Machine Learning | **Universität Hamburg**
Spring 2024 | Sören Laue

Exercise Sheet 7

Exercise 1

Your task is to (manually) create a Naive Bayes classifier that distinguishes between positive and negative movie reviews. Your training data consists of the following reviews (already split into individual tokens):

Positiv	**Negativ**
a, great, time	one, out, of, five
a, great, film	a, terrible, movie
a, great, movie	boring, film
amazing, movie	not, great
not, bad	not, great, time

Use Laplacian Smoothing with value 1. Now please classify the following reviews in your test data:

1. not, great, movie
2. a, boring, film

What do you notice about the results? Give a short guess why the results turned out like this.
Now assume you would try to use a bigram classifier to enhance the results. A bigram classifier uses two neighbouring words as units. This is the training data split in to bigrams. # denotes an artificial delimiter which allows all words to occur in exactly two bigrams.

Positiv	**Negativ**
# a, a great, great time, time #	# one, one out, out of, of five, five #
# a, a great, great film, film #	# a, a terrible, terrible movie, movie #
# a, a great, great movie, movie #	# boring, boring film, film #
# amazing, amazing movie, movie #	# not, not great, great #
# not, not bad, bad #	# not, not great, great time, time #

Use the same two test reviews and first split them into bigrams. Then do the classification again using your bigram classifier. Did this solve the problems from the first try? Which new problems do you see? What would happen if you would use arbitrarily large n-grams?

Exercise 2

In some cases, one wants to include *prior knowledge* into your classifier. This kind of knowledge is taken from experts or literature, and is particularly helpful if it concerns rare features, problems with sparse training data or knowledge which is not directly reflected in the data. Can you come up with an example for such prior knowledge? How would you include prior knowledge in a base classifier or in a decision tree? Why is this challenging in practice?

Exercise 3

Apply a Naive Base classifierto two classification problems. We provide you the datasets `20newsgroups.csv` and `spam_or_not_spam.csv` . You can use `scikit-learn` for this exercise.
For both datasets, use the methods of `scikit-learn` to split the data into training and test date (recomondation: 80% training). You can also find methods to turn the text into a vector representaion using the so-called TF-IDF measure. TF-IDF helps to identify words that are as distinctive as possible for the respective categories.
`spam_or_not_spam.csv` contains e-mails that should be classified in two categories (spam and not spam). `20newsgroups.csv` contains newsgroup texts that should be classified in 20 categories. Your classifier thus should work for an arbitrary number of labels.
Please provide Accuracy, Precision, Recall and F1 for both classification tasks.
We provide you an additional test dataset `test_dataset.csv` which you should be able to classifiy with an accuracy of about 0.75.

Exercise 4

Apply both a Decision Tree classifier and a Random Forest classifier to the classification problem we provide you with the dataset `adult_sourcedata.csv` . The datapoints contain features of an individual person, and the labels indicate whether or not the person earns more than 50.000$ Use the OneHotEncoder of `scikit-learn` to preprocess the data (what does it do?). Then apply both Decision Trees and Random Forests, and test which value for the maximum depth of the tree(s) (`max_depth`) between 1 and 20 works best. Visualize the evaluation as graph with maximum depth on one axis and accuracy on the other one. What happens with large values for the maximum depth? Try the algorithm without specifying `max_depth` and name the 10 most important features according to gini importance.
Hint: Cross-validation might take a while for Random Forest. If it takes longer than 10 minutes, reduce the number of folds or the number of trees (not less than 10).

Please turn in your solutions by Thursday, June 6th.