# DSS UE1 Part1: Experiment Description

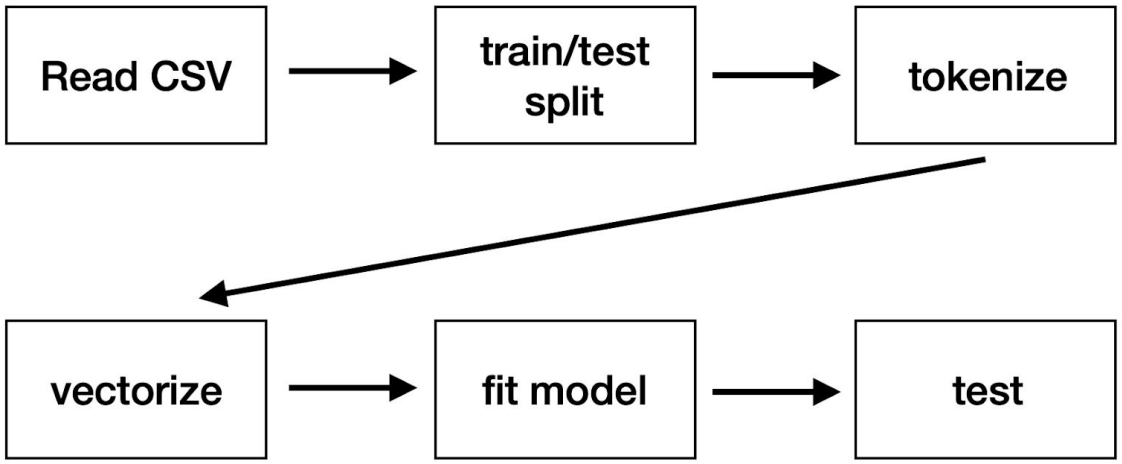
Summer Term 2020, Jacob Palecek 01526624

## Abstract

This experiment is concerned with the automatic detection of spam (i.e. unwanted/pernicious) text messages. It is based on a fully labelled dataset consisting of a little over 5000 text messages from various sources, with each of them marked as “ham” or “spam”, where “ham” denotes valid messages and “spam” unwanted ones. The data is then transformed to make it more suitable for machine learning. In the last step a sequential neural network is trained on the transformed input for 20 epochs. Keras with Tensorflow in the backend is used for the implementation of the neural network. Training and testing results, as well as the loss scores are then plotted to visualize the results and the best scores are printed to a text file. In this first basic evaluation an accuracy of over 98% was achieved when evaluating the 20% of data used for testing.

## Experiment steps

The data is sourced from Kaggle as after some searching this particular dataset provided a good balance of size and complexity as datasets consisting of hundreds of thousands of samples would have been infeasible to process. The exact dataset used can be found here: https://www.kaggle.com/uciml/sms-spam-collection-dataset
First the data is read, then split into training and testing data using a ratio of 80:20. Next the input data, which consists of a string containing the text message for each sample is tokenized using the Tokenizer provided by Keras. Tokenization is a process where a text is turned into a sequence of tokens, where a token can be a word, integer, vector or some other representation. One can imagine a text being turned into a sequence of numbers where each number represents a certain word. In a next step these token sequences are vectorized into occurrence vectors. The vector has to be at least as long as the number of distinct words in the dataset. In this case a vector size of 10000 was chosen as it was the first number tried that was able to fit the dataset. Each sample is thus turned into a vector of size 10000, where each location at the vector denotes a word. A value of 1 at a vector location means that the word occurs in the text of the sample, and a value of 0 means it does not. This way the datasets are turned into a fixed size vector input well suited for machine learning. It should be noted however that this form of transformation is relatively simple and information such as multiple occurences of the same word or the ordering of occurrences is lost.
Lastly a sequential neural network with 3 layers is trained for 20 epochs. The first two layers are dense with 4 neurons each and a ‘relu’ activation function, while the last one is also dense, but only consists of a single neuron with a sigmoid activation function. The optimizer used was rmsprop and the loss function binary cross entropy.




# Results

The experiment resulted in a validation accuracy of 0.982 and a loss score of 0.073 after 20 epochs of training. In the below graphics the loss and accuracy score progression of training and validation over the fitting period can be seen.