Machine Learning Project Report

IMT2022019

IMT2022087

IMT2022515

Project Overview

This project involved developing and tuning predictive models to analyse financial loan default risk. The dataset, sourced from Coursera's Loan Default Prediction Challenge, contains 255,347 rows and 18 columns, representing various customer attributes and loan-related information. The primary goal was to use machine learning to predict which individuals are at the highest risk of defaulting on their loans, helping financial institutions proactively identify and intervene with at-risk clients. To achieve this, we explored several machine learning models and configurations, including standalone XGBoost, XGBoost with grid search, and stacking, to identify the model that yielded the highest accuracy in predicting defaults.

Preprocessing Steps

After analyzing the dataset, it was determined that no substantial preprocessing was necessary. The dataset was already clean and standardized, and dropping any columns led to reduced accuracy, highlighting the importance of each feature. As a result, all columns were retained for the analysis.

Key Findings:

→No preprocessing required: The dataset's quality allowed for model training without additional data transformations.
→Dropping columns: Removal of columns resulted in worse performance, indicating that each feature contributed positively to model accuracy.

Model Selection and Tuning

Multiple models and optimization strategies were tested:

1. Grid Search: Applied to fine-tune parameters for decision trees and XGBoost. This systematic parameter search aimed to maximize performance via cross-validation.

2. Standalone XGBoost: This model, without additional tuning, outperformed the grid-searched version, yielding the highest accuracy in the project.

3. XGBoost with Grid Search: XGBoost was optimized with grid-searched hyperparameters, such as `max_depth`, `learning_rate`, and `n_estimators`, which improved accuracy, but the results did not surpass the performance of standalone XGBoost.

4. Stacking: Ensemble stacking was also implemented by combining multiple models to improve robustness. However, this approach did not outperform standalone XGBoost.

Final Results

→Best Model: Standalone XGBoost achieved the highest accuracy among all approaches.
→Performance Metrics: Standalone XGBoost consistently outperformed both the grid-searched and ensemble models, demonstrating superior predictive accuracy.

Possible Reason for Standalone XGBoost's Superior Performance

Grid search can sometimes lead to **overfitting**, particularly if hyperparameters are tuned specifically to maximize performance on cross-validation data. This may reduce generalization on the test set. The standalone XGBoost model, on the other hand, retained a balance in default parameter settings, likely resulting in better test performance.

Conclusion

Standalone XGBoost proved to be the most effective model for this dataset. While grid search provided modest improvements for other models, it did not outperform the default XGBoost configuration. This analysis underscores the robustness of XGBoost's default settings and the minimal need for feature engineering with this dataset.