Building a highly accurate, parsimonious model
Advanced Machine Learning - Project II

Jakub Piwko, 313451
Kacper Skonieczka, 313505
Grzegorz Zakrzewski, 313555

03.06.2024

1 Introduction

The objective of this project is to construct a highly accurate model using a limited number of variables from dataset that imitates problem of classifying bank clients that will subscribe to an offer. To achieve this, we've broken down the task into research pipeline. Firstly, we explored the data during exploratory analysis. Then we sorted the available variables based on their importance and relevance, selecting a small subset of features. This process involved employing feature selection methods such as information theory-based criteria and decision tree-based feature importance. Secondly, we aimed to develop the most effective classifier using the selected subset of features. This process consisted of many steps that used large search space considering different classifying algorithms, different sizes of features set, hyperparameter tuning and ensembles. The primary evaluation technique during this stage was the simulation of prediction profit, in accordance with the project instructions. In the upcoming paragraphs, we will introduce the applied techniques and describe the results.

2 Experiments

For this project we decided to experiment with many approaches available in classic machine learning field in step by step manner. In this section and following subsections we would like to present actions taken by us to find the best model.

2.1 Exploratory Data Analysis

Firstly we decided to investigate the data, mainly correlations between features and target variable, to gain some intuition about sets and facilitate selection of methods in further steps. In this report, all of the figures that we refer to are included in Appendix. The barplot presenting absolute value of correlation coefficient for all features is displayed on Figure 1. The correlation between all the features is presented on matrix on Figure 2. Unfortunately, we cannot see any outstanding relations in the data, especially if it comes to correlation with target variable. None of features has absolute value of correlation coefficient larger than 0.043, what indicates no relations. For matrix, we can see that beside some small areas, columns are also not correlated.

On the matrix on Figure 3 we can see correlation for features from 0 to 9 and 100 do 109. It is interesting that in these subsets of columns we can see significantly bigger positive correlation. We can assume that this features have some hidden importance.

We also investigated histograms of features to analyse possible distributions. On Figure 4 we can see such histogram for feature with index 109. From its shape we can assume that variable has Gaussian distribution. Same shape is present in other features. Also, the dependencies from train set are true also for test set. From this premises, we are assuming that dataset was artificially generated, with column randomly drawn from normal distribution. Therefore, it could be hard to find natural relation between predictive variables and target, taking into account low correlations.

2.2 Feature selection

After getting more intuition about the data, and incorporating the fact, that we cannot rely on reducing number of features based on correlation coefficient, we decided to implement and utilize other methods that will help us select the best features. The three lines of actions were set for this data set, namely Joint Mutual Information and Conditional Mutual Information Maximization, which we already described in the Section 4.1 and also feature importance gained from training Random Forest models. While the first two methods were building feature portfolio selecting one by one from the set based on the information criteria in each step, the Random Forest approach consisted of training model on 4-fold cross validation with grid hyperparameter search and obtaining feature importance index from the best estimator.

After gaining the ranking of features for each technique, we decided to combine them in one table that shows rank of feature for every method. Results truncated to only 15 columns are presented on Table 1

We can see that for information criteria methods, the ranking of features is very similar. With more permuted order, Random Forest was also focusing on the same features. We can see that the highest positions were achieved by the subsets of features with indexes from 0 to 9 and even higher places for indexes from 100 to 105. Such results are promising, giving the fact that these features were standing out when comparing correlation.

We have also checked the change of selection criteria when adding more features in each step. In case of RF technique, we showed importance for the following best columns. The results can be seen on the Figure 5. From

Table 1: Feature rank for each feature selection method

Feature	105	101	100	102	8	104	103	5	3	4	2	6	9	7	0
JMI	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
CMIM	1	2	3	4	5	6	7	11	10	8	12	9	13	23	35
RF	1	7	2	3	6	13	4	8	11	14	12	9	5	15	16

this plot we can see that strong elbow is visible close to 20th feature for every method, after which the progress is hardly visible. This indicates that for the problem where number of columns used while modelling is so important, we should consider smaller numbers.

In the next step, we wanted to see what is the progress of some evaluation metrics with growing subsets of features used during training, but already limiting its count to reasonable numbers. We checked metrics progress when training random forest on 4-fold cross validation with grid hyperparameters search with growing subset of features. Features were added one by one based on ranking for every method. On the Figure 6 we can see accuracy change for different size of set, and on the Figure 7 we placed effectiveness - custom metric that takes accuracy and number of features as a penalty into account. One can see that accuracy has reached peak values where set had around 7 features for two information criteria, but around 17 for Random Forest feature importance. On the other hand, when looking at effectiveness we can see that sets containing more than 4 features are already deteriorating the score, what is alarming.

Based on this experiments we decided to test some different classifying algorithms on subsets of features that had the highest ranking positions in Table 1.

2.3 Training classifiers

Searching of best classifier started with creating a pipeline that could explore as many combinations of classifiers, subsets of features and hyperaparameters settings as possible. That is why, we developed simple algorithm that we describe in points below:

1. Split the train set into train and validation set in the proportion of 80% of samples going to train set and 20% to validation with fixed value of random seed to ensure that every model in next step was trained and evaluated on the same data.

2. Randomly select umber of features to train model on between 2 and 7.

3. Select features from subset of 12 features selected in the feature selection step. The indexes of features that could be drawn: 105, 101, 100, 102, 8, 104, 103, 5, 3, 4, 2, 6.

