Machine Learning project report

Testing models on Twitter dataset

Katarzyna P, Julia N, Jakub M

Wroclaw, 01.30.2020

1. Objective

Checking performance of Naive-Bayes Classifier, Decision Trees, Logistic regression, Support vector machine and K-nearest neighbours in context of sentiment prediction on airline sentiment dataset. We wanted to compare their accuracies and behaviors on different variants of these models.

2. Dataset

Our dataset consists of tweets and target labels.

Chart 1.: Distribution of labels in a dataset

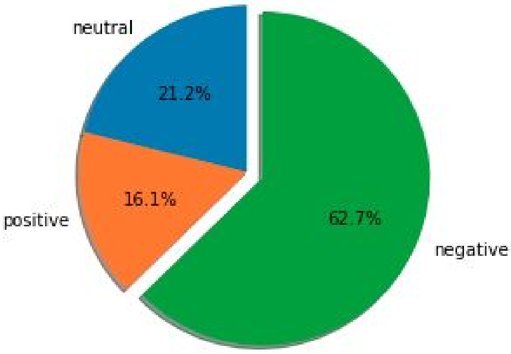
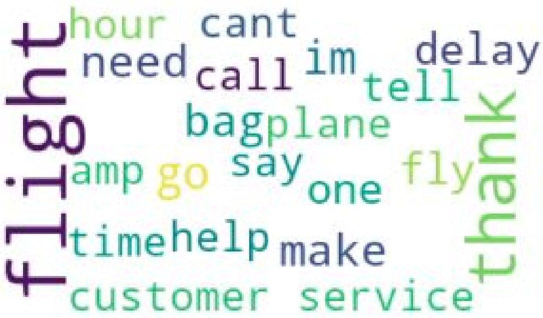


