Introduction to ML - report on experiments

1 Description of experiments

For machine learning experiments, I divided the data into two parts - the training and the heldout data.

The heldout data were the last 30 lines, the training were the first 220 lines. The data were shuffled "randomly" – the word "randomly" is in quotes, because I shuffled the array (1, 2, . . . 250) before conducting the experiment and I used the same shuffling everytime with every word.

For features, I used just the features, recommended in the PDF. Which subset is to be used exactly is described further in this report.

For all the words, I tried the following methods:

- Naive Bayes classifier from R's `e1071` package
- Support Vector Machine classifier from the same package
- Decision Tree classifier from the `rpart` package
- Bagging from `adabag` package
- Boosting from `adabag` package

Now, Bagging and Boosting methods use I believe Decision Trees "inside" them, but the `bagging` and `boosting` methods from adabag are to be used as a stand-alone classifiers, so I treated them as such.

For a feature selection, I first tried an evolutionary algorithm. Evolutionary algorithm are algorithms for optimizing some function (called *fitness*function in evolutionary algoritms) over a big solution space (called *population*), that's computationally hard to walk trough one by one; I thought that, taking all subsets of all features as a population and fitness function as a result of an experiment with this particular subset of features, evolutionary algorithm may work and give sufficiently good results.

It turns out feature selection is – at least in my experiments – not a particularly good problem for evolutionary algorithm. The results don't really benefit from any type of crossover and are actually getting worse from generation to generation. Plus, it takes unnecessarily big amount of time.

In Jan Václ's presentation, he talked about ignoring the features, that have the same value the whole time. I thought about taking it a bit further – I tried to first transform all features into binary and then ignoring all the features, that have the lesser used value less than 5-times.

The results were actually a bit better than using all the features. I called this "feature cutting". This implicates a question, where exactly to "cut" the features – if we should cut it at 5, 20, 50. . . Again, cutting at 20 means ignoring the features where the lesser used value is used less than 20.

I also found out that with parameter tuning, I got sometimes better results when doing the grid search *myself* than with a built-in parameter tuning. So, I tried both parameter tuning myself (it's possible with everything except naive Bayes), built-in parameter tuning (only possible with DT and SVM) and default parameters.

So, to summarize everything I wrote, for every word I tried the following:

- 5 possible classifier methods
- for each of these, I tried for feature searching:
  - evolution algorithm, which returns 1 result for the best feature set
    - for the algorithm to be quicker, I actually cut features even here at 3; the population are subsets of the set of features that has the lesser used value used more than 3 times
    - a fitness value for this was a result of 11-fold cross validation on the training data, tried on the classifier method with the default parameters
    - I skip evolution algoritm for bagging and boosting, since it simply takes too long
  - cut the features at 5, 10, 20, 50 and 70 (arbitrarily chosen values) - this returns 5 feature sets, independent on the classifier method
    - this step is actually done in the script that transforms the input data into R-readable table, called `transform.pl`
- for each combination of 5 classifier method and 6 feature sets from the last step, I used 3 forms of parameter searching (where it makes sense)
  - first called 0 is no parameter searching
  - second called 1 is built-in parameter searching
  - lastly I implement my own grid-search, which is again 11-fold cross validation on the training data for every possible combination of parameters; I call that 2

After these all elaborate steps, I have exactly 73 classifiers for every verb.

Originally, I thought I will take those 73 classifiers (which has not seen any of the "heldout" data in any step), try them on the heldout data and take the best one in every verb as final. The results are shown below, as Results 1.

However, I realized the resulting model might be too much over-fitted on the heldout data. Also, the results were the same for a lot of the best models. So, I thought about taking the best 20 models for every verb, once again cross-validating each of them – this time on the whole data, disregarding the testing/heldout split, using 10-fold cross-validation. This was actually quick, because I skip the "grid search" – the parameters are already "set in stone".

To my surprise, what almost always "won" this evaluation were bagging and boosting models, without any parameter tuning, with "cut" features. The results indicate that evolutionary algorithm and custom parameter tuning may be still a bit overfitting. The results are shown below, as Results 2.

I generalized a little here and decided to use **only** bagging and boosting for the "final" model that I hand in as a result – both for simplicity and for the fact that it generalizes nicely. So, the final question is only whether to use bagging on boosting models in each case, and where exactly to cut the features.

As you probably noticed, nothing in this report is verb-specific; I tried everything for every verb. So, if I need to chose a specific verb, I am choosing the ones that got the best difference between the final, 10-fold cross validation test, and the baseline. So, in other words, verbs plough, submit and ally.

2 Results

The results are always the number of right assignment, divided by the number of all assignments. They are always sorted from the best to the worst.

2.1 Results 1

In all tables, PS means "parameter search", lower means lower bound of 95% confidence interval, upper means upper bound of 95% confidence interval.

