Advanced Machine Learning Project 2

Łukasz Tomaszewski
Patryk Rakus
Michał Tomczyk

May 2024

Contents

1 Introduction

The task of the second project was to propose a model dealing with two problems at the same time: high qualit y of predictions with binary classification and reducing the dataset's number of variables used for training. The dataset was anonymized and consisted of 500 variables describing the customers, with the target variable being whether a customer has taken the advantage of the offer. We were supposed to predict 1000 observations most likely to be classified as positive (meaning the clients that used the offer). The results were validated using a customer metric, where 10 points were given for each true positive observation among the 1000 selected and there was a penalty of 200 for each variable used for training. We have selected various methods of feature selection and have trained numerous different models, evaluating them by both accuracy and the custom scoring method to find the best customers selection. To achieve

that, we have implemented custom evaluation methods. All of our work was written using Python.

2 Methodology

2.1 Feature selection methods

2.1.1 Random Forest

The Random Forest selection method works by fitting the scikit-learn's RandomForestClassifier and using its feature importances which are provided by the fitted attribute feature_importances_ and they are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. To obtain the threshold below which features are selected as unimportant, we sort the feature importances, and select the largest decrease in importance between consecutive scores.

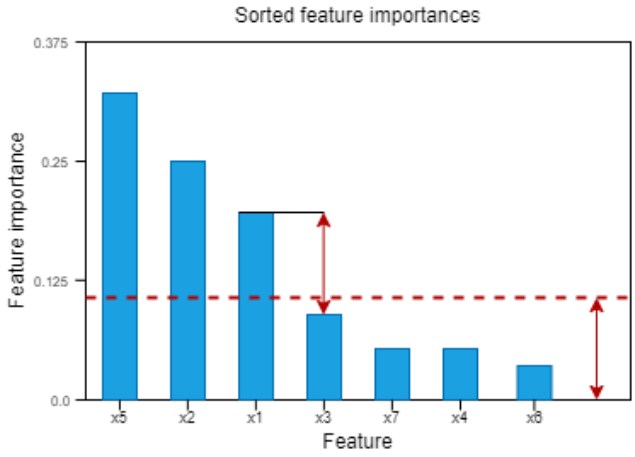

Figure 1: Example plot illustrating how the method works; the red arrow shows the largest consecutive decrease in feature importance and all feature importances below its length are dropped, in this case it's 'x3', 'x7', 'x4' and 'x6'

2.1.2 Logistic Regression Lasso

For this selection method we utilize the L1 penalty to zero the coefficients of unimportant features. As we are dealing with a classification task, we utilize a Logistic Regression model. All features, whose coefficient is equal to 0, are considered unimportant and therefore dropped.

2.1.3 Ensemble selection

To further improve the feature selection process, we have created a class for combining multiple selection methods in 2 different ways:

- bagging - provided feature selection methods choose important features via majority voting
- pipeline - provided feature selection methods are applied consecutively

2.1.4 SHAP

We calculated SHAP values for XGB and LGBM models, which were trained on the dataset with all columns. With the use of summary plot we could decide which variables in our dataset are relevant and have high influence on the prediction.

2.2 Classification models

Apart from applying different feature selection methods, we have also decided to experiment with various models, commonly used for binary classification. We have compared their results to choose the one which achieves the highest scores. The following models were compared:

- Extreme Gradient Boosting (XGBoost)
- Light Gradient Boosting Machine (LGBM)
- Adaptive Boosting (AdaBoost)
- Support Vector Machines (SVM)

Each model was trained and evaluated by both the custom metric score from the task and accuracy score. On XGBoost, LGBM and SVM hyperparameters Bayes optimization was also performed to extract model configuration that allows to achieve the best results.

3 Results

3.1 Selected variables

Less complex selection methods (Logistic Regression Lasso and Random Forest) left too many variables included, which led to high penalty for the number of variables and an unsatisfactory score as a result. The ensemble methods handled this problem better. They reached similar conclusion: variables with indexes 100-105 are the most important (using Python indexing convention, where the first index is 0). Among them, the least important turned out to be the feature with the index of 104 and it was not always included in the model. The results

from ensemble selection corresponded to the largest SHAP values of the full model, which confirmed our conclusions.

We also performed an iterative feature selection in which XGB hyperparameters were optimized on a set of variables, then the model was fitted with the best hyperparemeters that were found, SHAP values were calculated and the least significant variables were removed. We started from the dataset with all columns and halved the number of variables in each step until one column was left. This experiment gave similar results to the previous ones. Because of that we have decided to use columns 100, 101, 102, 103 and 105 when tuning different models.

3.2 Prediction quality

We defined a function for model evaluation. Similar to the given task, we selected only 20% of records in the validation set. These were the records with the highest probability. We calculated the fraction of the true positives in chosen records and scaled the results to get the number of true positives if we were selecting 1000 customers. At the end we added the penalty for the number of used variables. The scores of sample selection methods and models are show in Table 1.

Table 1: Scores of different feature selection methods and models

Selection method	**Number of columns**	**Model**	**Score**
L1	49	XGB	-5066
RandomForestSelector	14	XGB	4600
Bagging	8	XGB	5700
SHAP	5	AdaBoost	6366
Pipeline	5	XGB	6600
SHAP	5	XGB	6733
SHAP	5	LGBM	6766
SHAP	5	SVM	6966

Because the results obtained with the use of SVM were the best and the train accuracy was similar to the validation accuracy, we decided to use this model to make the final predictions.

4 Conclusion

We have applied numerous different feature selection methods to find the most important variables. We have decided to use 5 variables selected with SHAP for the final predictions. Numerous binary classifications models were trained and their performance was evaluated over the task's score and accuracy. After feature selection, the best model turned out to be SVM, which was used for the final prediction after optimizing its hyperparameters.