**Document classification**

**Task 1 – Spam filtering (optional)**

Try to manually prepare a set of rules that might be used to filter spam. Work with the data from the CSDMC2010\_SPAM archive, directory TRAINING (file “5 – spam). The file contains e-mail examples marked as spam / no spam, on each line one e-mail. When creating the rules you might rely on your experience or you might examine a few e-mails from the archive. Are there any preprocessing steps that are typical for the e-mail domain and not needed for another domain? Use the labeled e-mails to test your rules. Quantify the classification error. How to decrease it? Improve your rules and try them again.

From the same archive, choose first 10 e-mails. Use the provided labeling and train a classifier (Naive Bayes) in Weka. Use the rest of the data for testing. How do you assess the classification performance measures?

Add more data (the following 10 e-mails) to the training set. See how the classification results changed.

Change the label of the e-mail number 2 (Wanna see sexually curious teens playing with each other?...). Repeat the process of building a classifier and testing it.

Change also the label of the e-mail number 4 (Enhance your desire, pleasure and performance!...). Repeat the process of building a classifier and testing it.

See how the classifier performance changes.

**Task 2 – Classification of customer reviews**

The file with customer reviews from amazon.com (file “6 - reviews\_amazon”) consists of real reviews of cellphones. Each line contains a number of stars awarded by a customer and the review text. Perform and comment the results of the following tasks:

* Consider the numbers of stars to be class labels. We want to predict the number of stars for a new review. Create a dataset containing 1000 instances from each class (randomly select them). Convert the data to the ARFF format to be further processed by Weka. Train a classifier (Naive Bayes, Multinomial Naive Bayes, J48, SMO). Use percentage split (i.e., a part of the data will be used for training , the rest for testing) with the default percentage value to test the classifiers. What can you say about the results? Is the accuracy satisfactory? Compare it to the results of the ZeroR classifier. Look at the confusion matrix and detailed classification performance measures values by class.
* Transform 2 stars to 1 star (these reviews are negative), 4 stars to 5 stars (these reviews are positive), ignore reviews with 3 stars. Repeat the previous task with this data (you will have 2000 instances from each class). Look at the results.
* Take first 5000 reviews from the file and transform 2 stars to 1 star, 4 stars to 5 stars, ignore reviews with 3 stars. The classes in the training set will have different sizes. Train a classifier (NB, J48, SMO) and test it on a distinct test set consisting of 2000 instances from each class. Look at the detailed analysis of classifiers' performance.
* Create a training set similarly as in the previous step but balance the number of instances from each class (add additional instances of the minority class); test the classifier on the same distinct test set as in the previous step. See, how the results changed.