Report: The Language Classification Problem

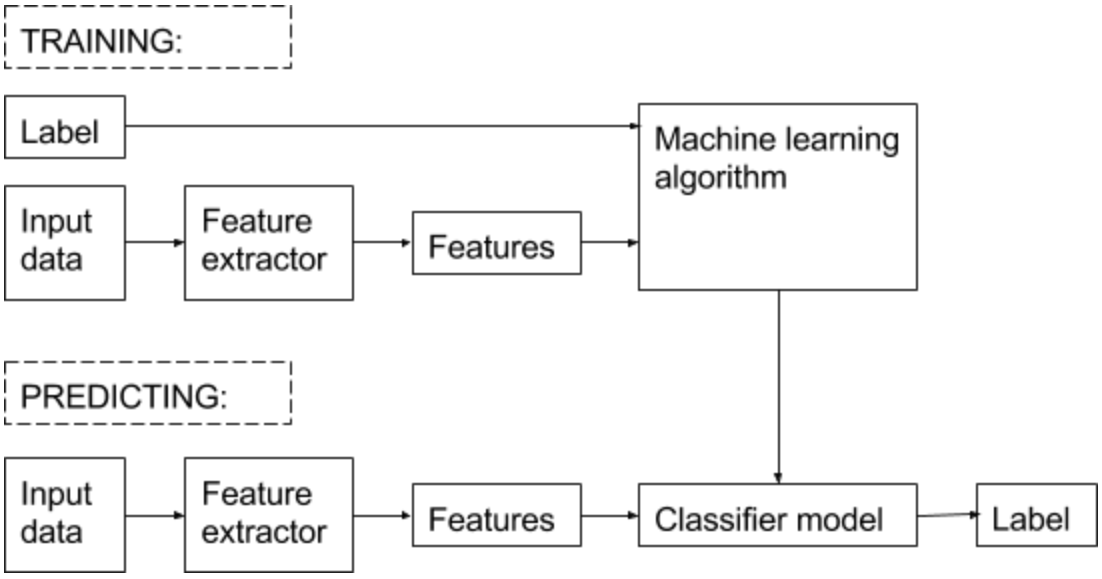
The aim of this project was to build and train and a machine learning model that can accurately distinguish between the three different languages, English, Afrikaans and Dutch. The model is trained and tested using the labeled data set of phrases provided in the document lang_data.csv.

This report will describe each stage of the program which was developed in python to solve this problem. After briefly discussing the models architecture, it will cover the following stages of the program, after which the results will be discussed:

1. Importing the libraries
2. Loading the data
3. Preparing the data
4. Training the classifier model
5. Testing and evaluating
6. Saving the classifier model
7. Loading the trained classifier model

Model architecture:

This problem falls in the category of a supervised learning classification problem and thus the algorithm is set up as follows:



After loading and cleaning the input data it is processed by the feature extractor, where the relevant features (in this case the words in the phrases) are converted into a feature set. Each feature with its accompanying label is then fed into the machine learning algorithm to generate a classifier model, this phase is known as the training phase. During the prediction phase the same feature extractor converts the unseen test data into a feature set. These are then fed into the classifier model which predicts labels. After which the models performance metrics can be measured.

Importing the libraries:

In order to construct the model, various libraries were imported so that specific pre built functions could be used, these include the following:

1. Pandas: for loading the data
2. Numpy: for reshuffling the data after it has been loaded
3. Nltk: for processing the strings and performing the machine learning algorithm
4. Pickle: for saving and loading the trained classifier

Loading the data:

This was performed using a panda command for importing csv files, which turns the csv data into a dataframe, which contains the two columns of data and has numbered rows and labelled columns.

Preparing the data:

Reshuffling the data:

Reshuffling was done to ensure that the data in the data set is evenly distributed. This has two primary advantages, firstly to ensure that each class is proportionally distributed in the test set as well as the training set, such that an even further imbalance of the classes can be avoided. The second advantage is that the robustness of the model can be tested by performing the test multiple times and comparing the performance metrics, since the training set and the test set will be different every time.

Summarizing the data:

To understand the data a bit more, a summary of the data was printed, which includes the first 20 rows of the dataframe, the shape of the dataframe, a description and a class distribution summary.

This revealed that the data is quite imbalanced since there are 2077 English phrases, 671 Afrikaans phrases and 91 Dutch phrases. This imbalance means that although there is a lot of data, the language set of Dutch phrases is rather small and thus a high accuracy for this set cannot be expected.

Printing the dataframe also gives a better idea of what needs to be cleaned out.

Cleaning the data:

Cleaning the data involves extracting all the missing data labelled ‘nan’ and everything that was not a letter or a space. This was done using one for loop within the other, the first one runs through every row of the array and filters out all the missing data labelled ‘nan’ and the second one runs through

each letter in the first element of the array, which is the language sample and it filters out all the letters and spaces and adds them to a new string, which then replaces the old string, which contained other characters.

Separating the data:

Since there is a trade-off between the amount of data available for testing and the amount available for training, selecting the ratio between the training set and the test set is important. Since the data in this assignment is quite unbalanced and small, a ratio of 4:1 was chosen. The test set is later split into three smaller test sets, which each consist of only English, Dutch or Afrikaans. This is done to determine the model's prediction accuracy for each language.