Chart 2.: The 20 most common words in dataset



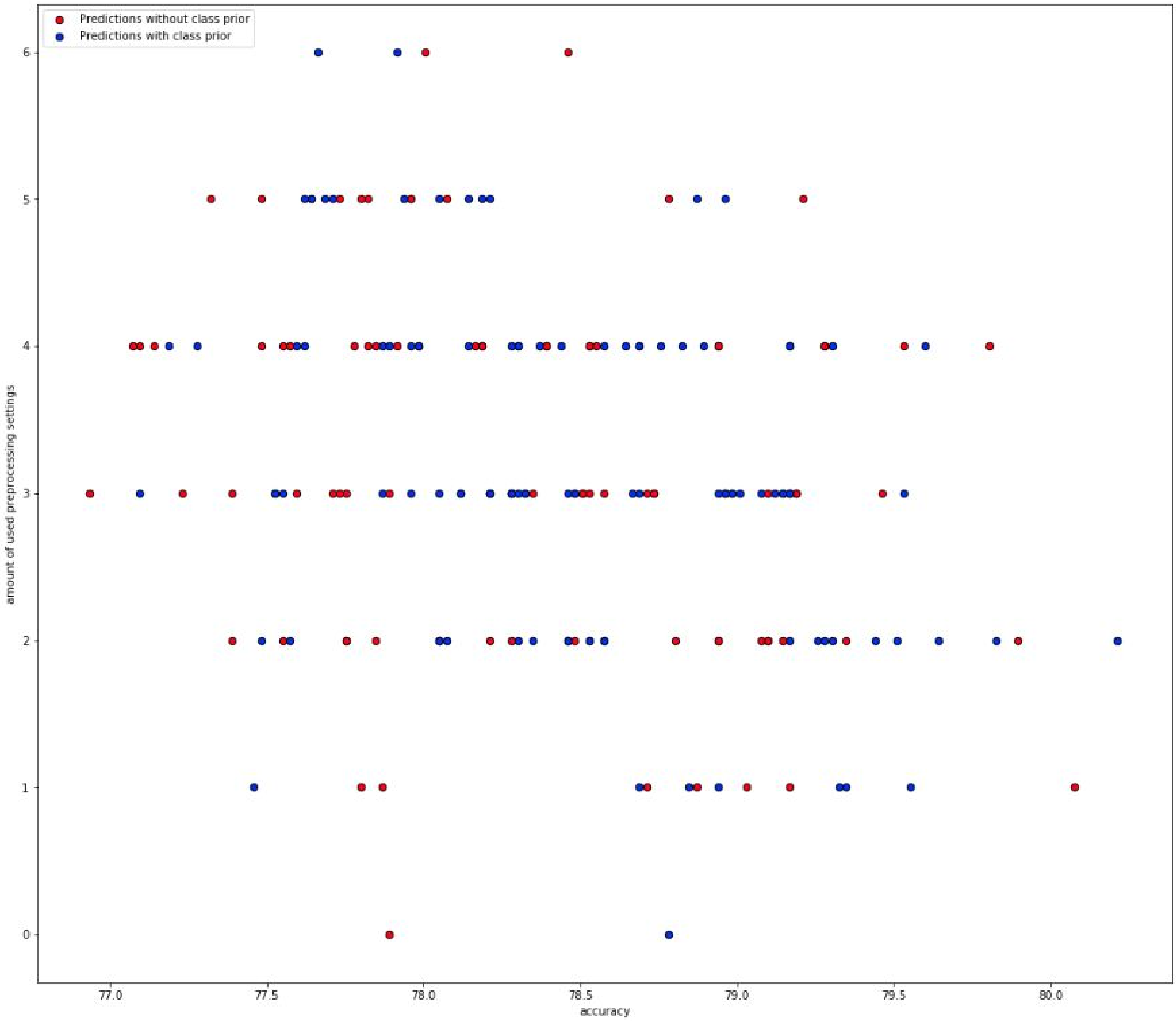
3. Results

We took implementations from sklearn library, preprocess data and test different combinations of parameters and preprocessing.

3.1. Naive-Bayes Classifier

First, we had made data preprocessing using all possible combination of our preprocessing parameters (changing emojis into text, removing punctuations, removing links, removing stopwords, removing negations words, using porte lemmatizer or lancaster lemmatizer). This gives us $2^7 - 2^5 = 96$ different combinations (because we never use two lemmatizer at the same time). For each preprocessed data we calculated accuracy of Naive-Bayes with and without class prior. We had calculated class prior by counted all targets in training set and divided it by amount of all targets in training set. (Best 20 Naive-Bayes Classifiers are shown on “NB_chart1.jpg” chart).

Chart 3.: Naive-Bayes Classifiers accuracy (x axis) depends of amount of used preprocessing settings (y axis). Blue dots represents classifiers with class prior, red dots classifiers without class prior.



Then that, we took preprocessing parameters for best classifier and used it to train classifier using N-grams parameter (equals 2, 3, 4 or 5, shown on “NB_chart2.jpg” chart). Next we took parameters for best classifier with and without class prior and change “artificial data duplication”, which means that we had copied rows from traint dataset with rarer sentiment while amount of sentiments in train dataset was the same. We compare accuracies of classifiers with and without “artificial data duplication” (shown on “NB_chart3.jpg” chart).

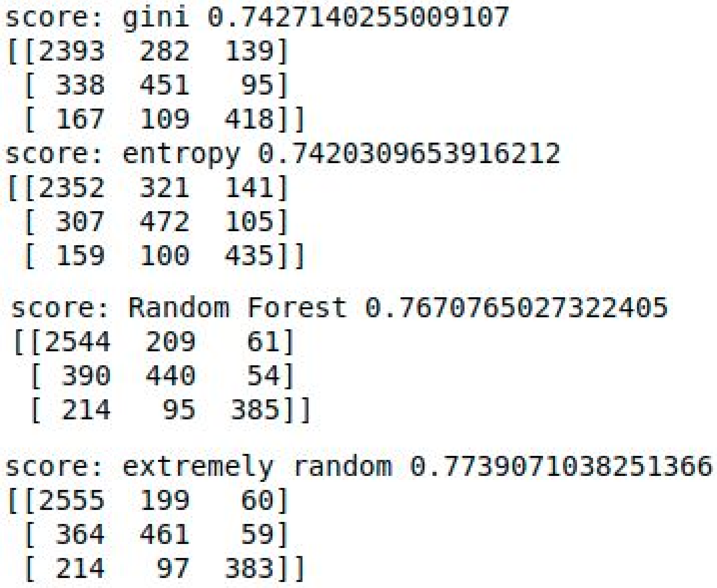
Results: best accuracy was achieved by using Naives-Bayes with class prior, with N_grams equalled 2, without “artificial data duplication” and data preprocessing using removing links and changing emojis into text. This accuracy was equalled **80.6466302368%**.

3.2. Decision Trees

Firstly, we had made data preprocessing using different preprocessing parameters (removing emojis, removing handle, removing punctuations, using lemmatizer, removing stop words and removing links and numbers). For each preprocessed data we calculated accuracy of Decision Tree Classifier with default parameters (shown on “DT_chart1.jpg” chart).

Next, we had tested different variants of Decision Trees algorithm (using Gini, Entropy and using Random Forest Classifier and Extremely Random Trees algorithms) on data preprocessed by best preprocessing from previous tests. We took accuracies and confusion matrices for all of these variants.