4. Train 4 different classifiers using 4-fold cross validation and grid hyperparameter search when possible. The classyfing algorithms were selected randomly from subsets containing 11 different models from scikit-learn package, namely: Random Forest, Multi-layer Perceptron, SVM, Gaussian Process, QDA, XGboost, Naive Bayes, K-nearest Neighbors, AdaBoost, Logistic Regression, Extremely Randomized Trees.

5. Evaluate the best estimator on validation set from grid search using accuracy, precision, recall and profit - custom metric that corresponds to metric that will be evaluation score for this project, including precision score, number of expected positive samples as 1000 and penalty from number of features. Its formula can be written as follows: $\text{Profit} = 10 \cdot 1000 \cdot Precision - 200 \cdot |FeatureSet|$

6. Save information about used model, used features and metric scores to create model library.

We repeated such process multiple times and obtained 400 different classifiers, which was the number that allows valuable analysis of their performance.

Firstly, we looked at general performance of first 9 best models when comparing generated profit. Their details are presented in Table 2

We can see that best models generated profit above 6000, with the best one being Random Forest with two features achieving score above 6400. As we can see, the majority of models have only features, which means that for 1000 positive observations in test set, it is hard to obtain larger profit with models using more features.

We decided to investigate what models, features and what feature sets sizes are the most common among best trained models. On the Figure 8 we checked which classifiers were most frequently present in top 100 in model library. We can see that QDA and SVM were those that generated the highest profit most often. On the plot 9 we summed occurrences of features in 100 best feature sets. From this plot it emerges that features 100 and 105

Table 2: List of 9 best classifiers by profit from the first step of experimenting with modelling

Model type	rf	svm	nb	rf	nb	qda	xgb	qda	xgb
Features	[105, 104]	[100, 2]	[105, 102, 100]	[100, 8]	[8, 101, 100, 105]	[102, 6, 100]	[100, 8]	[104, 100]	[104, 105]
Accuracy	0.618	0.603	0.650	0.605	0.658	0.638	0.606	0.617	0.609
Precision	0.682	0.676	0.689	0.668	0.706	0.684	0.664	0.658	0.656
Recall	0.465	0.419	0.567	0.443	0.559	0.533	0.455	0.512	0.484
Profit	6420.8	6361.9	6289.9	6276.5	6264.6	6243.4	6237.9	6182.2	6160.0

are particularly frequent. It can be said about the whole family of features from 100 to 105, but also about feature with index 2 and 8. Bar plot on Figure 10 indicates that it is most profitable to reduce number of features to only 2, as larger sets have larger penalty.

We decided to once again narrow down the experiment search space and repeat the process of training models.

2.4 Training classifiers - second iteration

With conclusions from first iterations of modelling experiments, we decided to create new model library, but with more strict assumptions about best possible classifier. In this step we limited feature sets to have maximum of 3 features. Based on performance of columns, we decided that this time only features with indexes 100, 101, 102, 103, 104, 105, 2 and 9 will be available to draw. Newly selected feature set will be always used to train 4 models: Random Forest, Naive Bayes, QDA and SVM. The training procedure stays the same - we are using 4-fold cross validation and grid hyperparameters search. In this step we obtained 200 different classifiers.

In addition, we wanted to ensure that every pair of columns from 100 to 105 will be trained by every classifier from 4 used in this step. That is why we extended model library with models trained specifically on every possible pair from this columns. This way we obtained 260 new models and we present best 10 based on profit in Table 3.

Table 3: List of 10 best classifiers by profit from the second step of experimenting with modelling

Model type	nb	svm	rf	rf	svm	qda	rf	qda	rf	rf
Features	[102, 100]	[105, 104]	[100, 104, 103]	[100, 103]	[100, 103]	[104, 102, 101]	[100, 102]	[104, 105]	[103, 101]	[102, 104]
Accuracy	0.610	0.609	0.621	0.637	0.620	0.652	0.617	0.581	0.627	0.584
Precision	0.710	0.704	0.716	0.693	0.689	0.709	0.685	0.685	0.684	0.679
Recall	0.480	0.459	0.478	0.459	0.452	0.546	0.480	0.427	0.479	0.409
Profit	6700.2	6640.2	6562.9	6534.9	6497.5	6495.9	6459.5	6452.9	6446.5	6394.8

We can see that we were able to find models that significantly boost profit and the best model from second iteration achieved score over 6700. We can also see that best models get precision score above 0.7 and it is crucial for final performance of the model, because this metric is directly linked to profit. Models with larger feature set size could achieve larger precision, but gain does not compensate penalty from number of columns used. On the other hand, we can see that recall is small, but we need to sacrifice its level if we want to maximise the profit.

With such huge library of models we also wondered if usage of ensembles could improve the precision score and therefore boost the profit.

2.5 Ensembles

Having large library of divergent models, we decided to experiment with creating ensembles from this classifiers. Ensembles are known for their robust and accurate performance because they combine predictive patterns from many models.

In our case, we decided to create committees with stacking technique. We firstly obtain meta-sets contain columns created from predictions and probability predictions from each component model. Then based on such data set, new meta-learner is fitted to target variable. In our case, we used XGBoost model as the one to aggregate the outputs of component models. We decided to build ensembles from library of models from second iteration of training. We were randomly selecting the size of ensemble from 2 to 7 models, then creating meta sets and train meta-learner. We also decided that it will be more efficient to focus on best models in process of selection components to committee. That is why we decided to select only 50 best models from library based on profit, and also give them exponentially decaying weights. We trained only 20 ensembles and 5 best based on profit with their size, all features used and performance are presented in Table 4.

