Literature Review 1

From Python to Julia: Feature Engineering and ML

By Wang Shenghao

Introduction

Shenghao authors an article to demonstrate model building in Julia, which is a high-level language like Python, but with the fast performance of a low-level language, like C. He builds a financial fraud detection model, and goes through the feature engineering, training, and testing of the model in both Julia and Python.

Summary

Shenghao uses a supervised credit card transaction dataset from Kaggle with 30 columns, 28 of which are obtained by PCA. He begins by feature engineering, starting with splitting his data, and making sure that it is stratified. Next, he performs a standard scaling of the data. According to him, scaling helps improve network convergence and prevents an individual feature dominating during training. Then he uses a technique called SMOTE to oversample the minority fraud class. The data is heavily imbalanced in favour of the negative class, and SMOTE needs to be used to synthetically create data for the positive class. Finally, he trains an XGBoost model and tests its precision and recall. He finds that the Julia implementation takes longer to train than the Python one, but it displayed slightly better metrics.

Analysis and Takeaways

A major challenge in financial fraud detection is class imbalance. In these datasets, there are huge imbalances, on the scale of 500:1 in favour of the negative class. It is imperative that counter strategy is employed during the data processing step. We intend to use SMOTE, just like the author, to generate synthetic samples of our minority class. Both Shenghao's and our datasets are similar, so we think it will be a necessary step for our project. We will also employ his strategy of scaling the data and stratifying it during splitting. In conclusion, this article gave our team a great foundation for data preprocessing: namely to stratify split, scale, and resample using SMOTE.

Literature Review 2

Credit Card Fraud Detection in Python

By Usevalad Ulyanovich

Introduction

Credit card fraud detection has become a critical application of machine learning, addressing the substantial economic losses businesses face due to fraudulent transactions. The guide by Fively explores how Python's rich ecosystem of libraries and tools supports the development of effective fraud detection models, specifically for financial, healthcare, and e-commerce

Summary

The guide outlines a structured approach to building fraud detection models using Python, starting with data preparation and analysis. Techniques like Exploratory Data Analysis (EDA) and dataset partitioning (train-test split) are utilized to identify patterns that differentiate fraudulent transactions from legitimate ones.

Model selection involves using six classification algorithms—K-Nearest Neighbors (KNN), Logistic Regression, Support Vector Machine (SVM), Random Forests, XGBoost, and Decision Tree—to predict fraudulent activity. These models are then evaluated using metrics such as Accuracy, F1 Score, and Confusion Matrix to ensure robustness and precision. Python libraries like Scikit-learn and XGBoost play a crucial role in efficient model development and testing, while additional tools enhance visualization and data manipulation. Furthermore, the blog highlights Python's versatility in automation, web development, and data analysis, showcasing its ability to create scalable and adaptable solutions for fraud detection.

Analysis and Takeaways

The methodology presented highlights Python's versatility and underscores the significance of a data-driven approach to fraud detection. By incorporating various classification models, the approach allows for performance comparison, while the use of evaluation metrics ensures actionable insights. The methodology's strengths include comprehensive coverage of fraud detection processes, a clear explanation of Python's role in simplifying complex tasks, and practical code examples for implementation. However, there are notable limitations, such as a reliance on high-quality data, which may not be accessible to all organizations, and the need for advanced technical expertise to build and maintain the models.

References

Shenghao, W. (2023, June 27). *From python to julia: Feature engineering and ML*. Medium. https://towardsdatascience.com/from-python-to-julia-feature-engineering-and-ml-d55e8321f888

Ulyanovich, U. (2022, August 19). 🔐 A Guide to Credit Card Fraud Detection in Python. https://5ly.co/blog/fraud-detection-in-python/