Machine Learning and Its Applications

Credit Card Fraud Detection Project

By
Kintali Sai Kiran
VU22CSEN0600082

Overview:

In this project, I tackled the challenge of detecting credit card fraud.I used the Credit Card Fraud Detection dataset from Kaggle, which contains 284,807 transactions with 31 features. Only a very small fraction of these transactions (about 0.17%) are fraud, making the dataset highly imbalanced.
My goal was to develop a supervised learning model that effectively identifies fraud transactions.

Implementation:

Data Loading and Exploration

I started by loading the dataset using Pandas. After examining the data's shape, summary statistics, and data types, I confirmed that there were no missing

values. I also visualized the class distribution using Seaborn, which clearly showed the severe imbalance between legitimate and fraud transactions.

Data Preprocessing

1. **Scaling:** I normalized all features using the StandardScaler.
2. **Train_Test_Split:** I split the dataset into a training set (70%) and a test set (30%) using stratified sampling.
3. **Class Imbalance:** I applied **SMOTE** (Synthetic Minority Over-sampling Technique) to the training data which allowed me to generate synthetic examples for the fraud class, which helped the classifier learn more effectively.

Model Training & Evaluation

I chose the **Random Forest classifier** for this project because of its robustness and ability to manage complex datasets. I trained the model on the **SMOTE** balanced training data using default hyperparameters.

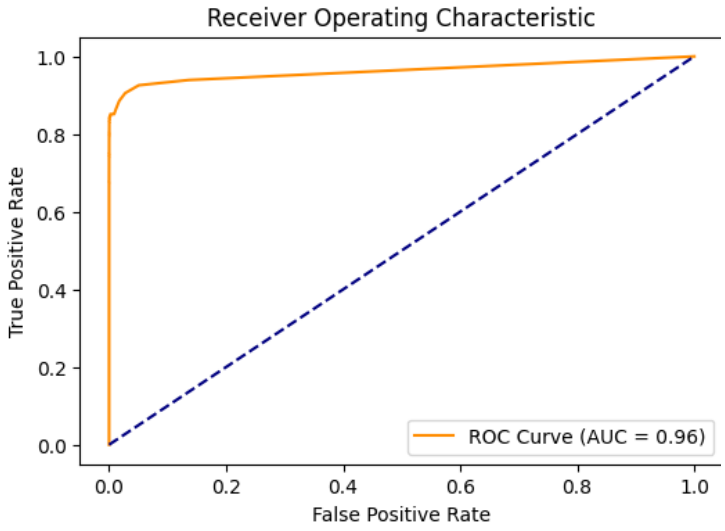
I adjusted the decision threshold from the **default 0.5 to 0.3**. This lower threshold helps increase the recall which means the model is more likely to identify a transaction as fraud even if it introduces more false positives.

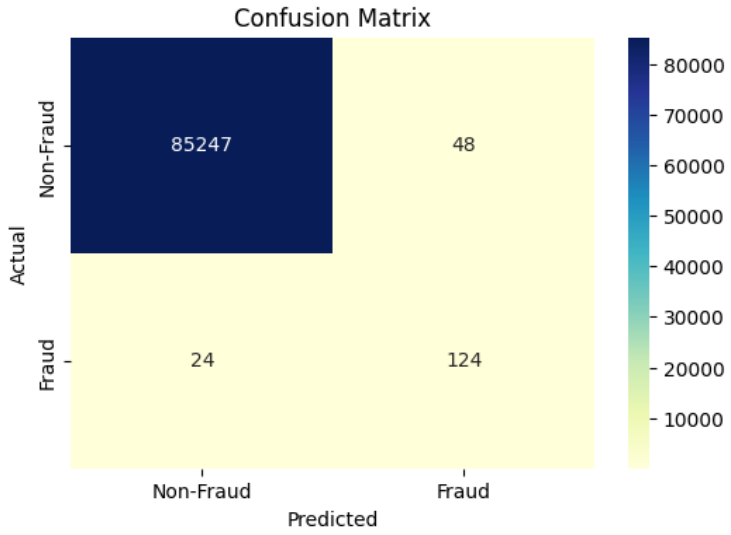
1. I generated Classification report for **Precision, F1 score, recall** etc
2. I Calculated the ROC AUC score
3. I visualized the ROC curve and Confusion Matrix.

Packages used

1. Pandas
2. NumPy
3. Matplotlib & Seaborn
4. Sklearn and Imblearn and
5. Imbalance-learn (SMOTE)
6. Pre-processing, Model_selection, ensemble, metrics, Over_sampling

Result:





Classification Report (threshold = 0.3):

	precision	recall	f1-score	support
0	1.00	1.00	1.00	85295
1	0.72	0.84	0.78	148
accuracy			1.00	85443
macro avg	0.86	0.92	0.89	85443
weighted avg	1.00	1.00	1.00	85443

Conclusion

Overall, I achieved strong performance in detecting credit card fraud. With a recall of 84% for the fraud class and a high ROC AUC score, the model demonstrates effective detection.

Improvements can be done:

1. I could experiment with even lower thresholds (e.g., 0.2) to potentially improve recall further.
2. Fine-tuning the Random Forest parameters or testing alternative algorithms like XGBoost might yield even better results.

This project provided me with an understanding on how to handle imbalanced datasets, model evaluation with respect to fraud detection, and the trade-offs between precision and recall and other Metrics.