Credit Card Fraud Detection

Github Repo Link:
https://github.com/atharva-diwan/credit_card_fraud_detection_cs419.git

Submitted By:-

Atharva Diwan (190100028)
Samyak Ajmera (19B080023)
Ankit Yadav (190100019)
Parag Bajaj (190100088)
Abhinav Singh (19D180002)

Motivation

- Cybersecurity is becoming increasingly important. When it comes to digital security, the most difficult task is detecting unusual activities.
- Credit limit in credit cards sometimes helps us make purchases even if we don’t have the amount at that time.
- These features are misused by cyber attackers
- We need a system that can abort the transaction if it finds fishy

Data Processing & Understanding

- The exact variables are not disclosed due to security concerns, however they have been modified versions of PCA. As a consequence, there are one time, 29 feature columns and one final class column to be found.
- The dataset is imbalanced towards a feature “legit transaction”.
- Our dataset has no null values
- The mean amount of Fraudulent transactions is greater than the legit
- We removed duplicate transactions

Training data & Test data - Splitting data

- Since our dataset is significantly unbalanced, we first undersample the data from the majority class.
- We upsample the minority class using SMOTE and build a sample dataset containing similar distribution of normal transactions and Fraudulent Transactions
- We divide the data into two datasets - training data and testing data

SMOTE (Synthetic Minority Oversampling Technique)

- SMOTE starts by picking a minority class instance at random and then looking for its k closest minority class neighbours.
- The synthetic instance is then constructed by randomly selecting one of the k nearest neighbours b and connecting a and b in the feature space to form a line segment.
- The synthetic instances are created by combining the two chosen examples a and b in a convex way.

Model Building - Logistic Regression, SVM

```
CONFUSION MATRIX
[[19797   204]
 [    9    89]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99955   0.98980   0.99465     20001
           1    0.30375   0.90816   0.45524        98

    accuracy                        0.98940     20099
   macro avg    0.65165   0.94898   0.72495     20099
weighted avg    0.99615   0.98940   0.99202     20099

SCALAR METRICS
          MCC = 0.52187
        AUPRC = 0.84508
        AUROC = 0.96483
Cohen's kappa = 0.45123
     Accuracy = 0.98940
```

- It is memory efficient as it uses a subset of training points in the decision function
- Uses SGD learning to create regularized linear models
- Data should have a zero mean and unit variance for the best results when using the default learning rate schedule

Model Building - Random Forest

```
CONFUSION MATRIX
[[19994     7]
 [   13    85]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99935   0.99965   0.99950     20001
           1    0.92391   0.86735   0.89474        98

    accuracy                        0.99900     20099
   macro avg    0.96163   0.93350   0.94712     20099
weighted avg    0.99898   0.99900   0.99899     20099

SCALAR METRICS
          MCC = 0.89469
        AUPRC = 0.89078
        AUROC = 0.97938
Cohen's kappa = 0.89424
     Accuracy = 0.99900
```

- Uses numerous decision trees to classify data
- It employs bagging and feature randomization in order to generate an uncorrelated forest of trees
- There needs to be some actual signal in our features
- The predictions made by the individual trees need to have low correlations with each other

Model Building - Decision Tree

```
CONFUSION MATRIX
[[19819   182]
 [   10    88]]

CLASSIFICATION REPORT
              precision    recall  f1-score   support

           0    0.99950   0.99090   0.99518     20001
           1    0.32593   0.89796   0.47826        98

    accuracy                        0.99045     20099
   macro avg    0.66271   0.94443   0.73672     20099
weighted avg    0.99621   0.99045   0.99266     20099

SCALAR METRICS
          MCC = 0.53782
        AUPRC = 0.65611
        AUROC = 0.94032
Cohen's kappa = 0.47450
     Accuracy = 0.99045
```

- Most powerful and popular tool for classification and prediction
- Each internal node denotes a test on an attribute
- Each branch represents an outcome of the test
- Each leaf node (terminal node) holds a class label

Conclusion

- We find that the best model which gives highest accuracy in test data is Random Forest

- Found that the five variables most correlated with fraud are, in decreasing order, V14, V10, V12, V4, and V17

- The decision tree achieved MCC score of 0.53, and a random forest achieved a cross-validated MCC score of 0.89

Contribution

NAME	CONTRIBUTION
Atharva Diwan	SMOTE, Random Forest
Samyak Ajmera	SMOTE, Random Forest
Ankit Yadav	Decision Tree
Parag Bajaj	Logistic Regression, SVM
Abhinav Singh	Logistic Regression, SVM & Data analysis

References and Resources

YouTube:
https://www.youtube.com/watch?v=NCgjcHLFNDg

Source Dataset:
https://www.kaggle.com/mlg-ulb/creditcardfraud

Websites:
https://www.geeksforgeeks.org/ml-credit-card-fraud-detection/

https://scikit-learn.org/

THANK YOU