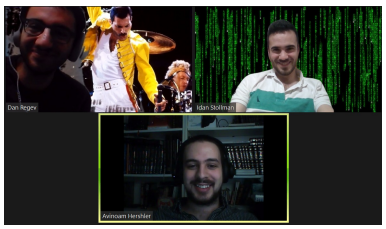
**67577 Introduction To Machine Learning | Hackathon**

**Dan Regev 208272518 | Avinoam Hershler 208926964 | Idan Stollman 315009480**

June 12, 2020




**Feature Selection**

The most interesting and exciting part of the task was extracting the features from the data.

We decided to split the data to three parts: train, validate and test (60%, 20%, 20%, correspondingly).

We first built a basic learner , then tried to boost it using Adaboost which didn't work beacuse our algorithm didn't minimize the empirical risk. The way it works is as follows: It treats the lines of code as samples and creats a histogram of the frequency of each word in each project. To predict to which project the lines of code belong it "gathers a commitee" of the words in the lines and asks to which project do they think that they belong. The decision is taken according to the "vote" of the majority. The results were quite good. We decided that it would be much clever to give weight to the amount of confidence a word has in its decision. We did so using a very cool function we wrote. If we had more time we would play a bit more with it. We tried to use the function 'split' with more delimiters other than whitespace so we could use more words from the text but it just made things much much worse - the error rate increased from 0.27 to 0.87!

We thought that if we normalise the histogram with the absolute norm we saw in the lectures then the performance will be better, and indeed it did.

For each line of code we got a vector of size 6 which contained the amount of matching of the line to each corresponding project (the projects being labeld 0 to 6).

The baseline learner took the argmax of the vector, and then we realised that we have a classification problem in $\mathbb{R}^7$.

**Classification in $\mathbb{R}^7$**

We tried using several classifiers, including logistic regression, SVM, k-nearest-neighbours, decision tree with boosting and random forest, all with different tuning parameters tested on the validation set (see figures and notes below).

Random Forest, depth - 25 score: 0.939 training set, 0.811 validate set, **0.813 test set**.

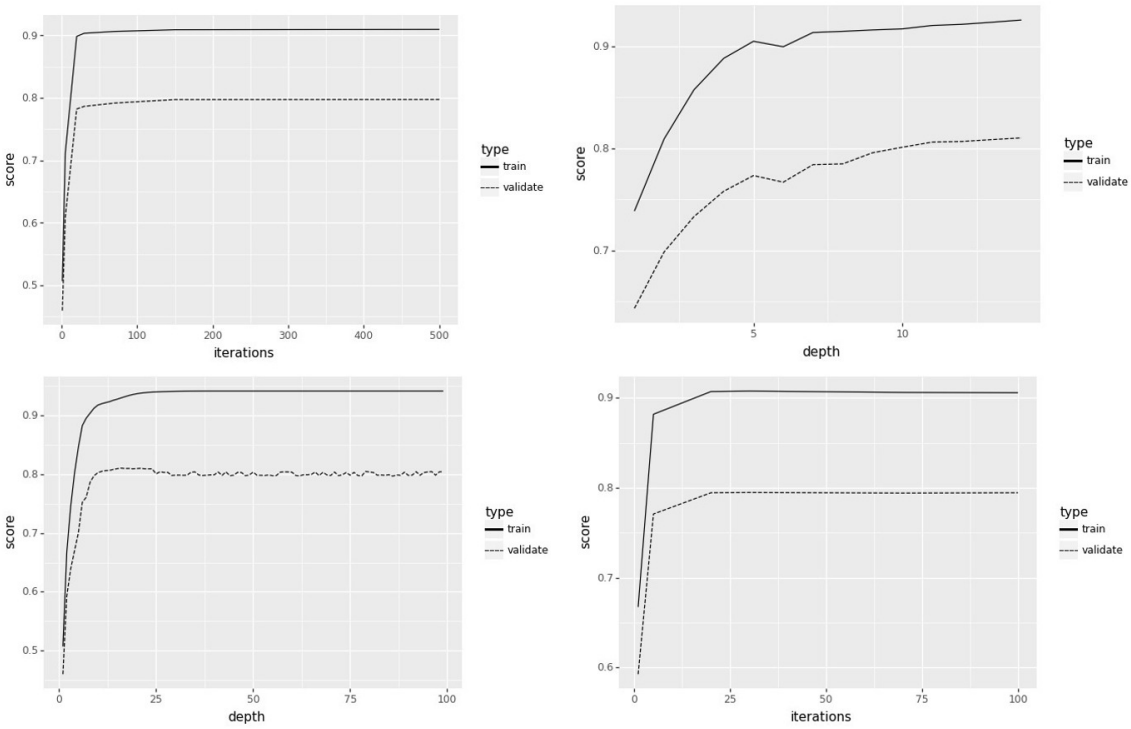
BaseLine Learner score: 0.89 training set, 0.77 validate set.

Logistic Regression score: 0.90 training set, 0.79 validate set.

Random Forest, depth - 30 score: 0.94 training set, 0.80 validate set.

Random Forest, depth - 20 score: 0.93 training set, 0.81 validate set.

Tree with depth 100, Random Forest, depth - 15 score: 0.92 training set, 0.81 validate set, Adaboost with 500 iterations and Decision stump, Adaboost with 100 iterations and Decision tree with max depth 2 (clockwise):




**Submission**

After serializing our classifier we realised that it greatly exceeded the bound of 20 MB - in fact it was of size 180 MB. So we moved the train procedure to the constructor of the classifier.