Warsaw University of Technology

FACULTY OF
MATHEMATICS AND INFORMATION SCIENCE

Advanced Machine Learning

Project 2

Michał Ciasnocha
Filip Kołodziejczyk
Jerzy Kraszewski

1 Introduction

This project is focused on the feature selection/importance. The goal is to propose the optimal model that combines high accuracy and utilizes few features for the given dataset. The optimization problem is expressed with the formula 1 , where the target is to maximize the net profit. The income for each correctly classified true sample equals 10, with each used feature cost of 200. There is an additional restriction. At most, 20 percent of the whole set of samples will provide the income. Hence, the model output should be 20 percent of the set's most probable true candidates.

$$\text{Net profit} = 10 \times \text{TP} - 200 \times \text{number of variables} \tag{1}$$

2 Data exploration

Note: In code, the index of features starts from 0, while this document is from 1.

The first stage of the process attempts to understand the nature of the dataset and its patterns. The problem is a binary classification problem, where the input consists of 500 numerical features. The target variable is either 1 or 0. The size of the training dataset is exactly 5000 samples.

The simple overview of features mean and standard deviation provides exciting insights. Despite the large number of features, only four patterns emerged:

1. Features 1-10 have mean ≈ 0 and std. dev. ranges from 1.6 to 2.17
2. Features 10-199 have mean ≈ 0 and std. dev. is ≈ 1
3. Features 200-399 have mean ≈ 0 and std. dev. is ≈ 0.29
4. Features 400-499 have mean ≈ 10 and std. dev. is ≈ 4.5

Those patterns lead to the suspicion that the data was generated artificially. Next, an outliers detection was performed. Their number is negligible for each feature (no more than 2.5%). However, correlation analysis showed an interesting observation: features from 1 to 10 are highly correlated, implying a high chance of using fewer features to represent this group. Following this lead, multicollinearity was checked. It comes out that features 1-10 and 101-110 have an infinite VIF score, i.e., they can be perfectly described by other variables using linear combinations. Usually, such features are treated as redundant, and they are removed. However, in this case, they offer combined knowledge from multiple variables, being perfect candidates to represent multiple variables at a low cost. No correlation was found between the target and any feature, indicating that the target must be a result of a non-linear function or . Based on that knowledge, the decision was made to utilize two models in this project: Random Forest and Gradient Boosting (XGBoost). While SVM is also a good candidate, it does not return probabilities and will not provide a ranking of best-matching samples. Neural networks also fit the problem, but was abandoned, due to risk of over-fitting (given the small dataset) and increased complexity (both implementation and time).

3 Initial model and feature selection

Since some feature selection methods require the classifier to work, one must be prepared first. A random forest was chosen for this stage. It's hyperparameters were tuned on the full dataset using grid search cross-validation. Compared with the default hyperparameters, the increase in precision was notable and presented in Table 1.

Model	Precision (%)	Top 20 % Precision (%)
Random Forest (no tuning, full data)	60.93 (±2.78)	67.10
Random Forest (tuned, full data)	65.04 (±2.04)	68.80
Random Forest (no tuning, top 12 feat.)	67.59 (±1.57)	73.60
Random Forest (tuned, top 12 feat.)	69.92 (±1.82)	74.80
XGBoost (no tuning, top 12 feat.)	65.55 (±1.79)	72.70
XGBoost (tuned, top 12 feat.)	71.26 (±1.04)	76.50
XGBoost (tuned, top 4 feat.)	69.32 (±1.82)	74.60

Table 1: Model performance (measured with cross-validation)

Pos	Random Forest	MIC	SHAP	XGBoost	Boruta	RFE	ReliefF	Ensemble
1	106	102	106	9	103	103	106	106
2	101	103	103	101	106	106	9	103
3	103	106	101	303	101	104	103	101
4	104	101	104	103	104	101	102	102
5	102	9	102	106	102	102	101	9
6	105	104	105	102	9	105	4	104
7	9	6	9	104	105	9	10	105
8	6	105	6	279	6	7	3	10
9	10	10	10	344	4	372	104	4
10	4	1	4	100	10	10	7	7
11	3	7	7	105	3	114	6	2
12	7	5	2	371	1	3	2	1
13	2	493	8	442	7	485	8	5
14	5	2	5	206	2	301	1	485
15	1	352	3	355	5	404	5	404
16	8	3	1	60	8	4	105	3
17	286	192	413	131	404	110	329	221
18	446	168	286	476	286	109	304	459
19	338	107	329	346	271	6	322	361
20	132	240	65	79	221	392	352	329

