PROJECT PHASE-II REPORT

ON

ANALYSIS AND PREDICTION OF EFFECT OF VIRAL DISEASE IN HUMAN +DEPLOYMENT

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted to:

Manish Sharma

Assistant Professor

Submitted by:

Ayan Bhatnagar- 190109900 - BDA
Kartikey Garg-180109045 - BDA
Pradyumn Bahukhandi-180109025 - BDA
Yash Sharma-180109032 - BDA

SCHOOL OF COMPUTING

DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

2020

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the Report, entitled **Analysis and Prediction of effect of Viral Diseases in Human + Deployment**, in partial fulfilment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of my work carried out during the period ***30/7/2020*** to ***22/10/2020*** under the guidance of **Manish Sharma Sir**.

Date: 15/03/2021

Signature of the Instructor

Manish Sharma

(Manish Sharma IBM Trainer)

Signature of the Candidates

Ayan Bhatnagar

Kartikey Garg

Kartikey Garg

Pradyumn

Pradyumn Bahukhandi

Yash Sharma

ACKNOWLEDGEMENT

It is a great opportunity for us to present this report on the partial completion of the University Group Project. We express our heartiest thanks to all the faculty members engaged directly or indirectly in guiding the making of this project. We would also like to thank **Manish Sharma Sir** who supervised and assisted us at every moment. We wish to utilize this valuable Project and the professional skills which we have learned in the Development and Deployment of this Project in our Professional Future.

Name: Ayan Bhatnagar

Roll No.: 190109900

SAP ID: 1000013728

ABSTRACT

This project will describe different aspects of virus, disease and presents a visualization of the spread of the infection and discusses the potential applications of data analytics components on this viral infection. Firstly, a literature survey and gathering of the dataset is done on viruses highlighting several factors including its origin, its similarity with the previous disease, its transmission capacity, its symptoms, etc. Secondly, data analytics is applied to a dataset of respective to data collection and find out the spread of the viral infection, the worldwide increase in the number of confirmed cases over time is modeled. Thirdly the application deployment will reduce our project on a platform, and it will work on real-time data.

LIST OF FIGURES

TABLE OF CONTENT

Chapter 1 -- Introduction

The project aims to provide details to each respective user about how infectious a virus on the human body with certain pre-assumed characteristics like an unhealthy body or under an age group that will help in predicting future occurrences and consequences.

Further, the reason that led us to choose this topic for Project is not only because it is trending but also with the help of this, we can provide a detailed analysis of further and ongoing diseases that can make research and knowledge better for the user as well as the application.

Detection of highly divergent or yet unknown viruses from datasets is a challenging task. When human samples are sequenced, a substantial proportion of assembled conventional methods find no similarity to known sequences. We wished to explore whether machine learning algorithms could improve the detection of viral sequences in metagenomic sequencing data by performing research on the dataset of viruses like Plague, Influenza, Dengue, Ebola, Hantavirus, COVID-19, etc.

Chapter 2 –Project Description

2.1. Purpose

- Analysis of this dataset can help us to understand human symptoms.
- It helps us figure out the future scope of viral diseases and their symptoms too.
- Not only this for an accuracy of 80% this limit bounded analysis can be helpful for further future research wherein they would have prospects like AI in health and neurology.

2.3. Problem statement

Fighting with the diseases is necessary for human Survival and Development, so a platform must be there that helps humans analysing the future scope of diseases.
Looking at current Scenarios we were eager to work on a project that has future used. Working on this project after completing this project will help in developing further research and development.

2.4. Special Features

This project deals with the understanding of diseases the cause of harmful effects on human civilization. It aims to visualize the future scope and present scope of viruses on humans.

Chapter 3 –Tools and Technologies

1. IBM Cognos

We will be using the IBM Cognos for the analysing the data and give the meaningful outcome. In IBM Cognos analysis studio we can use for multidimensional analysis and exploration of large data sources and it is user friendly interactive environment to answers the data query.

