Advanced Machine Learning – predictive model for marketing purposes

Project report

Dutt Salveen 317298
Prusak Patryk 305794
Zagórski Mateusz 313509

Instructor: mgr Anna Kozak

June 3, 2024

Contents

1 Dataset and assignment overview

The objective of Project No. 2 is to construct a model that achieves a high score while being economical in the utilization of variables from the dataset provided within the assignment materials. The task itself describes a business case study that requires determining the target group of a marketing campaign among a group of customers.

A dataset comprising 5000 records and 500 features detailing the characteristics of clients is employed to train an optimal model. The training set includes an array of labels that is a binary variable, where the positive class indicates that the corresponding customer took advantage of the company's offer. A binary classification will be conducted to predict as many potential customers likely to take advantage of the offer as possible. The aim is to use the minimal number of features necessary. A specific scoring algorithm has been provided as the efficiency indicator. There is a reward of \$10 for each correctly classified positive sample, with a limit of 1000 customers indicated by the classifier. On the other hand, there is a cost of \$200 for each variable used to train the classifier.

2 Methodology

To accomplish the project task, it was decided to evaluate various feature selection methods, scalers, a feature generator, and multiple popular models used for binary classification. Each method was thoroughly examined in the context of the project task. Model performance was evaluated using a scoring algorithm but additional metrics such as accuracy, precision, and number of true positives were also monitored.

To assess the models more accurately, we defined a similar scoring measure as per the task description. Instead of directly predicting classes from created models, probabilities of data points belonging to a positive class were collected. Next, a certain number of data points with the highest probabilities were determined as predicted positive class data points, whereas the rest were defined as negative predictions. Next, the amount of correct predictions is calculated and multiplied by 10. Afterward number of features used multiplied by 200 is subtracted, and finally, the result is divided by a score defined as ideal, that is 100% of correct predictions multiplied by 10 minus 2 features used times 200. This relation can be defined as:

$$score = \frac{10c - 200f}{10n - 400}$$

where c is the number of correct predictions, f is the number of features used, and n is the limiting number of predictions.

The X_{train} and y_{train} were divided into a train-test split (with 33% dedicated to the test split) that remained the same for all tested configurations to ensure the reliability of the results. Initially, the performance was evaluated without considering the task limitation regarding the maximum output size. However, we have then noticed the limitations of such an approach and decided to utilize the aforementioned scoring function limiting the number of correct true positives as 45% of all available positives contained in the split y_{test}.

For each method, a set of relevant parameters was determined. With this configuration prepared, experiments were conducted for each method such that while one parameter was tested across its entire range, the remaining variables (parameters, feature selector/model) were held constant. Initially, the range of parameters tested was

broad and sparse to ensure that the best result would not be overlooked. Subsequently, the best-performing methods were chosen for further investigation. The range of their parameters was narrowed around the most promising values, and the experiment was repeated.

The experiments have been performed in batches. Each batch contains a number of feature selectors, classifiers, data scalers (or none), and feature generators (or none). For each, different parameters are selected and each combination is measured in terms of score, accuracy, number of True Positives, and precision. In general, the aim of the first 5 batches was to compare the performance of different feature selectors: **SelectKBest**, **RFE**, **RFECV**, **FPR**, **FDR**, **FWE** and **PCA** with different parameters on classifiers with a default configuration: **Linear Discriminant Analysis**, **Quadratic Discriminant Analysis**, **K Neighbors Classifier**, **SVC**, **Gradient Boosting**, **Histogram-based Gradient Boosting**, and **MLP Classifier**. Next, the best feature selector candidates were chosen and we began to test different parameters for the classifiers, including **XGBoost**, **Voting**, **AdaBoost**, and all the previously mentioned classifiers. The two best candidates appeared to be Gradient Boosting and MLP Classifier, therefore the next couple of batches were dedicated to finding the best parameters for each of them. At this stage, we also included data scalers such as the **Standard Scaler** and **Robust Scaler**. We also tried to generate some additional features from the already selected features by using a **Polynomial Feature Generator**. Finally, we have chosen the MLP classifier with Robust data scaler, and SelectKBest feature selector as our final model and performed a few more batches to find the most optimal parameters. All the mentioned models apart from XGBoost (xgboost package) were provided by the SciKit library.

Some possible shortcomings of the above methodology are that the size of the split X_{test} dataset is much smaller than the X_{test} provided with the task description as a result the amount of positive class predictions is lowered and the penalty for feature usage becomes more significant. Perhaps, the increased precision of a model trained on a larger amount of features would outweigh the feature penalty on a larger dataset. Additionally, not enough attention has been given to the Voting classifier, with proper configuration it has the potential to outrank other classifiers used.

3 Results

Based on the research conducted during the project, the following conclusions can be drawn:

- Among the models tested, Gradient Boosting and MLP classifiers emerged as the top contenders.
- Both FDR and FWE feature selection methods did not yield any reasonable results.
- The Robust data scaler demonstrated a slight improvement in the performance score for the MLP Classifier.
- The features generated using the Polynomial Feature Generator did not provide substantial benefits.
- In total, 101,261 experiments were performed.

