year: 2021, Container: Kaggle.com, URL: Credit Card Fraud Detection (kaggle.com) )

# Software Requirements: -

1. Computer system with high internet facility.
2. Google Colab.
3. Dataset.

# Scope: -

The scope for Credit Card Fraud Detection encompasses a broad range of activities and objectives aimed at minimizing financial losses, enhancing security, and improving customer trust. Here are some key aspects: detection of fraudulent transactions, improvement of detection algorithms, handling imbalanced datasets, real-time fraud detection, customer education and awareness, continuous monitoring and improvement, etc.

# Objective: -

1. Implement effective fraud detection mechanisms to minimize the financial impact on both the credit card issuer and the cardholder.
2. To deal with the past and ongoing issues with credit card fraud detection techniques (I am using September 2013 by European cardholders' data). The future issues with credit card frauds will be solved. And to diminish the economic and financial losses for both cardholders and banking authorities.
3. Improve the accuracy of fraud detection models to reduce the number of legitimate transactions flagged as fraudulent, minimizing inconvenience to customers.

It is crucial for credit card companies to identify fraudulent transactions to ensure that customers are not billed for purchases they did not make. This project will address the historical and current challenges associated with credit card fraud detection methods (referencing European cardholder data from September 2013). Future credit card fraud issues will be resolved, aiming to reduce economic and financial losses for both cardholders and banking institutions.

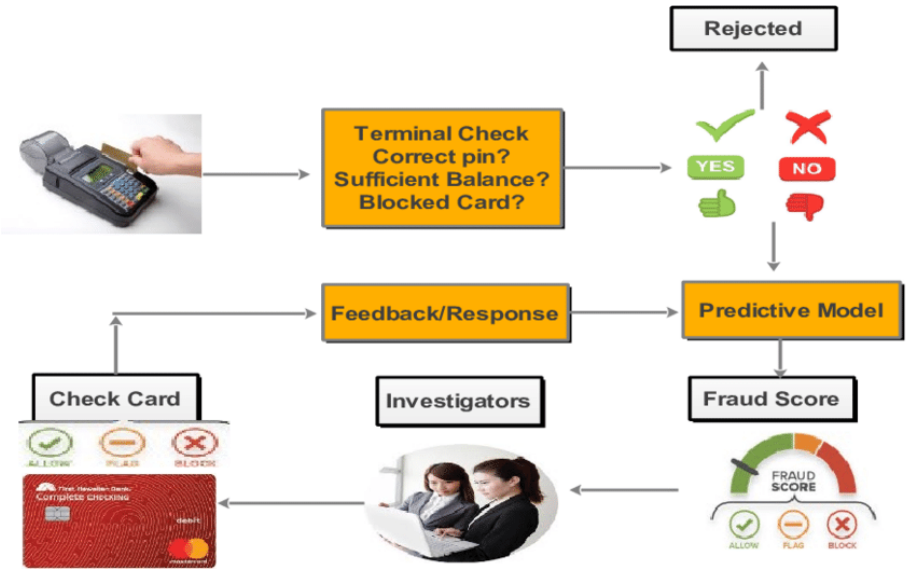
Many existing models struggle to adapt quickly to new and evolving fraud patterns, leading to delayed detection. Fraud detection datasets are often highly imbalanced, with fraudulent transactions representing a small fraction of the data. This can lead to biased models that underperform in detecting fraud. Many current systems generate a high number of false positives, flagging legitimate transactions as fraudulent, which frustrates customers and incurs additional costs.

Research could focus on developing adaptive models that learn from new fraud techniques in real-time, improving detection accuracy. Exploring advanced techniques to address data imbalance, such as synthetic data generation or cost-sensitive learning, could improve model performance. Aiming to develop more precise algorithms that reduce false positives while maintaining high fraud detection rates.

# Key features: -

Key features for credit card fraud detection typically include a combination of transactional, behavioral, and derived data points. Here are some of the essential features: transaction amount, transaction time, previous transaction history, transaction frequency, transaction sequence, etc. These features, when combined and analyzed, can significantly improve the detection of fraudulent transactions in real-time.

# Designing: -




(Reference By Learning Group Year: 2019, Container: researchgate.net, URL: [(PDF) Credit Card Fraud Detection using k-star Machine Learning Algorithm (researchgate.net)] )

# Implementation: -

The implementation of Credit Card Fraud Detection involves translating the design specifications into functional code. For implementing Credit Card Fraud Detection, I used Google Colab([Welcome To Colab - Colab (google.com)] ). I have implemented Credit Card Fraud Detection project by using following concepts.