Chart 4.: Accuracies and confusion matrices of Decision Trees variants



```
score: gini 0.7427140255009107
[[2393  282  139]
 [ 338  451   95]
 [ 167  109  418]]
score: entropy 0.7420309653916212
[[2352  321  141]
 [ 307  472  105]
 [ 159  100  435]]

score: Random Forest 0.7670765027322405
[[2544  209   61]
 [ 390  440   54]
 [ 214   95  385]]

score: extremely random 0.7739071038251366
[[2555  199   60]
 [ 364  461   59]
 [ 214   97  383]]
```

Then we had tried to use “artificial data duplication” on preprocessed data and train Extremely Random Trees on this data (because Extremely Random Trees had best accuracy in previous test, results of this test are show on “DT_chart2.jpg” chart).

Results: best accuracy was achieved by using Extremely Random Trees and data preprocessing using removing handles. Accuracies was the same with and without “artificial data duplication”. This accuracy was equalled **77.41347905282332%**.

3.3. Logistic Regression

Firstly, we had made data preprocessing using different preprocessing parameters (removing emojis, removing handle, removing punctuations, using lemmatizer, removing stop words and removing links and numbers). For each preprocessed data we calculated accuracy of Logistic Regression with default parameters.

Then, we had tried to use different values for C parameter (from 0.01 to 1.0) and train Logistic Regression Classifier with this parameter on on data preprocessed by best preprocessing from previous tests (confusion matrix for best C is shown on “LR_chart1.jpg” chart).

Chart 5.: Accuracy for different preprocessing (Logistic Regression)

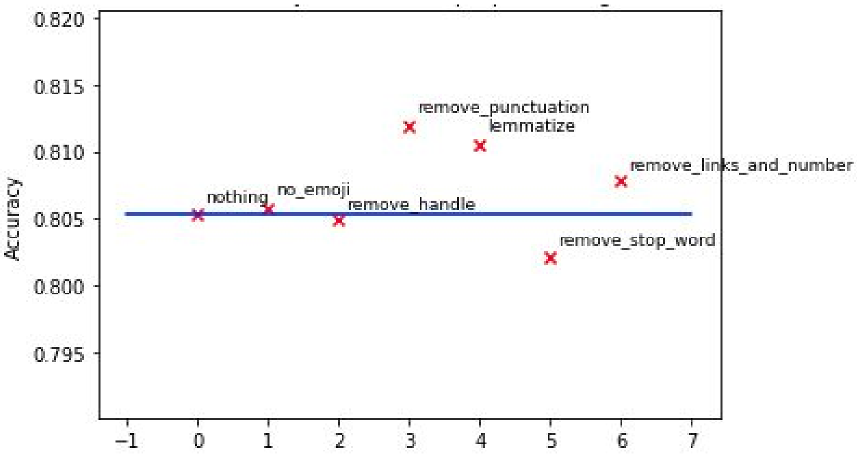
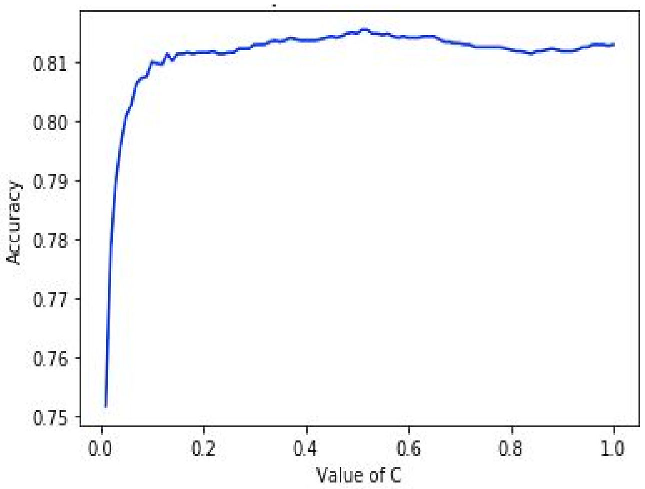


Chart 6.: Accuracy for different values of C (Logistic Regression)



Next, we had tried to use “artificial data duplication” on preprocessed data and train Logistic Regression with best C parameter on this data (confusion matrix of this test is shown on “LR_chart2.jpg” chart).

Results: best accuracy was achieved by using Logistic Regression without “artificial data duplication”, C parameter equalled to 0.51 and data preprocessing using removing punctuations. This accuracy was equalled **81.53460837887068%**.

