**Project Report- The Inspectors**

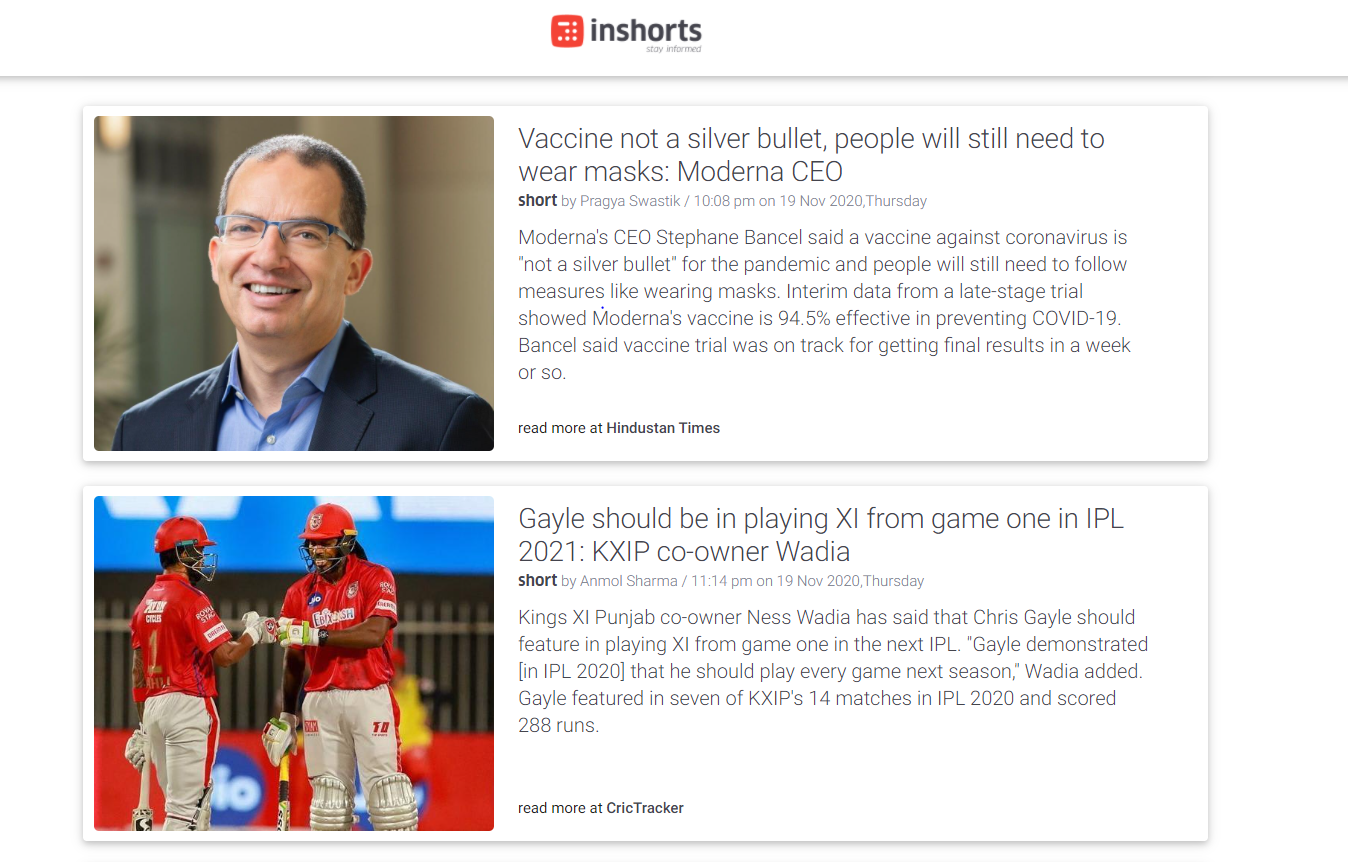
**Description of the website:**

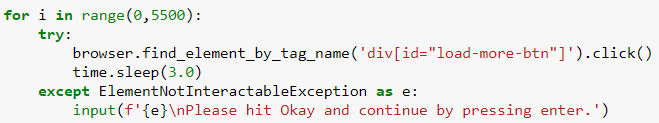
Inshorts keeps the user updated with latest news while saving user’s time by providing the news in less than 60 words or less. The main abstract or gist of the news is usually provided in the first 60 words and if the user is interested, they can click the article and they will be redirected to the full length article where they can read the news in detail. There are multiple articles on the general page that consist of the latest news in all genre. If the user wishes to read article specific to the topics such as science or technology than they can choose the respective filter and they will be redirected to the webpage where they are provided with the articles that matches their topics.

**Objective:**

The main objective is to scrape the data from the website in a tabular format and perform topic modelling to sub categorize each news article into its related domain then ingest sentiment analysis on the abstract of the article in order to know the tone of the news. E.g. Positive, negative or neutral.

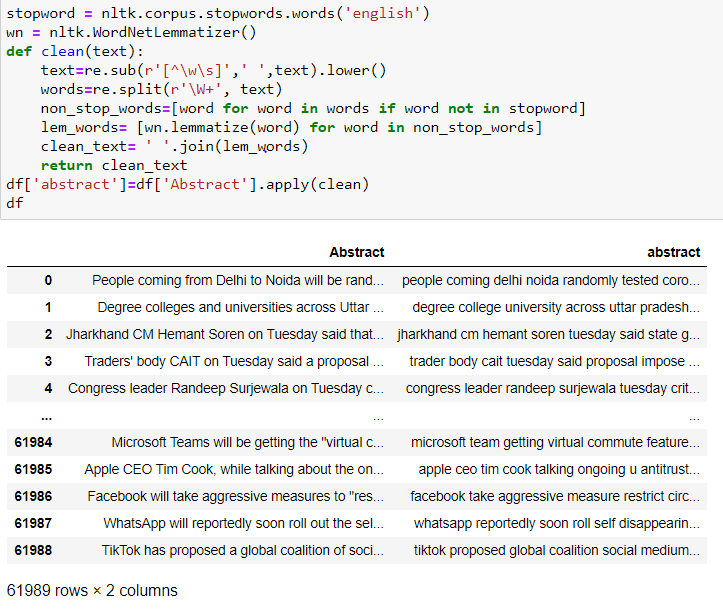
**Website Scraping:**

The look of the website is quite simple and the website itself is user-friendly. Along with that the code written to develop the website looks firm and is systematic. But the website has some JavaScript coding and the navigation through different pages is quite difficult and hence it was tricky to scrape article from multiple pages of the website. The image below shows the general feel of the website: -

In order to scrape the code from the website, we’ve used various libraries such as beautiful soup, re, numpy , pandas and selenium. We tried to extract the dataset while navigating through the pages which consumed almost 8 hours of processing and ended up in scraping multiple duplicates in the dataset, where we decided to firstly navigate through the pages and then scrape data on the next part, which eliminates the challenges of scraping duplicates, we faced during the first time. The second method was long and heavy as well, it took almost 9 hours of processing power, but we got the results we wanted we were able to scrape the required articles and form a dataset. The following images basically shows the code used to scrape the website and the output of the website

