Advanced Machine Learning
Project 2 – Cost-Aware Customer Selection

Bruno Tobiasz, Mikołaj Trębski

June 2025

1 Problem overview

The goal of this project is to identify a subset of electricity customers who should receive an energy-saving offer, based on past consumption patterns. The dataset contains 5,000 samples with 500 features and binary target variable. While the reward is €10 per correctly identified customer (true positive), each feature used in the final model incurs a cost of € 200. The goal is to maximize profit.

2 Evaluation metric

The score function used to evaluate each model is defined as:

$$\text{Score} = 10 \times \text{TP} - 200 \times \#\text{features} \tag{1}$$

Where TP refers to true positives on the test set, and the number of features is the count of input variables used by the model after feature selection. This metric balances model complexity (cost) against the value of correct predictions.

3 Feature selection

To determine the best subset of features to train the models on, two methods from Scikit-Learn package had been used:

1. `SelectKBest()` that chooses specified number k of features with highest scores,
2. `RandomForestClassifier()` that allows to determine subset of features based on calculated importance higher than threshold.

For both methods, different values of k and threshold were verified in terms of performance. Starting from bigger subsets (from up to 500) to smaller (up to 2), easy conclusion could be made, that for certain threshold (more than ∼25 features), the performance of all models was degrading or just constant on less satisfying level.

Apart from lower accuracies, the scores from score function were dramatically decreasing with higher number of features. That's why only smaller subsets were taken into account for finding the optimal solution.

The feature importance analysis revealed that only a fraction (13 out of 500) of the features had more significant importance, i.e., higher than 0.003, as shown in Figure 1.

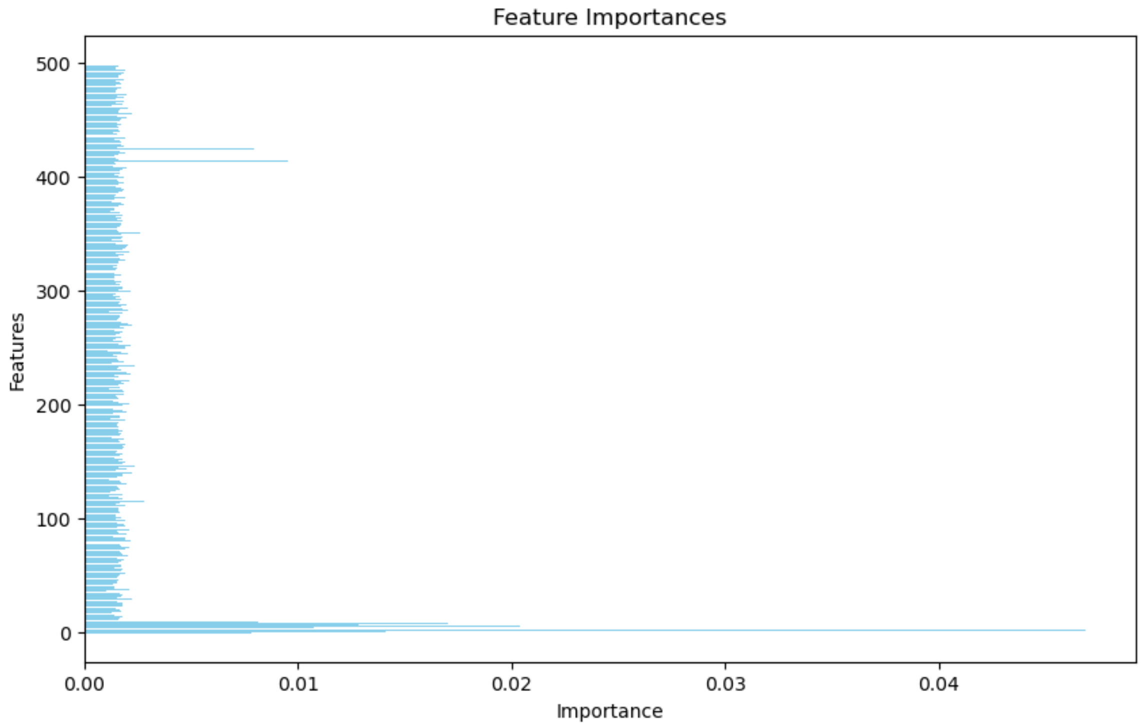

Figure 1: Feature importances distribution showing that only 13 out of 500 features have importance greater than 0.003

4 Models

To find the best possible model for the problem, 5 different algorithms had been tested with different combinations of parameters. All but AdaBoost were taken from Scikit-Learn library:

1. Logistic Regression
2. RandomForestClassifier
3. AdaBoost
4. GradientBoostingClassifier
5. BaggingClassifier

For LogisticRegression two hyperparameters were tested: C and penalty. For others: number of estimators, and maximum depth of tree.

