machine learning project assignment 1 report

Mathijs Afman

Feb 15, 2024

1 Introduction

For this assignment, the goal was to build a classifier that could distinguish two types of tweets. To do this, we were given the option to use a Naive Bayes (NB) classifier, or a K Nearest Neighbours (KNN) classifier. I chose the KNN classifier, because I hypothesized that the way we encoded the tweets would have a positive influence for this classifier. Namely, we encode the tweets based on what items of a vocabulary occur in every tweet, resulting in a vector space that encodes all items of all tweets. The classifier, for a new tweet, then tries to find the K tweets with the closest distance in the encoded vector space. The idea is that tweets that have a similar vector encoding, also have similar meaning, and should be classified with the same label. In this report I will walk you through what I did to achieve the final version of my classifier.

2 Data

The data that was provided, were a set of train data and two sets of test data, of which I used the longer one. The train data consisted of 1950 tweets, each with a label, either 1 or 0. The test data had a length of 534 tweets and labels. The train data was well balanced, with 48% being labelled 1, and the remaining 52% being labelled 0. This means that a model that always chooses randomly is right about 50% of the time.

3 Method

3.1 Pre-Processing

Before data can be trained on, it needs to be pre-processed. It needs to be cleaned of as much meaningless information as possible. This is done to ensure that our classifier will be able to extract actual meaningful information from the data, which it can then apply on new data. For pre-processing, I removed the following things:

- user tags
- hashtags
- hyperlinks
- punctuation and emojis
- letters with accents (e.g., ‘ë’ is turned into ‘e’)
- capitalization

Until just lower-case letters and the space ‘ ’ was left.

3.2 Features

This left me with a feature set of 27 characters. However, through N-grams, I introduced features that capture more meaning. With N=2 I had about 800 features, and with N=3, about 6000. Regrettably, this exponential growth of the number of features had a very negative impact on the time it would take to train the classifier. For N=1, I measured about a tenth of a second for training, but for N=3 this time had already grown to a full minute.

3.3 Cross Validation

You might be thinking, why characters? I chose characters over words, because they showed better results in early testing. I confirmed this by cross validating different versions of the model. In Fig. 1 you can see that for the same value of N, the model based on characters performed much better than the model based on words, for all values of K. Through cross validation I tested for the best type of tokenisation, best value for N and K and the best vector distance calculation method, 'euclidean' or 'cosine'. In this case, I measured performance by the F1 score, which is the harmonic mean of precision and recall.

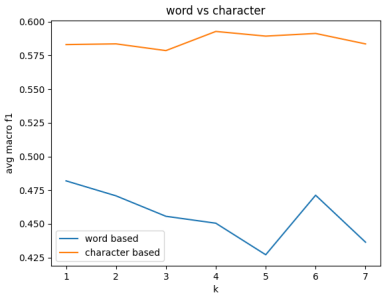

Figure 1: Tokenisation type

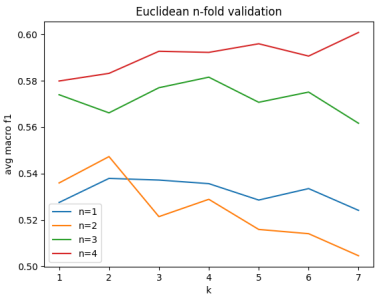

Figure 2: N and K optimisation

4 Results

In the limited time that I had to find the best possible model, I found that the following parameters worked best.

- Tokenisation: character based
- N for N-grams: 4
- K for KNN: 7
- Distance calculation method: 7

My final model with these settings performed like this:

- Accuracy: 0.674
- Macro F1 Score: 0.673

This is better than random (which should give an accuracy of about 0.5), but not by a lot.

5 Conclusion

In the end, I was not able to make my classifier perform better than the provided SVM classifier with default settings. Due to the time limitation, I was also unable to test whether my KNN classifier actually performs better than the other option, Naive Bayes. On a more positive note, I was able to find parameters for the KNN classifier well, and that make it perform better than random. It is unfortunate that I did not have time to test for higher values of N and K, because as you can see in Fig. 2, higher values for N and K tend to produce better classifiers. In future research, it would be interesting to test the limits of N and K.