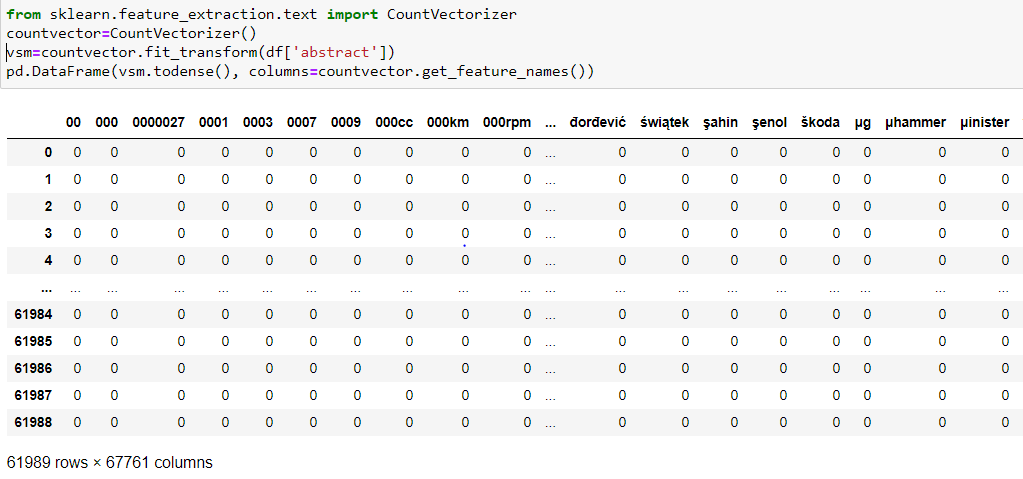
We were able to extract almost 62k rows of the data and successfully load into the csv file that we will further use for analysis.

**Cleaning Abstract for Topic Modelling:**

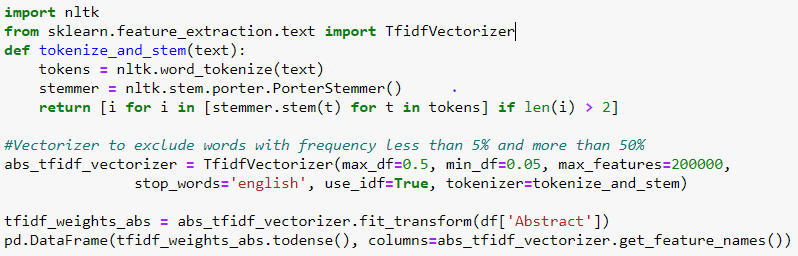
Since we must perform the analysis on the abstract, we will isolate the abstract from the rest of the dataset. We need to remove punctuations, alpha numeric characters, stopwords and apply the cleaning process to every single abstract in the dataset. The following pictures depict the process of cleaning and the result obtained by the cleaning process.

**Topic Modelling:**

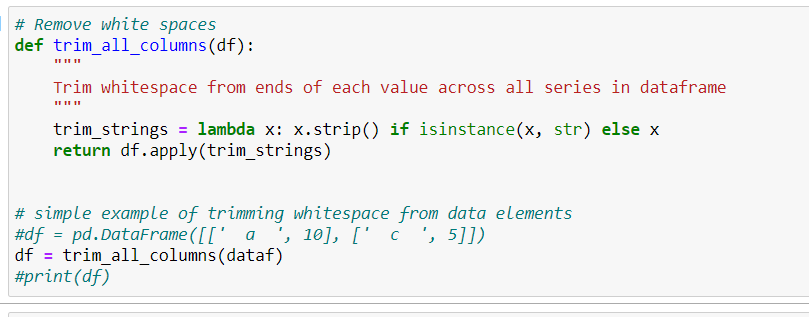
We tried to directly apply the topic modelling, but the results contained majority of common words and hence with the help of professor, we decided to first exclude extremely rare and common words from our list to enhance our topic model. In order to do that, first we created a vector of all the models by using count vectorizer to get the frequency of the words occurring in the abstract. We will use sklearn library and token and stemmers method to match the frequency and form a separate database.

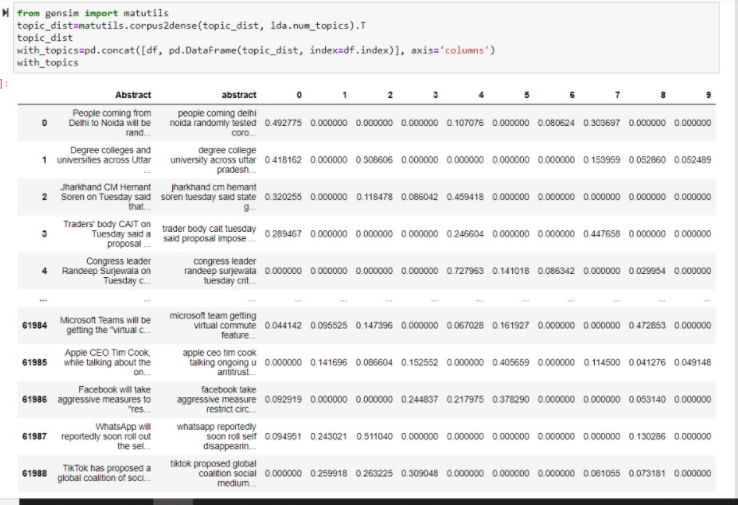
The following picture depicts the formation of count vectorizer: -