5 Modeling strategies

Five distinct strategies were evaluated, combining different classifiers with combinations of hyperparameters and feature selection techniques with various thresholds. To allow easier tests, each model had been checked individually, as seen in the logistic regression example in Table 1.

The results, comprising of both scores and accuracies achieved by cross-validation on training set, allowed to find the model with best ratio between high score and high accuracy, as seen in Table 2.

Table 1: Example results for model=LogisticRegression

Score	Accuracy	Params	Feature_Selector	Num_Features
4265.0	0.6918	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=2	2
4265.0	0.6918	{'C': 0.01, 'penalty': 'l2'}	SelectFromModel_thresh=99.7th_percentile	2
4195.0	0.6996	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=2	2
4195.0	0.6996	{'C': 0.01, 'penalty': 'l1'}	SelectFromModel_thresh=99.7th_percentile	2
4012.5	0.6892	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=3	3
4012.5	0.6892	{'C': 0.01, 'penalty': 'l2'}	SelectFromModel_thresh=99.5th_percentile	3
3995.0	0.6996	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=3	3
3995.0	0.6996	{'C': 0.01, 'penalty': 'l1'}	SelectFromModel_thresh=99.5th_percentile	3
3797.5	0.7014	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=4	4
3787.5	0.6986	{'C': 0.01, 'penalty': 'l1'}	SelectFromModel_thresh=99.2th_percentile	4
3722.5	0.6798	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=4	4
3717.5	0.6728	{'C': 0.01, 'penalty': 'l2'}	SelectFromModel_thresh=99.2th_percentile	4
3710.0	0.7076	{'C': 0.1, 'penalty': 'l2'}	SelectKBest_k=2	2
3710.0	0.7076	{'C': 0.1, 'penalty': 'l2'}	SelectFromModel_thresh=99.7th_percentile	2
3652.5	0.7064	{'C': 0.1, 'penalty': 'l1'}	SelectKBest_k=2	2
3652.5	0.7064	{'C': 0.1, 'penalty': 'l1'}	SelectFromModel_thresh=99.7th_percentile	2
3635.0	0.7062	{'C': 1.0, 'penalty': 'l2'}	SelectFromModel_thresh=99.7th_percentile	2
3635.0	0.7062	{'C': 1.0, 'penalty': 'l2'}	SelectKBest_k=2	2
3627.5	0.7058	{'C': 1.0, 'penalty': 'l1'}	SelectFromModel_thresh=99.7th_percentile	2
3627.5	0.7058	{'C': 1.0, 'penalty': 'l1'}	SelectKBest_k=2	2
3597.5	0.7014	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=5	5
3587.5	0.6986	{'C': 0.01, 'penalty': 'l1'}	SelectFromModel_thresh=99th_percentile	5
3522.5	0.7072	{'C': 0.1, 'penalty': 'l2'}	SelectKBest_k=3	3
3522.5	0.7072	{'C': 0.1, 'penalty': 'l2'}	SelectFromModel_thresh=99.5th_percentile	3
3507.5	0.6734	{'C': 0.01, 'penalty': 'l2'}	SelectFromModel_thresh=99th_percentile	5
3500.0	0.6784	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=5	5
3477.5	0.7080	{'C': 0.1, 'penalty': 'l1'}	SelectFromModel_thresh=99.5th_percentile	3
3477.5	0.7080	{'C': 0.1, 'penalty': 'l1'}	SelectKBest_k=3	3
3462.5	0.7082	{'C': 1.0, 'penalty': 'l2'}	SelectKBest_k=3	3
3462.5	0.7082	{'C': 1.0, 'penalty': 'l2'}	SelectFromModel_thresh=99.5th_percentile	3
3450.0	0.7072	{'C': 1.0, 'penalty': 'l1'}	SelectFromModel_thresh=99.5th_percentile	3
3450.0	0.7072	{'C': 1.0, 'penalty': 'l1'}	SelectKBest_k=3	3
3400.0	0.7012	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=6	6
3372.5	0.7164	{'C': 0.1, 'penalty': 'l2'}	SelectKBest_k=4	4
3342.5	0.7172	{'C': 0.1, 'penalty': 'l1'}	SelectKBest_k=4	4
3327.5	0.7062	{'C': 0.1, 'penalty': 'l2'}	SelectFromModel_thresh=99.2th_percentile	4
3310.0	0.7156	{'C': 1.0, 'penalty': 'l2'}	SelectKBest_k=4	4
3307.5	0.7156	{'C': 1.0, 'penalty': 'l1'}	SelectKBest_k=4	4
3267.5	0.7070	{'C': 0.1, 'penalty': 'l1'}	SelectFromModel_thresh=99.2th_percentile	4
3262.5	0.6758	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=6	6
3250.0	0.7062	{'C': 1.0, 'penalty': 'l2'}	SelectFromModel_thresh=99.2th_percentile	4
3240.0	0.7062	{'C': 1.0, 'penalty': 'l1'}	SelectFromModel_thresh=99.2th_percentile	4
3187.5	0.6996	{'C': 0.01, 'penalty': 'l1'}	SelectKBest_k=7	7
3182.5	0.7170	{'C': 0.1, 'penalty': 'l2'}	SelectKBest_k=5	5
3130.0	0.7164	{'C': 0.1, 'penalty': 'l1'}	SelectKBest_k=5	5
3127.5	0.7078	{'C': 0.1, 'penalty': 'l2'}	SelectFromModel_thresh=99th_percentile	5
3100.0	0.7146	{'C': 1.0, 'penalty': 'l2'}	SelectKBest_k=5	5
3097.5	0.7144	{'C': 1.0, 'penalty': 'l1'}	SelectKBest_k=5	5
3062.5	0.6732	{'C': 0.01, 'penalty': 'l2'}	SelectKBest_k=7	7
3062.5	0.7078	{'C': 0.1, 'penalty': 'l1'}	SelectFromModel_thresh=99th_percentile	5