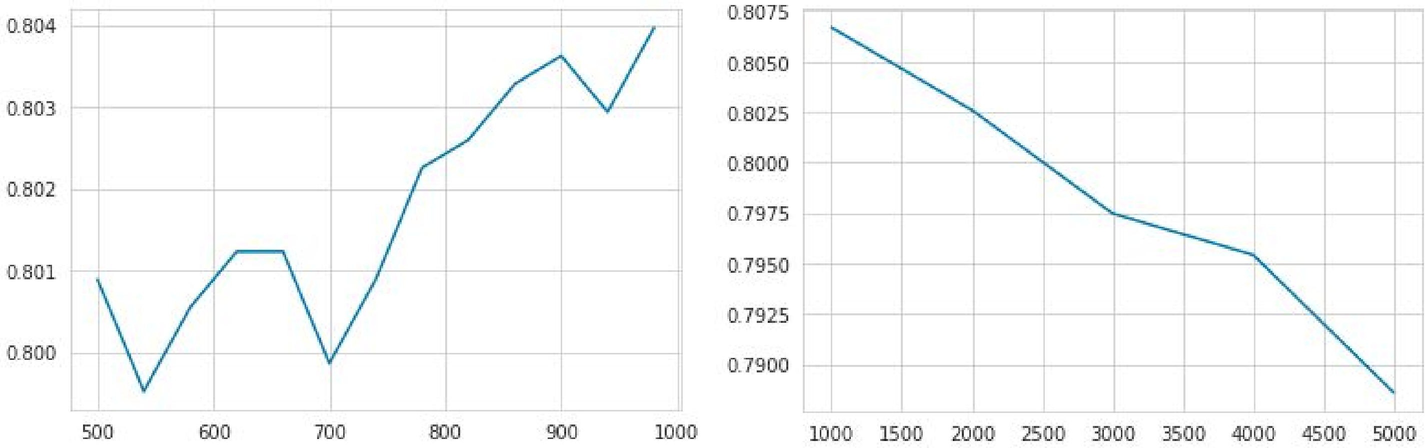
3.4. Support vector machine

Firstly, we trained Support vector machines classifier with default parameters on data without any preprocessing. This classifier had 63.79% accuracy, but it was predicted everything as 'negative'. Then we had tried to use “artificial data duplication”, which lead to overfitting.

Next, we tested different values of C parameter. After that, we tested SVM classifier on preprocessed data (removing punctuations and numbers) using best C parameter from previous test and with ‘scale’ and ‘auto’ gamma parameter values. Accuracy for ‘scale’ gamma was equalled 79.91%, and for ‘auto’ gamma was equalled 80.84%.

Then we had repeated all of previous tests using Linear Kernel, Polynomial Kernel and Sigmoid Kernel, additionally using different preprocessing parameters (we had added tokenizations to preprocessing and test all combinations of all three parameters).

Chart 6 and 7.: Default SVM accuracy (y axis) for different C parameter values (x axis).

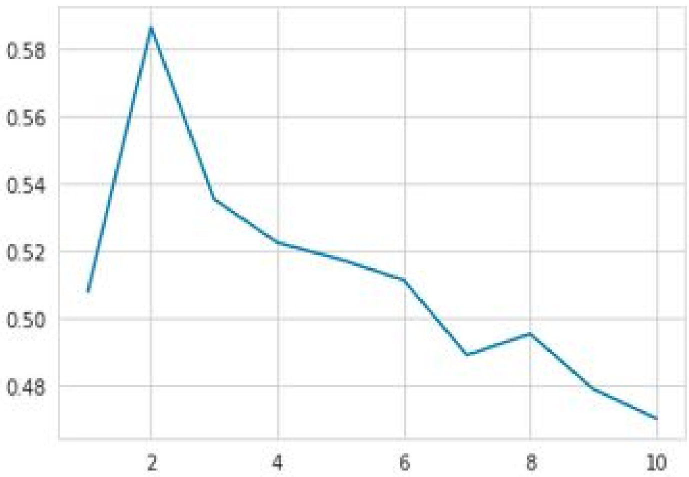


Results: best accuracy was achieved by using SVM with sigmoid kernel, C parameter equalled to 2000, gamma parameter equalled to ‘auto’, and data preprocessing using tokenization, removing punctuations and removing numbers. This accuracy was equal to **81.625%**.

3.5. K-nearest neighbours

We tested KNN classifier on data without any preprocessing and using different K parameter values. Then we made this again, but this time using “artificial data duplication”, which returned accuracy near 80%, which was overfitting and still worse than SVM.

Chart 8.: KNN accuracy for different K parameter values



After that preprocessed data (removed punctuations and numbers) and using best k from previous tests. Then we had tried to made preprocessing only on training data, reduced number of features and reduced dimensions with CountVectorizer.

Results: best accuracy was achieved by using KNN simple knn with 2 neighbors and dimensionality reduction which was based on removing most and least frequent words using CountVectorizer. This accuracy was equal 0.6369.

4. Final results

Ranking of accuracies achieved by algorithms:

1. Support vector machine - 81.6256%
2. Logistic regression - 81.5346%
3. Naive-Bayes classifier - 80.6466%
4. Decision trees - 77.4134%
5. K-nearest neighbours - 63.6953%