Into the Pandemic

Antara Arun Agarwal (2019CSB1076)
Ishika Chaudhary (2018MMB1284)
Ruheen Zeba (2019CEB1029)

Date : 23/11/2020

Abstract

Here we thought to analyze the reliability of news channel coverage on daily cases of covid-19 pandemic. We compare the graph obtained by plotting daily cases data given by worldometer (considering it as a reliable source of information) with the graph obtained by plotting daily news articles and webpages present on the internet archive with the help of Pearson's coefficient.

In our project we will be trying to find a direct relation between information on coronavirus on the internet platform and the number of cases of coronavirus with data of the past few months.

1 Introduction

Since Covid pandemic has stretched for too long, masses in general have become much more careless than before. Without taking the proper precautions, covid cases are booming. The news information or data does not reveal the correct scenario, but it is the most viewed by almost all the General Public, which in turn misleads them. So, we planned on to create a comparison between what is real and what is shown. The implementation of the same follows !

1.1 Problem

How does the data available on news websites or wikipedia, provide misleading information about the situation of the Pandemic? And how does this affect the number of coronavirus cases?

1.2 Literature

Pearson's Correlation : https://en.wikipedia.org/wiki/Pearson_correlation_coefficient

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

r = correlation coefficient
x_i = values of the x-variable in a sample
$\bar{x}$ = mean of the values of the x-variable
y_i = values of the y-variable in a sample
$\bar{y}$ = mean of the values of the y-variable

Worldometer Data - https://www.worldometers.info/coronavirus/
Internet Archive - https://archive.org/advancedsearch.php

1.3 New idea

For this project, to find the relationship between the information and the cases, we made use of the Pearson's coefficient, which tells us about the linear correlation between the two graphs. This is a new way, because the other ways of comparing two graphs included making lists of fixed size and computing the difference between the values of y coordinates [elements of the lists], or by integrating the two graphs [finding the area under the graphs] & computing difference. These methods would have involved equivalent scaling between the two graphs. But, Pearson's Method helps us directly establish a relation between the two variables.

2 Method

We extracted the data for the total number of coronavirus cases on a daily basis, and all the news articles related to the coronavirus on different graphs. Then, we compared them with the help of the Pearson's coefficient. Pearson's Correlation Coefficient, is a statistical value that measures linear correlation between two variables X and Y. It gives a value between +1 and −1. A value of +1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation. We used the Y-coordinates of both the data sets [Graphs], and used Pearson's correlation coefficient to find correlation between them.

2.1 Implementation details

Programming language used is Python.
Libraries used:

1). requests
2). bs4
3). Json
4). datetime
5). pandas
6). matplotlib
7). numpy
8) scipy.stats

The essential data for initiating with the project was, the number of new coronavirus cases on a daily basis. To obtain this data, we selected "https://www.worldometers.info/" website. Using the beautiful soup (bs4) library, we scraped the data from the webpage "https://www.worldometers.info/coronavirus/". For the purpose of comparing the news articles published on the internet, we needed the number of articles/ web pages/pages related to coronavirus along with their published dates. To acquire this data, we searched "archive.org" which keeps an archive of the internet. By making use of the 'advanced search feature’ of archive.org, we can search for articles and web pages with specific keywords along with the date they were published on.

To search for coronavirus related articles we used the query *“coronavirus” OR “covid"*. This searched for all the articles containing either of these two above mentioned words. The process of getting the number of articles manually would take a lot of time. The website uses a "get" request to https://archive.org/advancedsearch.php constructed on the basis of the user’s input data, to get the result and we established the get request from python using the "requests" library. Using the datetime module, we extracted the data for each day.

We plotted the graph between a particular week and average no. of cases in that week by using matplotlib library.

After plotting the data on the graphs, the next step was to compare them, for this we used Pearson’s correlation method, the libraries used were numpy, scipy.stats. The function that plots the cross correlation between two variables is Matplotlib.pyplot.xcorr(). The value of Pearson's coefficient was extracted with the help of scipy.stats library importing pearsonr.

