FRAUD DETECTION IN CREDIT CARD TRANSACTIONS

1. Introduction

Fraud detection in credit card transactions is an essential task to safeguard financial systems and customer security. The goal of this project is to develop a machine learning model that can effectively identify fraudulent transactions. By leveraging transactional data and advanced techniques, the model aims to detect fraud with high accuracy, reducing financial losses for institutions and customers alike.

2. Data Understanding and Preprocessing

The dataset used for this project contains both fraudulent and legitimate transactions. Given that fraudulent transactions represent a smaller proportion of the dataset, the class imbalance was carefully managed.

Missing or inconsistent data were addressed by imputing missing values and correcting or removing erroneous data. Outlier analysis was conducted to ensure that extreme values did not skew the model's predictions.

3. Feature Engineering

Feature engineering plays a critical role in improving the model's ability to detect fraud. The following features were created:

1. **Transaction Time**: Categorized the transaction times into day segments (e.g., Monday, Tuesday, etc.).
2. **Week Day**: Captured the day of the week the transaction took place.
3. **Transaction_Amount_Bin**: Transaction amounts were binned into ranges to group similar transaction amounts.
4. **Amount_to_Mean_Ratio**: A calculated feature that compares the transaction amount to the overall mean transaction amount.
5. **Amount_to_Global_StdDev_Ratio**: The ratio of the transaction amount to the global standard deviation, identifying outliers in transaction amounts.
6. **Risk**: Assigned a risk score to each transaction type. For example:
    - Online Purchase: 0.504
    - In-Store Purchase: 0.426
    - ATM Withdrawal: 0.577

These engineered features provided valuable insights that enhanced the model's detection capabilities.

4. Categorical Columns Handling

To make categorical data usable in machine learning models, the following transformations were applied:

- **One-Hot Encoding**: Applied to categorical features such as 'Transaction Type', 'Card Type', 'Transaction Time', and 'Week Day', ensuring they were represented as binary variables.
- **Ordinal Encoding**: Used for 'Transaction_Amount_Bin' to preserve the order of transaction amount bins.
- **Label Encoding**: The target variable, 'Is Fraudulent', was encoded as 0 (legitimate) and 1 (fraudulent) for binary classification.

5. Feature Importance

Feature importance was calculated using the absolute correlation with the target label (Is Fraudulent). Only the top 11 features, with correlation values above 0.05, were selected for model development out of the original 19 features. This helped reduce dimensionality and focused the model on the most relevant features.

6. Model Development

Two machine learning models were employed to develop the fraud detection system:

- **Logistic Regression**: A linear model for binary classification, particularly useful for problems where the outcome is either 0 or 1. It estimates the probability of fraud based on input features.
- **Naïve Bayes**: A probabilistic classifier based on Bayes' theorem. It assumes that the presence of a particular feature is independent of other features, which can be useful in detecting fraud when such assumptions hold.

Both models were trained on the preprocessed and feature-engineered dataset.

7. Model Evaluation

The models were evaluated using key performance metrics such as accuracy, precision, recall, and F1 score. The results were as follows:

Logistic Regression Performance:

- **Accuracy**: 0.6712
- **Precision**: 0.6970
- **Recall**: 0.6216
- **F1 Score**: 0.6571

Naive Bayes Performance:

- **Accuracy**: 0.6849
- **Precision**: 0.7188
- **Recall**: 0.6216
- **F1 Score**: 0.6667

While both models showed reasonably good performance, **Naïve Bayes** slightly outperformed **Logistic Regression** in terms of accuracy and precision. However, the recall values for both models were identical, indicating that they were equally capable of identifying fraudulent transactions.

8. Recommendations

To further enhance the effectiveness of the fraud detection system, the following recommendations are made:

- **Ensemble Learning**: Combining models such as Random Forest, Gradient Boosting, or a Voting Classifier could leverage the strengths of multiple algorithms and improve overall performance.
- **Data Augmentation**: To improve the model's robustness, additional features such as geographic location, device type, and IP addresses could be integrated into the dataset.
- **Increase Dataset Size**: The dataset used contained only 363 rows, which is insufficient for training a robust model. Expanding the dataset by including more transactions, especially fraudulent cases, would significantly improve the model's performance. Larger datasets would allow the model to learn better from patterns and reduce the impact of overfitting.
- **Continuous Model Updating**: Given that fraud patterns evolve, regular updates and retraining of the model with fresh data would help maintain its predictive power.
- **Provide Real Values for Merchant and Location**: Instead of dummies 'Merchant' and 'Location' values, it is crucial that datasets provide us with the real values for these fields. Having access to actual merchant names and locations would allow for deeper insights, such as identifying merchants or regions with higher fraud risks. This would enable more sophisticated feature engineering, like calculating risk scores based on past fraudulent behavior for specific merchants or geographic areas, ultimately leading to more accurate predictions.

9. Conclusion

In conclusion, this project successfully developed and evaluated two machine learning models for credit card fraud detection: Logistic Regression and Naive Bayes. Both models demonstrated reasonable performance, with Naive Bayes slightly outperforming Logistic Regression. The model performance metrics such as accuracy, precision, recall, and F1 score indicate that the system is capable of identifying fraudulent transactions while minimizing false positives and false negatives. Future work could focus on improving the model through ensemble techniques, real-time implementation, and the inclusion of additional features for more precise fraud detection.