Table 4: List of 5 best ensembles by profit with their size and all feature used

Size	3	5	4	2	3
Features	[100, 102, 103, 104]	[100, 101, 102, 103, 104, 105]	[100, 102, 103, 104, 105]	[100, 102, 103, 104]	[100, 101, 102, 103]
Accuracy	0.650	0.724	0.694	0.652	0.659
Precision	0.684	0.723	0.702	0.682	0.681
Recall	0.576	0.738	0.688	0.588	0.618
Profit	6045.7	6039.3	6028.1	6026.4	6011.2

Although we can observe a huge boost of performance if it comes to recall, the rest of evaluation metrics did not improve significantly. With the fact that ensemble is using models that were trained based on different feature subsets, all of used features need to be considered during profit calculation. That is why scores of this metric are so small in comparison to single models. It means that we need to resign from using ensemble technique and rather focus on models from second training iteration.

2.6 Evaluation

Final step of our model selection is evaluation of best models from library. It is important to mention that in previous steps all classifiers where trained on one split of original train data set into train and test. This means that obtained results need verification to ensure models robustness what is very crucial when the final test set has the same number of observations as whole train sample.

This is why for 10 best models from model library obtained in second training iterations we were training them for 50 different splits with partitioning in proportion 80% to 20% and then we shown results on boxplots. On Figure 11 we can see distribution of accuracy and on Figure 12 distribution of profit. One can observe that accuracy of models is not a good indicator of which model should be selected as the best, because accuracy is naturally higher for models with 3 features, while profit is larger for smaller subset of features. Looking at the boxplot we could assume that possibly the best model is Random Forest trained on columns 100 and 102, which was 7th on initial ranking.

But apart from that we also wanted to evaluate models in environment that would imitate final test set that is unlabelled and of the same size as train test. That is why we repeated the process of training models on multiple splits, but this time proportion of split was 50% to 50%. Also, we decided to evaluate only 500 observations that had the highest probability of belonging to class 1. This is because in final test set we are searching for 20% of observations that we want to classify as positives. We want to check if in such conditions, results will change.

This time we included boxplots of count of correct positive classifications on Figure 14 and profit on Figure 12. We can observe that the highest ratio of positive observations among 500 samples that model scored the highest probability was achieved by QDA model. Unfortunately, this model was trained on 3 features, what is reflected in its worsened profit. In case of the latter metric, neither QDA nor Random Forest selected before scored the highest value. It turns out that in more adjusted scenario, the first model - Naive Bayes yielded the best profits on multiple splits.

3 Final model selection and conclusions

After complex and exhaustive process of selecting features and classifiers we decided to select Naive Bayes Classifier trained on features 100 and 102. It will be trained on whole train set and applied to predict probabilities on test set. Then 1000 records with highest probability of belonging to class 1 will be chosen to assess. This model spawned the most promising performance during experiment steps. Its evaluation in environment applying project requirements proved that it is the best model that we developed during the whole venture. In general we are satisfied with results of our experiments, even though more valuable afterthoughts could be drawn after obtaining score of our results.

References

[1] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13(1):27–66, 2012.

4 Appendix

4.1 Feature selection - Information theory based criteria

Information theory measures can be incorporated as a classifier-independent feature selection technique. Using information theory measures for feature selection involves quantifying the amount of information each feature provides about the target variable. Features that provide more information are considered more relevant and are selected for use in predictive modelling or analysis, while irrelevant features are discarded.

Information theory is grounded in several theoretical concepts, *entropy*, *conditional entropy*, *Mutual Information* and *Conditional Mutual Information.* The mathematical formulas for these four measures are presented as follows:

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{1}$$

$$H(X|Y) = -\sum_{x \in Y} p(y) - \sum_{x \in X} p(x|y) \log p(x|y) \tag{2}$$

$$I(X;Y) = H(X) - H(X|Y) \tag{3}$$

$$I(X;Y|Z) = H(X|Z) - H(X|YZ) \tag{4}$$

To briefly recap the core idea behind these expressions: entropy measures uncertainty. When all events x in X are equally likely, entropy is high because the outcome is uncertain. Conditional entropy indicates how much uncertainty remains in X when we know Y. Mutual Information represents the difference between these two values, revealing how much information Y provides about X.

A mature yet comprehensive comparison of information theory-based feature selection criteria was conducted by Brown et al. [1]. In the final section of their article, they presented a study encompassing several different criteria. These criteria were evaluated based on three conditions. Firstly, a criterion needed to balance the terms of relevance and redundancy to avoid ignoring features that are relevant but redundant during the selection process. Secondly, it had to include a reference to a conditional redundancy term to address the issue of correlated features that could be useful and should not be omitted. The third condition focused on whether the criterion was estimable with small samples. According to their study, three criteria satisfied these properties. Among them were Joint Mutual Information (JMI) and Conditional Mutual Information Maximization (CMIM).

The formulas for JMI and CMIM are presented as follows:

$$J_{jmi}\left(X_k\right) = \sum_{j \in S} I\left(X_k, X_j; Y\right) = I\left(X_k; Y\right) - \frac{1}{|S|} \sum_{j \in S} \left[I\left(X_k; X_j\right) - I\left(X_k; X_j | Y\right)\right] \tag{5}$$

$$J_{cmim}\left(X_k\right) = \min_{j \in S} \left[I\left(X_k; Y | X_j\right)\right] = I\left(X_k; Y\right) - \max_{j \in S} \left[I\left(X_k; X_j\right) - I\left(X_k; X_j | Y\right)\right] \tag{6}$$

Those criteria were implemented and used to sort variables in the project's dataset according to their relevance.

