Advanced Machine Learning - Feature selection

Wiktor Jakubowski, Zuzanna Kotlińska, Jan Kruszewski

May 2024

1 Introduction

The primary goal of the project was to build a model on training data and then identify 1000 customers in the test set who would benefit from the offer the most. The effectiveness of the model was evaluated in the following way:

- For each designated customer who actually took advantage of the offer, we gain €10.
- For each variable used, we must pay €200.

Combining these two factors we obtain a function which we strive to maximize:

$$f(estimator) = 10 \cdot \mathbb{1}_{y=\hat{y}=1} - 200 \cdot n \tag{1}$$

where n is the number of selected features.

2 Methodology

We explored various combinations of feature selector and classifier, using 5-fold cross validation. The averaged results are shown in Table 1.

Table 1: Results of scoring function for each pair of feature selector and classifier

classifier\selector	*KBest*	*RF*	*DT*	*XGB*	*ET*	*PCA*	*Lasso*
Logistic Regression	4690	5000	5060	4550	5060	4110	4990
LDA	4740	5030	5060	4520	5060	4110	4990
QDA	4540	6920	**6930**	6720	**6930**	5060	4990
SVM	4610	6900	6810	6650	6810	4870	4940
Decision Tree	4270	4860	4890	4620	4900	4220	4850
Random Forest	4380	6390	6330	6020	6170	4720	5070
Neural Network	4620	6830	6880	6630	6820	4870	4990
XGBoost	4160	6630	6310	6000	6310	4270	4800
LightGBM	4270	6550	6430	6120	6430	4580	4910
Extra Trees	4230	6340	6180	5980	6320	4660	5090

More detailed results, in the form of boxplots for selected models, are available in Appendix A.

We can observe that certain combinations outperformed the rest. When it comes to the feature selection, model-based selections from tree classifiers outclassed other methods. On the other hand, when we take a look at the performance of classifiers, on average top three models were QDA, SVM and MLP. In fact, combination of these models coupled with tree-based selectors yielded the best results. Thus, for further investigation, we will perform hyperparameter tuning using these pairs exactly.

3 Hyperparameter tuning

3.1 Feature selection

Figure 1 presents an example of feature importance ranking determined by the Random Forest algorithm, which is the base for our further experiments.

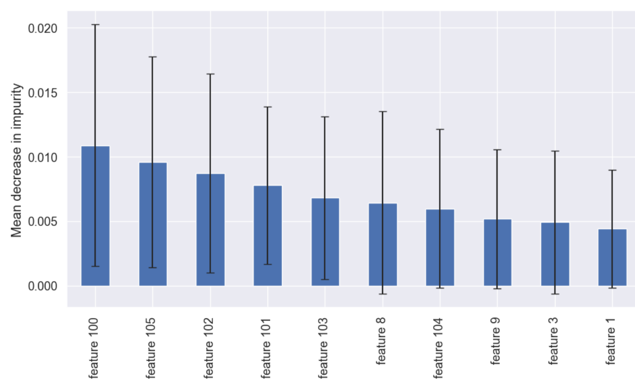

Figure 1: Top 10 feature importances using Random Forest.

Firstly, we explored whether varying the number of features for our top-performing models could enhance results. Experiments were conducted using k-fold cross-validation with feature counts ranging from 3 to 6. The plots visualizing our feature selection experiments are available in Appendix A.

The baseline value of k, set at 4, achieved the best balance between scoring function value and result stability. Thus, we selected 4 features for further experiments. While tree-based models produced similar results, there were slight variations in feature importance. We evaluated model performance using different combinations of the top features, specifically 100, 101, 102, 103, and 105.

To ensure robustness, we performed 5-fold cross-validation and repeated the experiments ten times. The feature set (101, 102, 103, 105) consistently provided the best performance across all tested models. Therefore, we will now focus on hyperparameter tuning for models trained with these features.

3.2 Grid-search results

In order to increase the quality of the selected models, we decided to use the grid search method, with 5-fold cross validation.

3.2.1 Parameters exploration

Below we present results for 6 example sets of parameters for SVM and MLP. We decided not to carry out a grid search for the QDA model because of the severely limited potential for parameter rotation.

Table 2: SVM Parameter Sets

Index	C	coef0	degree	gamma	kernel	shrinking	mean_score
0	0.0001	1.5	2	3	poly	False	7010
1	0.00015	1.5	3	1.25	poly	True	6950
2	0.0003	2.5	2	0.5	poly	True	6640
3	5e-05	1	2	0.5	poly	True	4390
4	5e-05	1	3	1	poly	False	6900
5	5e-05	2	2	1.25	poly	True	**7060**

Table 3: MLP Parameter Sets

Index	activation	alpha	hidden_layer_sizes	learning_rate	mean_score
0	logistic	0.001	(100, 100, 100)	adaptive	4790
1	logistic	0.1	(100,)	adaptive	5030
2	relu	0.0001	(100, 100, 100, 100)	adaptive	6780
3	relu	0.1	(100, 100, 100)	constant	**6950**
4	tanh	0.001	(100, 100, 100)	constant	6730
5	tanh	0.001	(100,)	adaptive	6410

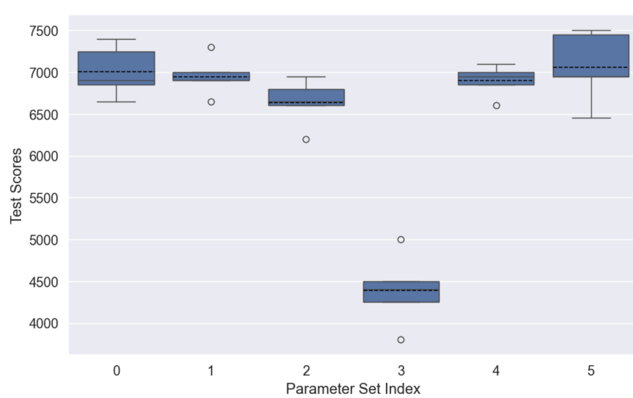

Figure 2: Results for example SVM parameters.