Code for the above implementation:-
https://docs.google.com/document/d/1dIpVA6YMeIOU7TkGixzBvDmjY4pkvQmKR-8rFQjjTEw/edit?usp=sharing

3 Results

3.1 Experiment findings

Observations regarding the graph :-

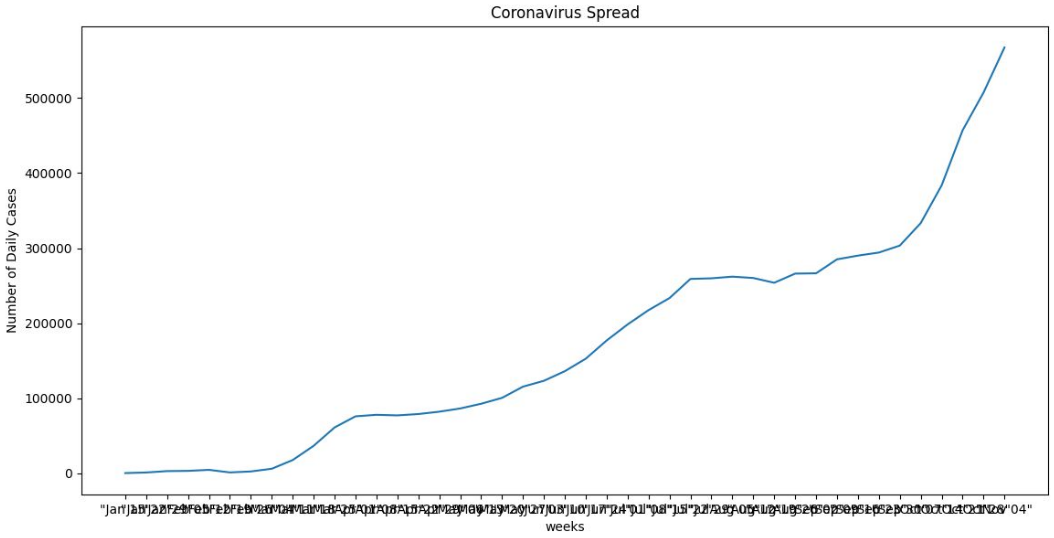


Fig 1: Graph of Daily Covid Cases (obtained from Worldometer)

