Report: Fraud Detection Model

Problem Statement

The goal of this project is to build a fraud detection model for financial transactions. The dataset contains information about transactions made by users, including details such as transaction amount, type, and balances of both the sender and receiver. The target variable is `isFraud`, which indicates whether the transaction was fraudulent (`1`) or not (`0`). Fraud detection in financial transactions is critical to preventing financial loss and ensuring system security.

Approach

1. **Data Preprocessing:**
    - The dataset contains both numerical and categorical features.
    - For categorical features (e.g., `type`, `nameOrig`, `nameDest`), I used **Label Encoding** to convert them into numerical representations that the model can understand.
    - Numerical features were scaled using **StandardScaler** to normalize the data and ensure that all features are on the same scale, which helps improve model performance.
2. **Model Selection:**
    - I experimented with several classification models to predict fraudulent transactions. The models considered include:
        - Logistic Regression
        - Random Forest Classifier
    - After training and evaluating all the models using cross-validation and accuracy scores, **Random Forest Classifier** was selected as the best model based on its superior performance.
3. **Model Training and Evaluation:**
    - I used the **Stratified K-Fold cross-validation** technique to ensure that the target variable distribution was preserved in both training and testing datasets.
    - The model was evaluated using an accuracy and **classification report**, which provided insights into precision, recall, and F1 score for detecting fraud.
4. **Handling Imbalanced Dataset:**
    - The dataset was imbalanced, with a higher number of non-fraudulent transactions compared to fraudulent ones. To address this, I applied **Random Under-Sampling** to balance the dataset and ensure that the model wasn't biased toward the majority class.

    - However, further steps such as **SMOTE** (Synthetic Minority Over-sampling Technique) or **ensemble methods** can be explored in future iterations to further improve model performance.
5. **Model Saving and Deployment:**
    - After finalizing the model, I saved the trained Random Forest model, **scaler**, and **label encoder** into separate files using **pickle**.
    - The model and associated files were then used for making predictions on new datasets.
6. **Prediction on New Data:**
    - For predicting new transactions, the saved model was loaded, the new data was preprocessed (encoded and scaled), and predictions were made for whether a transaction is fraudulent or not.

Challenges Faced

1. **Imbalanced Dataset:**
    - One of the main challenges was the imbalanced nature of the dataset, with far fewer fraudulent transactions compared to non-fraudulent ones. This imbalance can lead the model to predict the majority class (non-fraudulent transactions) more frequently.
    - Although I used random under-sampling to address this, further exploration of advanced techniques like SMOTE or ensemble methods will be necessary to enhance the model's ability to detect fraud.
2. **Feature Encoding:**
    - Encoding categorical features, such as `type`, `nameOrig`, and `nameDest`, was essential for the model but needed to be done consistently across the training and prediction phases. This required the use of `LabelEncoder` for transforming categorical data into numerical form.
3. **Data Preprocessing Pipeline:**
    - A robust preprocessing pipeline was required to ensure that all preprocessing steps (encoding, scaling) were applied consistently to both training and new data. This added complexity in ensuring that the transformations did not result in mismatched feature names or inconsistencies.
4. **Overfitting and Underfitting:**
    - Overfitting was a potential risk due to the high variance in the data. Careful tuning of the hyperparameters and using cross-validation helped mitigate this issue.

Future Work

- **Handling Imbalanced Data:**

    - Future updates will include experimenting with more advanced methods to handle class imbalance, such as **SMOTE** or **ensemble techniques**, to better predict fraudulent transactions.
- **Model Improvement:**
    - The current Random Forest model shows promising results, but further hyperparameter tuning and model comparison (e.g., with XGBoost or LightGBM) will help improve accuracy and robustness.
- **Deployment for Real-Time Prediction:**
    - The model can be deployed as part of a web application or service that processes financial transactions in real-time to detect fraud. Further work will be needed to integrate this model into production environments and monitor its performance over time.

Conclusion

The project successfully developed a fraud detection model using the Random Forest algorithm, achieving decent accuracy. However, the imbalance in the dataset remains a challenge, and further techniques to handle this issue are planned for future iterations. Additionally, the saved model, scaler, and encoder allow the application of this model to new, unseen transaction data, enabling real-time fraud detection in financial systems.