Introduction

Machine Learning is a novel approach to automating complex problems with a long list of elaborate rules. Instead of explicitly telling the computer what to do and hardwiring every rule, data is fed to the computer with appropriate identifiers. Minimising a loss function, the computer is guided toward the rules in a hot/cold method. This minimization of the loss function is attributed to the 'learning' ability of a computer. This is an example of supervised learning.

Machine Learning can be used for a never-ending list of problems including various types of Natural Language Processing (NLP). NLP is the process of wrangling natural text used by humans and converting it into a more computer friendly format which retains the meaning of the text. This mined text can then be used for various purposes such as exploring the common words around a theme, or even a plethora of machine learning uses.

The machine learning NLP applications include, but are not limited to, classifying various texts into their respective categories, automatically flagging offensive comments on a public form or flagging spam email, summarizing long documents automatically and even creating human-like chatbots for specific purposes like customer support of a product. Each application utilizes a different machine learning algorithm such as Recurrent Neural Networks (RNNs).

Problem

The task for the project was to classify various texts pertaining to health-related Sustainable Development Goals (SDGs), outlined by the United Nations (UN). An example of the text is 'Quality Improvement Initiatives for Diabetes' and it's corresponding SDG would be '3.4.1 - Mortality rate attributed to cardiovascular disease, cancer, diabetes or chronic respiratory disease'. This is an example of a text that only has one label. This is not the case for the entire dataset.

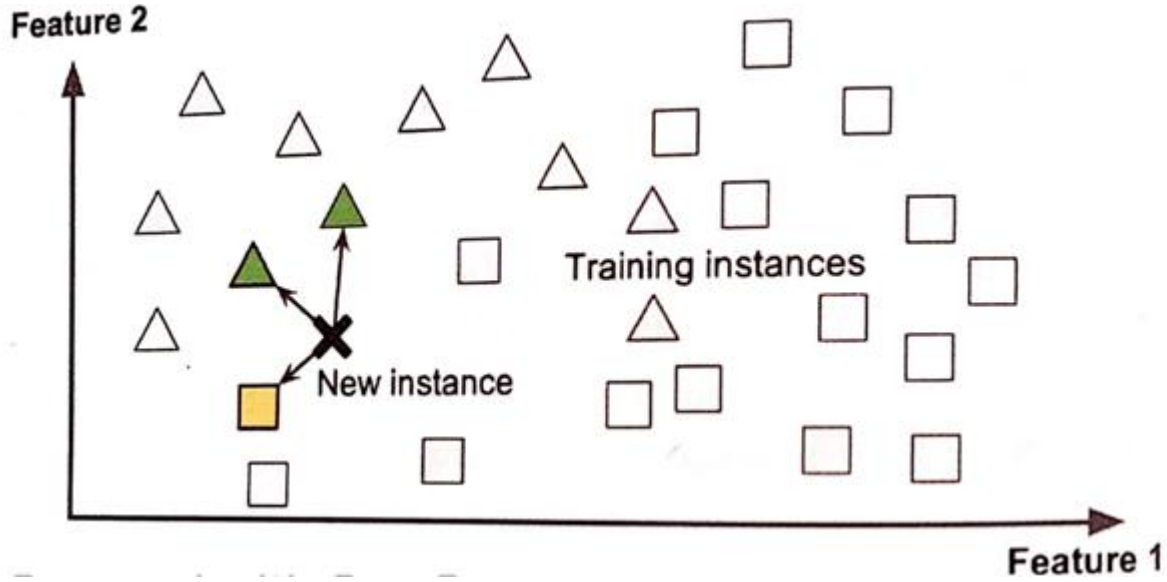
Most applications of machine learning are geared towards mutually exclusive classification. Implying that it can only have one type of label at a time. The prototype programmed in the Python language for the project is a general solution for all multi-class and multi-label texts. Meaning that a single piece of text can have more than one label at a time and the possibility of labels are more than one two. For example, an essay could refer to either poverty or employment or both.