Table 2: Top 20 most important features per feature selection method

With the baseline model ready, feature selection was handled using some of the most popular methods. Those are: **Random forest feature importance**, **Maximal Information Coefficient**, **SHAP**, **XBoost feature importance**, **Boruta**, **RFE**, **ReliefF**, and, **Ensemble**.

After the importance rankings generation, the top 20 features were chosen from each method for further analysis and selection of the best candidates for exhaustive search. The ranking is given in Table 2. Clear dominance of features 101-106 is visible, followed by the features 1-10. In order not to risk losing less important yet relevant information, the decision was made to keep all the features that appeared in the ranking more than twice: 1-10, 101-106, 286, 329, 352, and 404 (there are a few more but their second occurrence was in the ensemble).

The features above were explored once again. Their correlation was visualized in Fig. 1. In this correlation matrix, it is evident that all features 1-10 are highly correlated. Multicollinearity was reevaluated, too. Both features 1-10 and 101-106 achieved infinite VIF factor. Those groups measurement was repeated separately. Then, features 101-106 gave VIF 1, meaning describing them linearly with others within the group is impossible. At the same time, group 1-10 still hold high VIF values, but not infinity. It hints that features 101-106 are linear combinations of features 1-10 and features 1-10 are highly dependent on each other. Thus, only one feature from group

