Credit Card Fraud Detection

Submitted By - Chahit Kumar Gaware

Overview :-

- This project involves building a machine learning model to detect fraudulent credit card transactions. The dataset used for this project is highly imbalanced, with only a small fraction of transactions being fraudulent. We address this class imbalance using Synthetic Minority Over-sampling Technique (SMOTE) and build a Logistic Regression model to classify transactions as either fraudulent or not.

Key Steps:

1. Dataset Loading and Preprocessing
2. Handling Imbalanced Data
3. Building and Training a Machine Learning Model
4. Evaluating Model Performance
5. Saving and Loading the Model
6. Testing the Model on New Data

Prerequisites:

- Python 3.x
- Libraries:
  Pandas
  Scikit-learn
  Imbalanced-learn
  joblib

Dataset:

- The dataset used for this project is sourced from the Kaggle Credit Card Fraud Detection Dataset. It contains anonymized credit card transaction data with the following details:
- Features: Numerical values resulting from a PCA transformation (for privacy reasons, the original features are not disclosed).
- Label (Class):
  `0` for non-fraudulent transactions.
  `1` for fraudulent transactions.

Workflow:

1. **Dataset Loading and Preprocessing-**
   The dataset is first loaded and inspected for any issues. Since the data has been preprocessed (anonymized and scaled), no significant transformations are needed. However, we handle any missing or irrelevant data, ensuring the features are in the appropriate format.

2. **Handling Imbalanced Data-**
   Since fraudulent transactions are a minority in the dataset, we handle this imbalance using SMOTE (Synthetic Minority Over-sampling Technique). This technique generates synthetic examples of the minority class (fraudulent transactions) to balance the dataset, ensuring the model does not become biased toward non-fraudulent transactions.

3. **Model Building and Training-**
   The machine learning model is built using Logistic Regression, a popular algorithm for binary classification. The model is trained on the resampled dataset after applying SMOTE, and the target variable is the class label (fraudulent or non-fraudulent).

4. Model Evaluation-

To evaluate the model's performance, we use the following key metrics:

- Accuracy: The overall correctness of the model.
- Precision: The proportion of predicted fraudulent transactions that are actually fraudulent.
- Recall: The proportion of actual fraudulent transactions that are correctly identified by the model.
- F1-Score: The harmonic mean of precision and recall, especially important for imbalanced datasets like this.
- Confusion Matrix: Provides a summary of the model's predictions, showing true positives, true negatives, false positives, and false negatives.

5. Model Saving and Loading-

Once trained, the model is saved using joblib for future use. This allows us to reuse the model without retraining. The saved model can be loaded at any time to make predictions on new or unseen data.

6. Testing on New Data-

The saved model is tested using new or test data. This process ensures that the model is generalizing well and can make accurate predictions on unseen transactions. Predictions will output whether a transaction is classified as fraudulent or not.

Evaluation Metrics

- Accuracy: Measures the proportion of correctly classified instances (fraud or non-fraud) out of the total instances.
- Precision: The proportion of correctly predicted fraud cases out of all predicted fraud cases.
- Recall: The proportion of actual fraud cases that were correctly identified by the model.
- F1-Score: The harmonic mean of precision and recall, providing a balance between both metrics in cases of imbalanced data.

- Confusion Matrix: A summary of the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

Confusion Matrix Breakdown:

- True Positives (TP): Correctly identified fraud cases.
- True Negatives (TN): Correctly identified non-fraud cases.
- False Positives (FP): Non-fraud cases incorrectly classified as fraud.
- False Negatives (FN): Fraud cases incorrectly classified as non-fraud.

Future Work

- Experiment with different models: Explore other models like Random Forests, XGBoost, and Neural Networks to see if they provide better results.
- Improve preprocessing: Investigate further improvements to data preprocessing techniques, including feature engineering and selection.
- Real-time fraud detection: Extend the model to work in real-time environments, where transactions are continuously monitored for potential fraud.
- Model optimization: Fine-tune hyperparameters or experiment with ensemble techniques for better performance.

Conclusion

This project demonstrates how to tackle imbalanced datasets with a machine learning model to predict credit card fraud. By leveraging techniques such as SMOTE and Logistic Regression, we effectively addressed class imbalance and built a model that can predict fraudulent transactions with reasonable accuracy and precision.