Feature extractor:

The features which are of the highest interest are the words within the phrases and by feeding them through a tokenizer, these are separated such that they can be fed into the classifier. This was done for both the training set and the test set. In addition each word was matched with the label for the phrase it was apart of.

Training the classifier model:

The machine learning algorithm that was chosen was the Naive Bayes classifier, since it is simple and effective. In the Naive Bayes classifier every feature gets a say in determining which label should be assigned to a given input value. To choose a label for an input value, the Naive Bayes classifier calculates the prior probability of each label, which is determined by determining the frequency of each label in the training set. The contribution from each feature is then combined with this prior probability, to arrive at a likelihood estimate for each label. To identify the most informative features, these are printed out as a sanity check.

Testing and evaluating:

Once the test set is fed into the classifier the label whose likelihood estimate is the highest is then assigned to the input value. After every feature has made its contribution, the classifier checks which label it is closest to, and assigns that label to the input. Individual features make their contribution to the overall decision by "voting against" labels that don't occur with that feature very often. In particular, the likelihood score for each label is reduced by multiplying it by the probability that an input value with that label would have the feature.

It is important that the test set has the same format as the training set and that the test set is distinct from the training set, otherwise the model simply memorized its input, without learning how to generalize to new examples, this would result in a misleadingly high accuracy.

Finally to test the model's accuracy, four tests were performed to determine the overall accuracy and the accuracy of each language. These test were then also performed several times to check the robustness of the model by determining the standard deviation of accuracy of the model for the four different test sets.

Saving the trained classifier:

This was done by creating a pickle file and dumping the classifier data into it.

Loading the trained classifier:

To use the trained classifier model, a file has been prepared called (NB_Trained_Model.py), which first imports the libraries required, then loads the trained classifier, then loads, cleans and prepares the the test data using the same methods as for the original data set and finally it feeds the test set into the classifier model, afterwhich it calculates the overall accuracy and the accuracies for test sets containing only one of the three languages at a time.

Results and discussion:

The overall accuracy for a test set which includes all the three languages in the same proportion as the training set is approximately 95%, with a standard deviation of around 2%. After assessing the accuracy for the languages individually, by filtering each language out of the test set to create a new test set which only contains that specific language and then performing the same accuracy test as before, the following results are obtained:

English accuracy: 93% Standard deviation: 2%

Afrikaans accuracy: 85% Standard deviation: 3%

Dutch accuracy: 41% Standard deviation: 6%

From these results it becomes clear that the models prediction accuracy is lower for the languages which it had been given less training data. It can also been seen that the accuracy varies a slightly each time that the classification is performed and that it varies slightly more for the languages which it had been given less training data. This can be attributed to the fact that there is a small amount of data and the reshuffling of the data has an affect on which words are picked up in the training set and which are left in the test set, and the more evenly the same words are distributed between those the higher the accuracy will be. The lower accuracy for Afrikaans and Dutch phrases is also partially due to the bias of the classifier towards English, based on its dominant occurrence.

Bonus Questions:

1. Discuss two machine learning approaches (other than the one you used in your language classification implementation) that would also be able to perform the task. Explain how these methods could be applied instead of your chosen method.

The first algorithm that would also work very effectively is the K-Neighbours Classifier. Although this algorithm isn't trained before testing begins it simply compares to the test data to the training data and finds the most similar feature to it, after which it gives it the same label. The second classifier is the Decision Tree Classifier, which maps observations about a feature into conclusions about the feature target value, it then classifies the feature based on these observations.

2. Explain the difference between supervised and unsupervised learning.

In supervised learning an input and output dataset is used to train the machine learning algorithm and in unsupervised learning only an input dataset is provided and the machine learning algorithm has to learn how to organize this data itself.

3. Explain the difference between classification and regression.

Classification is used to predict which class a data point is part of and regression is used to predict continuous values.

4. In supervised classification tasks, we are often faced with datasets in which one of the classes is much more prevalent than any of the other classes. This condition is called class imbalance.

5. Explain the consequences of class imbalance in the context of machine learning.

Imbalance can lead to a classification model that is bias towards the dominant class in the data set. This can become evident by evaluating the prediction accuracy of each class, since this will shed more light on whether or not the less dominant classes are being classified correctly or if the classifier has simply chosen to classify the feature as the dominant class. To further investigate this the prediction precision can be examined.

6. Explain how any negative consequences of the class imbalance problem (explained in question 4) can be mitigated.

Possible solution 1) Collect More Data

A larger dataset might expose a different and more balanced dataset and more examples of minor classes may be useful when resampling the dataset.

Possible solution 2) Changing Your Performance Metric

Since accuracy can be obscured by the class imbalance, one could have a look at the precision, which shows the exactness of a classifier.

7. Provide a short overview of the key differences between deep learning and any non-deep learning method.

In essence the differences is that deep learning methods offer a set of techniques and algorithms that aim to parameterize deep neural network structures with many hidden layers and parameters, whereas simple machine learning algorithms don’t.