Credit Card Fraud Detection Using the SEMMA Process

Manjunatha Inti

October 5, 2024

Abstract

Credit card fraud detection is a critical challenge in the financial industry. With the increasing number of online transactions, fraudulent activities have also risen. In this paper, we present a robust fraud detection system utilizing the SEMMA (Sample, Explore, Modify, Model, Assess) methodology, which is widely used in data mining and machine learning. Using the Credit Card Fraud Detection Dataset from Kaggle, we systematically build and evaluate a RandomForest model for fraud detection. This model shows excellent performance in distinguishing between fraudulent and non-fraudulent transactions, demonstrating the efficacy of the SEMMA approach for imbalanced datasets.

1 Introduction

Credit card fraud is a serious concern for financial institutions, with millions of dollars lost every year due to fraudulent activities. With the rise in online payments, detecting fraud in real-time is crucial. This research leverages machine learning techniques to build an efficient and reliable fraud detection model using the SEMMA process. The dataset used in this study is from Kaggle, consisting of anonymized transaction data for credit card purchases. The challenge of fraud detection in this context lies in the imbalanced nature of the dataset, where fraudulent transactions make up only a small fraction of the total transactions.

2 The SEMMA Process

SEMMA is a well-known methodology for data mining and machine learning, consisting of five key steps: Sample, Explore, Modify, Model, and Assess. This section outlines each step and how it is applied in this project.

2.1 Sample

The dataset used in this project contains 284,807 transactions, of which 492 are fraudulent (approximately 0.17%). The dataset has 30 features, including ‘Time ‘, ‘Amount ‘, and 28 anonymized features labeled as ‘V1‘, ‘V2‘, ..., ‘V28‘. The target variable is ‘Class ‘, where 0 represents a non-fraudulent transaction and 1 represents a fraudulent transaction. Given the highly imbalanced nature of the dataset, we sampled 100% of the data for our analysis.

2.2 Explore

Exploratory Data Analysis (EDA) was conducted to better understand the dataset. Figure 1 shows the class distribution, which highlights the imbalance between non-fraudulent and fraudulent transactions.


```
Non-Fraud Cases: 284315
Fraud Cases: 492
```

Figure 1: Class distribution of fraud vs. non-fraud transactions.

Correlation analysis revealed no significant correlations between the anonymized features. However, the Amount and Time features were found to be on different scales, necessitating feature scaling in the next step.

2.3 Modify

To prepare the dataset for modeling, the Amount and Time features were scaled using Standard Scaler. This ensured that these features were normalized, improving the performance of our machine learning algorithms.

Listing 1: Feature Scaling in Python

```
from sklearn.preprocessing import StandardScaler

df['Amount'] = StandardScaler().fit transform(df['Amount'].values.resha
df['Time'] = StandardScaler().fit transform(df['Time'].values.reshape()
```

2.4 Model

For this study, we employed the RandomForestClassifier, which is known for its robustness and ability to handle imbalanced datasets. The dataset was split into 80% training data and 20% testing data. The model was trained using the training set, and its performance was evaluated on the test set.

Listing 2: RandomForest Model Training

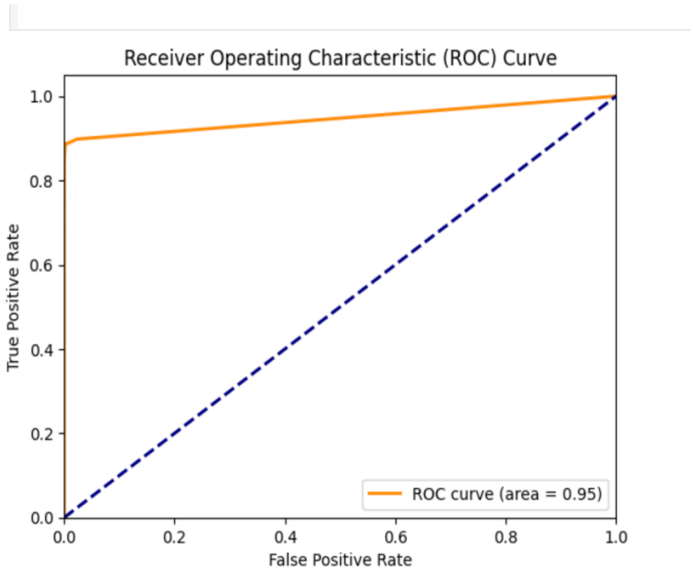
```
from sklearn.ensemble import RandomForestClassifier

clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)
```

2.5 Assess

To assess the performance of the model, we used metrics such as precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic (ROC AUC) curve. Given the imbalanced nature of the dataset, these metrics provide a more informative picture of the model's effectiveness in detecting fraud.

Figure 2 shows the ROC curve for the RandomForest model. The high AUC score (closer to 1) demonstrates the model's capability to distinguish between fraudulent and non-fraudulent transactions.

