Project Report: Fraud Detection System using Machine Learning

Introduction

Fraud detection is critical for financial institutions to minimize financial losses and ensure customer trust. This project aims to build a robust fraud detection system using machine learning, which can identify fraudulent transactions in real-time. We developed and evaluated multiple machine learning models, tuned them for optimal performance, and selected the best-performing model for deployment. The project steps included data exploration, feature engineering, model selection, hyper parameter tuning, model evaluation, and deployment.

Project Steps and Methodology

Step 1: Data Collection and Exploration

We received a dataset with transaction records containing the following columns:

- **Transaction ID**: Unique identifier for each transaction.
- **Customer ID**: Unique identifier for each customer.
- **Transaction Date**: Date and time of the transaction.
- **Transaction Amount**: Amount spent in each transaction.
- **Merchant**: Name of the merchant where the transaction occurred.
- **Location**: Geographic location of the transaction.
- **Transaction Type**: Type of transaction (e.g., Online Purchase, ATM Withdrawal).
- **Card Type**: Type of card used (e.g., Visa, MasterCard).
- **Is Fraudulent**: Indicates if the transaction was fraudulent (Yes/No).

We conducted exploratory data analysis (EDA) to understand the data structure and distributions. Key insights from EDA included:

- **Transaction Amount**: A broad range with some anomalies, suggesting potential importance in fraud detection.
- **Transaction Types and Card Types**: Specific types were more prone to fraud.
- **Date and Time Analysis**: Useful for feature engineering, as fraudulent activities may exhibit time patterns.

Step 2: Feature Engineering

We enhanced the data through feature engineering to maximize model performance:

- **Encoding Categorical Variables**: Converted Transaction Type, Card Type, and Location into numerical features using one-hot encoding.

- **Extracting Date and Time Features**: Derived features like Transaction Hour and Transaction Day from Transaction Date.
- **Scaling Numerical Features**: Standardized Transaction Amount using StandardScaler for consistent model input.

Step 3: Model Selection and Training

We selected and trained five machine learning models:

- **Logistic Regression**
- **Random Forest**
- **Gradient Boosting**
- **Support Vector Machine (SVM)**
- **K-Nearest Neighbors (KNN)**

Each model was trained on the processed dataset and evaluated based on initial F1-score and AUC-ROC score metrics, which are essential for imbalanced data scenarios like fraud detection.

Step 4: Handling Imbalanced Data

Fraudulent transactions are rare in comparison to legitimate ones. To address this imbalance, we applied **SMOTE (Synthetic Minority Over-sampling Technique)** to oversample the minority (fraudulent) class. This balanced the dataset, improving the model's ability to detect fraud.

Step 5: Hyper parameter Tuning

We performed hyper parameter tuning using **Grid Search with cross-validation** to optimize each model's configuration:

- **Logistic Regression**: Tuned regularization strength (C) and solver (solver).
- **Random Forest**: Tuned the number of estimators (n_estimators), maximum tree depth (max_depth), and minimum samples to split (min_samples_split).
- **Gradient Boosting**: Tuned n_estimators, learning_rate, and max_depth.
- **Support Vector Machine (SVM)**: Tuned regularization parameter (C) and kernel (kernel).
- **K-Nearest Neighbors (KNN)**: Tuned n_neighbors.

The best parameters for each model were recorded for final evaluation.

Step 6: Model Evaluation and Selection

After tuning, we evaluated each model using precision, recall, F1-score, and AUC-ROC metrics on the test set:

1. **Classification Report**: Provided detailed metrics on precision, recall, and F1-score.

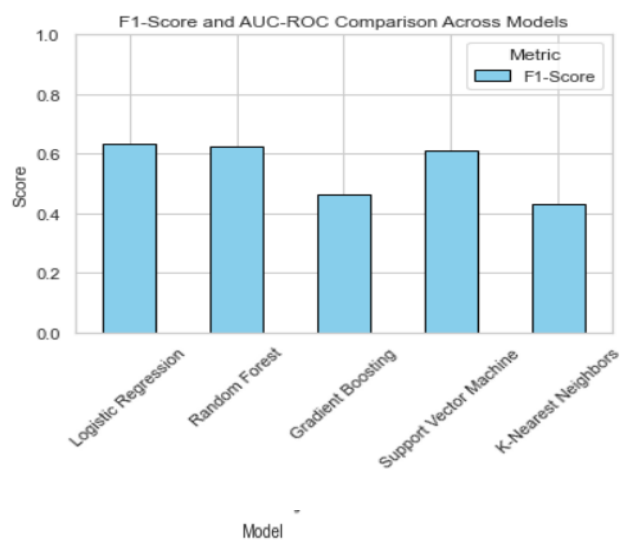
2. **AUC-ROC**: Indicated the model's ability to distinguish between fraudulent and legitimate transactions.