4.2 Figures

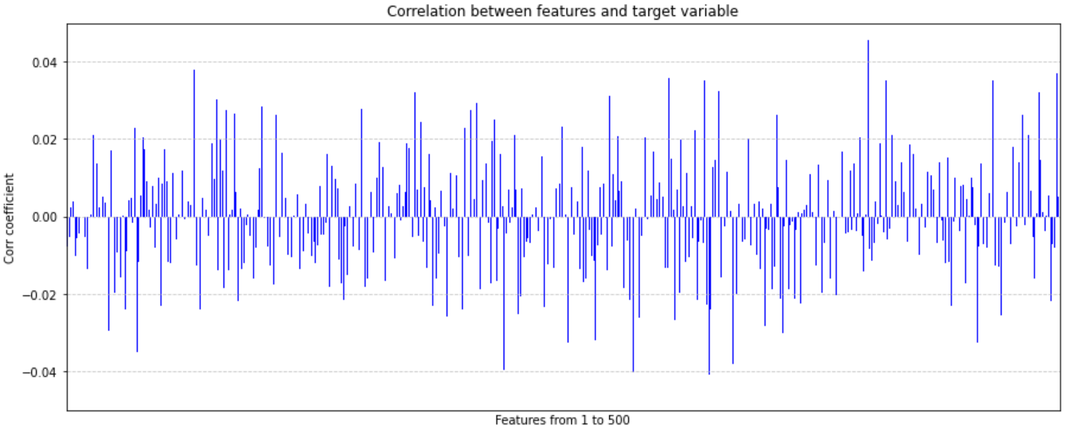

Figure 1: Correlation coefficient between all features and target variable.

