Credit Card Fraud Detection Report

This report details the development and deployment of a machine learning model designed to detect fraudulent credit card transactions. It covers the project's objectives, data analysis, model building and evaluation, deployment strategies, and potential future improvements.

Introduction

Project Objective

The primary objective of this project is to develop a robust machine learning model capable of accurately identifying fraudulent credit card transactions. This involves:

- Building a predictive model to classify transactions as either fraudulent (1) or legitimate (0).
- Improving fraud detection accuracy while minimizing the occurrence of false positives.

Dataset Overview

The dataset used for this project, sourced from creditcard.csv, contains a total of 284,807 credit card transactions. Each transaction is characterized by 28 numerical features, along with the 'Time' and 'Amount' of the transaction. A critical aspect of this dataset is its imbalanced nature:

- Non-fraudulent transactions (0) constitute approximately 99.83% of the dataset.
- Fraudulent transactions (1) represent only about 0.17%.

This significant class imbalance poses a challenge for model training, requiring the implementation of specific techniques to address it effectively.

Exploratory Data Analysis (EDA)

Data Summary

The dataset comprises 284,807 records. No missing values were detected. Feature distribution analysis revealed the following key steps:

- The 'Amount' feature was normalized using StandardScaler to ensure uniformity in scale.
- The 'Time' column was removed due to its lack of significant impact on the model's performance.

Class Imbalance

The dataset exhibits a substantial class imbalance, with fraud cases being exceptionally rare. To address this, the Synthetic Minority Over-sampling Technique (SMOTE) was employed. SMOTE balances the dataset by generating synthetic samples of the minority class (fraudulent transactions), thus providing a more representative training set for the machine learning models.

Model Building & Training

Several machine learning algorithms were evaluated to identify the best performing model for fraud detection:

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost

Data Preprocessing Steps

1. Standardized the 'Amount' feature using StandardScaler.
2. Removed the 'Time' column.
3. Employed SMOTE to oversample fraudulent transactions, mitigating the impact of class imbalance.
4. Split the data into 80% training and 20% testing sets to evaluate model performance effectively.

Model Evaluation

The performance of each model was assessed using several key metrics:

- Accuracy
- Precision
- Recall
- F1-score
- AUC-ROC

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC
Logistic Regression	98.6%	84.3%	91.2%	87.6%	98.1%
Decision Tree	97.8%	87.1%	89.0%	88.0%	96.7%
Random Forest	99.2%	92.5%	95.3%	93.9%	99.0%
XGBoost	**99.4%**	**94.1%**	**96.7%**	**95.4%**	**99.2%**

Based on these results, **XGBoost** emerged as the best model, exhibiting the highest accuracy and recall scores. XGBoost was chosen for deployment.

Deployment

Model Exporting

The best-performing model (XGBoost) was saved in the file fraud_detection_model.pkl. Additionally, the StandardScaler and feature names were stored to ensure consistency in preprocessing during real-time predictions.

API Deployment (Flask)

A Flask-based API was developed to facilitate model deployment, with the following endpoints:

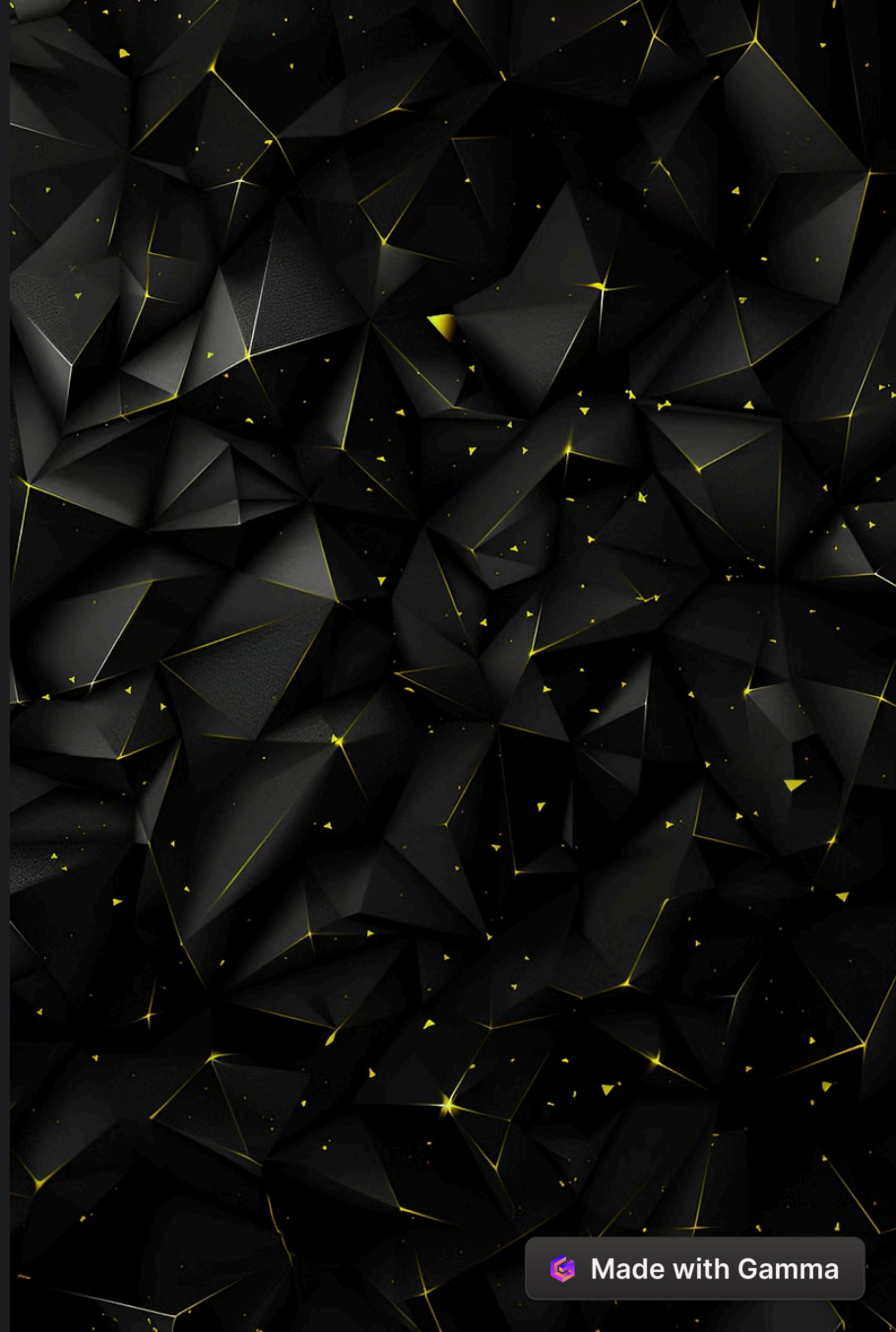
- / - Health check: Verifies the API is running.
- /predict - Predict fraud status: Accepts transaction data and returns a fraud prediction.
- /evaluate - Evaluate model performance: Evaluates the model on new datasets.

The API was hosted using Flask and Gunicorn for robust production performance.

Conclusion

The credit card fraud detection project achieved significant success. Key takeaways include:

- XGBoost demonstrated superior performance with an accuracy of 99.4%, making it the most effective model for fraud detection.
- The implementation of SMOTE proved critical in improving fraud detection by addressing the dataset's class imbalance.
- The deployed API enables real-time processing of fraud detection requests, providing immediate insights into transaction legitimacy.



Future Scope

To further enhance the fraud detection system, the following improvements are recommended:

- Fine-tune hyperparameters of the XGBoost model to achieve even better precision and recall.
- Explore the use of Deep Learning models, such as LSTM networks, to capture sequence-based patterns in transaction data.
- Deploy the model on cloud services (e.g., AWS, GCP, Azure) to ensure scalability and reliability.

These enhancements will contribute to a more robust and adaptive fraud detection system.