2.1.1 Ally

Baseline is 0.7.

method	feature selection	PS	result	lower	upper
bagging	cut 10	2	0.9	0.7926464	1.0073536
bagging	cut 10	0	0.9	0.7926464	1.0073536
bagging	cut 5	2	0.9	0.7926464	1.0073536
SVM	evolution	1	0.8666667	0.7450227	0.9883107
SVM	evolution	0	0.8666667	0.7450227	0.9883107
SVM	cut 2	1	0.8666667	0.7450227	0.9883107
bagging	cut 2	2	0.8666667	0.7450227	0.9883107
bagging	cut 20	0	0.8666667	0.7450227	0.9883107
SVM	cut 2	0	0.8666667	0.7450227	0.9883107
bagging	cut 20	2	0.8666667	0.7450227	0.9883107
SVM	cut 2	2	0.8666667	0.7450227	0.9883107
bagging	cut 2	0	0.8666667	0.7450227	0.9883107
SVM	evolution	2	0.8666667	0.7450227	0.9883107
SVM	cut 10	2	0.8333333	0.6999722	0.9666944
boosting	cut 10	0	0.8333333	0.6999722	0.9666944
boosting	cut 10	2	0.8333333	0.6999722	0.9666944
SVM	cut 5	0	0.8333333	0.6999722	0.9666944
SVM	cut 50	1	0.8333333	0.6999722	0.9666944
SVM	cut 70	1	0.8333333	0.6999722	0.9666944
bayes	evolution	0	0.8333333	0.6999722	0.9666944
SVM	cut 50	0	0.8333333	0.6999722	0.9666944
SVM	cut 50	2	0.8333333	0.6999722	0.9666944
SVM	cut 70	0	0.8333333	0.6999722	0.9666944
bagging	cut 50	0	0.8333333	0.6999722	0.9666944
SVM	cut 5	2	0.8333333	0.6999722	0.9666944
SVM	cut 70	2	0.8333333	0.6999722	0.9666944
SVM	cut 5	1	0.8333333	0.6999722	0.9666944
boosting	cut 5	0	0.8333333	0.6999722	0.9666944
bagging	cut 5	0	0.8333333	0.6999722	0.9666944
bagging	cut 70	0	0.8333333	0.6999722	0.9666944
SVM	cut 10	0	0.8333333	0.6999722	0.9666944
bayes	cut 2	0	0.8333333	0.6999722	0.9666944
boosting	cut 5	2	0.8333333	0.6999722	0.9666944
SVM	cut 10	1	0.8333333	0.6999722	0.9666944
boosting	cut 50	0	0.8	0.6568618	0.9431382
boosting	cut 50	2	0.8	0.6568618	0.9431382
SVM	cut 20	1	0.8	0.6568618	0.9431382
boosting	cut 2	2	0.8	0.6568618	0.9431382
DT	cut 70	1	0.8	0.6568618	0.9431382
SVM	cut 20	2	0.8	0.6568618	0.9431382
DT	cut 70	0	0.8	0.6568618	0.9431382
SVM	cut 20	0	0.8	0.6568618	0.9431382
boosting	cut 20	2	0.8	0.6568618	0.9431382
bayes	cut 5	0	0.8	0.6568618	0.9431382
boosting	cut 20	0	0.8	0.6568618	0.9431382
boosting	cut 2	0	0.8	0.6568618	0.9431382
bayes	cut 70	0	0.7666667	0.6153151	0.9180183
DT	cut 70	2	0.7666667	0.6153151	0.9180183
DT	cut 50	1	0.7666667	0.6153151	0.9180183
bagging	cut 50	2	0.7666667	0.6153151	0.9180183
boosting	cut 70	2	0.7666667	0.6153151	0.9180183
DT	cut 50	2	0.7666667	0.6153151	0.9180183
bagging	cut 70	2	0.7666667	0.6153151	0.9180183
DT	cut 50	0	0.7666667	0.6153151	0.9180183
boosting	cut 70	0	0.7666667	0.6153151	0.9180183
DT	cut 20	1	0.7333333	0.5750881	0.8915785
bayes	cut 50	0	0.7333333	0.5750881	0.8915785
DT	cut 20	2	0.7333333	0.5750881	0.8915785
bayes	cut 10	0	0.7333333	0.5750881	0.8915785
DT	cut 20	0	0.7333333	0.5750881	0.8915785
bayes	cut 20	0	0.7333333	0.5750881	0.8915785
DT	evolution	1	0.6666667	0.4979768	0.8353566
DT	evolution	2	0.6666667	0.4979768	0.8353566
DT	cut 10	2	0.6666667	0.4979768	0.8353566
DT	cut 10	1	0.6666667	0.4979768	0.8353566
DT	evolution	0	0.6666667	0.4979768	0.8353566
DT	cut 5	1	0.6666667	0.4979768	0.8353566
DT	cut 2	2	0.6666667	0.4979768	0.8353566
DT	cut 5	2	0.6666667	0.4979768	0.8353566
DT	cut 5	0	0.6666667	0.4979768	0.8353566
DT	cut 2	0	0.6666667	0.4979768	0.8353566
DT	cut 2	1	0.6666667	0.4979768	0.8353566
DT	cut 10	0	0.6666667	0.4979768	0.8353566

2.1.2 Arrive

Baseline is 0.5333333.

