**Extracting Article Keywords**

Using Newspaper3k library, keywords were extracted from any article, provided the user has the URL of the article.[Beautifulsoup 4.0](https://www.crummy.com/software/BeautifulSoup/bs4/doc/#porting-code-to-bs4)

The function is then defined that when called; extracts and prints out the keyword from the article.

**Tokenize and remove stopwords**

Using scikit module, tokenization and removing stop words is carried out by converting our textual data into a vector matrix product as specified in above formula since the tfidf vectorizer will automatically tokenize the title passed to it Then I got the keywords using the get\_feature\_names method which returned a string of keywords. And I joined the keywords together making it a string.

In summary, the function requires the scikit learn module to accept a title(string) and return a string of keywords from that title.

**Clustering and Exporting the model**

Using kmeans clustering of articles is carried out to get clusters through the function which returns keywords from the title of posts. The posts are sourced directly from sql database for the model to create clusters from the returned keywords of titles. 5 clusters were adopted to fit in the 10 titles to be returned. The number of titles is required to be greater than or equal to the number of clusters for it to be functional, In this case 5 were adopted to get more clusters.

kmeans was imported from scikit learn clusters for this task.

**Predicting similar articles**

Used tfidf to identify keywords and remove stopwords from post titles, applied cosine similarities to calculate the frequency of likelihood of such keywords appearing in other titles using their post\_id as pointers. With 2 functions, one to fetch post\_ids to be re-used in the second function to recommend similar articles.

Function **post** has one parameter that's **post\_id**

Function **recommend** has 2 parameters **posi\_id** and **num** i.e. number of recommendations.The second function call returns the expected output. i.e. similar post titles