**CMA1151 Associate Data Scientist**

**Gilad Amzaleg**

**Part 1 – Summary of Work**

**Introduction**

The purpose of this task is to build an accurate model that can predict whether a short summary of text is about a merger or whether it is not. This was done by using a machine learning model that trains and tests itself on data provided.

The data was in the form of two columns. One column containing a short summary of a news item from the Competition and Markets Authority’s (CMA) homepage. The second column contains a binary labelling system that describes if the short text summary is about a merger (1) or not (0).

**Data Pre-processing**

Before any modelling can be done, a pre-processing/cleaning regime was implemented on the data provided. This is to insure a more accurate model can be built before predictions can be made.

The first step was to remove all rows in the data that did not contain any information as this would potentially skew the model slightly and make the processing easier.

Natural language and free text is easy for humans to interpret and understand, however much more difficult for a computer. This means Natural Language Processing (NLP) is carried out. NLP is a series of actions that can be carried out on free text data that makes if computer readable.

First, all the words containing capital letters were replaced with their lower-case equivalents. This was done to ensure the model does not think the same words are different due to the fact they have different case of letters. An example, the word “competition” is the same as “Competition”. However, a computer wouldn’t be able to see them as the same word due to the discrepancy in the casing.]

Tokenisation of words is an important next step in the pre processing regime. Tokenisation is the action of breaking down a piece of text into smaller units called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Tokens can either be words, character or sub words. These are the building blocks of Natural Language and the most common way of processing the raw text data.

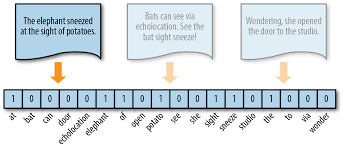
Here is an example of tokenisation:

Input: Friends, Romans, Countrymen, lend me your ears;

Output: \framebox{Friends\weestrut} \framebox{Romans\weestrut} \framebox{Countrymen\weestrut} \framebox{lend\weestrut} \framebox{me\weestrut} \framebox{your\weestrut} \framebox{ears\weestrut}

The final step in the cleaning process is to remove all stop words. Stop words are the most common words in language that do not possess any relevant information about the text. I.e. they do not tell us anything useful about the document of text itself. Examples include “a”, “an”, “and”, “the”, “he”. They are common to all text however do not provide any sort of classification information so are hence removed from the text.

As mentioned before computers find it very difficult to interpret natural language (one, two, three…) however do find it much easier to deal with and numerical data (1,2,3…). The internal processing of the computer can work with numerical data much easier, so we apply a process called vectorisation to the text data.

Vectorisation is the process of representing a document of text as a numerical number or vector. This is done for every event in the data set. As we are using a bag of words approach, we are essentially counting every word from the text. Below is a small schematic representation:

For the above example, vectorisation creates a vocabulary of all the words throughout the entire data set. Then for each event counts how times that word appears in the individual text resulting in a vector that represents what words are present in the text and what words are not present.

The output from this vectorisation is a matrix where the rows contain the entire vocabulary from all the data and the rows represent each individual data event. The values inside the matrix correspond to the frequency of that word appearing in the data event.

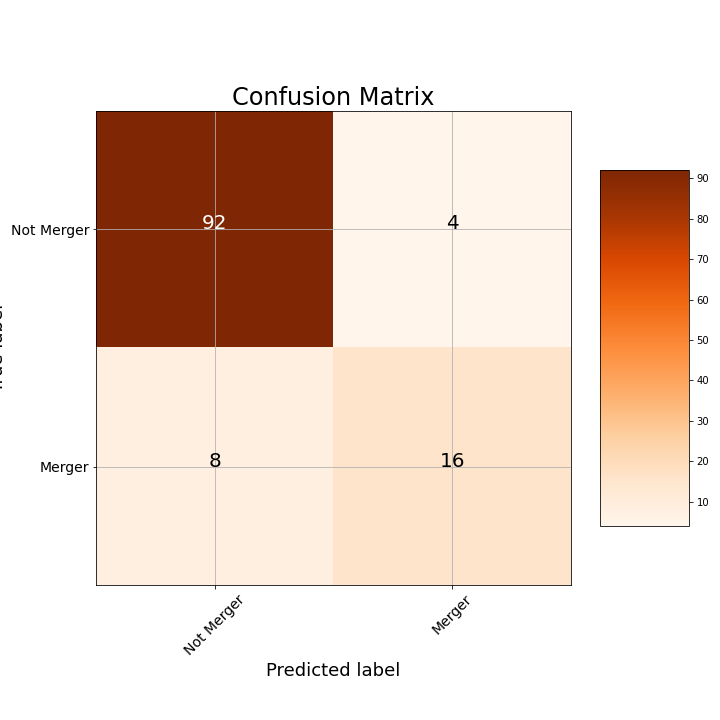
Now that we have converted are textual information into a numerical representation that can be processed by a computer, we can build our model to predict.

**Model**

First step in building a model is the shuffle the data to avoid order bias and then split the data into a training and test set. This was done 80% training and 20% test.

A two-class random forest classifier was chosen as it results in accurate predictions and fast training times. Random forest, like its name implies, consists of a large number of individual decision trees that operate as an [ensemble](https://en.wikipedia.org/wiki/Ensemble_learning). Each individual tree in the random forest spits out a prediction and the class with the most votes becomes our model’s prediction. Essentially creates multiple outcomes and votes on the most post method of getting to that outcome.

The model was trained on the training set and tested on the test set. To test the performance of the model, a [Receiver Operating Characteristic Area Under the Curve](https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc) (ROC AUC) was used. The ROC AUC score is a measure of how the much the model can determine between our two possible outcomes (1-merger and 0-non merger). The scoring system is such that 0 poor performance and 1 is perfect performance. The model created for this project scored a ROC AUC score of **0.945** suggesting our model is accurate.

A further visual method of evaluation is to plot the confusion matrix as shown below.

True Label

This shows the predictions the model got correct in the top left and bottom right corners and the predictions missed by the model in the lower left and upper right. We can clearly see that for most of our predictions the model was corrected compared to the test data with 92% correct for non-mergers and 80% correct for mergers.

Hence, our model appears to be an effective method of predicting whether a short summary of text is detailing a merger or a non-merger, with an ROC AUC score of 0.945

**Part 2**

**Alternative approaches**

The method of vectorisation used here was a bag of words approach which takes the number of words in the text as set, i.e. imagine a bag full of words. BOW counts the frequency of words present in the bag however does not give any weight to how significant the words actually are in the text.

Another way of judging the topic of an article is using [Term-frequency-inverse document frequency](https://en.wikipedia.org/wiki/Tf%E2%80%93idf) (TF-IDF). TF-IDF assigns different words weight and measure relevance and not frequency. Instead of word counts a score is given across the whole data set.

Anther method of NLP is word2vector which produces a different vector per word whereas BOW produces one number (a wordcount). Word2Vec is far greater for digging into documents content and subsets of content. Its vectors represent each word’s context.