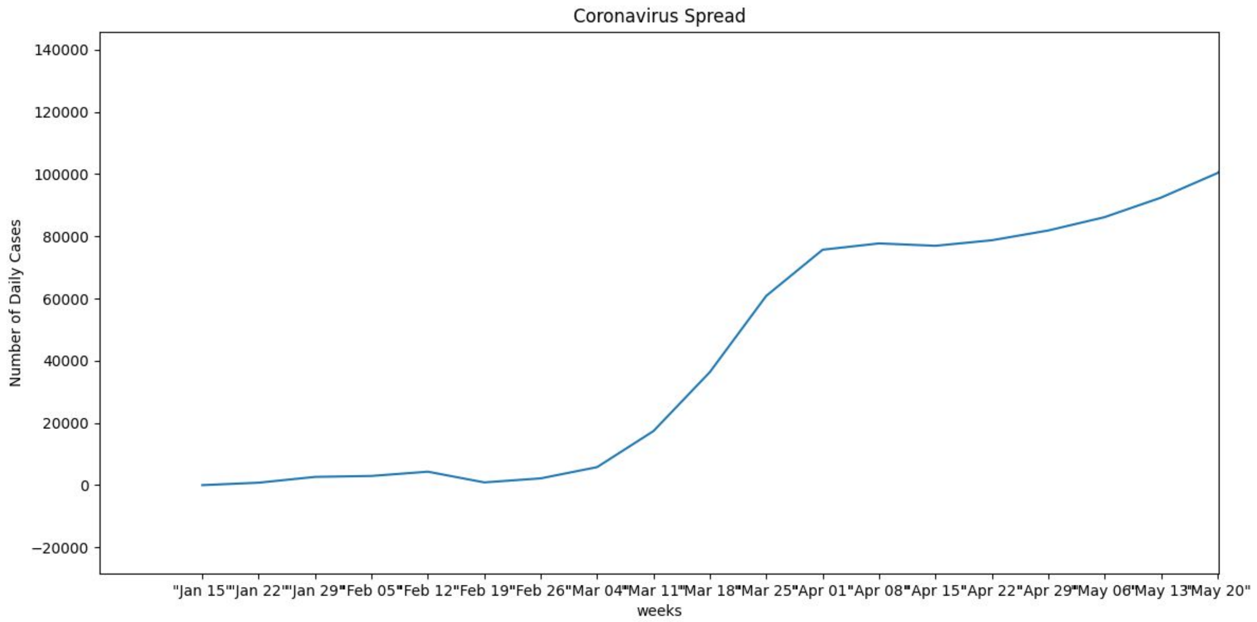


Fig 2: Zoom in version of Fig 1

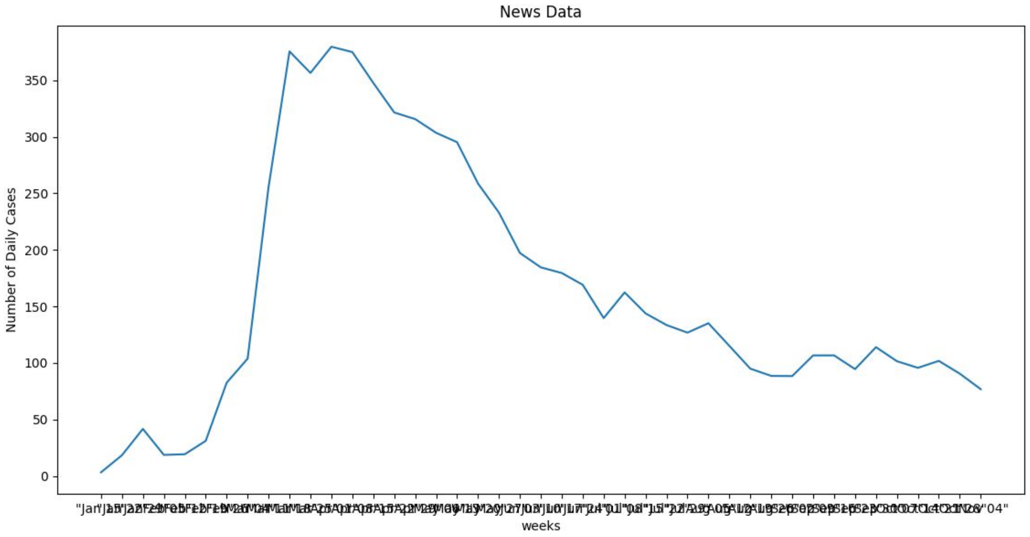


Fig 3: Graph of News Data

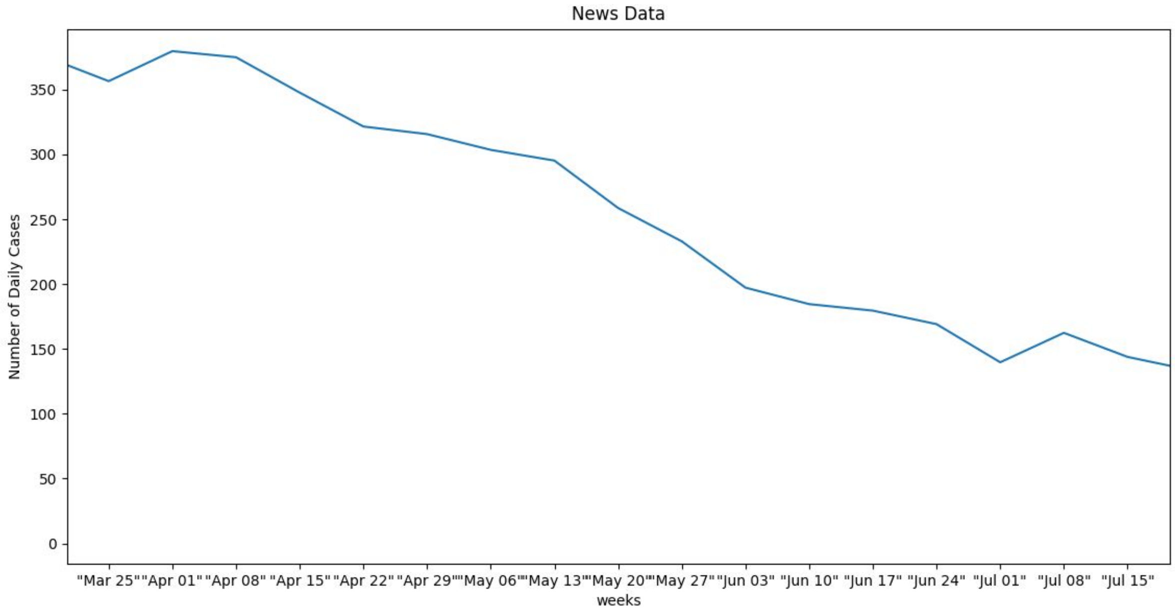
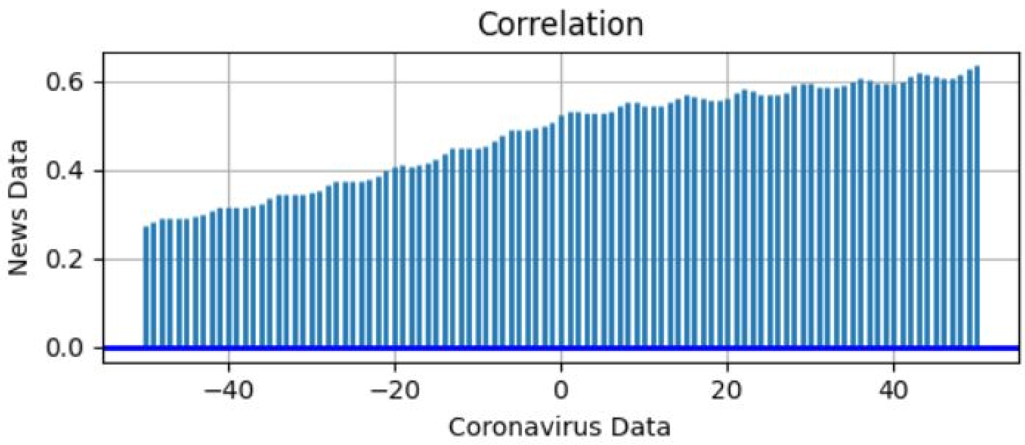


