knowledgehut upGrad

Data Science Bootcamp

Capstone Project

By : Rishab Mattoo

Email: rishabmattoo5@gmail.com

FindDefault (Prediction of Credit Card fraud)

Problem Statement:

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash.

It is important that credit c ard companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase.

The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

1. Introduction

This report details the development and evaluation of a K-Nearest Neighbors (KNN) model for credit card fraud detection. The project addressed a highly imbalanced dataset with a significant number of legitimate transactions and a small minority of fraudulent ones.

2. Design Choices

- **Data Balancing:** SMOTE (Synthetic Minority Oversampling Technique) was employed to address class imbalance and prevent bias towards the majority class.
- **Model Selection:** KNN was chosen for its effectiveness with numerical data and interpretability. Hyperparameter tuning with GridSearchCV optimized the distance metric (Euclidean distance with p=2).
- **Evaluation Metrics:** Accuracy, F1-Score, and AUC-ROC were used to assess model performance.

3. Performance Evaluation

- **Accuracy:** 99.83% (achieved with SMOTE and KNN with Euclidean distance)
- **F1-Score:** 63.18% (Balances precision and recall for imbalanced data)
- **AUC-ROC:** 92.83% (Measures the model's ability to differentiate b/w classes)

4. Discussion

- The high accuracy and F1-score suggest the model effectively distinguishes fraudulent transactions.
- However, real-world application requires a balance between accuracy and false positives (declining legitimate transactions).
- Generalizability to unseen data needs further evaluation.
- Cost-sensitive learning, considering the financial impact of misclassifications, is crucial for practical implementation.
- KNN's computational and memory demands necessitate optimization for deployment in production environments.

5. Model Deployment

Deploying the KNN model for credit card fraud detection requires careful consideration of its strengths and weaknesses:

- **Strengths:**
    - Interpretability: KNN allows understanding the factors influencing fraud classification.
    - Effectiveness with numerical data: Well-suited for credit card transaction data.

- **Weaknesses:**
    - Computational demands: KNN can be computationally expensive for large datasets.
    - Memory requirements: Storing all training data for nearest neighbor search can be memory-intensive.

Deployment Strategies:

- **Approaches to reduce computational cost:**
    - Utilize techniques like k-d trees or ball trees for efficient nearest neighbor search.
    - Limit the number of neighbors considered during prediction.
- **Approaches to reduce memory usage:**
    - Implement incremental learning, where the model updates with new data without storing everything.
    - Explore dimensionality reduction techniques to reduce feature space size.
- **Deployment Options:**
    - Cloud-based deployment: Leverages cloud resources for scalability and handling large datasets.
    - On-premise deployment: Suitable for scenarios with stricter data privacy requirements.

6. Future Work

- Evaluate the model on unseen data to assess real-world generalizability.
- Implement cost-sensitive learning to optimize the model for practical applications.
- Explore alternative models like Random Forest or Gradient Boosting for potential improvements.
- Develop a comprehensive deployment plan considering KNN's computational and memory requirements, including chosen optimization strategies and deployment options.
- Continuously monitor and refine the model with new data to maintain accuracy.

7. Conclusion

The KNN model with SMOTE and Euclidean distance optimization achieved excellent performance on the test data. However, further exploration is necessary to ensure the model's effectiveness in real-world scenarios and to optimize its practicality for credit card fraud detection. Careful consideration of deployment strategies is crucial to address KNN's computational and memory limitations.