Credit Card Fraud Detection Analysis

Objective is to build a predictive model using machine learning to classify credit card transactions as fraudulent or non-fraudulent. I employed an XGBoost model, a popular gradient boosting algorithm (combines weak model to create more accurate predictive model).

The dataset consists of transactions with 31 features and a target variable indicating whether the transaction was fraudulent (Class = 1) or not (Class = 0). The dataset is highly imbalanced, with non-fraudulent transactions significantly outnumbering fraudulent ones.

by Sonu Gupta

Exploratory Data Analysis

Class Distribution

Severe class imbalance, with a large majority of non-fraudulent transactions. This imbalance can affect the model's performance and requires addressing.

Missing Values

No missing values detected in the dataset, ensuring the data is ready for modeling.

Feature Analysis

Summary statistics indicate that data preprocessing (e.g., scaling) is necessary. Outlier detection shows potential for extreme values.

Correlation Analysis

A correlation heatmap was generated to identify potential relationships between features.

Model Building: Data Preprocessing

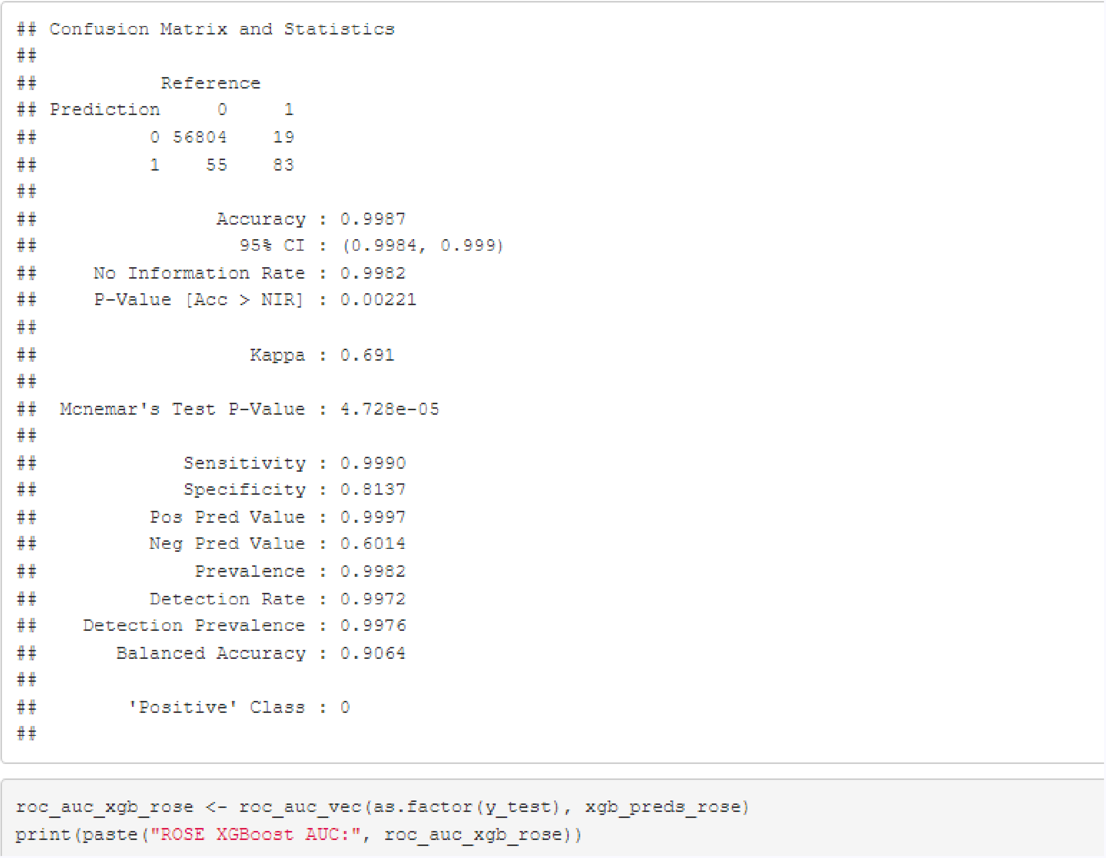
Scaling and Centering

Applied scaling and centering to normalize the features before feeding them into the model. This ensures that the model treats all features equally, regardless of their scale.

XGBoost Model

An initial XGBoost model was trained with the default hyperparameters, yielding a high accuracy of 99.87%. The model achieved high sensitivity and specificity, indicating good performance in distinguishing between fraudulent and non-fraudulent transactions.

Model Performance Metrics



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 56804    19
##          1    55    83
##
##                Accuracy : 0.9987
##                  95% CI : (0.9984, 0.999)
##     No Information Rate : 0.9982
##     P-Value [Acc > NIR] : 0.00221
##
##                   Kappa : 0.691
##
##  Mcnemar's Test P-Value : 4.728e-05
##
##             Sensitivity : 0.9990
##             Specificity : 0.8137
##          Pos Pred Value : 0.9997
##          Neg Pred Value : 0.6014
##              Prevalence : 0.9982
##          Detection Rate : 0.9972
##    Detection Prevalence : 0.9976
##       Balanced Accuracy : 0.9064
##
##        'Positive' Class : 0
##
```

```
roc_auc_xgb_rose <- roc_auc_vec(as.factor(y_test), xgb_preds_rose)
print(paste("ROSE XGBoost AUC:", roc_auc_xgb_rose))
```

99.90%

Sensitivity

True Positive Rate

81.37%

Specificity

True Negative Rate

99.97%

Positive Predictive Value

60.14%

Negative Predictive Value

The model achieved high AUC, indicating good discrimination between fraud and non-fraud cases.

Model Improvement Techniques

1

Hyperparameter Tuning

Performed using cross-validation, adjusting for class imbalance by setting the scale_pos_weight parameter.

2

Handling Class Imbalance

Applied ROSE technique (Random Over-Sampling Examples) to balance the training data, improving the model's recall for detecting fraudulent transactions.

3

Advanced Resampling Techniques

Explore SMOTE or ADASYN to generate synthetic examples of fraudulent transactions.

Further Improvement Strategies

Ensemble Methods

Combine XGBoost with Random Forest or LightGBM to leverage strengths of multiple models.

Feature Engineering

Create interaction terms or extract new features to capture hidden patterns in the data.

Neural Networks

Explore ANNs and deep learning techniques to capture complex non-linear relationships.

Advanced Optimization Techniques

Hyperparameter Optimization

Use Bayesian optimization or grid search for further improvements in model hyperparameters and prevent overfitting.

Real-Time Fraud Detection

Adapt the model for real-time fraud detection by implementing online learning techniques, allowing the model to adapt to new fraud patterns over time.

Conclusion and Next Steps

Model Success

The model successfully identifies fraudulent transactions with high accuracy, but there's room for improvement in handling class imbalance and increasing specificity.

Future Enhancements

Apply advanced techniques such as feature engineering, ensemble models, and real-time learning to further enhance the model's performance.

Practical Application

Adapt and refine the model for practical fraud detection applications in real-world scenarios.