Advanced Machine Learning

Project 2
Feature Selection

Albert Roethel, Marcel Affi

316061,316056

MSc Data Science

Faculty of Mathematics and Information Systems

Warsaw University of Technology

2021

Introduction

Feature selection is the process of selecting a subset of relevant features used to train a machine learning model. Raw data is the basic building block of ML algorithms. But on its own it can't be used to accurately train models. Instead, it must be refined to "features" – variables or attributes that can be used for analysis. The features we use to train a ML model are crucial to its performance. This means selecting the most relevant ones possible is absolutely vital.

For this project we were given two data sets with the following shapes :

- Artificial : Train (2000,500), Test (600, 500)
- Digits : Train (6000, 5000), Test (1000, 5000)

Our project will use the following feature extraction methods to select the most important features: (1) ElasticNet, (2) Extremely Randomized Trees, (3) Permutation Feature Importance, (4) Recursive feature elimination, (5) Boruta, (6) XGBoost

General approach

Main goal of this project was to select best features for two datasets (digits and artificial) and then based on those features make a prediction for the validation set.

Our approach looked as follows. At first we standardized our data (subtracted mean and divide by standard deviation for each predictor). Then we splitted randomly our datasets into two non overlapping sets - train and test set ($\frac{2}{3}$ of all observations belonging to training data and the rest for test). Then we considered several methods of selecting features. For each of them we used training data to pick up only relevant features. Then using 5-fold cross validation on training set we selected best parameters for SVM with rbf kernel (we searched for best gamma and C hyper-parameters). Finally we tested our approach on independent test data which wasn't seen on any of the previous steps. Model and feature selection method performing best on test data was selected for the submission prediction. Model was train on the whole data. Before final prediction we decor-related co-linear features, so that number of predictors was smaller but without sacrificing the accuracy of the model.

In this work we used Python libraries for ML algorithms: `sklearn`, `xgboost`, `Boruta`.

ElasticNet

Elastic Net is an embedded method for feature selection. It uses a convex combination of L1 and L2 penalty to shrink the coefficients of the unimportant features to 0 or near zero. During learning it also uses the penalty parameter α which indicates how strong the penalty should be. We used cross validation to find best α and proportion of L1 to L2. In case of artificial dataset we ended up with just Lasso model (no L2 penalty included), $\alpha = 0.065$ and 19 features. The accuracy of such Lasso model was however only 58%. possibly due to nonlinear relation between X and y. Best SVM reached 70% on test set. In the case of digits dataset, the lasso model performed much better with the same penalty values and α

used as in the artificial dataset, resulting in a accuracy of 97% with 68 features selected on the best SVM selected hyper-parameters.

Extremely Randomized Trees

Compared to other trees enables this method goes further in the way it tries to reduce the bias by introducing some randomness into the model. As in random forests, a random subset of candidate features is used, but instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule. Each predictor is assigned its feature importance which is associated with how close to the root the decision based on this predictor was made. Following algorithm returned 20 features which had significantly better feature importance score. Best SVM resulted in 90.6% of accuracy on test set.

Permutation feature importance

In this method we picked trained SVM on training data containing all predictors with optimized hyper parameters. It achieved 67% of accuracy on test set. Then we calculated permutation feature importance, which is a method that for each predictors permutes randomly the predictors and calculates how much the accuracy dropped. If it drops significantly then this features is important. With this method for artificial dataset we got 3 features, based on which best SVM produced 64% of accuracy on test set.

Recursive feature elimination

Recursive feature elimination is a method which recursively trains selected model (i.e. Gradient Boosting Classifier as in our case) starting from all the features and then iteratively removes the least significant features based on metric returned by the model. Procedure is done after reaching target number of features. We let the algorithm to select 20 features. Best SVM yielded 89.4% of accuracy for artificial dataset.

Boruta

Boruta iteratively removes features that are statistically less important than a random probe (artificial noise variables introduced by the Boruta algorithm). In each iteration, rejected variables are removed from consideration in the next iteration. In our case Boruta returned 20 features with Extremely Randomized Trees and SVM yielded 90% of accuracy for test data.

XGBoost

Boruta iteratively removes features that are statistically less important than a random probe (artificial noise variables introduced by the Boruta algorithm). In each iteration, rejected variables are removed from consideration in the next iteration. In our case Boruta returned

feature selection method	dataset			
	artifitial		digits	
	accuracy	# of features	accuracy	# of features
ElasticNet	69.9%	19	97.0%	68
Extremely Randomized Trees	90.6%	20	97.7%	43
Permutation importance	64.2%	3	50.2%	all
Recursive feature elimination	89.5%	20	95.7%	20
Boruta	89.6%	20	97.9%	1088
XGBoost	89.6%	9	92.2%	8

20 features with Extremely Randomized Trees and SVM yielded 90% of accuracy for test data.

Final model

Final selection of features was performed by the Extremely Randomized Trees as SVM build on the top of features selected by this method produced best results. It was noticed that among 20 features most of them are heavily correlated with other features (see Fig. 1). Because of this we selected 101features that are not highly correlated with any other feature. Then we trained the SVM model on full data with only 101features and then we made prediction of posterior probability of class +1 for validation set.

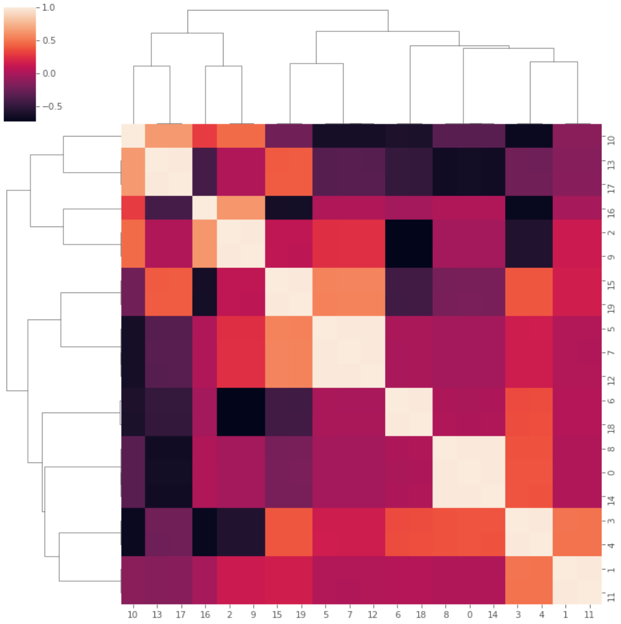

Figure 1: Correlogram of 20 features selected by Extremely Randomized Trees.