FRAUD DETECTION IN FINANCIAL TRANSACTIONS

Safeguarding Your Finances with Intelligent Insights

PROBLEM STATEMENT

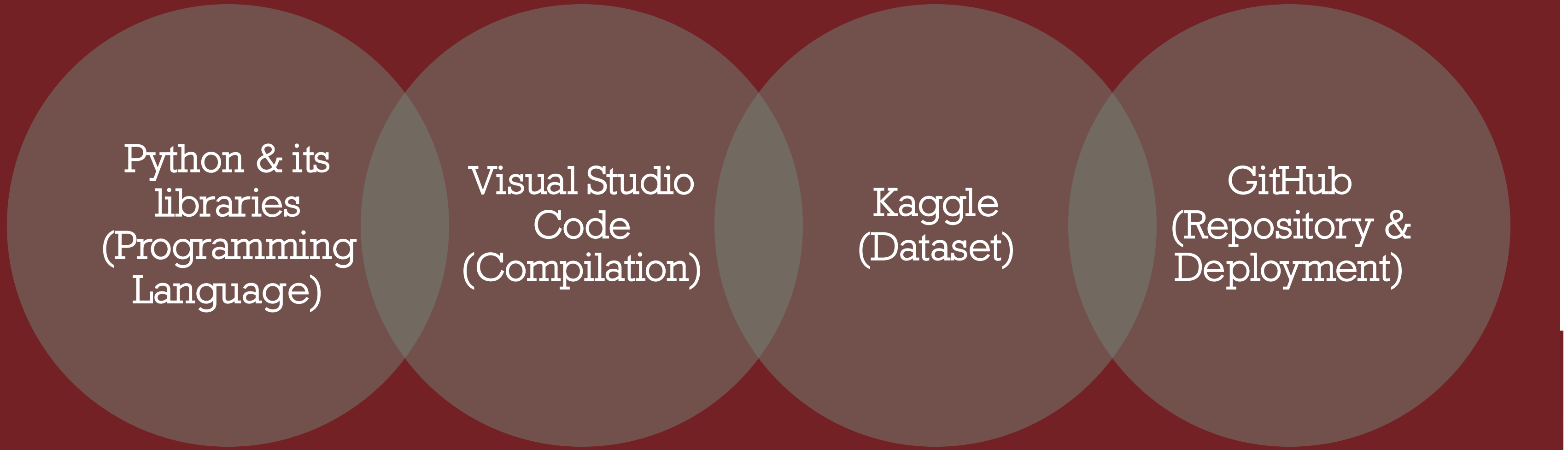
Financial institutions face challenges in detecting and preventing fraudulent transactions, which can result in significant financial losses. Develop a machine learning model that analyzes transaction patterns and user behaviors to detect and flag potentially fraudulent activities in real-time. Through this machine learning model, we aim to enhance the security of financial transactions through credit cards by providing an advanced fraud detection system that identifies suspicious activities, quickly and accurately.

ABOUT THE PROJECT

This project focuses on creating a model to detect fraudulent credit card transactions using machine learning. By analyzing patterns in historical transaction data, the model learns to identify suspicious activities. The dataset used includes both legitimate and fraudulent transactions, balanced to ensure the model is trained effectively. This balanced approach helps the model accurately differentiate between normal and fraudulent transactions.

The model employs logistic regression, a common technique for binary classification, to make predictions. It has been evaluated rigorously and has demonstrated high accuracy on both training and test datasets, meaning it can reliably flag potential fraud. This tool is essential for financial institutions, helping them protect customers by detecting and preventing fraudulent transactions in real-time. By catching suspicious activities early, the model contributes to reducing financial losses and increasing security in the digital banking environment.

