Project 2

Jakub Seliga, Krzysztof Wolny

June 2025

1 Introduction

Github: `https://github.com/kseligga/AML-proj2`

The goal of this project was to compare different classification models and feature selection methods to find the best-performing combination on the given dataset. The performance of your model was scored as follows:

- For each correctly identified household (i.e., one that did indeed exceed the threshold), the utility company pays you EUR 10.
- For each variable used in your model, you must pay EUR 200 to simulate the cost of acquiring and processing that data.

We started with checking what is the best model we could create while using all of the availiable features. Then, we performed experiments with different models, number of features, and methods of selecting those features.

2 Methodology

2.1 Dataset and scoring

Dataset combined of 500 features. Training and testing sets had 5000 observations. We decided to scale our dataset with StandardScaler from sklearn python library.

In our problem, both the training and test datasets contain 5 000 rows, and our ultimate goal is to identify 1 000 (20 %) of the test-set households most likely to exceed the usage threshold. To simulate how that 20 % selection will work on unseen data, we first split the 5 000 rows training set into 4 000 rows for training and 1 000 rows for validation set. On the validation fold of 1 000, we rank by each model's predicted probabilities and pick the top 20 % - the 200 highest-probability rows. Because we are selecting 200 out of 1 000 on validation, which corresponds proportionally to selecting 1 000 out of 5 000 on test, we simply multiply achieved profit by 5, to simulate what would be the expected profit on test dataset. After choosing the winning model and feature set, we retrain it on all 5 000 training rows and apply it to the 5 000 test rows. We sort the test predictions by probability and take the top 1 000 as our final threshold-exceeding households.

The best results we could get in validation set were:

- **Maximum reward on validation:** $200 \times 10 \times 5 = 10000$
  (All 200 cases were completed successfully.)
- **Minimum variable cost on validation:** 200
  (Using one variable)
- **Maximum final score on validation:** $200 \times 10 \times 5 - 200 = 9800$
  (All 200 cases were completed successfully with the minimum variable cost.)

2.2 Methods used

Initially, we tested multiple classification models using all available features. The models included:

- Logistic Regression,
- Random Forest Classifier,
- Support Vector Machine (SVM),
- XGBoost Classifier.

We performed randomized hyperparameter search with 5-fold cross-validation on four classifiers. The best accuracy scores on the training data were: Logistic Regression (0.693), Random Forest (0.690), SVM (0.655), and XGBoost (0.512). In further exploration we used models with their best parameters.

We evaluated cost for every combination of feature selection methods and classification models. We selected four feature selection methods for this task, to see which give the best results:

- Recursive Feature Elimination (RFE) with XGBoost Classifier,
- Shap with XGBoost Classifier,
- Mean Decrease in impurity with XGBoost Classifier,
- Univariate feature selection with F-test for feature scoring.

Four classification methods we explored during search were listed above. We are calculating probability that it is eqaul to 1. We take 200 of those indices that have biggest probability. We also did ensembling on our models and all feature selection methods.

3 Results

3.1 Overall Results

The complete evaluation results for all model-feature selection combinations are available in Table[1].

Overall results show, that we achieve best score for model with a small number of selected features - the very best configurations are the ones with number of features equal to only 1.

- Best overall accuracy: **0.732** with **SHAP + SVM**.
- Best overall score: **7600** with **SHAP + Ensemble soft svm+lr**.

feat sel	name	sel index	nr of feat	acc	recall	precision	reward	var cost	final score
shap	Ensemble soft svm+lr	2	1	0.709	0.719836	0.695652	7600.0	200	7400.0
rfc	Ensemble soft lr+svm	2	1	0.709	0.719836	0.695652	7600.0	200	7400.0
shap	LogisticRegression	2	1	0.707	0.695297	0.702479	7600.0	200	7400.0
rfc	LogisticRegression	2	1	0.707	0.695297	0.702479	7600.0	200	7400.0
ufs	LogisticRegression	2	1	0.707	0.695297	0.702479	7600.0	200	7400.0
rfc	Ensemble soft rf+xgb+lr	2, 6	2	0.698	0.707566	0.685149	7750.0	400	7350.0
ufs	Ensemble soft rf+lr	2	1	0.625	0.615542	0.616803	7550.0	200	7350.0
ufs	Ensemble soft lr+rf	2	1	0.625	0.615542	0.616803	7550.0	200	7350.0
rfc	Ensemble soft lr+svm	2, 3	2	0.710	0.721881	0.696252	7700.0	400	7300.0
ufs	Ensemble soft xgb+svm+rf	2, 6	2	0.700	0.709611	0.687129	7700.0	400	7300.0
ufs	Ensemble soft xgb+svm+lr	2	1	0.710	0.719836	0.697030	7450.0	200	7250.0
rfc	Ensemble soft rf+svm+lr	2, 3	2	0.709	0.719836	0.695652	7650.0	400	7250.0
shap	Ensemble soft xgb+rf+svm+lr	2	1	0.691	0.693252	0.680723	7450.0	200	7250.0

Table 1: Model performance including accuracy, recall, and precision, with cost metrics for various feature selection and ensemble methods

3.2 Final Models

Based on accuracy, consistency, and interpretability, we selected the following models:

- **Final Model 1**: SHAP + Ensemble soft svm+lr | Indexes: 2
- **Final Model 2**: RFC + Ensemble soft lr+svm | Indexes: 2
- **Final Model 3**: SHAP + LogisticRegression | Indexes: 2
- **Final Model 4**: RFC + Ensemble soft rf+xgb+lr | Indexes: 2, 6

We used **Final Model 1** to calculate test data and send it in our solution.