Lab 3 TDT4310

Vebjørn Ohr

March 5 2021

Exercise 1

A 80/20 split was chosen for the training and test data. Accuracy was used to measure the performance. The feature set used contained the one and two letter suffixes of the names. The accuracies for the Naive Bayes, the Decision Tree, and the Max Entropy classifiers can be seen in Table 1. As seen the accuracies are very similar which is not surprising with classification on such a simple feature set. Choosing another feature set would yield different results, determined by how well the features help differentiate a name between female and male. For example looking at the length of the name would probably not be a good feature, while looking at the entire name as a feature could yield better results if the training data was large enough which would then basically yield a lookup table for each name for classification (except unisex names).

Classifier	Accuracy
Naive Bayes	0.7873
Decision Tree	0.7823
Max Entropy	0.7809

Table 1: Accuracies for name classifiers

Exercise 2

I used scikit-learn to implement the classifier and split the data into training and test sets. I used the English NLTK stopwords in the vectorizer when calculating the TF-IDF weights. Using 500 tweets from both Elon Musk and Donald Trump yielded a model with 0.915 accuracy. Testing for the tweet ”Just agree to do Clubhouse with @kanyewest” (an Elon Musk tweet) gave a probability of 0.61 for Musk, and 0.39 for Trump.

Exercise 3

The movie review classifier was copied from the NLTK book. Wordnet was used to find synonyms of each word, and the boolean value of each synonym's presence in the document (review) was added to the feature set. The accuracy seems to increase slightly when using the synonyms.

Exercise 4

I made all words lowercase, removed numbers, punctuation, and white space to remove noise. This was done using basic python methods. The sentences was also tokenized for removing stopwords (NLTK english stopwords) and using Lemmtization (NLTK WordNetLemmatizer) as normalization.