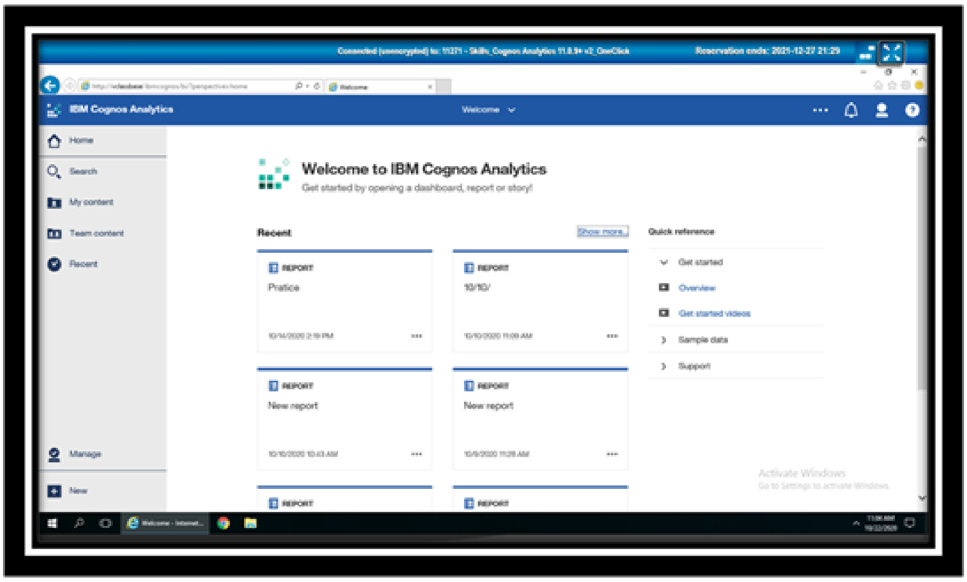


Figure 1: IBM Cognos

2. Exploration

During our Phase 2 process we will be using OLAP (Online Analytical Processing) exploration refer to the term dicing and slicing of the data. For example, we may look the death ratio for year 2006 to 2008 by gender. You may notice the rise or drop in the death ratio by clicking on the 2006 we can drill down to show the death result by quarter to 2006. It will help us to focus on the data that can answer our queries.

- Visualization

We will be using the visualization to communicate comparisons between relationships and our query. It will emphasize and clarify our data. Forecasting our data modelling which will be corresponding to visualization.

3. Analysing Large Data

In Analysis studio it will helps us to find the meaningful details while keeping summaries in view to maintain a clear overview of our data.

- **NumPy**

NumPy full form is Numerical Python is a perfect tool for performing basic and scientific computing and advanced array operations.

Figure 2: Python NumPy

It offers many useful features for performing many operations on n-arrays and matrices in Python. It helps to process arrays that store same data type values, and makes math operations on arrays much easier. The vectorization of mathematical operations on the NumPy array type can increases performance and accelerates the execution time of any data.

- **Pandas**

Pandas is a library will helped us to work with labelled and relational data and query counterintuitive. It is based on two data structures: "Series" which is one-dimensional, like items list and "Data Frames" – which is two-dimensional, like a table with multiple columns. It enables us to convert the data structures to Data Frame objects, and also allows the handling of missing data and adding/deleting columns from the Data Frame, imputing the missing files, and plotting the data with histogram or plot box. It’s a must-have for data manipulation, visualization, and wrangling.

Figure 3: Python Pandas

- **Matplotlib**

  Matplotlib is a library used in python programming language and it is also used for the numerical, mathematical extension of the NumPy. It also provides an object-oriented API for embedding plots into data sets and the application for using general purpose GUI toolkit

