**Introduction:**

My part was basically consists of sentiment analysis of Media report to the common people of India over Covid Pandemic situation.

It consists of 4 groups:

1. Sentiment analysis of death due to Covid issues
2. Sentiment analysis of Economical aspect of Covid
3. Sentiment analysis of Education scenario of Covid
4. Sentiment analysis of Public Health Awareness due to covid

I have used “The Telegraph” e-newspaper for sentiment analysis for the period of 24th March – 30th June 2020

**Preprocessing and Data Extraction:**

The e-newspaper provided by the “The Telegraph” is a scanned version of English daily. It can be viewed from the e-paper website.

Step-1: The first step is to generate the URL access date wise for the English daily.

Step-2: Next for each day there are ‘N’ numbers of pages, so we need to iterate every page.

Step-3: On each page we need to find the particular ‘div’ section where all the data are kept

Step-4: Next we need to find the particular key for accessing the data

Step-5: As it is a scanned version of the daily so all the necessary data are in .jpg format.

Step-6: As the data are in printed version, no handwriting analysis required. We have used “pyTesseract”. It is a open source python driven virtual machine which is used for Optical Character Recognition. We need to set up this virtual system in our local machine and provide the path to the system environment variable so that we can access its functionality.

Step-7: Now with the help of pyTesseract we can able to convert the content of .jpg into .txt format for further analysis.

**Selection of Article based on Probabilistic Approach:**

Now the task is to choose the correct article for analysis. As in English daily there are ‘N’ numbers of article having different kind of news (from state to country, from business to politics, from sports to science etc.), our task is to choose the correct article so that it is relevant to Covid for our further analysis.

So we have taken few predictable keywords need to be present in every article such that we can identify an article for each group.

As the list of predictable keywords are more in number and it is not necessary that every keyword should be present in every article for every group. So, We have checked whether that keyword is present in the article for that group and count the total number of keywords matched.

As, we got the count we can divide it by the total number of keywords taken for that group and adjust the threshold value for the selection of the article.