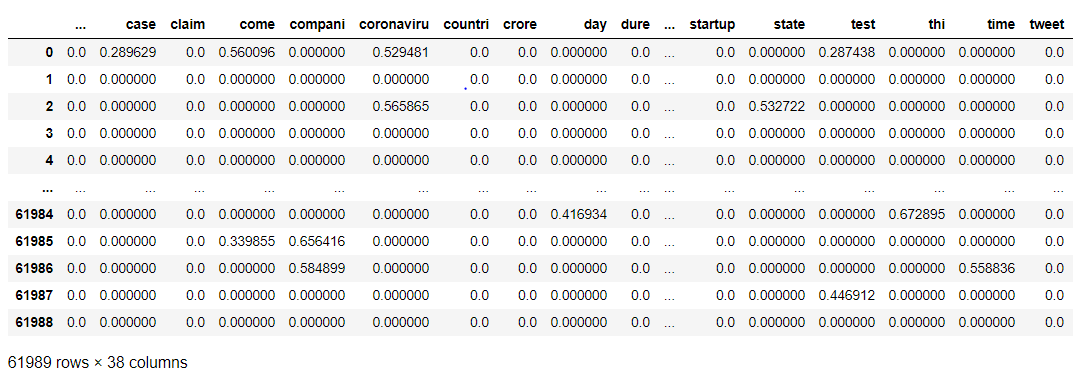
After the formation of the count vectorizer, we defined a method which will eliminate the words that occur less than 5% which are considered extremely rare and words that appear more than 50% which are considered extremely common. This small change made our model clearer and more effective. The following pictures depicts the code and the output of the final model.



Removed White spaces from the data frame

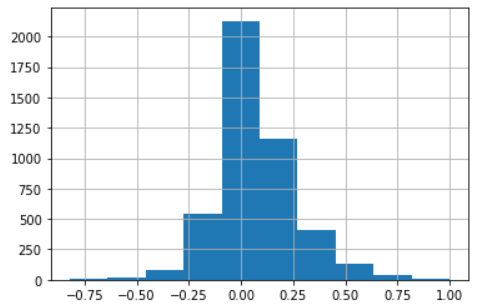


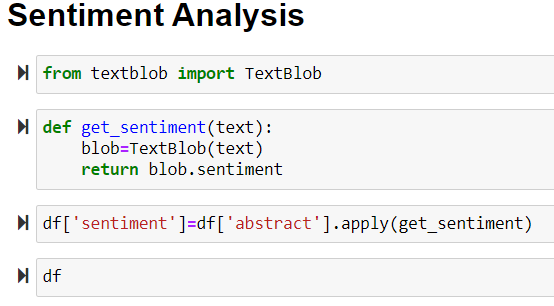


Above we have concatenated the two abstracts and merged it into a dataframe with reference to the topics. Next, we will perform sentiment analysis over this abstract in order to get the tone of each articles.

**Sentiment Analysis:**

After topic modelling we performed sentiment analysis over the abstract data in order to term the sub categorized topics as positive, negative or neutral in terms of tone of the article. Eg if there is a news of burglary the algorithm will term this news as negative whereas if the news is about a team winning a championship it will mention as positive. Generally, this type of sentiment analysis is useful for consumers who are trying to research a product or service, or marketers researching public opinion of their company. We found out that majority of the articles have a neutral tone.





Conclusion - We successfully scraped the dataset from the website, performed Topic modelling and Sentiment Analysis on the dataset.