PLATFORMS AND TECHNOLOGIES USED

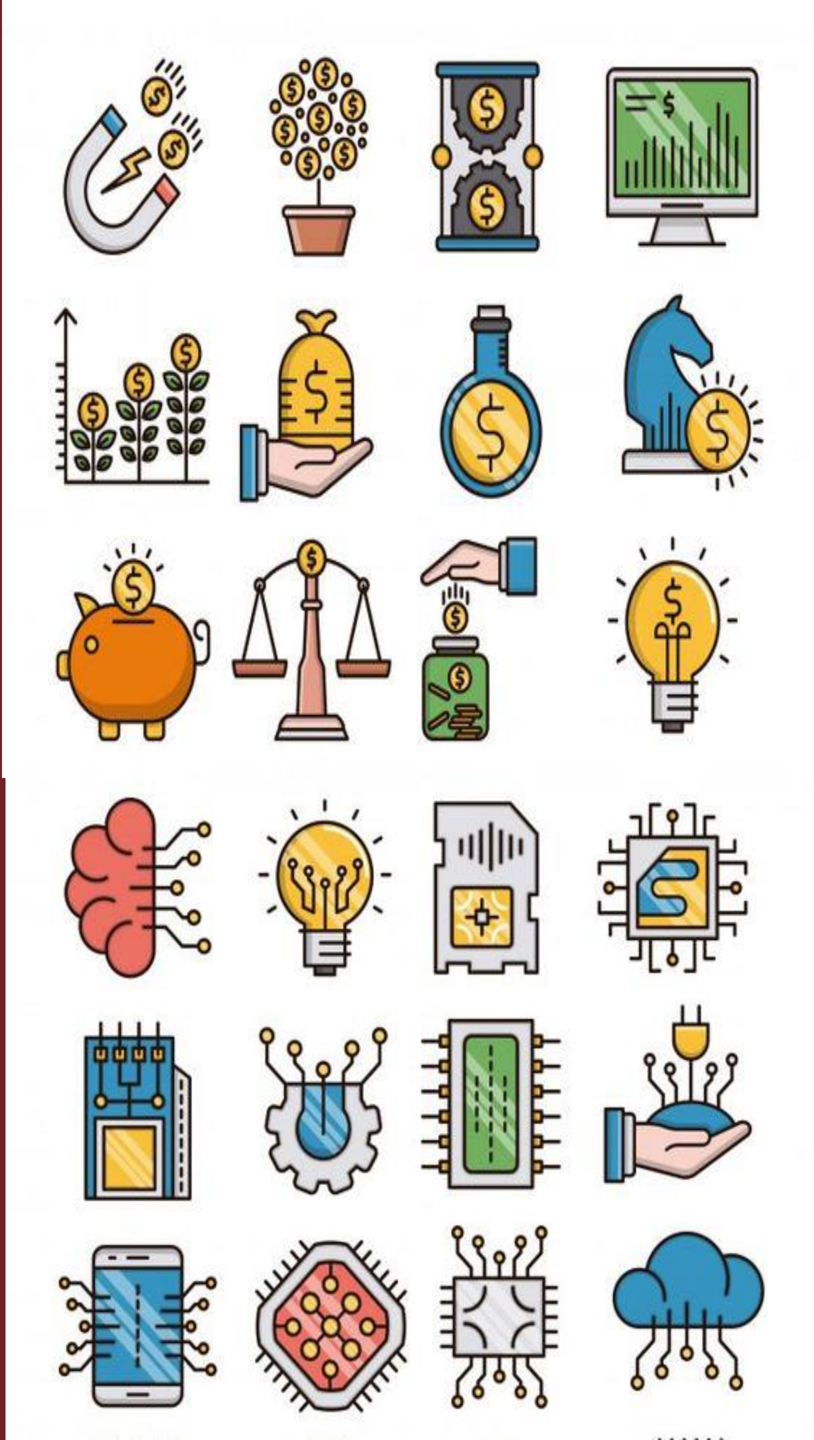


Python & its libraries (Programming Language)