Figure 1 presents the results of all experiments for each of the tested models. Some outliers were removed for the clarity of the diagram. Overall the results are quite similar for all tested models, however, the diagram clearly highlights that the best classifiers for the given task are Gradient Boosting and MLP Classifier, with **MLP Classifier** achieving higher results.

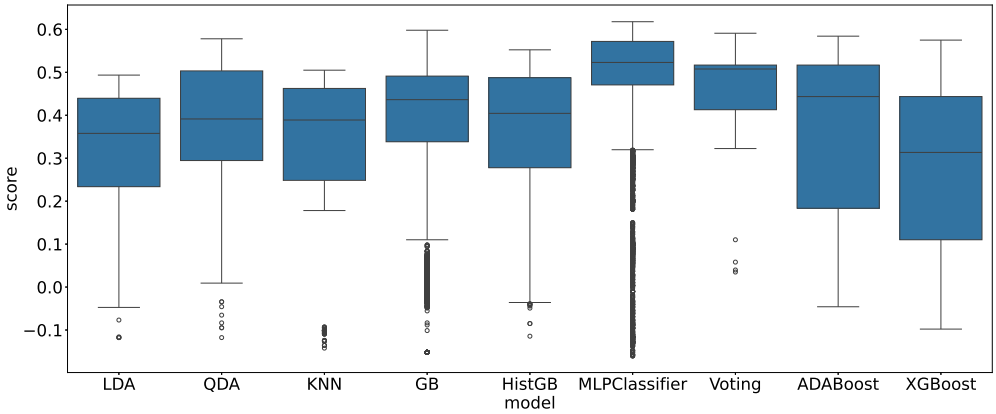

Figure 1: Task results for different classifiers (outliers removed). The score is represented as a percentage of the maximum score.

Figure 2 presents the results of all experiments for each of the tested feature selectors. Some outliers were removed for clarity, as was done in Figure 1. In this case, however, the **FWE** method failed to produce any results and the method of no feature selection was omitted to present the results more clearly. Keeping in mind the definition of the scoring function and the fact that the X variables originally contained 500 features one can imagine that the scores for no feature selection were extremely low. The diagram illustrates the performance of the SelectKBest, RFE, FPR, FDR, RFECV, and PCA methods in feature selection for the project assignment. FDR achieved unsatisfactory results, whereas SelectKBest, especially with 2 and 3 features, performed notably better than other methods.

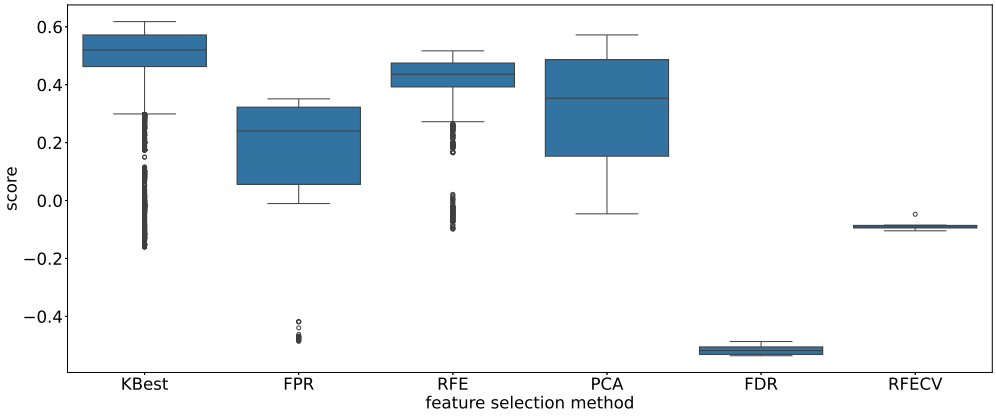

Figure 2: Task results for different feature selection methods (outliers removed). The score is represented as a percentage of the maximum score.

Through extensive experimentation that has been conducted, the **MLP Classifier** combined with the **Robust data scaler** with default parameters and **SelectKBest feature selector** with 2 features and *mutual_info_classif* as the score function was identified as the optimal solution for the project needs achieving 61% of the perfect solution. Tables 1 and 2 present, respectively, a detailed description of the methods that achieved the final result and the parameters of the classifier used to achieve it.

Table 1: Characteristics of the final solution.

Parameter	**Value**
Score	0.617737
Number of true positives	242
Accuracy	0.56303
Precision	0.572519
Number of features	2
Model	MLPClassifier
Feature selector	KBest
Scaler	Robust

Table 2: MLP Classifier parameters for the final solution.

Parameter	**Value**
activation	relu
solver	adam
beta_1	0.95
beta_2	0.999
alpha	0.3
learning_rate	adaptive
learning_rate_init	0.008
hidden_layer_sizes	(13,)
max_iter	1600
random_state	42