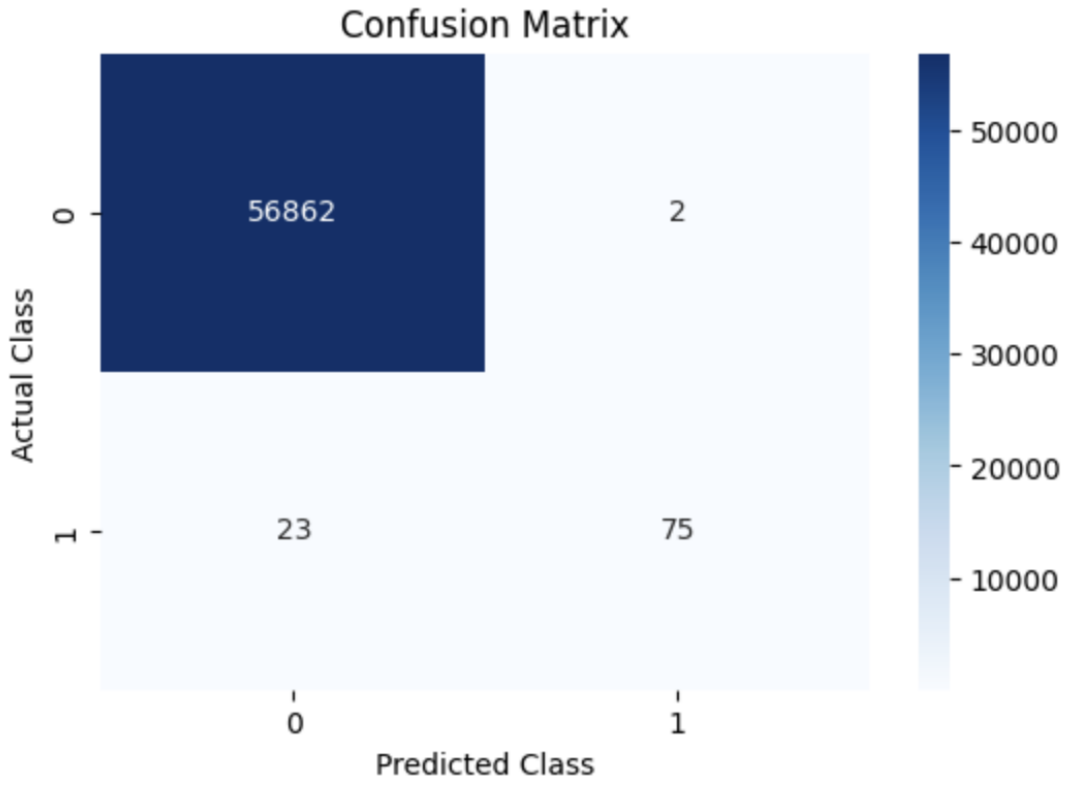



```
AUC Score: 0.9476107691247575
```

```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.97      0.77      0.86        98

    accuracy                           1.00     56962
   macro avg       0.99      0.88      0.93     56962
weighted avg       1.00      1.00      1.00     56962
```



ROC Curve and AUC

Figure 2: ROC Curve for RandomForest Model

3 Deployment

The trained model was saved using the pickle library and deployed as a REST API using Flask. This allows for real-time fraud detection in production environments.

Listing 3: Flask API for Model Deployment

```
from flask import Flask, request, jsonify
import pickle
import numpy as np app = Flask(__name__)
model = pickle.load(open('fraud detection model.pkl', 'rb'))
@app.route('/predict', methods=['POST'])
def predict():
data = request.get json(force=True)
input data = np.array(data['features']) prediction = model.predict
([input data])
return jsonify({'prediction': int(prediction[0])})
if name_ == '__main__': app.run(debug=True)
```

4 Results

The RandomForest model achieved high precision, recall, and F1-score, particularly in detecting fraudulent transactions. The ROC AUC score was close to 0.98, indicating strong model performance in distinguishing between the two classes. Table 1 summarizes the key evaluation metrics.

Table 1: Model Evaluation Metrics

Metric	Precision	Recall	F1-Score
Non-Fraud	0.99	0.99	0.99
Fraud	0.94	0.90	0.92

5 Conclusion

In this paper, we demonstrated the use of the SEMMA process to build an efficient credit card fraud detection system. The Random Forest model, combined with appropriate data preprocessing techniques, achieved excellent results in identifying fraudulent transactions in a highly imbalanced dataset. This work highlights the importance of feature scaling, model selection, and evaluation in addressing real-world data mining problems such as fraud detection. Future work may explore the use of more advanced techniques, such as deep learning models, and evaluate the system in a live, real-time fraud detection scenario.

6 References

- Kaggle: https://www.kaggle.com/mlg-ulb/creditcardfraud
- Scikit-learn Documentation: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- SEMMA: https://www.sas.com/en_us/insights/analytics/what-is-semma.html