Table 2: Best model combinations for each algorithm

Algorithm	Score	Accuracy	Hyperparameters	Features
Logistic Regression	3710	0.7076	{C: 0.1, penalty: l2}	2
	3342.5	0.7172	{C: 0.1, penalty: l1}	4
Random Forest	3500	0.7174	{n_estimators: 500, max_depth: 5}	4
	3807.5	0.7048	{n_estimators: 500, max_depth: 3}	2
AdaBoost	3860	0.7060	{n_estimators: 400, max_depth: None}	2
GradientBoosting	3342.5	0.7172	{C: 0.1, penalty: l1}	4
Bagging	3055	0.7068	{n_estimators: 500}	6

With these models and used parameters, we evaluated the subset of variants on the same dataset split into training and testing sets.

6 Final model

The final, most promising selected model was a **RandomForestClassifier** with 400 estimators and a maximum depth of 5, using only 4 features selected via SelectKBest method. This model produced the highest cost-adjusted score of **4380** and accuracy of **72.4%**.

The selection of this model was based on achieving the optimal balance between prediction accuracy and feature cost, maximizing the profit function defined in the evaluation metric. The RandomForest approach proved particularly effective for this cost-sensitive classification problem, demonstrating that ensemble methods can achieve high performance while maintaining parsimony in feature selection.

7 Conclusion

This project successfully demonstrated the importance of cost-aware feature selection in machine learning applications. The key findings include:

- Feature selection is crucial when features have associated costs
- A small subset of features (4 out of 500) can achieve competitive performance
- RandomForest with proper hyperparameter tuning provides the best cost-benefit trade-off
- The evaluation metric effectively balances accuracy against feature acquisition costs

The final model achieves a cost-adjusted score of 4380, representing an effective solution for the utility company's customer targeting problem while maintaining operational efficiency through minimal feature requirements.