2IMM20 - Foundations of datamining

Assignment 2

Students:

Joris van der Heijden	(0937329)
Bram van der Pol	(0780042)

Email addresseses:
j.j.m.v.d.heijden@student.tue.nl
a.f.v.d.pol@student.tue.nl

Supervisors:
Dr.ir. Joaquin Vanschoren

Eindhoven, March 16, 2018

Contents

1 A data mining challenge

This sections shows the steps that are done to improve the model for the datamining challange. This is done in a small report to show the different steps with the generated figures.

1.1 Detects accents in speech data

First three previously used models are fitted on the data to check which model works best in this case.

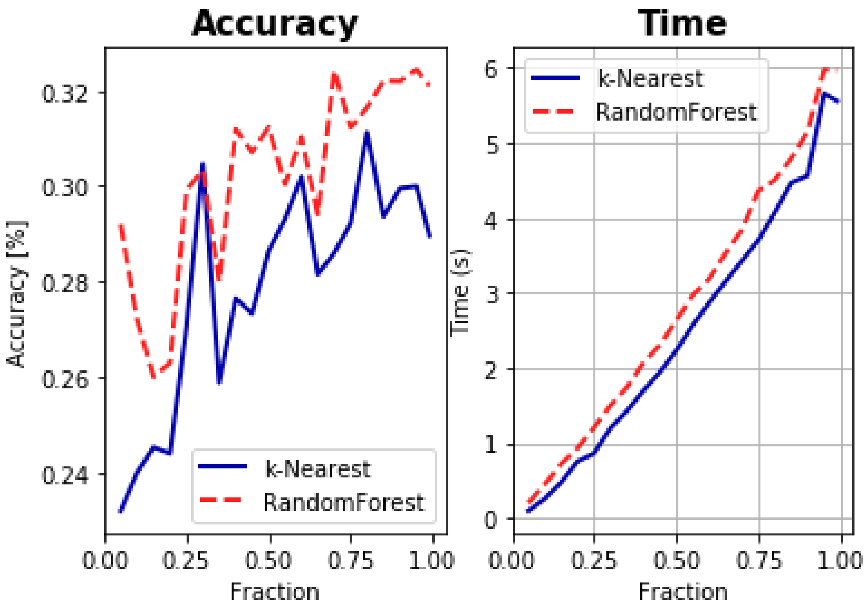


Figure 1: *Performance of the k-Nearest and Random forest algorithms.*

The RandomForest classifier is chosen because this gives the best accuracy with the default parameters. The computing time for both algorithms is almost equal for bot algorithms. The logistic regression (not shown) has a large computation time with low performance, so therefore this model is not

chosen.
In order to save computation time chosen is to start with a coarse grid and make this mesh finer and finer to come to the optimal solution. The grid search is done for only 1% of the data to save computing time. After the optimal solution is found the model is fitted on the entire data set using the obtained hyperparameters. The coarse grid search for 1% if the data is shown is figure **??**.

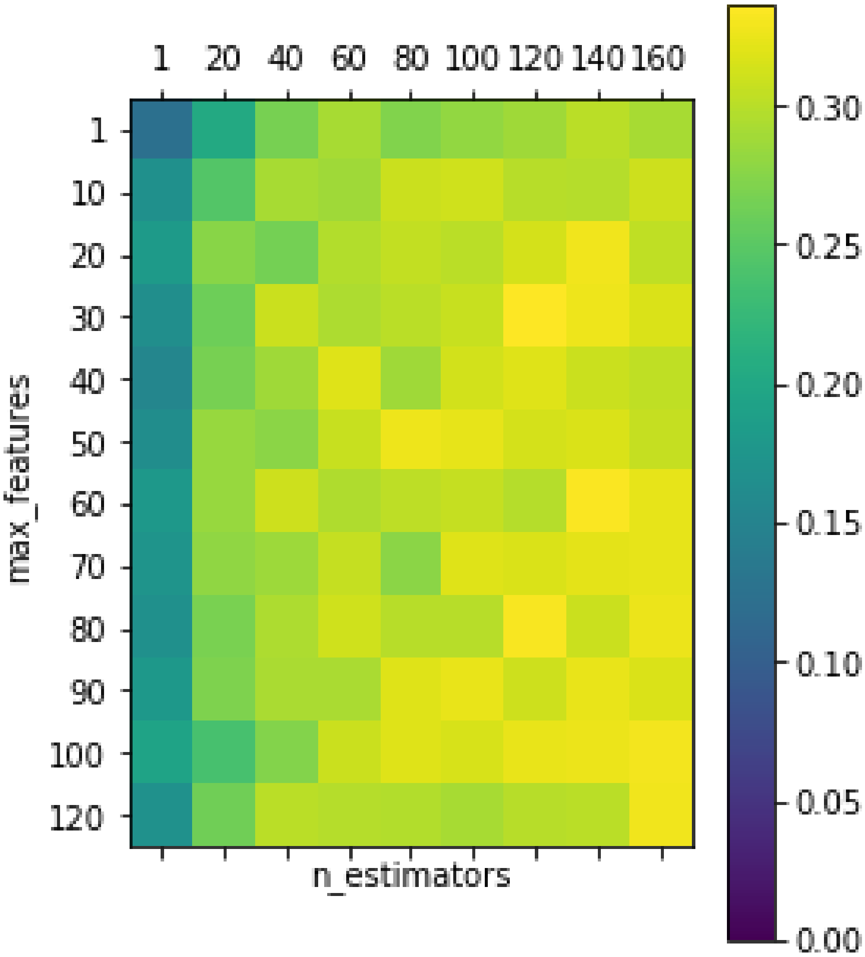


Figure 2: *A coarse grid search for 1% of the data with the Random Forest classifier. The optimal parameters are:* $n_{estimators} = 120$ *and* $max features = 30$