## Data Analysis

PANDAS:
Pandas provide high performance, fast, easy to use data structures and data analysis tools for manipulating numeric data and time series. Pandas is built on the NumPy library and written in languages like Python, and C. In pandas, we can import data from various file formats like JSON, SQL, Microsoft Excel, etc. (Reference By Anon, Container: pandas.pydata.org. URL: [API reference — pandas 2.2.2 documentation (pydata.org)] )

NUMPY:
It is the fundamental library of python, used to perform scientific computing. It provides high-performance multidimensional arrays and tools to deal with them. A NumPy array is a grid of values (of the same type) that are indexed by a tuple of positive integers, NumPy arrays are fast, easy to understand, and give users the right to perform calculations across arrays. (Reference By Anon, Container: numpy.org URL: [NumPy reference — NumPy v2.0 Manual])

## Data Visualizations

Data Visualization is the graphic representation of data. It converts a huge dataset into small graphs, thus aids in data analysis and predictions.

MATPLOTLIB:
It is a Python library used for plotting graphs with the help of other libraries like Numpy and Pandas. It is a powerful tool for visualizing data in Python. It is used for creating statical interferences and plotting 2D graphs of arrays. (Reference By Anon, Container: Scipy-lectures.org, URL: [1.5. Matplotlib: plotting — Scipy lecture notes (scipy-lectures.org)] )

SEABORN:

It is also a Python library used for plotting graphs with the help of Matplotlib, Pandas, and Numpy. It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data. (Reference By Anon, Container: seaborn.pydata.org, URL: API reference — seaborn 0.13.2 documentation (pydata.org) )

Train_Test_Split:
The train-test split is a technique for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. (Reference By scikit-learn, Year: 2018, Container: Scikit-learn.org, URL: train_test_split — scikit-learn 1.5.1 documentation)

KNeighborsClassifier:
The K-Nearest Neighbors classifier (KNN) is one of the simplest yet most commonly used classifiers in supervised machine learning. KNN is often considered a lazy learner. It doesn't technically train a model to make predictions. (Reference By scikit-learn, Year: 2019, Container: Scikit-learn.org, URL: KNeighborsClassifier — scikit-learn 1.5.1 documentation)

## Data Modelling

K-nearest Neighbors:
It is a pattern recognition algorithm that uses training datasets to find the k closest relatives in future examples. (Reference By Onel Harrison, Year: 2018, Container: Medium Publisher: Towards Data Science, URL: Machine Learning Basics with the K-Nearest Neighbors Algorithm | by Onel Harrison | Towards Data Science )

Decision Tree:
A decision tree is a supervised learning algorithm that is perfect for classification problems, as it's able to order classes on a precise level. (Reference By scikit-learn, Container: Scikit-learn.org, URL: 1.10. Decision Trees — scikit-learn 1.5.1 documentation )

Random Forest:
The random forest algorithm is an expansion of decision tree, in that you first construct a multitude of decision trees with training data, then fit your new data within one of the trees as a "random forest". (Reference By Scikit-learn, Year: 2018, Container: Scikit-learn.org URL: RandomForestClassifier — scikit-learn 1.5.1 documentation )

# Testing: -

Testing is a crucial phase in every model to ensure the quality, reliability, and correctness of Credit Card Fraud Detection model. Various types of testing can be applied at different stages of development. Here I used Google Colab for compilation and execution. To execute the model, we need to insert that dataset correctly, such that no errors will be found.

# References: -


- → *"Python programming for the absolute beginner" by Micheal Dawson in the year 2023, Boston: Cengage.*
- → *scikit-learn: machine learning in Python, By Scikit-learn, Year: 2019, Container: Scikit-learn.org, URL: https://scikit-learn.org/stable/*
- → *Future Issues with Credit Card Fraud Detection Techniques By Marvin Namanda Year: 2016 Publisher: GRIN Verlag ISBN: 9783668222588*
- → *Amin, A., Akhtar, S., & Mian, M. A. (2019). "A Review of Machine Learning Techniques in Credit Card Fraud Detection." International Journal of Computer Science and Information Security, 17(2), 1-9. DOI: 10.5281/zenodo.3406360*
- → *Credit Card Fraud Detection using Machine Learning from Kaggle (2024). - YouTube. Available at: https://youtu.be/frM_7UMD_-A?si=8cGsCCt4xocnyjXg [Accessed 14 Aug. 2024].*

# Conclusion: -

In the culmination of the Credit Card Fraud Detection model, a transformative solution emerges, reshaping the landscape of fraud detection and losses causing due to frauds. From its inception, the project aimed to address the limitations of detecting frauds during credit card transactions, introducing innovative features, and leveraging Python technology for enhanced efficiency, security, and adaptability.