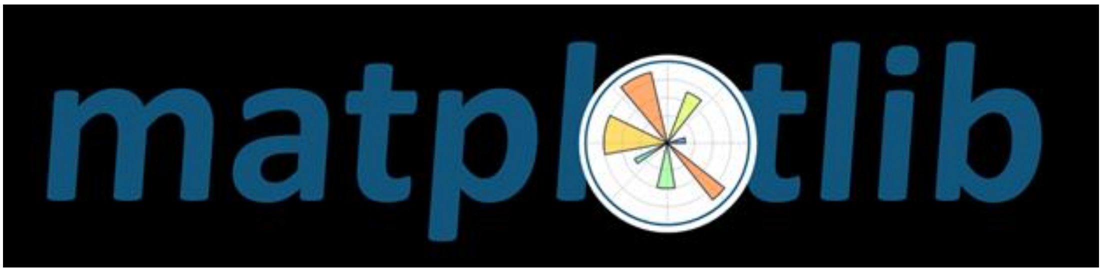


Figure 4: Python matplotlib

- **Tkinter**

Figure 5: Python Tkinter

- **wxPython**

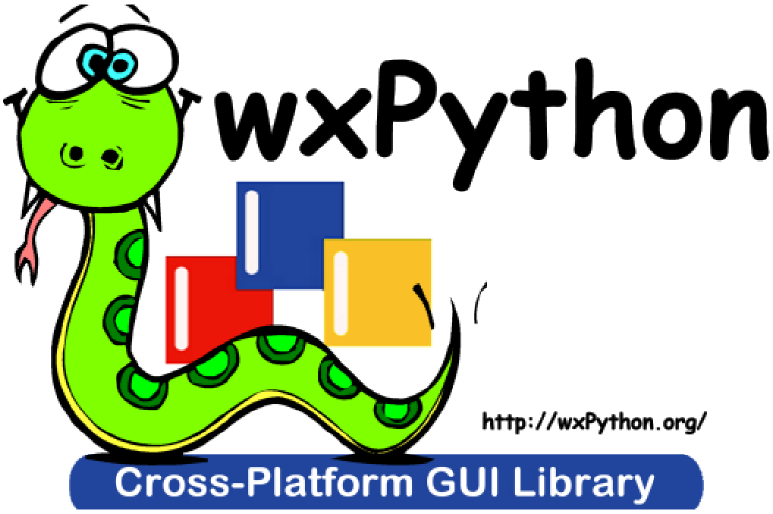


Figure 6: wxPython

- **Qt**

Figure 7: Python Qt

- **GTK+.**

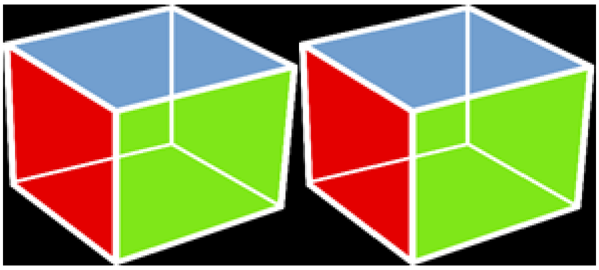
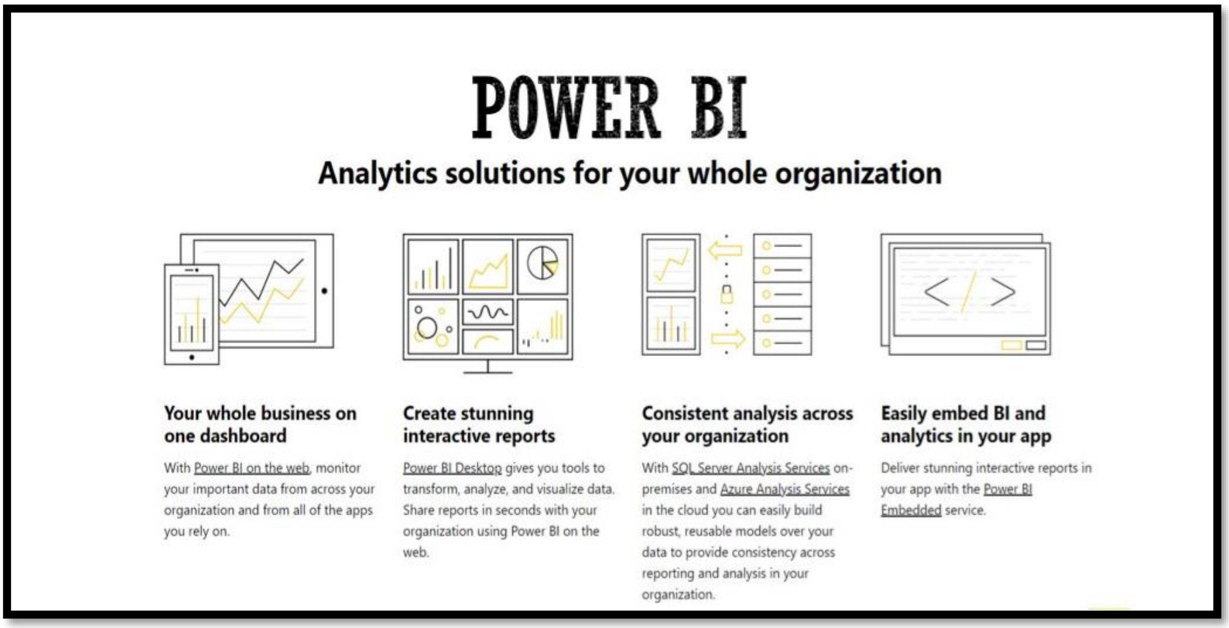