method	feature selection	PS	result	lower	upper
boosting	cut 50	0	0.6333333	0.4608896	0.8057770
bayes	cut 70	0	0.6333333	0.4608896	0.8057770
boosting	cut 50	2	0.6333333	0.4608896	0.8057770
SVM	cut 50	0	0.6	0.4246923	0.7753077
boosting	cut 70	0	0.6	0.4246923	0.7753077
SVM	cut 50	2	0.6	0.4246923	0.7753077
bayes	cut 50	0	0.6	0.4246923	0.7753077
SVM	cut 70	2	0.6	0.4246923	0.7753077
DT	cut 70	2	0.6	0.4246923	0.7753077
bagging	cut 50	0	0.6	0.4246923	0.7753077
bagging	cut 70	0	0.6	0.4246923	0.7753077
bagging	cut 50	2	0.6	0.4246923	0.7753077
SVM	cut 70	0	0.6	0.4246923	0.7753077
SVM	cut 50	1	0.6	0.4246923	0.7753077
SVM	cut 70	1	0.6	0.4246923	0.7753077
DT	cut 20	2	0.6	0.4246923	0.7753077
DT	cut 50	2	0.6	0.4246923	0.7753077
bagging	cut 70	2	0.6	0.4246923	0.7753077
boosting	cut 70	2	0.6	0.4246923	0.7753077
boosting	cut 10	2	0.5666667	0.3893416	0.7439918
DT	cut 50	1	0.5666667	0.3893416	0.7439918
boosting	cut 2	2	0.5666667	0.3893416	0.7439918
bagging	cut 10	0	0.5666667	0.3893416	0.7439918
boosting	cut 5	0	0.5666667	0.3893416	0.7439918
SVM	cut 5	0	0.5666667	0.3893416	0.7439918
bagging	cut 5	0	0.5666667	0.3893416	0.7439918
DT	cut 50	0	0.5666667	0.3893416	0.7439918
bagging	cut 10	2	0.5666667	0.3893416	0.7439918
boosting	cut 20	0	0.5666667	0.3893416	0.7439918
boosting	cut 5	2	0.5666667	0.3893416	0.7439918
bayes	cut 20	0	0.5666667	0.3893416	0.7439918
SVM	cut 5	1	0.5666667	0.3893416	0.7439918
DT	cut 70	1	0.5666667	0.3893416	0.7439918
boosting	cut 2	0	0.5666667	0.3893416	0.7439918
bagging	cut 2	0	0.5666667	0.3893416	0.7439918
SVM	cut 5	2	0.5666667	0.3893416	0.7439918
boosting	cut 20	2	0.5666667	0.3893416	0.7439918
bayes	cut 10	0	0.5666667	0.3893416	0.7439918
DT	cut 70	0	0.5666667	0.3893416	0.7439918
bagging	cut 2	2	0.5666667	0.3893416	0.7439918
boosting	cut 10	0	0.5666667	0.3893416	0.7439918
SVM	evolution	1	0.5333333	0.3548086	0.7118580
SVM	cut 20	1	0.5333333	0.3548086	0.7118580
DT	evolution	2	0.5333333	0.3548086	0.7118580
SVM	evolution	0	0.5333333	0.3548086	0.7118580
DT	cut 2	1	0.5333333	0.3548086	0.7118580
DT	cut 20	0	0.5333333	0.3548086	0.7118580
SVM	cut 10	2	0.5333333	0.3548086	0.7118580
DT	cut 2	2	0.5333333	0.3548086	0.7118580
DT	cut 10	0	0.5333333	0.3548086	0.7118580
SVM	cut 2	1	0.5333333	0.3548086	0.7118580
bagging	cut 20	2	0.5333333	0.3548086	0.7118580
SVM	cut 20	0	0.5333333	0.3548086	0.7118580
SVM	cut 10	0	0.5333333	0.3548086	0.7118580
DT	cut 20	1	0.5333333	0.3548086	0.7118580
SVM	cut 2	2	0.5333333	0.3548086	0.7118580
DT	evolution	0	0.5333333	0.3548086	0.7118580
SVM	cut 10	1	0.5333333	0.3548086	0.7118580
bayes	evolution	0	0.5333333	0.3548086	0.7118580
DT	cut 2	0	0.5333333	0.3548086	0.7118580
DT	cut 5	0	0.5333333	0.3548086	0.7118580
bagging	cut 20	0	0.5333333	0.3548086	0.7118580
bayes	cut 2	0	0.5333333	0.3548086	0.7118580
DT	cut 10	1	0.5333333	0.3548086	0.7118580
SVM	cut 20	2	0.5333333	0.3548086	0.7118580
DT	cut 5	2	0.5333333	0.3548086	0.7118580
SVM	evolution	2	0.5333333	0.3548086	0.7118580
DT	evolution	1	0.5333333	0.3548086	0.7118580
SVM	cut 2	0	0.5333333	0.3548086	0.7118580
DT	cut 5	1	0.5333333	0.3548086	0.7118580
bagging	cut 5	2	0.5333333	0.3548086	0.7118580
bayes	cut 5	0	0.5	0.3210773	0.6789227
DT	cut 10	2	0.4333333	0.2560082	0.6106584

2.1.3 Cry

Baseline is 0.4333333.

