Advanced Machine Learning - Project 2

Mieszko Mirgos, Tomasz Siudalski, Piotr Robak

March 2024

Contents

1 Introduction

This report contains the results of our work for Project 2 of the Advanced Machine Learning course. The goal of the project was to select the top 1000 data points most likely to belong to class one (the problem is a binary classification) of the data while utilizing as few features as possible, specifically to maximize the score $10 *$ correct positive class predictions $- 200 *$ Number of features . To achieve the objective we were supposed to utilize at least 5 different methods of model/feature selection.

2 Methodology

In this section, we will describe the methods used to obtain the most relevant features subset.

2.1 Correlation

We checked for any correlation between the variables as for some methods that may impact the selection process. It turned out that features from 0 to 9 have a high correlation over 0.7. For methods other than Boruta we left only feature number 0.

2.2 Variable selection methods

We utilized several methods for the calculation of variable significance, which will be presented in this section. With each method, we selected the top 10 most significant features. Then for each method, we tested different subsets of the variables to further narrow down the search. We created a framework that tested all subsets counting 1-5 variables (it quickly turned out more does not help). For measuring the performance it used the Gaussian Naive Bayes classifier as it deemed very good results compared to other models and was very fast.

2.2.1 Boruta

The first tested method was Boruta. It consistently selected features 0-9 and 100-105 as the most significant. The framework with Gaussian Naive Bayes proven 3-4 features subsets of 100-105 yield the best results with scores approximating 6800-7000. The variables 0-9 turned out not to be so helpful. The best configuration was [101, 102, 103, 105], but with very small advantage over other 100-105 subsets.

2.2.2 Feature importance from xgboost

The second tested approach was based on feature importance from the XGBoost Classifier. In this case, the most important features were also 100-105 and additionally 0, 285, 391, 182. After running the additional experiments they returned very similar results, subsets of features 100-105 again turned out to be the most important scoring in the 6800-7100 range.

2.2.3 Boruta + SequenceFeatureSelector

Other methods like the SequenceFeatureSelector are computationally expensive therefore we run them on the top 50 features selected from Boruta. We paired the algorithm with the LogisticRegression and ran to choose the top 10 variables. The results are: 101, 403, 285, 155, 337, 471, 412,

131, 241, 335 which differ from the previous methods. The scores obtained from the framework are much lower, scoring 5900-6200. The best configuration was a single 101 variable.

2.2.4 Boruta + RecursiveFeatureElimination

The next algorithm is Recursive Feature Elimination paired with Linear SVM. The selected features are: 458, 131, 215, 316, 63, 360, 328, 133, 75, 412. The scores in this case are very low, in the range 4900-5100, often returning single variables which suggests they don't work well together.

2.2.5 Boruta + Mutual Info score

The last investigated method was choosing features based on the Mutual Information score. The results are: 101, 296, 328, 103, 412, 105, 0, 131, 351, 323. The two best configurations obtained after running the framework were [101, 103, 105] and [101, 105] scoring respectively 6740 and 6670.

2.3 Final features

Based on the experiments, the features 100-105 were most promising as they achieved the best results, other features were not even close. Therefore we decided to focus mostly on these features in the next experiments, trying to build more advanced models to further improve the score.

3 Results

We decided to probe the performance of models on the problem by selecting a few methods and optimizing their hyperparameters, then we built ensembles of the best-performing methods. We initially used both Bayesian (BOHB) and non-Bayesian(random points in search space) algorithms to tune the hyperparameters, both methods worked similarly so we decided to go forward with BOHB. Ray, the library we used allowed us to define a custom score function so all values will be presented in terms of this score.

When it comes to ensembles the class probability is the mean of classifiers in the ensemble

- XGBoost, max score: 7150, 4 features used
- Random Forrest, max score: 7050, 4 features used
- Naive Bayes, max score 7500, 3 features used
- Ensemble (XGBoost, ExtraTrees, SVC, Logistic Regression), max score: 7800, 4 features used
- Ensemble (XGBoost, ExtraTrees, SVC, Logistic Regression) x3, each with independent hyperparameters, max score 7750, 3 features used
- Ensemble (XGBoost, ExtraTrees, SVC, Logistic Regression, KNN), max score: 7650.0, 3 features used,
- Ensemble (XGBoost, SVC, Logistic Regression, Naive Bayes), max score: 7500.0, features used 3 to 4 depending on exact trial
- AutoMl using mljar-supervised (competitive mode, 600 seconds per feature set), max score: 7400.0, features used 3 to 4 depending on the exact trial.

- AutoMl using mljar-supervised (competitive mode, 3000 seconds per feature set), max score: 7300.0, features used 3 to 4 depending on the exact trial.

Then we conducted further testing for the most promising models, by repeatedly training the best configuration on 200 random 80/20 splits. Results are presented below

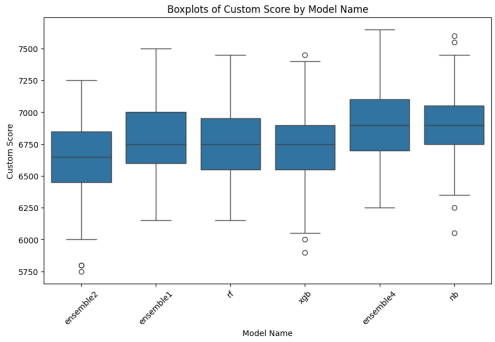

Figure 1: Estimated earnings across different top configurations of different models

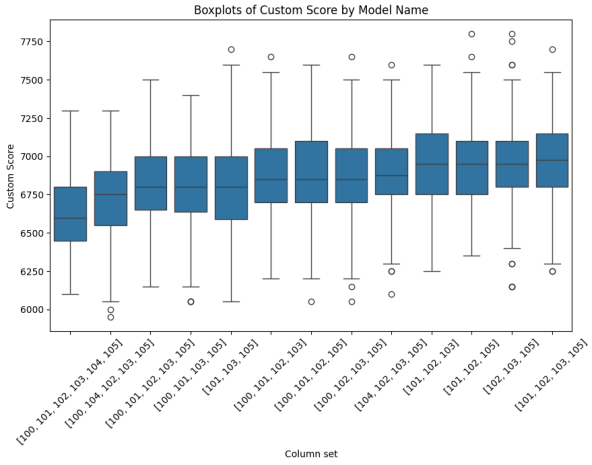

Figure 2: Estimated earnings across different column sets for Naive Bayes

4 Conclusion

Based on the results of the experiments we decided to select features: 102, 103, 105. The model that we chose is an ensemble consisting of: GaussianNB, XGBoost, SVM, and LogisticRegression.