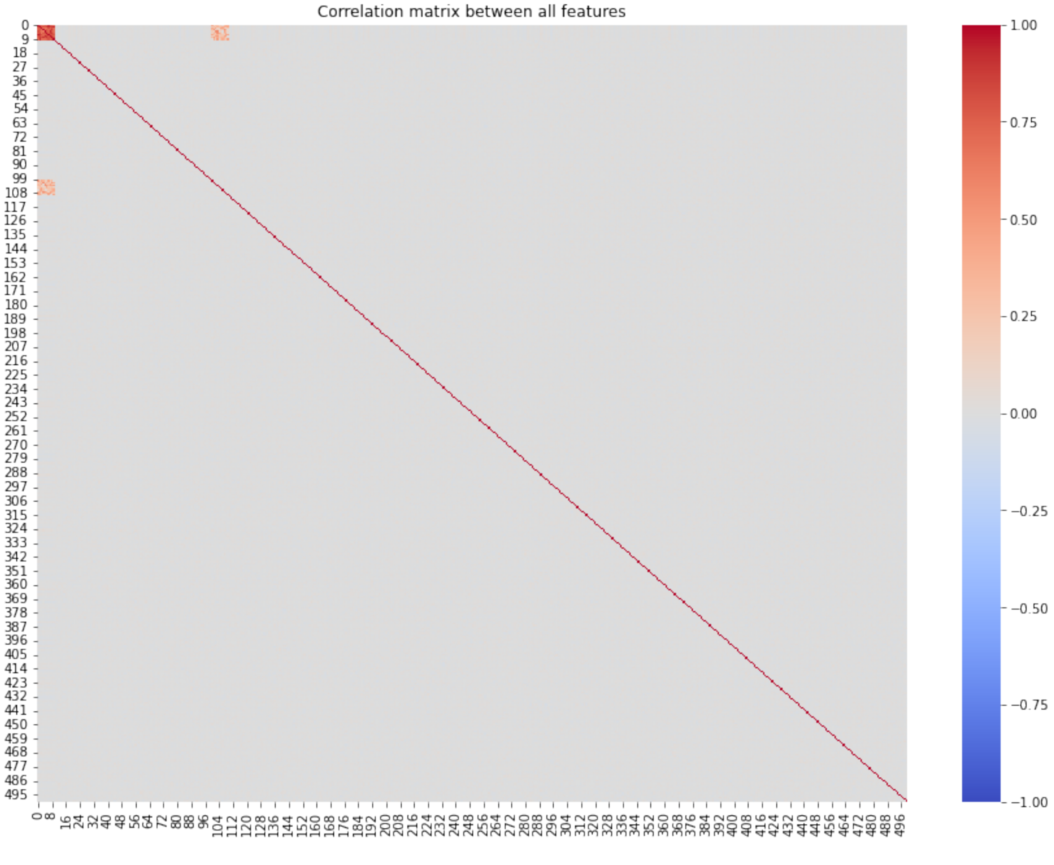

Figure 2: Correlation matrix for all features in train data set

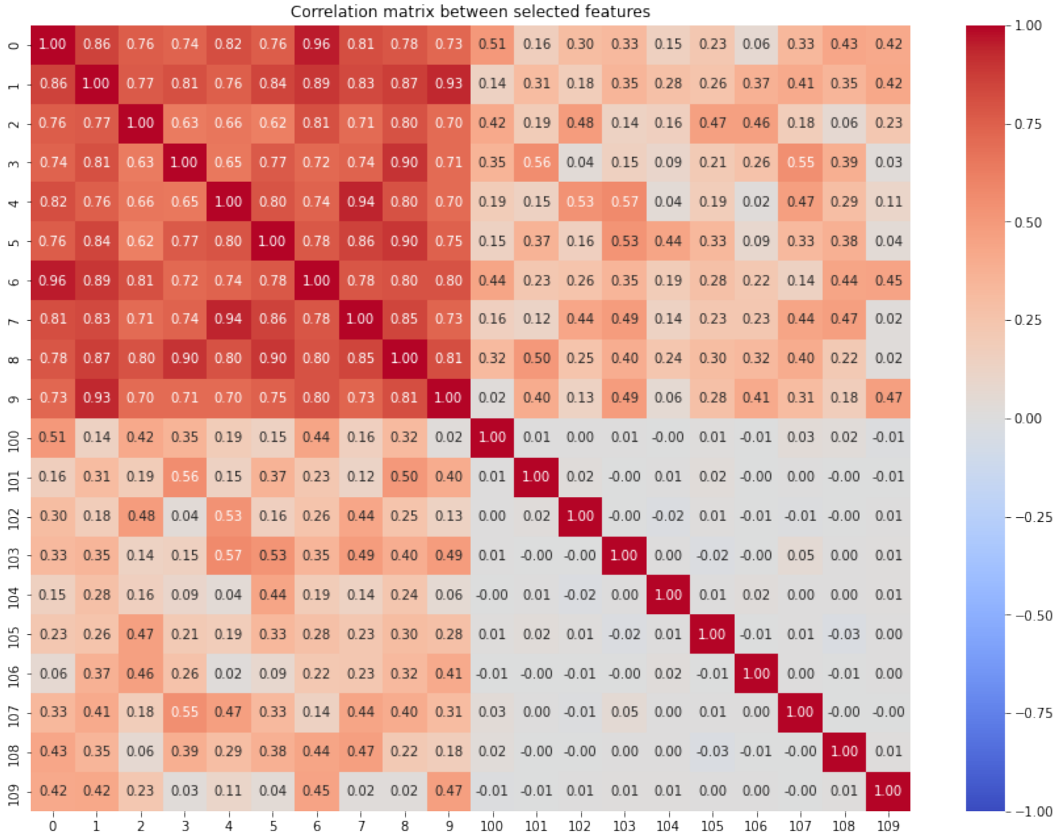

Figure 3: Correlation matrix for selected subset of features in train set

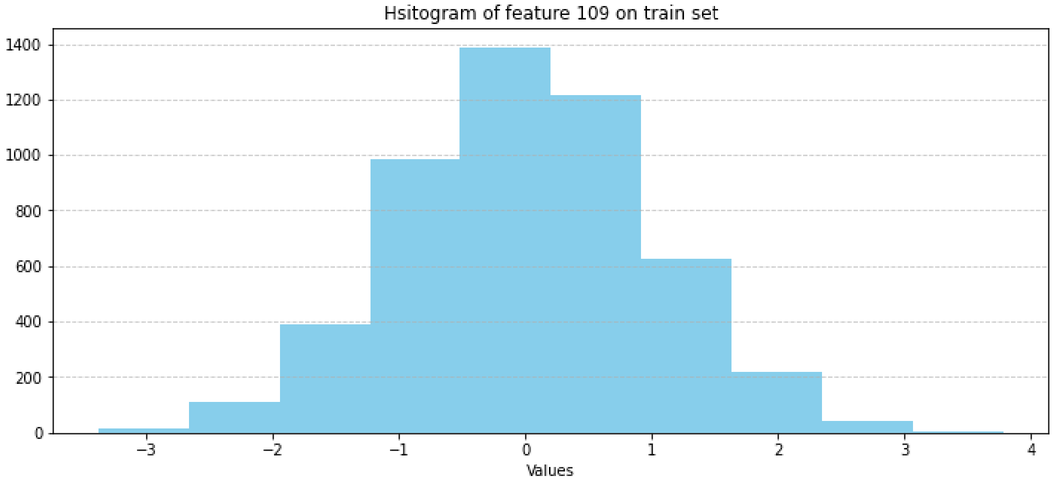

Figure 4: Histogram for feature 109 in train set

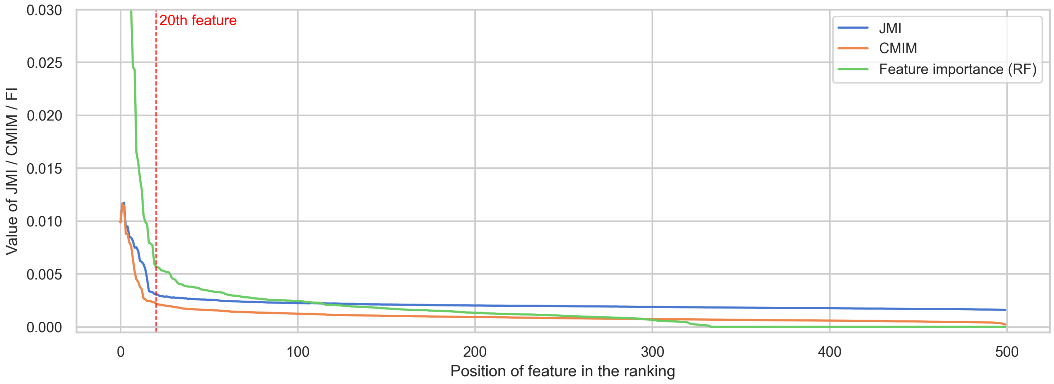

Figure 5: Line chart presenting improvement of selection criteria or feature importance for addition of following best features for each FS method.