method	feature selection	PS	result	lower	upper
SVM	cut 50	0	0.7333333	0.5750881	0.8915785
DT	cut 2	2	0.7333333	0.5750881	0.8915785
bayes	cut 5	0	0.7333333	0.5750881	0.8915785
SVM	cut 5	1	0.7333333	0.5750881	0.8915785
SVM	cut 50	1	0.7333333	0.5750881	0.8915785
bayes	evolution	0	0.7333333	0.5750881	0.8915785
DT	cut 5	2	0.7333333	0.5750881	0.8915785
SVM	cut 5	0	0.7333333	0.5750881	0.8915785
DT	evolution	2	0.7333333	0.5750881	0.8915785
SVM	cut 50	2	0.7333333	0.5750881	0.8915785
SVM	cut 5	2	0.7333333	0.5750881	0.8915785
boosting	cut 10	0	0.7	0.5360146	0.8639854
bagging	cut 50	0	0.7	0.5360146	0.8639854
SVM	cut 10	1	0.7	0.5360146	0.8639854
bayes	cut 10	0	0.7	0.5360146	0.8639854
DT	cut 10	2	0.7	0.5360146	0.8639854
boosting	cut 10	2	0.7	0.5360146	0.8639854
SVM	cut 20	1	0.7	0.5360146	0.8639854
boosting	cut 20	0	0.7	0.5360146	0.8639854
bagging	cut 10	0	0.7	0.5360146	0.8639854
bagging	cut 50	2	0.7	0.5360146	0.8639854
SVM	cut 20	2	0.7	0.5360146	0.8639854
boosting	cut 20	2	0.7	0.5360146	0.8639854
boosting	cut 5	2	0.7	0.5360146	0.8639854
SVM	cut 10	2	0.7	0.5360146	0.8639854
bagging	cut 10	2	0.7	0.5360146	0.8639854
boosting	cut 5	0	0.7	0.5360146	0.8639854
SVM	cut 20	0	0.7	0.5360146	0.8639854
SVM	cut 10	0	0.7	0.5360146	0.8639854
SVM	evolution	2	0.6666667	0.4979768	0.8353566
bagging	cut 5	2	0.6666667	0.4979768	0.8353566
bayes	cut 20	0	0.6666667	0.4979768	0.8353566
boosting	cut 2	2	0.6666667	0.4979768	0.8353566
bagging	cut 20	0	0.6666667	0.4979768	0.8353566
bagging	cut 5	0	0.6666667	0.4979768	0.8353566
boosting	cut 50	2	0.6666667	0.4979768	0.8353566
SVM	evolution	0	0.6666667	0.4979768	0.8353566
boosting	cut 50	0	0.6666667	0.4979768	0.8353566
DT	cut 20	2	0.6666667	0.4979768	0.8353566
SVM	evolution	1	0.6666667	0.4979768	0.8353566
boosting	cut 2	0	0.6666667	0.4979768	0.8353566
bayes	cut 2	0	0.6666667	0.4979768	0.8353566
DT	cut 50	2	0.6333333	0.4608896	0.8057770
DT	cut 50	1	0.6333333	0.4608896	0.8057770
DT	cut 50	0	0.6333333	0.4608896	0.8057770
bagging	cut 20	2	0.6333333	0.4608896	0.8057770
bagging	cut 2	2	0.6333333	0.4608896	0.8057770
bagging	cut 2	0	0.6333333	0.4608896	0.8057770
DT	cut 10	1	0.6	0.4246923	0.7753077
DT	cut 20	1	0.6	0.4246923	0.7753077
DT	cut 20	0	0.6	0.4246923	0.7753077
bayes	cut 50	0	0.6	0.4246923	0.7753077
DT	cut 5	0	0.6	0.4246923	0.7753077
DT	evolution	1	0.6	0.4246923	0.7753077
DT	cut 2	1	0.6	0.4246923	0.7753077
DT	cut 10	0	0.6	0.4246923	0.7753077
DT	cut 2	0	0.6	0.4246923	0.7753077
DT	cut 5	1	0.6	0.4246923	0.7753077
DT	evolution	0	0.6	0.4246923	0.7753077
SVM	cut 2	0	0.5666667	0.3893416	0.7439918
SVM	cut 2	2	0.5666667	0.3893416	0.7439918
SVM	cut 2	1	0.5666667	0.3893416	0.7439918
bagging	cut 70	2	0.5333333	0.3548086	0.7118580
bayes	cut 70	0	0.5333333	0.3548086	0.7118580
boosting	cut 70	2	0.5	0.3210773	0.6789227
boosting	cut 70	0	0.5	0.3210773	0.6789227
bagging	cut 70	0	0.4333333	0.2560082	0.6106584
DT	cut 70	1	0.4333333	0.2560082	0.6106584
DT	cut 70	2	0.4333333	0.2560082	0.6106584
DT	cut 70	0	0.4333333	0.2560082	0.6106584
SVM	cut 70	0	0.4	0.2246923	0.5753077
SVM	cut 70	1	0.4	0.2246923	0.5753077
SVM	cut 70	2	0.4	0.2246923	0.5753077

2.1.4 Halt

Baseline is 0.7333333.