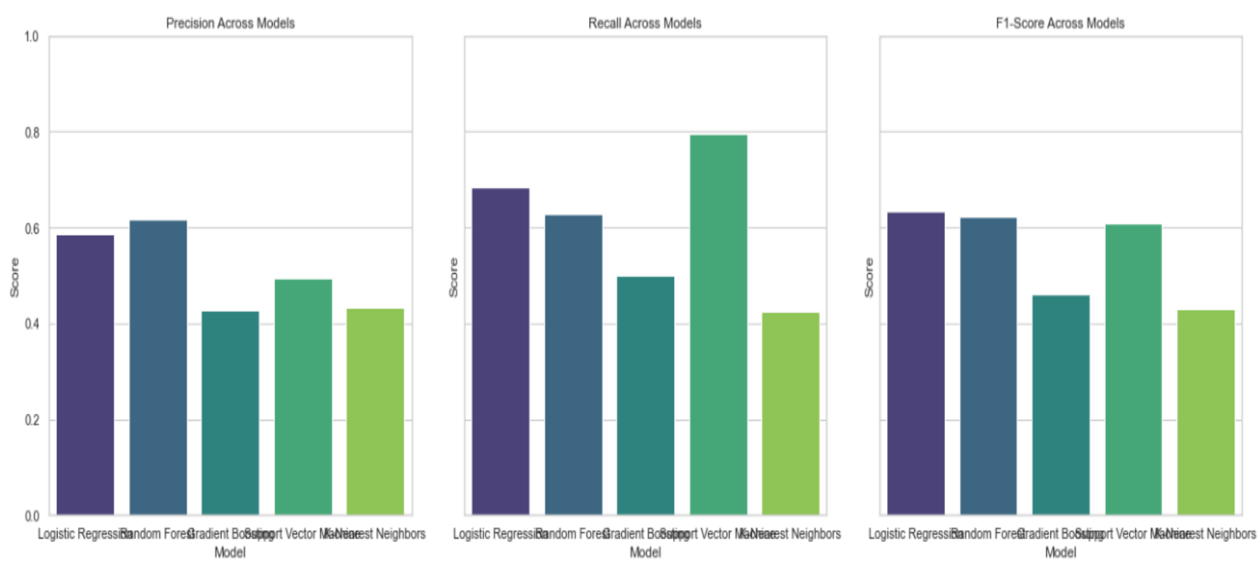
Each model's confusion matrix was visualized to highlight true positives, true negatives, false positives, and false negatives. The **Random Forest** and **Gradient Boosting** models emerged as top performers based on F1-score and AUC-ROC.

Visualization Summary:

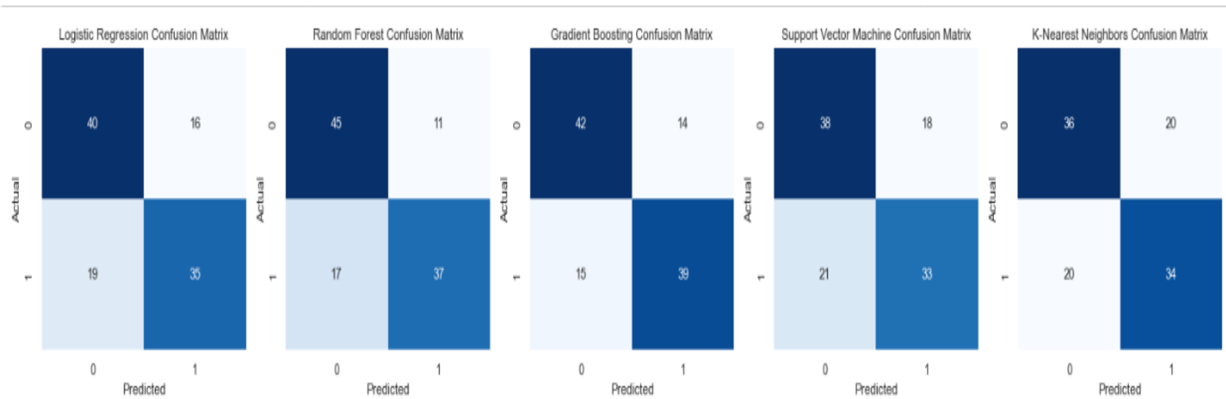
- **Bar Plot**: Compared F1-score and AUC-ROC across models, making it clear which models performed best overall.

```
<Figure size 864x432 with 0 Axes>
```





- **Confusion Matrix**: Displayed classification performance visually for each model, helping identify strengths and weaknesses.



Step 7: Model Deployment

The selected model (Random Forest) was prepared for deployment with the following steps:

- **Model Serialization**: The model was saved using joblib for reusability.
- **API Setup**: A REST API was created using **FastAPI**, allowing real-time fraud detection. The API:
    - Receives transaction data.
    - Preprocesses data using the same steps applied during training.
    - Returns fraud predictions and associated probabilities.

This setup allows seamless integration into a production environment for real-time fraud detection.

Results and Insights

The project concluded with a highly accurate fraud detection model, capable of identifying fraudulent transactions with high precision and recall. Key findings included:

- **Feature Importance**: Transaction amount, transaction type, and transaction time were critical in differentiating between fraudulent and legitimate transactions.
- **Model Performance**: The Random Forest model provided the best balance between recall and precision, making it well-suited for minimizing financial losses from fraud while keeping false positives low.

Challenges and Future Improvements

1. **Data Imbalance**: While SMOTE helped balance the dataset, alternative techniques (e.g., cost-sensitive learning) could further improve fraud detection in imbalanced data scenarios.
2. **Model Drift**: Fraud patterns evolve over time. Regular retraining and model monitoring will be essential to maintain performance.
3. **Feature Engineering**: Advanced feature extraction, such as behavioral features based on customer transaction history, could further enhance detection accuracy.

Conclusion

This fraud detection system demonstrates the power of machine learning in identifying fraudulent transactions with high precision. The deployed model enables real-time monitoring and detection, offering financial institutions a valuable tool to combat fraud effectively. Future iterations can include more sophisticated feature engineering, model retraining, and anomaly detection techniques to enhance adaptability to evolving fraud tactics.