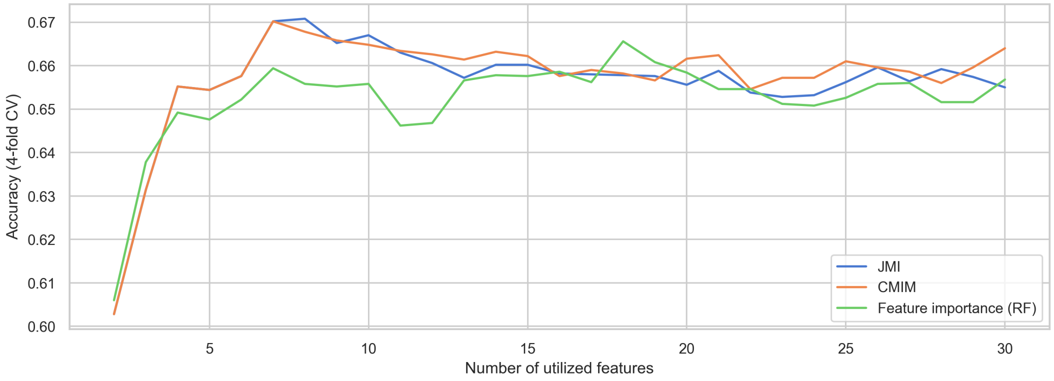

Figure 6: Line chart presenting accuracy metric for models with expanding subset of features used during training based on each feature selection method

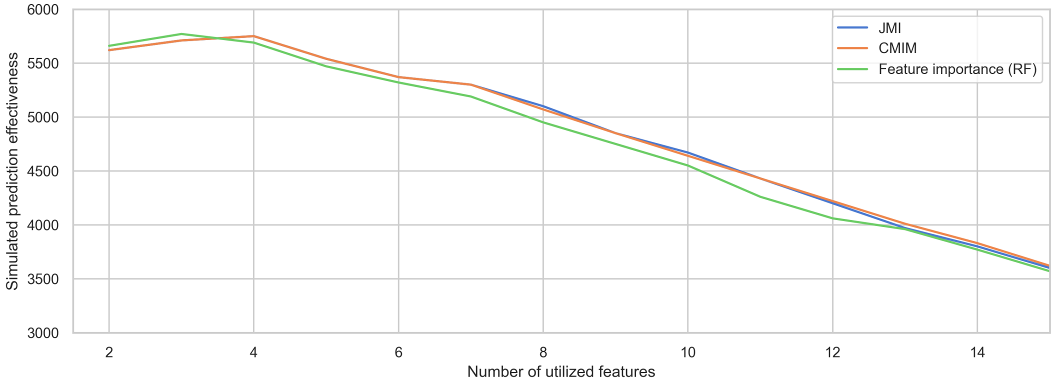

Figure 7: Line chart presenting effectiveness metric for models with expanding subset of features used during training based on each feature selection method

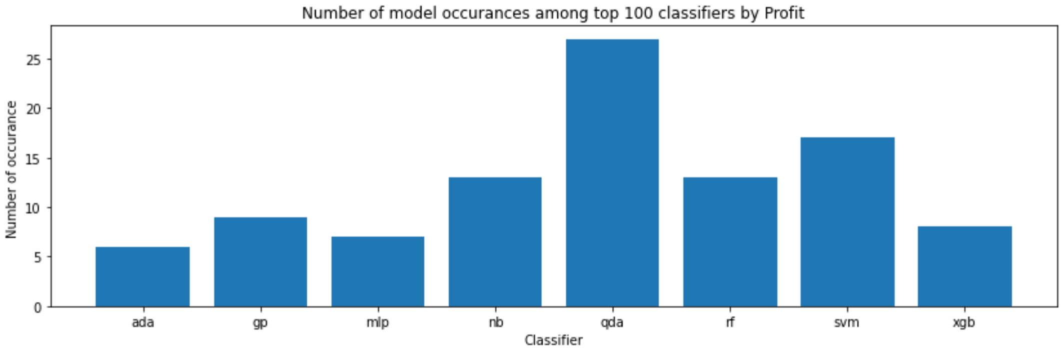

Figure 8: Distribution of occurrence of models used among 100 best performing classifiers from first step model library

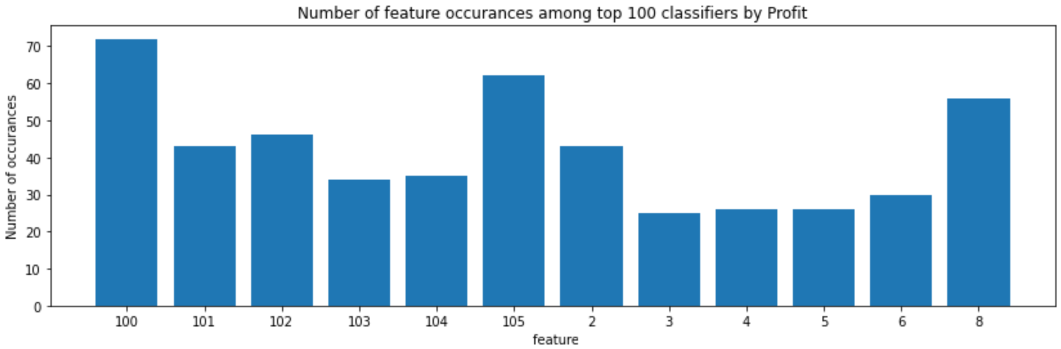

Figure 9: Distribution of occurrence of feature used among 100 best performing classifiers from first step model library