method	feature selection	PS	result	lower	upper
boosting	cut 5	2	0.8	0.6568618	0.9431382
boosting	cut 10	0	0.8	0.6568618	0.9431382
boosting	cut 2	2	0.8	0.6568618	0.9431382
boosting	cut 10	2	0.8	0.6568618	0.9431382
bagging	cut 50	2	0.8	0.6568618	0.9431382
boosting	cut 20	2	0.8	0.6568618	0.9431382
boosting	cut 20	0	0.8	0.6568618	0.9431382
boosting	cut 5	0	0.8	0.6568618	0.9431382
boosting	cut 2	0	0.8	0.6568618	0.9431382
bagging	cut 50	0	0.7666667	0.6153151	0.9180183
bagging	cut 2	0	0.7666667	0.6153151	0.9180183
bagging	cut 20	2	0.7666667	0.6153151	0.9180183
boosting	cut 70	2	0.7666667	0.6153151	0.9180183
bagging	cut 5	0	0.7666667	0.6153151	0.9180183
bagging	cut 10	0	0.7666667	0.6153151	0.9180183
bagging	cut 5	2	0.7666667	0.6153151	0.9180183
bagging	cut 2	2	0.7666667	0.6153151	0.9180183
boosting	cut 50	2	0.7666667	0.6153151	0.9180183
bayes	cut 50	0	0.7666667	0.6153151	0.9180183
boosting	cut 70	0	0.7666667	0.6153151	0.9180183
boosting	cut 50	0	0.7666667	0.6153151	0.9180183
SVM	cut 2	1	0.7333333	0.5750881	0.8915785
SVM	cut 50	0	0.7333333	0.5750881	0.8915785
DT	cut 10	2	0.7333333	0.5750881	0.8915785
SVM	cut 10	1	0.7333333	0.5750881	0.8915785
SVM	cut 70	2	0.7333333	0.5750881	0.8915785
SVM	evolution	0	0.7333333	0.5750881	0.8915785
SVM	cut 2	2	0.7333333	0.5750881	0.8915785
DT	cut 2	2	0.7333333	0.5750881	0.8915785
DT	cut 50	1	0.7333333	0.5750881	0.8915785
bagging	cut 20	0	0.7333333	0.5750881	0.8915785
bagging	cut 10	2	0.7333333	0.5750881	0.8915785
SVM	cut 70	1	0.7333333	0.5750881	0.8915785
SVM	cut 5	1	0.7333333	0.5750881	0.8915785
DT	cut 20	1	0.7333333	0.5750881	0.8915785
DT	cut 10	0	0.7333333	0.5750881	0.8915785
DT	cut 2	1	0.7333333	0.5750881	0.8915785
DT	cut 70	1	0.7333333	0.5750881	0.8915785
SVM	cut 20	2	0.7333333	0.5750881	0.8915785
DT	cut 5	0	0.7333333	0.5750881	0.8915785
SVM	cut 50	1	0.7333333	0.5750881	0.8915785
DT	cut 2	0	0.7333333	0.5750881	0.8915785
DT	evolution	0	0.7333333	0.5750881	0.8915785
SVM	cut 20	1	0.7333333	0.5750881	0.8915785
SVM	cut 2	0	0.7333333	0.5750881	0.8915785
DT	cut 5	1	0.7333333	0.5750881	0.8915785
SVM	cut 10	2	0.7333333	0.5750881	0.8915785
DT	evolution	1	0.7333333	0.5750881	0.8915785
SVM	cut 20	0	0.7333333	0.5750881	0.8915785
DT	cut 50	2	0.7333333	0.5750881	0.8915785
DT	cut 20	2	0.7333333	0.5750881	0.8915785
DT	cut 5	2	0.7333333	0.5750881	0.8915785
SVM	cut 5	0	0.7333333	0.5750881	0.8915785
SVM	evolution	1	0.7333333	0.5750881	0.8915785
bagging	cut 70	2	0.7333333	0.5750881	0.8915785
DT	evolution	2	0.7333333	0.5750881	0.8915785
SVM	cut 70	0	0.7333333	0.5750881	0.8915785
bagging	cut 70	0	0.7333333	0.5750881	0.8915785
DT	cut 20	0	0.7333333	0.5750881	0.8915785
SVM	cut 50	2	0.7333333	0.5750881	0.8915785
DT	cut 10	1	0.7333333	0.5750881	0.8915785
SVM	cut 10	0	0.7333333	0.5750881	0.8915785
SVM	cut 5	2	0.7333333	0.5750881	0.8915785
DT	cut 70	2	0.7333333	0.5750881	0.8915785
DT	cut 70	0	0.7333333	0.5750881	0.8915785
SVM	evolution	2	0.7333333	0.5750881	0.8915785
DT	cut 50	0	0.7333333	0.5750881	0.8915785
bayes	cut 2	0	0.7	0.5360146	0.8639854
bayes	cut 10	0	0.7	0.5360146	0.8639854
bayes	cut 5	0	0.7	0.5360146	0.8639854
bayes	evolution	0	0.7	0.5360146	0.8639854
bayes	cut 70	0	0.6666667	0.4979768	0.8353566
bayes	cut 20	0	0.6666667	0.4979768	0.8353566

2.1.5 Plough

Baseline is 0.3. (sic)

