Advanced Machine Learning Project 2

Paulina Pacyna, 290600
Mateusz Wójcik, 290639

30 May 2021

1 Introduction

The purpose of this project is to explore and compare different feature selection algorithms. To achieve this, we have tested Lasso logistic regression, Gini importance from Random Trees, Boruta algorithm and Recursive Feature Elimination algorithm.

2 Lasso logistic regression

Lasso regression analysis is based on the logistic regression model, but with a penalization term added. In Lasso regression we try to minimize the following function

$$\min_{\alpha,\beta} L(x,y) = \min_{\alpha,\beta} \Big\{ \sum_{i=1}^{N} (y_i - \alpha - \beta^T x_i) + \frac{1}{C}(||\beta||_{l1} + |\alpha| \Big\}),$$

where C is a constant.

We used LogisticRegression model from scikit learn with $l1$ penalization, and we examined the relation between the constant the accuracy, number of features selected and the constant C. The results are presented below (Figure 1). We used this model with constant $C = 10$ to evalueate validation data.

3 Random forest

Another option to select features is to use the importance based on Random Forests. This importance3 is calculated based on Gini importance.
We've performed this test using `RandomForestClassifier` from `scikit learn`. We've obtained 0.669 accuracy based on 143 features on the artificial dataset, and 0.972 accuracy on digits dataset, with 647 features selected.

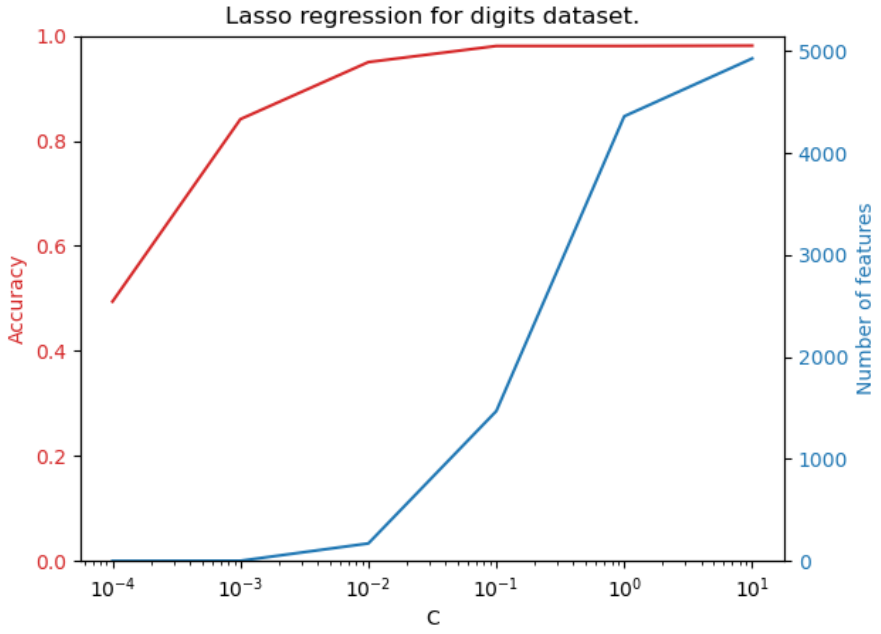

Figure 1: Relationship between constant C, accuracy and number of selected features for the digits dataset

4 Boruta Algorithm

As a third algorithm, we applied Boruta Algorithm. We have used *scikit-learn*-based implementation of the algorithm which is inspired by the equivalent implementation in R.

Boruta is mostly based on iteratively removing features that appear to be statistically less relevant than a random probe. They are commonly called *artificial noise variables* introduced by the algorithm. Unlike other algorithms, Boruta tries to find all features carrying information useful for prediction, rather than finding a possibly compact subset of features on which some classifier has the least error. In particular, in each iteration, rejected variables are removed from consideration in the next iteration of the algorithm. The algorithm is based on *Random Forest Classifier* that we introduced with default *scikit-learn* parameters.

The results we obtained are presented as follows.

The resulting accuracy was obtained by splitting the given training data. We obtained that for this algorithm standardizing variables did not produce significantly more reliable results.

Dataset	Accuracy	Number of Features
Artificial	0.892	19
Digits	0.979	920

Table 1: Boruta Algorithm Results

5 Recursive Feature Elimination

Recursive Feature Elimination is a wrapper-type feature selection algorithm that works by searching a subset of features by starting with all of the variables and successively removing features. This is achieved by fitting a specific classification algorithm (*Random Forest Classifier* in our case), ranking by importance, discarding the least important, and re-fitting the model (Machine Learning Mastery). We applied this algorithm alongside with $10-$fold cross-validation in order to prevent from over-fitting.

To implement the algorithm, we again used *scikit-learn* Python module. We standardized the variables before learning and used default parameters in the classifier. The results are presented as follows. Since the data was high-dimensional, in each iteration of the algorithm we decreased the number of features by 3 for artificial data set and by 10 in digits data set.

Dataset	Accuracy	Number of Features
Artificial	0.896	20
Digits		

Table 2: RFE Algorithm Results

It is worth mentioning that finding the optimal model with $10-$fold cross-validation was highly time-consuming. The execution of the algorithm lasted for about 3 hours with about 4.6GHz speed on 11th-generation Intel CPU.