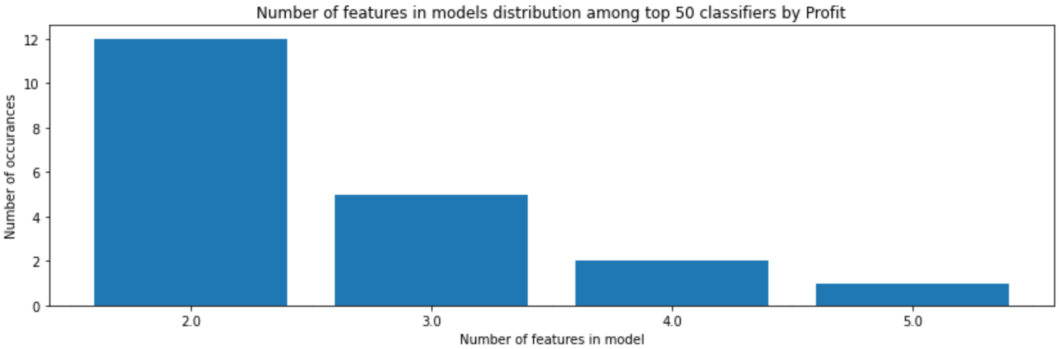

Figure 10: Distribution of accuracy set sizes among 50 best performing classifiers from first step model library

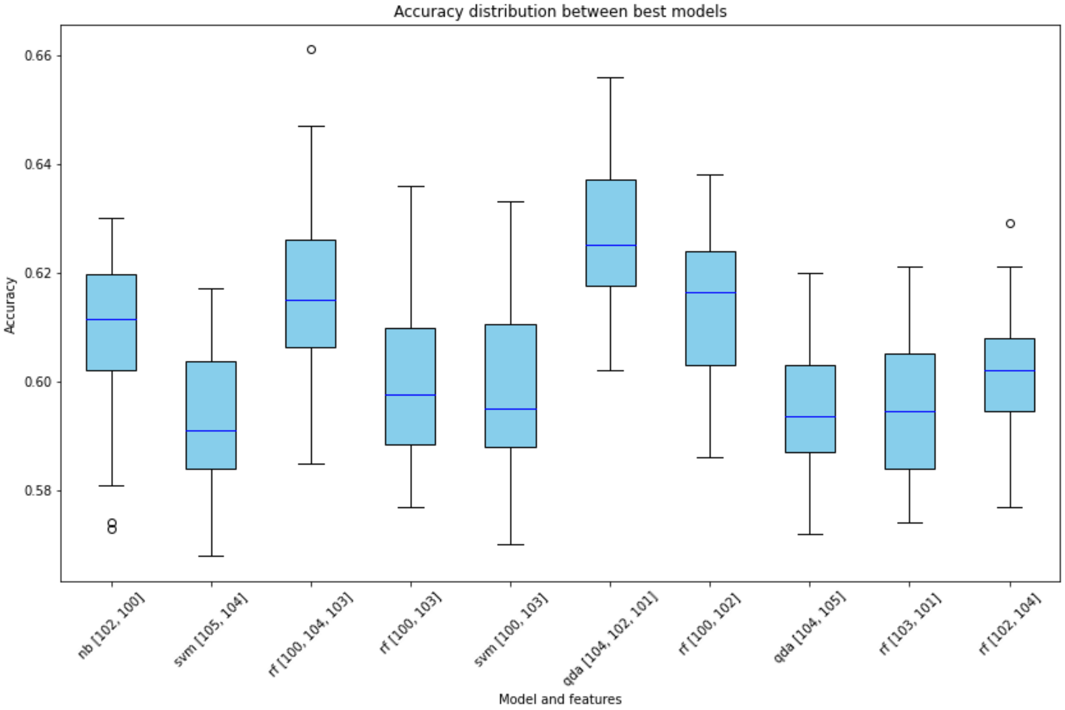

Figure 11: Distribution of accuracy for best models trained on 50 different data splits

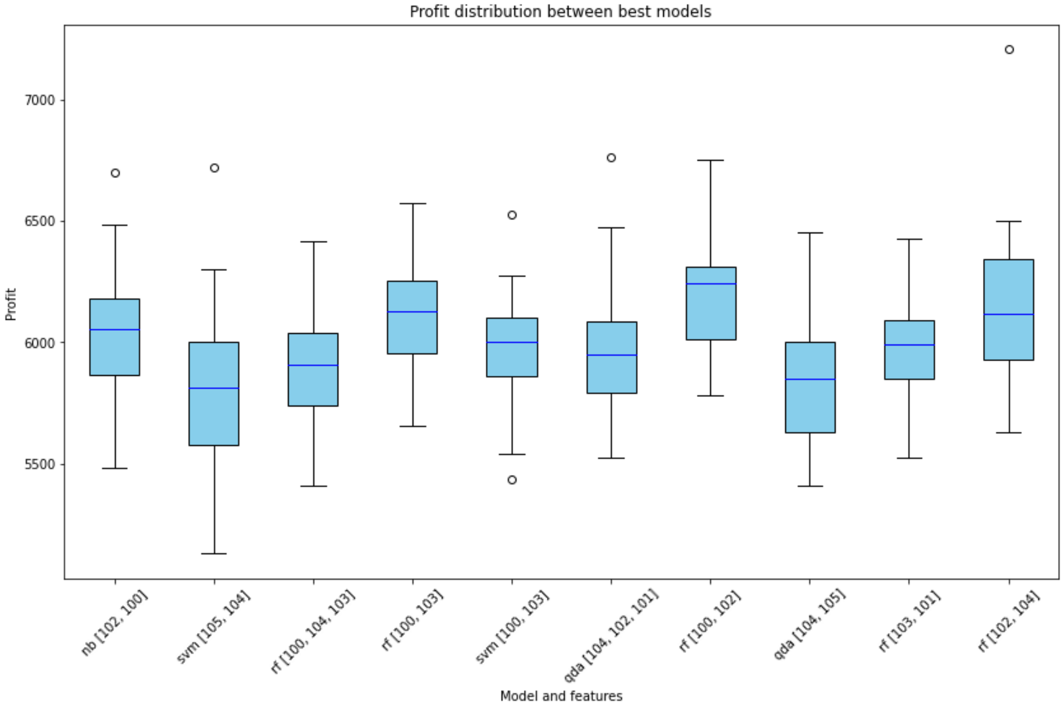

Figure 12: Distribution of profit for best models trained on 50 different data splits