method	feature selection	PS	result	lower	upper
bagging	cut 5	0	0.7	0.5360146	0.8639854
boosting	cut 2	0	0.7	0.5360146	0.8639854
bagging	cut 5	2	0.7	0.5360146	0.8639854
boosting	cut 2	2	0.7	0.5360146	0.8639854
bagging	cut 10	0	0.7	0.5360146	0.8639854
boosting	cut 5	2	0.7	0.5360146	0.8639854
boosting	cut 5	0	0.7	0.5360146	0.8639854
boosting	cut 10	0	0.6666667	0.4979768	0.8353566
boosting	cut 10	2	0.6666667	0.4979768	0.8353566
bayes	cut 5	0	0.6666667	0.4979768	0.8353566
bagging	cut 2	2	0.6666667	0.4979768	0.8353566
bayes	evolution	0	0.6666667	0.4979768	0.8353566
bagging	cut 2	0	0.6666667	0.4979768	0.8353566
bayes	cut 10	0	0.6333333	0.4608896	0.8057770
bagging	cut 10	2	0.6333333	0.4608896	0.8057770
bayes	cut 2	0	0.6333333	0.4608896	0.8057770
bagging	cut 20	0	0.6	0.4246923	0.7753077
boosting	cut 20	0	0.6	0.4246923	0.7753077
boosting	cut 20	2	0.6	0.4246923	0.7753077
SVM	cut 10	0	0.5666667	0.3893416	0.7439918
DT	evolution	2	0.5666667	0.3893416	0.7439918
DT	cut 2	2	0.5666667	0.3893416	0.7439918
SVM	cut 20	1	0.5666667	0.3893416	0.7439918
SVM	cut 20	0	0.5666667	0.3893416	0.7439918
SVM	cut 10	2	0.5666667	0.3893416	0.7439918
SVM	cut 20	2	0.5666667	0.3893416	0.7439918
SVM	cut 10	1	0.5666667	0.3893416	0.7439918
DT	cut 20	2	0.5666667	0.3893416	0.7439918
DT	cut 5	2	0.5666667	0.3893416	0.7439918
DT	cut 2	1	0.5333333	0.3548086	0.7118580
DT	cut 10	2	0.5333333	0.3548086	0.7118580
DT	cut 20	0	0.5333333	0.3548086	0.7118580
DT	cut 10	0	0.5333333	0.3548086	0.7118580
bagging	cut 20	2	0.5333333	0.3548086	0.7118580
DT	cut 20	1	0.5333333	0.3548086	0.7118580
bayes	cut 20	0	0.5333333	0.3548086	0.7118580
DT	evolution	0	0.5333333	0.3548086	0.7118580
DT	cut 2	0	0.5333333	0.3548086	0.7118580
DT	cut 5	0	0.5333333	0.3548086	0.7118580
DT	cut 10	1	0.5333333	0.3548086	0.7118580
DT	evolution	1	0.5333333	0.3548086	0.7118580
DT	cut 5	1	0.5333333	0.3548086	0.7118580
boosting	cut 70	2	0.5	0.3210773	0.6789227
SVM	cut 5	1	0.5	0.3210773	0.6789227
SVM	cut 5	2	0.5	0.3210773	0.6789227
SVM	cut 5	0	0.5	0.3210773	0.6789227
boosting	cut 70	0	0.5	0.3210773	0.6789227
bayes	cut 50	0	0.4666667	0.2881420	0.6451914
SVM	evolution	2	0.4666667	0.2881420	0.6451914
SVM	evolution	1	0.4666667	0.2881420	0.6451914
SVM	evolution	0	0.4666667	0.2881420	0.6451914
bagging	cut 50	2	0.4333333	0.2560082	0.6106584
SVM	cut 2	2	0.4333333	0.2560082	0.6106584
bagging	cut 70	2	0.4333333	0.2560082	0.6106584
SVM	cut 2	0	0.4333333	0.2560082	0.6106584
bayes	cut 70	0	0.4333333	0.2560082	0.6106584
SVM	cut 2	1	0.4333333	0.2560082	0.6106584
DT	cut 50	0	0.4	0.2246923	0.5753077
SVM	cut 50	2	0.4	0.2246923	0.5753077
bagging	cut 70	0	0.4	0.2246923	0.5753077
bagging	cut 50	0	0.4	0.2246923	0.5753077
DT	cut 70	2	0.4	0.2246923	0.5753077
DT	cut 50	1	0.4	0.2246923	0.5753077
SVM	cut 70	0	0.4	0.2246923	0.5753077
SVM	cut 70	1	0.4	0.2246923	0.5753077
SVM	cut 50	1	0.4	0.2246923	0.5753077
SVM	cut 50	0	0.4	0.2246923	0.5753077
boosting	cut 50	0	0.4	0.2246923	0.5753077
SVM	cut 70	2	0.4	0.2246923	0.5753077
boosting	cut 50	2	0.4	0.2246923	0.5753077
DT	cut 70	0	0.3666667	0.1942230	0.5391104
DT	cut 50	2	0.3666667	0.1942230	0.5391104
DT	cut 70	1	0.3666667	0.1942230	0.5391104

2.1.6 Submit

Baseline is 0.6666667.