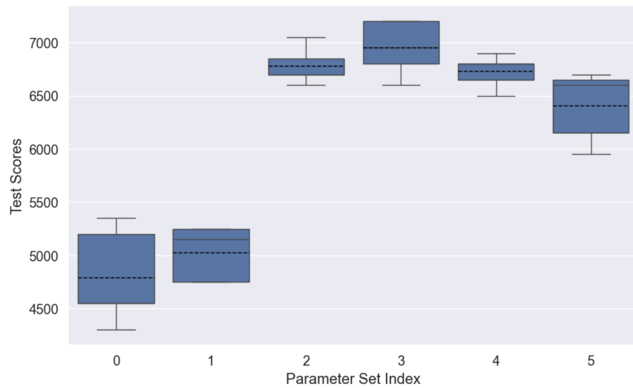

Figure 3: Results for example MLP parameters.

3.2.2 Best models

In the end, we have included a comparison of the models in the Figure 4, which shows the advantage of the finally obtained SVM model over the other two, which perform similarly.

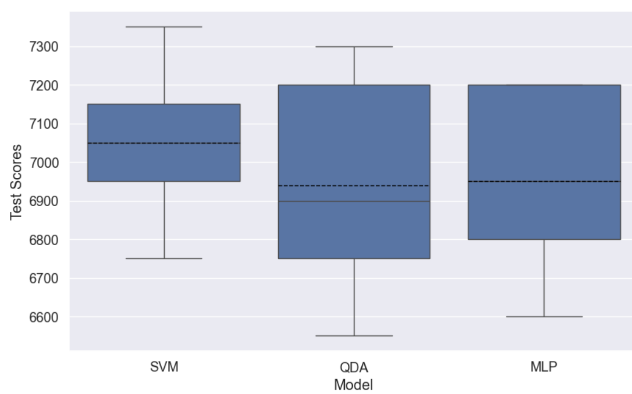

Figure 4: Comparison of the best models.

4 Conclusion

During the project, we conducted a comprehensive analysis of several feature selection techniques and machine learning models, successfully maximizing our gains while minimizing feature usage costs. Our results demonstrated that strategic feature selection and robust model evaluation are crucial for achieving cost-effective and accurate predictions, which is a key aspect of real-world machine learning problems.

A Appendix: plots

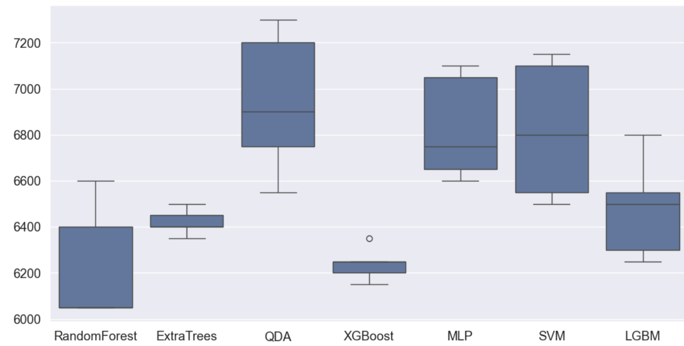

Figure 5: Comparison of cross validation results for different models.

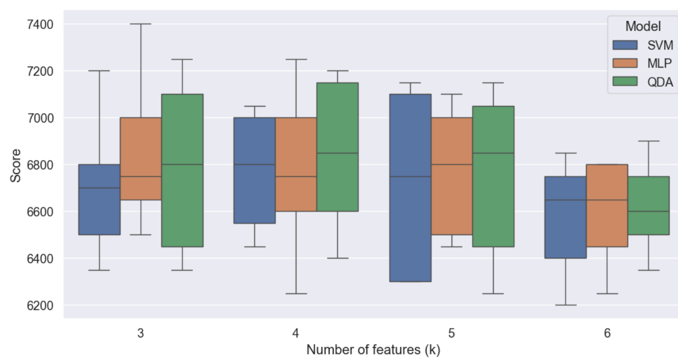

Figure 6: Models' performance across different feature counts.

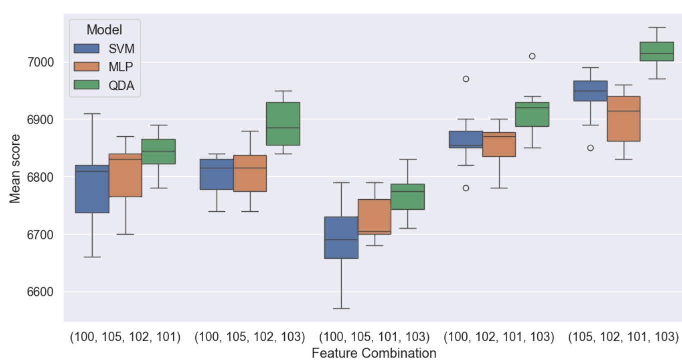

Figure 7: Models' performance across different feature combinations.