CSCU9YE Assignment Report

Student no: 2519302

November 17th 2018

Contents

1 Introduction

This project is tasked with the classification of spam emails, to a high degree of accuracy using machine learning models. This report will aim to provide an overview of the datasets used, an understanding of how the models used work and the results obtained from running these models on the datasets mentioned.

2 Specification

This section will walk through the flowchart shown in *Figure 1*.

Load Emails

Firstly, the emails need to be loaded into the program in order to parse through them and begin the pre-processing stage. This involves reading in all the files in the folder and saving them to a variable for manipulation further on in the program.

Pre-Process Data

This stage is where the data is processed to convert case, remove elements such as punctuation and stop words and is where the process of lemmatization happens. This process is also referred to as normalization. These entities are removed due to their inherent lack of relevance to the end classification. Lemmatization on the other hand is done in order to reduce the complexity of the generated dictionary in the next step and thus the overall runtime of the algorithms on the dataset.

Dictionary Creation

This process, mentioned briefly above, entails the tallying of the most commonly occurring words in all the emails in the dataset. This is done such that a correlation can be obtained when reading through similar pieces of text. This allows us to see if the emails share similarities and thus may be classed under a label. The generated dictionary is then compared to the features extracted from each email.

Feature Extraction

The feature extraction phase entails extracting a feature vector for all emails in the dataset. The features contain the number of occurrences of each entity in the email, with respect to the dictionary generated.

Split Dataset

The dataset is split into a test set and a training set. This is done such that the model can be trained on a subset of the dataset as a whole and the model can then be evaluated on the same data. The split is done at approximately 70 / 30, where the training set is 70% of the full dataset and the remaining 30% is used as the test set.

Train Model

The models are trained with a subset of the dataset as mentioned above.

Pre-Trained Model

This process is where the pre-trained model is tested with the test set obtained from when the dataset was split.

Evaluate Trained Model

The outputs of the models used here are evaluated using the performance matrices: Accuracy, Precision, Recall and F1-Score. The unlabelled dataset used here is the same test set, but without any accompanying labels designating the entities classification.

2.1 Pseudocode

```
begin

    begin load_emails
        load emails into program
        split dataset into test set and training set
    end load_emails

    begin pre_processing
        convert text to lower case
        remove stop_words
        lemmatize text
    end pre_processing

    begin create_dictionary
        generate dictionary
    end create_dictionary

    begin feature_extraction
        extract feature vectors for all emails
    end feature_extraction

    Train models on training set
    Test models on test set
    Generate predictions using models
    Evaluate model predictions

end
```

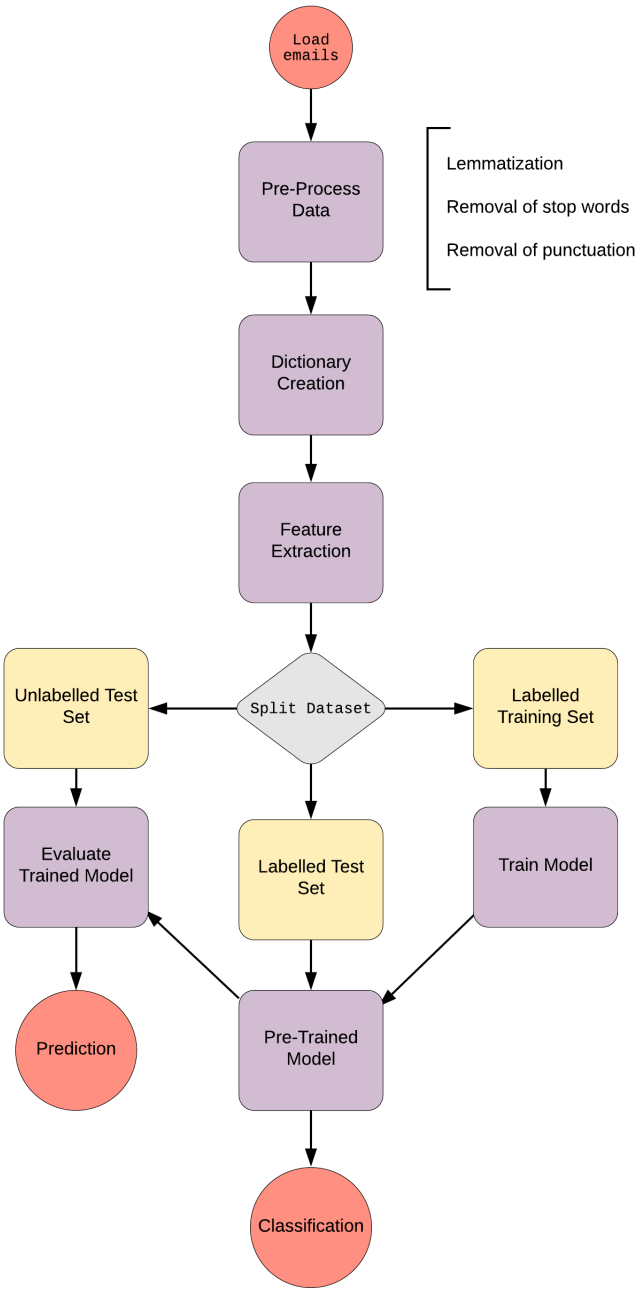

Figure 1: Flowchart for the general Spam Classification process

3 Pre-Processing

This section will aim to cover the different aspects of pre-processing employed in this project. Within the datasets used in this project, we can see that the general steps one would need to take are as follows:

- Lemmatization
- Removal of stop words
- Removal punctuation

Along with the aforementioned steps, the text will be processed to handle case. This is done by converting the text to lower case before any text normalization is done.

3.1 Lemmatization

Lemmatization is a form of text normalization where the aim is to remove inflection and/or derive the base word from a family of words in the dataset [1, 2, 3]. The base of a word, is also known as the **lemma** or the **dictionary form**. This process also leads to a decrease in complexity during the runtime of the algorithm. An example of which is shown in *Table 1* below.

Such mapping also allows us to find all relevant documents using a specific word.

Word		**Lemma**
Cleaning	$\rightarrow$	Clean
Cleaner	$\rightarrow$	Clean
Cleanliness	$\rightarrow$	Clean

Table 1: Showing the mapping of words to its lemma

3.2 Removal of Stop Words

Stop words are words such as "the", "he" or "in". These words do not contribute to the overall meaning of the text and are thus removed during the pre-processing stage. This also tends to be done in order to improve indexing times for larger datasets.

3.3 Removal of Punctuation

Punctuation, like stop words are removed due to their lack of contribution to the overall understanding of the text.

4 Machine Learning Methods

This section will aim to provide some knowledge on the machine learning models employed in this project.

4.1 Support Vector Machine

A Support Vector Machine (SVM) is a discriminative classification method. This method produces a classification by finding an optimal hyperplane that segregates the data points and thus returns classifications [4]. This can be visualized easiest on a 2-Dimensional plot as can be seen below in *Figure 2*:

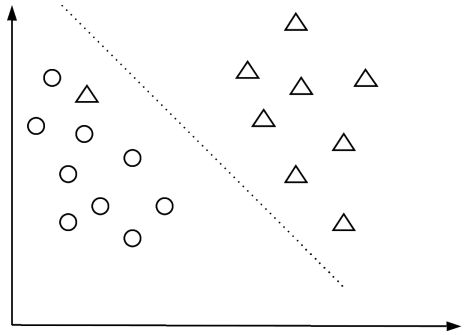

Figure 2: SVM on 2-D plot area

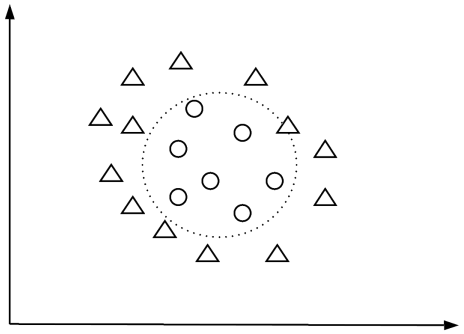

Figure 3: Kernel feature of SVM

Figure 3 above shows an SVM making a classification on data where it would seem like there is no *linear* hyperplane to be found. In order to make a classification, the SVM employs the kernel method [5]. This transformation, introduces a new dimension, by way of suggesting that for every **x** and **x'**, there is a function *k* such that *k* is equivalent to the sum of the squares of **x** and **x'** [6]. This transformation then allows the model to find an optimal hyperplane within this third dimension. Transforming this back into a 2-Dimensional plot, the hyperplane is mapped as a circular boundary around the classified data points.

4.2 Random Forest

Random Forest is a supervised learning algorithm. The way it works is by generating a number of decision trees, all of which generate a classification. This is analogous to each decision tree in the forest 'voting' on a classification. The algorithm obtains a final singular classification by choosing the classification with the most 'votes'. To note, the decision trees generated by this algorithm are not pruned.

4.3 Multilayer Perceptron

A Multilayer Perceptron (MLP), is a form of feedforward artificial neural network.

References

[1] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*, vol. 39. Cambridge University Press, 2008.

[2] H. Jabeen, "Stemming and lemmatization in python." https://www.datacamp.com/community/tutorials/stemming-lemmatization-python, 2018 (accessed November 17, 2018).

[3] K. Fortney, "Pre-processing in natural language machine learning." https://towardsdatascience.com/pre-processing-in-natural-language-machine-learning-898a84b8bd47, 2017 (accessed November 17, 2018).

[4] S. Patel, "Chapter 2 : Svm support vector machine theory." https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72, May.

[5] S. Ray, "Understanding support vector machine algorithm from examples (along with code)." https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/, 2017 (accessed November 17, 2018).

[6] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.