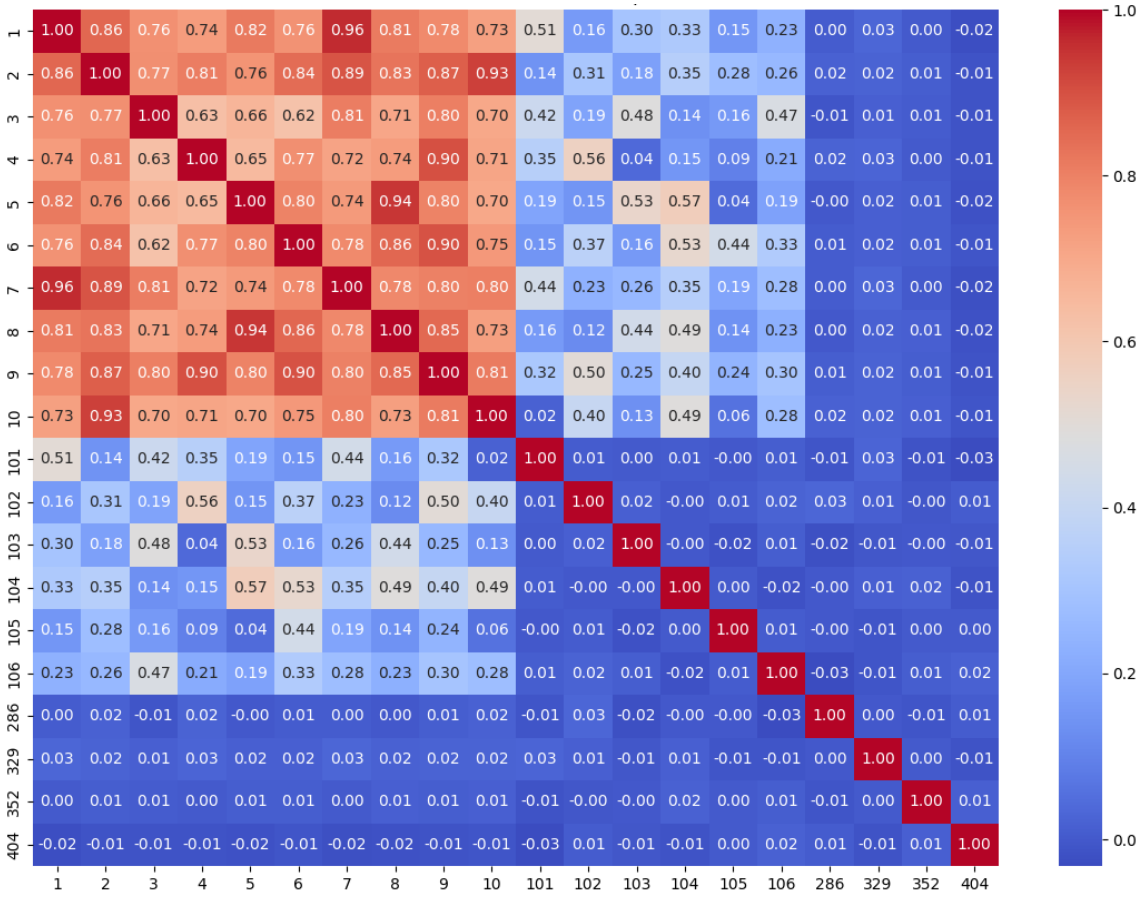

Figure 1: Most important features correlation matrix

1-10 was kept, hoping that one feature could describe this group well enough. The choice was feature 9, which presented the highest correlation with other features from its group. Those four extra features, 286, 329, 352, and 404, persisted for now, too. Therefore, selected 11 features are 9, 101-106, 286, 329, 352, and 404.

4 Final model and feature selection

At the final stage, the best set of features was selected via maximization of the Formula 1. Net income was calculated for each combination by the cross-validation process, where for each fold, the proposed metric was measured, and then their mean value was returned.

However, for that process, the classifier itself was required. The previous hyperparameter set was expected to be non-matching for the new set of features due to the tremendous difference between the number of features. Thus, tuning was repeated. After the hyperparameter tuning, the random forest achieved a precision of 69.92%. As mentioned in Sec. 2, the XGBoost model was tuned too. While its performance without hyperparameter tuning was lower than random forest, after proper tuning, it achieved the best result so far, 71.26%, with a minor variance too. The performance of all the models, along with the top 20% prediction precision, is shown in Table 1. Based on those results, the XGBoost model replaced random forest in the following experiments.

Before the exhausting search, selected feature importance methods were reevaluated for the pre-selected set of features to analyze the impact on their results after the pre-selection of the best

Pos	XGBoost	ReliefF	RFE	SHAP	MIC	Ensemble
1	106	106	103	103	9	103
2	101	103	101	106	101	101
3	103	101	106	101	102	106
4	9	9	104	104	103	9
5	104	102	102	102	104	102
6	102	104	105	105	105	104
7	105	105	9	9	106	105
8	404	286	286	329	329	329
9	329	352	404	404	404	404
10	286	329	329	286	352	286
11	352	404	352	352	286	352

Table 3: Ranking of pre-selected features per feature selection method

Ranking	Subset	Net Income
1	102, 103, 104, 106	6930.0
2	9, 101, 103, 104	6840.0
3	103, 104, 106	6820.0
4	101, 103, 104	6820.0
5	102, 103, 104	6810.0
⋮	⋮	⋮
2043	286, 352	4700.0
2044	286, 352, 404	4690.0
2045	286, 329, 352	4620.0
2046	286, 329, 352, 404	4610.0
2047	286	4600.0

Table 4: Ranking of pre-selected features per feature selection method

candidates. Results, presented in Table 3, were not affected much, with features 286, 329, 352, and 404 being the least important. The most significant change shown in XGBoost was caused either by removing nonmeaningful features, hyperparameter tuning, or both (previous XGBoost feature importance was measured on the model without tuning).

Finally, a brute force search was carried out. Table 4 presents five best and worst combinations of features. While this ranking holds the general conclusions of feature selection methods, none of the methods indicated the best subset (4-element) as the most critical four features. It suggests that whenever possible, it is a good idea to do the feature screening using efficient methods and then, on the most promising candidates, utilize an exhaustive search. Unfortunately, it is rarely possible due to a significant increase in computation requirements. This dataset was favorable for the scenario, thanks to only a tiny subset of meaningful features. In real scenarios, the dependencies are often much more intricate, and it is not easy to entirely ignore given features.

With the best combination of features 102, 103, 104, and 106, the XGBoost model was tuned once again. Final precision compared to a model with 12 features did not degrade much, as presented in Table 1. This explains good results in net income. A considerable reduction of features used did not affect precision much.