Now we ran a finer grid search near the domain $n_{estimators} = 120$ and $max features = 30$. The results is shown in figure 3

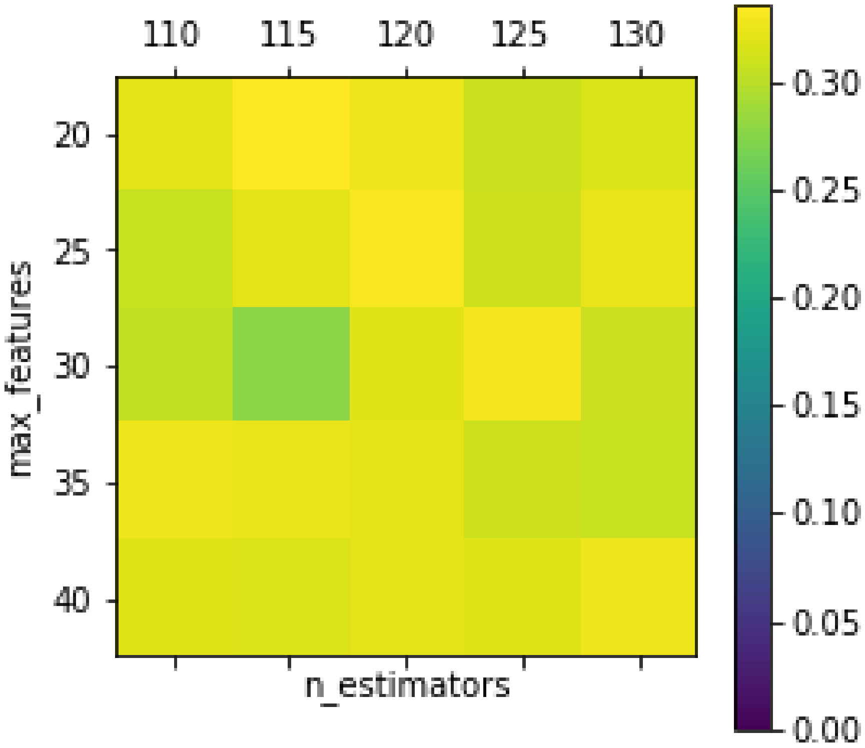
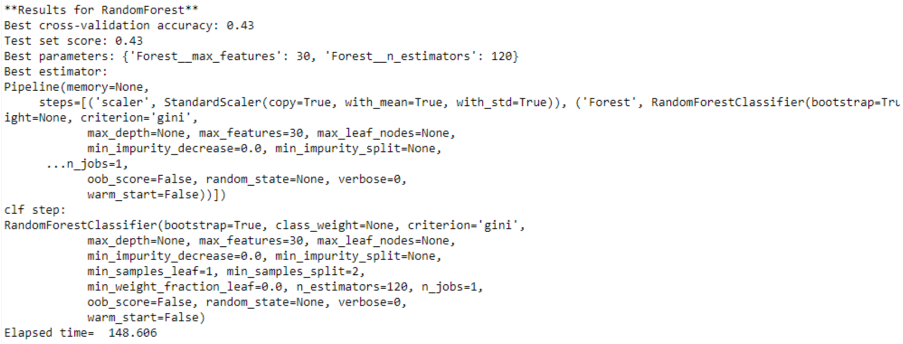


Figure 3: *A coarse grid search for 1% of the data with the Random Forest classifier. The optimal parameters are:* $n_{estimators} = 115$ *and* $max features = 30$*. This gave a score of 0.34 (for 1% of the data)*

Now we ran the model with the obtained parameters and achieved a score shown in figure 4.



```
**Results for RandomForest**
Best cross-validation accuracy: 0.43
Test set score: 0.43
Best parameters: {'Forest__max_features': 30, 'Forest__n_estimators': 120}
Best estimator:
Pipeline(memory=None,
     steps=[('scaler', StandardScaler(copy=True, with_mean=True, with_std=True)), ('Forest', RandomForestClassifier(bootstrap=Tru
ight=None, criterion='gini',
            max_depth=None, max_features=30, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
      ...n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False))])
clf step:
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=None, max_features=30, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=120, n_jobs=1,
            oob_score=False, random_state=None, verbose=0,
            warm_start=False)
Elapsed time=  148.606
```

Figure 4: *RandomForest score for:* $n_{estimators} = 115$ *and* $maxfeatures = 30$. *This gave a score of 0.43 for 100% of the data*

In order to improve the results, the leaderboard was checked to look for the best algorithm. This was the $Keras$ classifier and therefore decided was to use this neural network API. Unfortunately this model took too much time to run, and therefore the optimization of the Keras algorithm did not have the desired outcome.