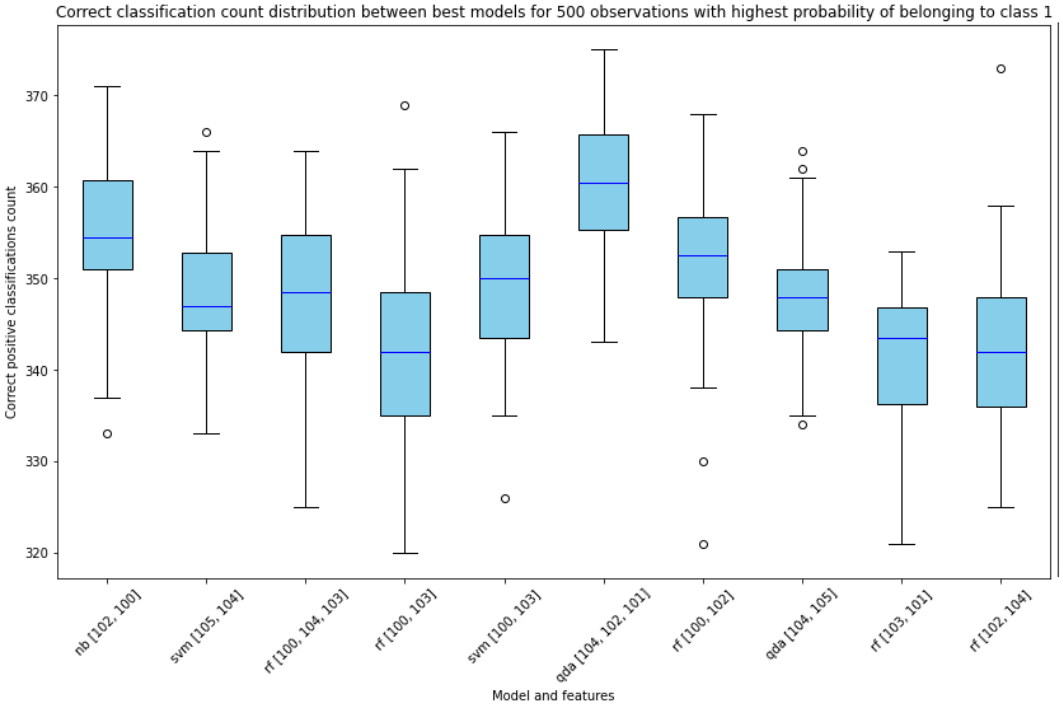

Figure 13: Distribution of number of correct classifications to positive class for best models for 500 observations with highest probability of belonging to positive class trained on 50 different data splits

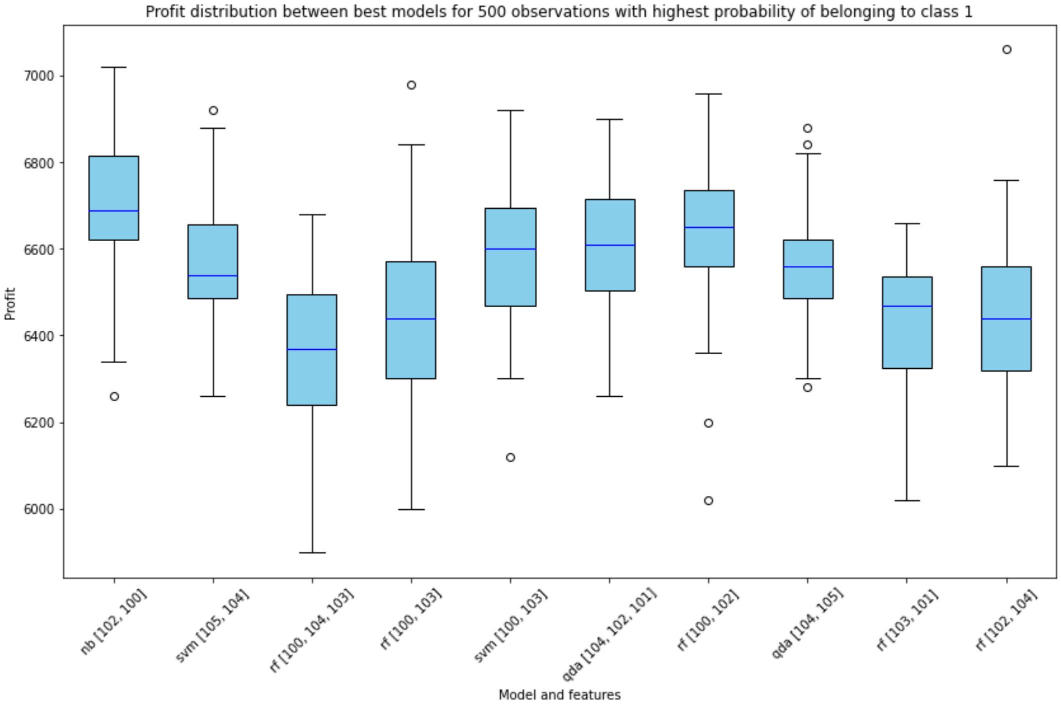

Figure 14: Distribution of profit for best models for 500 observations with highest probability of belonging to positive class trained on 50 different data splits