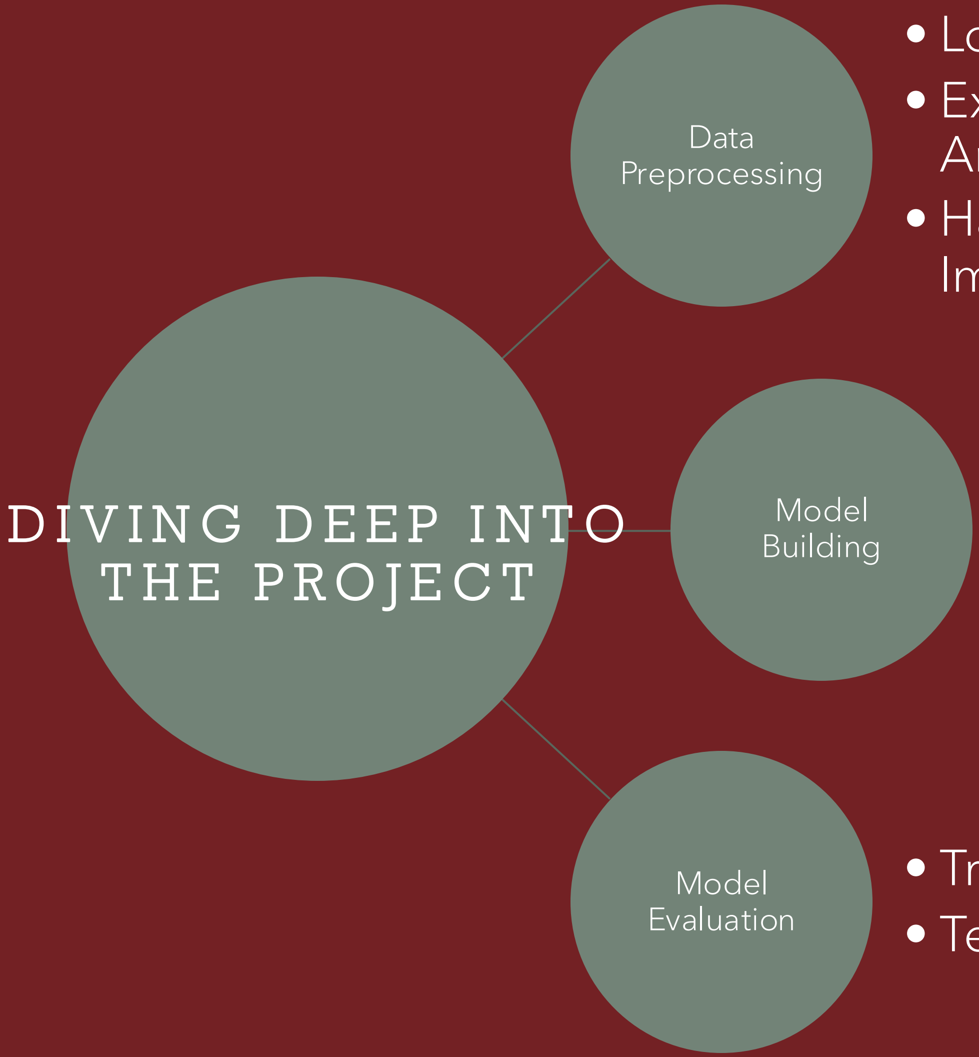
Visual Studio Code (Compilation)

Kaggle (Dataset)

GitHub (Repository & Deployment)



DIVING DEEP INTO THE PROJECT



Data Preprocessing

- Loading the Data
- Exploratory Data Analysis
- Handling Imbalance

Model Building

- Splitting Data
- Logistic Regression Model

Model Evaluation

- Training Accuracy
- Testing Accuracy

DIVING DEEP INTO THE PROJECT

DATA PREPROCESSING

Loading the Data:

- The dataset is loaded into a Pandas DataFrame for analysis.

Exploratory Data Analysis:

- The dataset contains 31 columns, including features named V1 to V28, the time of the transaction, the amount, and a class label indicating whether the transaction is legitimate (0) or fraudulent (1).
- Statistical summaries and the distribution of the 'Amount' feature are reviewed for both legitimate and fraudulent transactions.

Handling Imbalance:

- Given the imbalance, undersampling is used to create a balanced dataset. A random sample of legitimate transactions is selected to match the number of fraudulent transactions (492).
- The new dataset is created by concatenating the undersampled legitimate transactions with the fraudulent transactions.

