Diego Valdes

IST 736

Home Work 2

10/19/2019

**Introduction**

If one were to consider for a moment, the amount of data they generate in a day, they may be surprised at the footprint they are leaving without even knowing. Consider, for a moment, all of the internet searches a person does in a day for shopping, dining, and random questions. Then add the social media posts. Top that off with any pictures taken that day. Lastly, lets add any third party indirectly collecting data, like traffic lights as an example. Repeat that for everyone in the developed world, and suddenly that’s a whole lot of information to comb through. Now imagine, if through all of that data, a company was tasked with finding a particular piece of information to assist them is improving sales. The analogy ‘needle in a haystack’ may seem appropriate.

That is where artificial intelligence comes into play and why it’s one of the buzz words in every industry today. It is a possible solution to problems that didn’t exist in the past, as well as for situations where humans are simply not the right tool to accomplish the task. In most of these situations, humans are the best suited, but the volume of the work and the time constraints make it impossible to rely on humans. Think of the man power required to read all the tweets posted in one day to search for something inappropriate; there may not be enough humans on the planet to accomplish that task in a timely manner.

In the past few years, social media corporations have been at the forefront of criticism for their inability (or lack of effort) in curbing hate speech, or other forms of speech that can be deemed harmful. While there are many avenues to examine this issue, the one at heart is if these corporations can use A.I. to target it’s customers for advertisements, then the same technologies need to be used to analyze user produced content to find anything that is inappropriate.

**Analysis**

**About the Data**

The dataset is a collection of movie reviews provided by the course that was cleaned and formatted as part of an initial sentiment analysis. The data has two columns (review, sentiment), where each review was stripped of punctuation and special characters. This dataset was used for the vectorization process.

To further prepare the data for vectorization, upon reading in, the sentiment column was removed and stored separately from the reviews. No stop words were removed from the reviews. This decision was made due to previous analysis of The Federalist Papers, where it was discovered that some stop words were essential to predicting authorship. Instead, the approach was taken to remove words of less than four characters for initial vectorization. The end result was a list, where each element was one review and no words less than four characters where present.

**Models**

The CountVectorizer from sklearn was used to vectorize the reviews. Implementation was a small matter of passing the list of reviews to a function. The package vectorizes each review, then all of those vectors are converted to a dataframe. The resulting data frame was 2000 x 42926. The matrix was extremely sparse.

**Results**

First attempt at trimming it down was to examine the instances of each word appearing in the collection of reviews in more than ten occurrences. This created a significant size decrease to 9071 columns. Further limits on the threshold were attempted with the accepted result being 83 columns (threshold 1000).

**Conclusion**

CountVectorizer is a great option for vectorizing text documents. The package comes with many options to either read in a list of words or a collection of files. Once the matrix has been created, manipulating it with frequencies instead of word counts is straight forward. Trimming the matrix down to get it to a desired size is also simple to achieve.

While it would be a programming challenge worth undertaking, it would be possible to create the matrix from scratch. This approach would require the use of each document to be mapped out with a dictionary, then the dictionaries of each document converted into the matrix. The keys of the dictionaries would be the words, which would then be converted to column headers. While the challenge of this endeavor would be a creative use of a computer science student’s time, a full time graduate student holding down a fulltime job may not be able to spare the time.

Due to the ease of implementation, especially when considering the alternative of creating the matrix from scratch, CountVectorizer is the best solution for vectorizing text. The entire process can be wrapped in one’s own function to make the process easier and not have to constantly write out code.