The task at hand had 27 possible labels with no upper limit to how many of the SDGs are incorporated in a particular piece of text. The dataset that we were working on had up to 10 labels simultaneously, implying that a single piece of text is related to 10 distinct SDGs simultaneously. The text was muddled with HTML code and other ASCII characters not containing useful information. To deal with this data mining problem the Natural Language ToolKit 3 (NLTK) library for python was used. NLTK, originally created in 2001, is a comprehensive natural language processing library that includes vital functions used in almost all text analytics projects. Popular functions such as removing stop words (punctuation or common words that do not add information to the sentence such as 'the', 'a', 'if', 'then' etc.), lowercasing the entire text and removing unknown words such as

HTML code. These functions are used every time a natural text is analyzed to extract the valuable information from the clutter.

Solution

The model solves the problem in a generalized manner because the features used to train to model is the frequency of words categorized by label. This is an example of Instance Based Learning (IBL). IBL is a trivial form of learning which is to simply 'learn by heart' and then generalizes to new cases by using a similarity measure to compare them to the learned examples (Geron, 2019). This means the model consumes examples with labels, then when a new instance is fed without a label, the algorithm evaluates the closest possible label.



The figure above is an illustration of instance-based learning. The new instance (black cross) will be classified as a triangle because most of the similar instances belong to that class. Where the features would be the frequency of particular words. The words would be chosen with the frequency of their occurrence with respect to a particular label. For example, biomedical terms like 'neuroscience' were a pivotal feature for the label '3.b.2 - Total net official development assistance to medical research and basic health sector'. This is expected considering the SDG 3.b.2 pertains to texts discussing medical research. Once the features (popular words with respect to the label) are selected, when a new instance is processed, and it contains pivotal features, a classifier will be able to assign a probability of whether the text contains the label or not.

The classifier used to generate these probabilities in the developed model is the Naïve Bayes Classifier. In this multi-class classifier, every feature is incorporated into the model. This classifier first calculates the prior probabilities of the label. This is determined by the popular words. The more common the word, the higher the prior probability. When a new instance is fed, the words are examined and a likelihood estimate for each label is calculated. The label with the highest probability is assigned to the text.

Naïve Bayes Classifier

The working of the Naïve Bayes algorithm implies that for the model developed, the different labels were created for a single Naïve Bayes Classifier. This is untrue because the classifier outputs a single label meaning the labels are mutually exclusive. To deal with the issue, a separate classifier is instantiated for every single label. This translates into every classifier deciding whether a label is triggered for a text or it is not, making each classifier a binary classifier. Finally, to make the classifier into a multi-class and multi-label classifier, an ensemble of all 27 label classifiers was pieced together. Finally, a single piece of text is processed by every single Naïve Bayes model outputting a true/false for every label.

Output

2900 pieces of texts were used to train the models with each label getting its own set of texts. Individual Training accuracies of some of the models are shown below.

Label	Train Accuracy	Number of Texts
3.d.1	0.43	217
3.8.1	0.47	531
3.7.1	0.51	189
3.7.2	0.53	165
3.9.2	0.55	218
3.b.3	0.58	397
3.3.2	0.64	174
3.3.3	0.65	158
3.3.5	0.66	156
3.1.1	0.75	218
3.4.1	0.75	485
3.c.1	0.75	232
3.b.2	0.78	1044

Some labels train fine, others either underfit or overfit. The accuracy of these models can be vastly improved using better data wrangling techniques. Using more regular expressions to remove unknown words that the model can be sensitive towards.

The ensemble was then used to test 100 instances of data it had never seen before from the same dataset. The accuracy was a ratio calculated using the number of labels predicted correctly to total actual labels. This gave us an accuracy of ~62%. **While the method to calculate the accuracy was more lenient than conventional means, it was used as a quick and general idea of how the overall model was performing. Since this is not the final face of the ensemble and more work is required till it is ready for launch.**