method	feature selection	PS	result	lower	upper
DT	cut 10	0	0.8	0.6568618	0.9431382
bagging	cut 10	0	0.8	0.6568618	0.9431382
bagging	cut 20	0	0.8	0.6568618	0.9431382
DT	cut 50	1	0.8	0.6568618	0.9431382
DT	cut 50	0	0.8	0.6568618	0.9431382
DT	cut 20	2	0.8	0.6568618	0.9431382
bagging	cut 50	0	0.8	0.6568618	0.9431382
bagging	cut 5	0	0.8	0.6568618	0.9431382
DT	cut 20	1	0.8	0.6568618	0.9431382
bagging	cut 2	2	0.8	0.6568618	0.9431382
bagging	cut 20	2	0.8	0.6568618	0.9431382
DT	evolution	1	0.8	0.6568618	0.9431382
DT	cut 2	1	0.8	0.6568618	0.9431382
DT	evolution	0	0.8	0.6568618	0.9431382
bagging	cut 50	2	0.8	0.6568618	0.9431382
DT	cut 10	2	0.8	0.6568618	0.9431382
bagging	cut 10	2	0.8	0.6568618	0.9431382
bagging	cut 2	0	0.8	0.6568618	0.9431382
DT	cut 50	2	0.8	0.6568618	0.9431382
DT	cut 5	1	0.8	0.6568618	0.9431382
DT	cut 20	0	0.8	0.6568618	0.9431382
DT	cut 2	0	0.8	0.6568618	0.9431382
DT	cut 10	1	0.8	0.6568618	0.9431382
DT	cut 5	0	0.8	0.6568618	0.9431382
boosting	cut 5	0	0.7666667	0.6153151	0.9180183
SVM	cut 5	1	0.7666667	0.6153151	0.9180183
DT	cut 70	2	0.7666667	0.6153151	0.9180183
DT	cut 70	0	0.7666667	0.6153151	0.9180183
bagging	cut 70	0	0.7666667	0.6153151	0.9180183
SVM	cut 50	1	0.7666667	0.6153151	0.9180183
DT	cut 5	2	0.7666667	0.6153151	0.9180183
boosting	cut 10	2	0.7666667	0.6153151	0.9180183
DT	cut 2	2	0.7666667	0.6153151	0.9180183
boosting	cut 2	0	0.7666667	0.6153151	0.9180183
SVM	evolution	2	0.7666667	0.6153151	0.9180183
DT	cut 70	1	0.7666667	0.6153151	0.9180183
SVM	cut 2	1	0.7666667	0.6153151	0.9180183
SVM	evolution	0	0.7666667	0.6153151	0.9180183
boosting	cut 2	2	0.7666667	0.6153151	0.9180183
SVM	cut 50	0	0.7666667	0.6153151	0.9180183
SVM	cut 50	2	0.7666667	0.6153151	0.9180183
bagging	cut 5	2	0.7666667	0.6153151	0.9180183
boosting	cut 10	0	0.7666667	0.6153151	0.9180183
boosting	cut 20	2	0.7666667	0.6153151	0.9180183
SVM	cut 2	2	0.7666667	0.6153151	0.9180183
SVM	evolution	1	0.7666667	0.6153151	0.9180183
DT	evolution	2	0.7666667	0.6153151	0.9180183
SVM	cut 5	2	0.7666667	0.6153151	0.9180183
boosting	cut 5	2	0.7666667	0.6153151	0.9180183
boosting	cut 20	0	0.7666667	0.6153151	0.9180183
SVM	cut 2	0	0.7666667	0.6153151	0.9180183
SVM	cut 5	0	0.7666667	0.6153151	0.9180183
bayes	cut 2	0	0.7333333	0.5750881	0.8915785
SVM	cut 10	1	0.7333333	0.5750881	0.8915785
SVM	cut 20	2	0.7333333	0.5750881	0.8915785
SVM	cut 20	1	0.7333333	0.5750881	0.8915785
SVM	cut 10	2	0.7333333	0.5750881	0.8915785
bayes	evolution	0	0.7333333	0.5750881	0.8915785
SVM	cut 20	0	0.7333333	0.5750881	0.8915785
bagging	cut 70	2	0.7333333	0.5750881	0.8915785
SVM	cut 10	0	0.7333333	0.5750881	0.8915785
SVM	cut 70	1	0.7	0.5360146	0.8639854
SVM	cut 70	2	0.7	0.5360146	0.8639854
bayes	cut 70	0	0.7	0.5360146	0.8639854
boosting	cut 50	0	0.7	0.5360146	0.8639854
boosting	cut 70	0	0.7	0.5360146	0.8639854
boosting	cut 50	2	0.7	0.5360146	0.8639854
boosting	cut 70	2	0.7	0.5360146	0.8639854
SVM	cut 70	0	0.7	0.5360146	0.8639854
bayes	cut 20	0	0.6666667	0.4979768	0.8353566
bayes	cut 5	0	0.6666667	0.4979768	0.8353566
bayes	cut 10	0	0.6666667	0.4979768	0.8353566
bayes	cut 50	0	0.5333333	0.3548086	0.7118580

2.2 Results 2

Since these results are 10-fold cross validation, I am not sure how to cound the confidence intervals in these; the baseline is counted for every one of the 10 trials in 10-fold validations separately and then averaged as in 10-fold cross validation.

The model I chose for final handing in is in italics.

2.2.1 Ally

Baseline is 0.476.

Note: this is the only model that, in final form, does use custom parameters.

method	**feature selection**	**PS**	**result**
bagging	*cut 20*	*2*	*0.652*
boosting	cut 10	0	0.648
bagging	cut 20	0	0.644
SVM	evolution	2	0.644
bagging	cut 2	0	0.644
bayes	cut 2	0	0.632
boosting	cut 10	2	0.632
boosting	cut 5	0	0.632
SVM	evolution	1	0.632
SVM	cut 2	2	0.632
bagging	cut 10	0	0.632
SVM	evolution	0	0.632
bayes	evolution	0	0.628
bagging	cut 2	2	0.62
bagging	cut 5	2	0.62
SVM	cut 2	0	0.604
boosting	cut 5	2	0.604
SVM	cut 2	1	0.604
bagging	cut 70	0	0.6
bagging	cut 10	2	0.6

2.2.2 Arrive

Baseline is 0.68.

method	**feature selection**	**PS**	**result**
boosting	*cut 50*	*0*	*0.716*
SVM	cut 70	0	0.712
SVM	cut 50	2	0.712
SVM	cut 70	1	0.712
boosting	cut 70	0	0.704
bagging	cut 70	2	0.704
bagging	cut 70	0	0.704
bagging	cut 50	0	0.704
SVM	cut 70	2	0.704
bagging	cut 50	2	0.7
SVM	cut 50	0	0.696
SVM	cut 50	1	0.696
DT	cut 70	2	0.688
DT	cut 50	2	0.684
bayes	cut 70	0	0.676
boosting	cut 70	2	0.676
bayes	cut 50	0	0.66
boosting	cut 5	2	0.64
DT	cut 20	2	0.64
boosting	cut 50	2	0.6