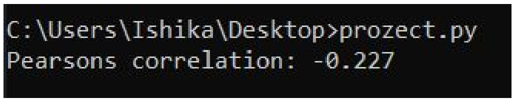
Fig 4: Zoom in Version of Fig 3 (from 25th March till 15th July)

For clear visualization of the above graphs visit:
Graphs - Google Docs

From the graphs we observe that at the very beginning when cases were quite low both the graphs were almost the same.
During April and May, news articles and web pages were very hyped about the corona as this was a new disease with no cures and no vaccines.
During June-July, we see that there is a drop in the news articles. It can be concluded that due to lax by them, people got misled into thinking that the virus is not as dangerous as before, thus, killing the authenticity of the information an individual might gather from the news. Which resulted in an increase in covid cases.
According to the worldometer graph the cases are increasing continuously, still news articles are decreasing as they are not covering the covid pandemic much during the last few months as it is quite old news and covering other things will help them in increasing their TRP.

Observations regarding the Pearson correlation coefficient:



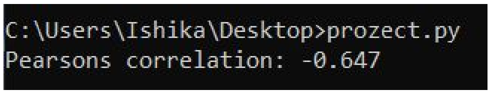


```
C:\Users\Ishika\Desktop>prozect.py
Pearsons correlation: -0.227
```

The comparison between the two above observed results (Graph), is conducted by Pearson's correlation coefficient, the value of the correlation coefficient as seen comes out to be negative, which shows negative correlation.

This implies that the two datasets vary very differently, and move in two opposite directions i.e while the actual data of cases is increasing the news data is revealing wrong stats.

3.2 Interpretation of findings



```
C:\Users\Ishika\Desktop>prozect.py
Pearsons correlation: -0.647
```

We found out that both the graphs were opposite in nature. After April, we can see that while the cases increase continuously, there is a decrease in the news articles. The overall correlation came out to be -0.227. Considering that initially, the news articles were increasing, the correlation value came out closer to 0 than expected. If we consider the correlation of both the graphs after April 20th, we see that the correlation value decreases to -0.647, which is expected, as the correlation must come closer to -1 for negative linear relation. This shows the difference between the slopes of the graphs. We can interpret that the information spread about the deadly virus is decreasing continuously. This, as expected, results in carelessness amongst the people regarding proper precautions, and thus, increase in the number of cases.

4 Conclusion

We learnt about a lot of new libraries used in python, their implementations, a whole lot of new errors & handling them. About how JSON format is used to store huge amounts of data online, and how to make requests to extract them. We also learned about how “payload” can be used to extract some specifically defined data.
Apart from the technical learnings from the project, our key takeaways include the realisation of how misleading information on the internet can be, if not analysed & viewed properly. And how this affects the society with so many hazardous consequences. The project brought awareness to us, for normal/general information ahead.

4.1 Team Work

We worked together as a team remarkably well, starting from shortlisting potential ideas for the project, to converging with a feasible one. We were really comfortable communicating with each other, which helped us a lot during the course of the project. We divided our chores always, before starting anything, even answering this question involved all of us opinions. We, in our perception,certainly did a great job.

Antara extracted the news data and coronavirus data used for the project.
Ruheen plotted the graphs of the data extracted
Ishika correlated the graphs between the data extracted.

We also planned on to showing the impact of the saturating data/stats i.e. how does the saturating data on news/wikipedia pages, is related to the number of coronavirus cases(increase or decrease) ; since information saturation leads to carelessness among people, them creating false interpretation of the hazards of the disease. But, we were not able to implement it because of constraints of correct data availability.

References

- Benesty J., Chen J., Huang Y., Cohen I. (2009) Pearson Correlation Coefficient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol 2. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-00296-0_5
- https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.xcorr.html
- https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html