MODEL BUILDING

Splitting Data:

- The dataset is split into features (X) and target (Y). The features include all columns except the 'Class' column, which is the target.
- The data is further split into training and test sets using an 80-20 split, maintaining the class distribution using stratification.

Logistic Regression Model:

- A logistic regression model is chosen for its simplicity and effectiveness in binary classification problems.
- The model is trained on the training data.

MODEL EVALUATION

Training Accuracy:

- The accuracy of the model on the training data is calculated by comparing the predicted values to the actual values. The training accuracy achieved is approximately 94.40%.

Testing Accuracy:

- The model's performance on the test data is similarly evaluated. The test accuracy is found to be approximately 91.40%, indicating that the model generalizes well to unseen data.

FUTURE ADVANCEMENT PROSPECTIVES

Enhancing Fraud Prevention with Cutting-Edge Technologies

GENERIC IMPROVEMENTS

Feature Engineering:

- Time-based Features: Create additional features based on the timestamp, such as day of the week, hour of the day, or transaction intervals, which might provide more context for detecting fraud.
- Aggregated Features: Include aggregated statistics like average transaction amount per day/week or the number of transactions within a specific period.

Data Augmentation:

- Synthetic Data Generation: Use techniques like SMOTE (Synthetic Minority Over-sampling Technique) to generate more synthetic fraudulent transaction data to address the class imbalance issue more effectively.
- Anomaly Detection: Implement unsupervised learning techniques to detect anomalies in transactions that could indicate fraud, providing an additional layer of security.

Model Ensemble:

- Combine multiple machine learning models (e.g., Random Forest, Gradient Boosting, SVM) to create an ensemble model that may improve prediction accuracy and robustness compared to a single model.

Real-time Processing:

- Adapt the model to process and evaluate transactions in real-time, providing immediate fraud detection and response capabilities.

User Behavior Analysis:

- Incorporate user behavior analytics to track spending patterns over time, which can help in identifying deviations that may indicate fraud.

TECHINICAL IMPROVEMENTS

Hyperparameter Tuning:

- Use techniques like Grid Search or Random Search to find the optimal hyperparameters for the logistic regression model or other machine learning models, improving their performance.

Cross-Validation:

- Implement cross-validation techniques to ensure the model's robustness and generalizability across different subsets of the data, reducing the risk of overfitting.

Advanced Models:

- Explore and implement more advanced models like XGBoost, LightGBM, or deep learning models (e.g., neural networks) that may capture complex patterns in the data more effectively.

Feature Selection:

- Apply feature selection methods to identify the most significant features contributing to the prediction, which can simplify the model and reduce the risk of overfitting.

Pipeline Automation:

- Create an automated machine learning pipeline using tools like Scikit-learn Pipelines or TensorFlow Extended (TFX) to streamline data preprocessing, model training, evaluation, and deployment.

Model Interpretability:

- Implement interpretability techniques (e.g., SHAP values, LIME) to understand how the model makes decisions, which can help in explaining the model's predictions to stakeholders and identifying any biases.

Performance Monitoring:

- Set up continuous monitoring of the model's performance in a production environment to detect and address any degradation in accuracy over time, ensuring the model remains effective.

Scalability:

- Optimize the model and the data processing pipeline for scalability, ensuring that the system can handle large volumes of transaction data efficiently.

REFERENCE LINKS

Below mentioned links aim to fulfil the function of a bibliography and information associated with the participants for maximum transparency.

Reference / Documents	Links
PPT Link	https://1drv.ms/p/c/e13a62664012fae4/EdmcuaRd13FInHd6Zxz2uDwBpg2rE4dqa1Eb5QAo2egK0w?e=M0Kgoe
Kaggle Dataset	https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download
Demo Video Link	https://drive.google.com/file/d/1d0y8CN7e-_hfYgmi6sfJTJEg_afUE27h/view?usp=sharing
GitHub Repository	https://github.com/Medhamishraa/ML_Error404

THANK YOU

Medha Mishra

medha.mishra.ug22@nsut.ac.in

+91 98118 56856

Anav Sobti

anav.sobti.ug22@nsut.ac.in

+91 98112 31517