Figure 8: GTK

- **Power BI**

In Power Bi we are able to apply forecasting prediction, and hindcasting, while visualizing and exploring the information. Forecasting in Power Bi utilizes inherent predictive statement models using exponential smoothing to mechanically notice seasonality within the knowledge to supply forecast results from a series of information. We can also explore forecast results by adjusting the specified confidence interval or by adjusting outlier knowledge to examine how they have an effect on results. we can conjointly hindcast to examine whether Power Bi would have expected this and up to date past supported older knowledge.

The best knowledge for statement is statistic knowledge or uniformly increasing whole numbers.



POWER BI

Analytics solutions for your whole organization

Your whole business on one dashboard

With Power BI on the web, monitor your important data from across your organization and from all of the apps you rely on.

Create stunning interactive reports

Power BI Desktop gives you tools to transform, analyze, and visualize data. Share reports in seconds with your organization using Power BI on the web.

Consistent analysis across your organization

With SQL Server Analysis Services on-premises and Azure Analysis Services in the cloud you can easily build robust, reusable models over your data to provide consistency across reporting and analysis in your organization.

Easily embed BI and analytics in your app

Deliver stunning interactive reports in your app with the Power BI Embedded service.

Figure 9: Power BI

- **Scikit-Learn (Sklearn)**

Scikit-learn (Sklearn) is the most helpful and sturdy library for machine learning in Python. It provides a variety of economical tools for machine learning and applied math modelling together with classification, regression, clustering and spatial property reduction via a consistence interface in Python. This library, that is essentially written in Python, is constructed upon NumPy, SciPy and Matplotlib.

Figure 10: Scikit – Learn

- **Seaborn Library**

Seaborn is a library for creating applied math graphics in Python. It builds on high of matplotlib and integrates closely with pandas knowledge structures.
Seaborn helps you explore and perceive your knowledge. Its plotting functions operate data frames and arrays containing whole datasets and internally perform the required linguistics mapping and applied math aggregation to supply informative plots. Its dataset-oriented, declarative API allows you to specialize in what the various parts of your plots mean, instead of on the small instruction or the way to draw them.

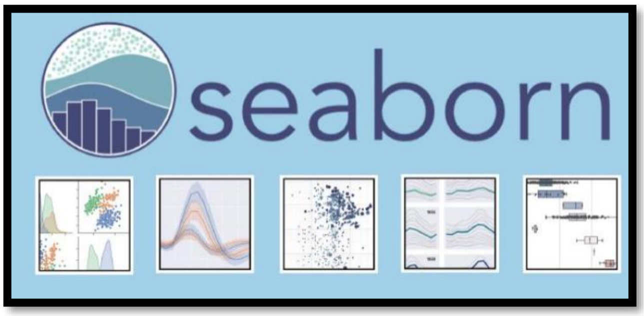


Figure 11: Seaborn

Chapter 4- Implementation Modules and Screen Shots

Pneumonia and diarrhoea

PenuCare is the name used for Pneumonia similarly Zinc name is used for Diarrhoea

The concept used here is Many-to-Many Joins in Tableau software and the primary key is Countries and Areas. The platform, used here is Tableau. The data is imported from online sources.