2.2.3 Cry

Baseline is 0.524.

method	feature selection	PS	result
bayes	cut 5	0	0.676
SVM	cut 50	2	0.676
bayes	evolution	0	0.676
boosting	*cut 5*	*0*	*0.668*
boosting	cut 20	2	0.668
bayes	cut 10	0	0.664
boosting	cut 10	0	0.66
boosting	cut 20	0	0.66
boosting	cut 10	2	0.66
SVM	cut 5	0	0.652
SVM	cut 5	1	0.652
SVM	cut 50	1	0.632
SVM	cut 50	0	0.632
SVM	cut 5	2	0.632
DT	evolution	2	0.624
DT	cut 2	2	0.624
DT	cut 5	2	0.624
boosting	cut 5	2	0.624
bagging	cut 50	0	0.604
bagging	cut 50	2	0.596

2.2.4 Halt

Baseline is 0.836.

method	**feature selection**	**PS**	**result**
bagging	*cut 2*	*0*	*0.844*
bagging	cut 2	2	0.844
bagging	cut 5	0	0.84
bagging	cut 5	2	0.84
bagging	cut 20	2	0.836
boosting	cut 70	0	0.832
boosting	cut 50	2	0.832
boosting	cut 20	0	0.828
boosting	cut 2	2	0.828
boosting	cut 10	0	0.828
boosting	cut 70	2	0.828
bagging	cut 50	2	0.828
boosting	cut 5	0	0.824
bagging	cut 50	0	0.824
boosting	cut 20	2	0.824
boosting	cut 10	2	0.816
boosting	cut 2	0	0.816
boosting	cut 50	0	0.796
bayes	cut 50	0	0.704
boosting	cut 5	2	0.604

2.2.5 Plough

Baseline is 0.324.

method	**feature selection**	**PS**	**result**
boosting	*cut 5*	*2*	*0.844*
bagging	cut 5	2	0.576
bagging	cut 10	0	0.568
boosting	cut 2	2	0.568
boosting	cut 10	2	0.568
boosting	cut 5	0	0.568
bagging	cut 5	0	0.568
bagging	cut 2	2	0.568
boosting	cut 2	0	0.564
bagging	cut 2	0	0.56
boosting	cut 10	0	0.56
boosting	cut 20	0	0.556
bagging	cut 10	2	0.548
boosting	cut 20	2	0.548
bagging	cut 20	0	0.54
SVM	cut 20	2	0.54
bayes	cut 2	0	0.536
bayes	evolution	0	0.536
bayes	cut 5	0	0.528
bayes	cut 10	0	0.524

2.2.6 Submit

Baseline is 0.708.

method	**feature selection**	**PS**	**result**
bagging	*cut 2*	*0*	*0.864*
bagging	cut 10	0	0.864
bagging	cut 5	0	0.864
bagging	cut 20	0	0.864
bagging	cut 2	2	0.86
bagging	cut 10	2	0.86
DT	cut 2	1	0.856
DT	evolution	0	0.856
DT	cut 5	0	0.856
DT	cut 20	1	0.856
DT	evolution	1	0.856
DT	cut 5	1	0.856
DT	cut 20	0	0.856
DT	cut 20	2	0.84
bagging	cut 20	2	0.84
bagging	cut 50	2	0.832
bagging	cut 50	0	0.828
DT	cut 50	1	0.824
DT	cut 50	0	0.824
DT	cut 50	2	0.816

3 Description of the files

3.1 Final models

All the final models are in the directory `hotovo`. To run the model, you need to run something like following in bash:

```
perl do.pl ../data/development.instances/ally.txt ../data/test.instances/ally.txt ally
```

Where the first argument is address to the train file, the second is address to test file, the thirs is the verb.

The script runs `transform.pl` to transform the instances into a table, readable by R. Then it runs the `train_and_test` file, which trains the model and test it on the example data, writing out the result of the test.

3.2 Other files

In `current_results`, there are intermediate results that are needed for the training, but can be deleted at any time when no intermediate models are trained.

In `results` directory, there are results that are needed for comparison of models, and also a particular feature sets/tuned parameters.

Scripts `baseline.pl` and `baseline.R` show a baseline (for Results 1) for any given verb.

Script `complete.pl` does all the possible combinations of verb/parameter search/feature search.

`evoluce.pl` and `evolution_try.R` are for evolutionary algorithm.

`jednoduchy_try.R` is for simple trying of model with no parameter search.

`muj_grid_search.R` and `muj_grid_search` are implementing my own grid searcg of parameters.

`napis_secres.pl` is showing results of the second, 10-fold crossover experiments in a somehow nice way; `napis_vysledky.pl` is showing results of the first experiments in a somehow nice way. The same files with `_latex` are exporting it in a latex table.

`report.latex` is this report in latex.

`second_baseline.pl` and `.R` is showing baseline of the second set of experiments.

`second_evaluation.pl` and `.R` are doing the second sets of experiments.

`shared.R` is a file with everything "important" – all the possible models are defined there and are just called from other `.R` files via a procedure, called `try`.

`transform.pl` is doing the transformation of the data file to R matrix.