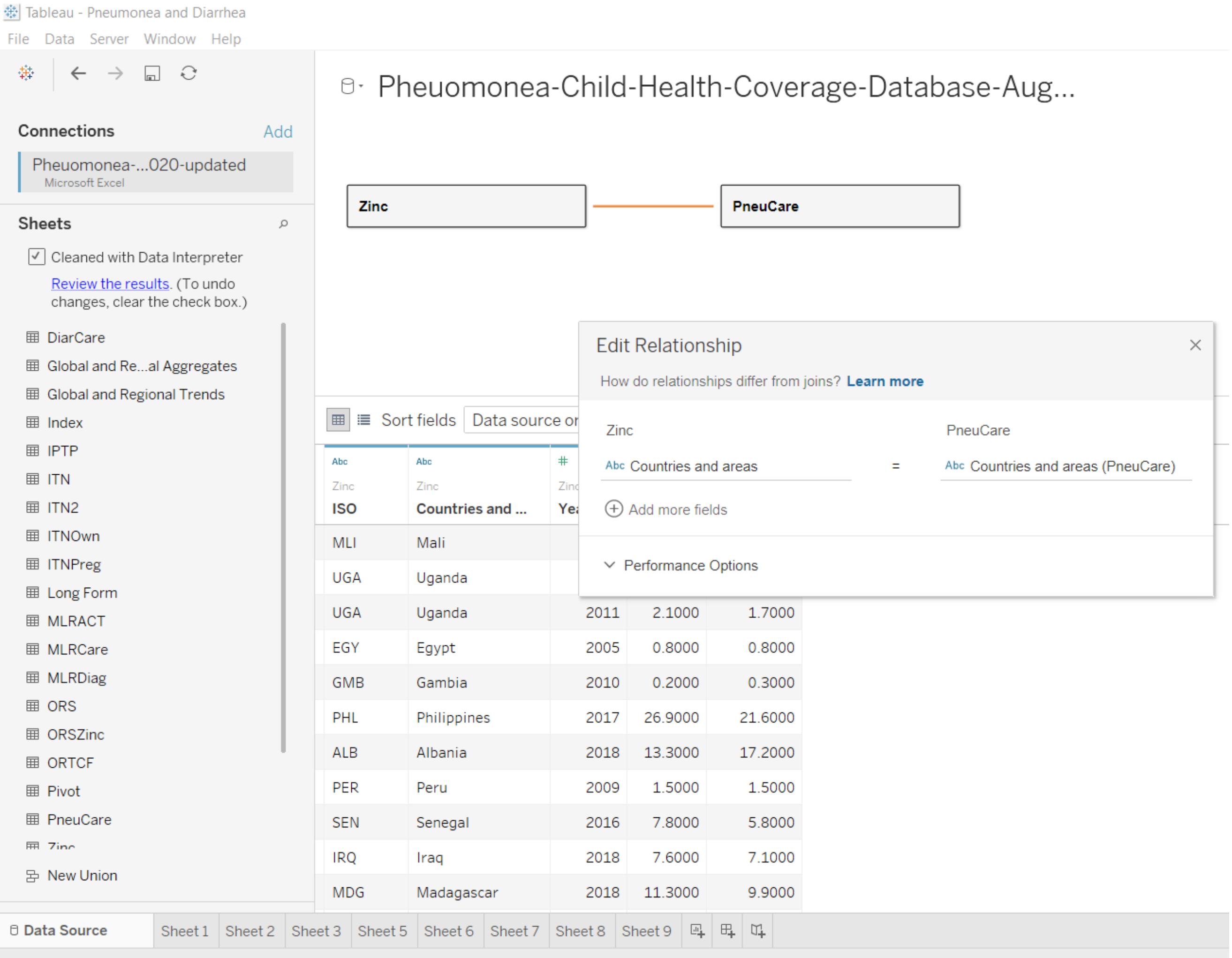


Tableau - Pneumonea and Diarrhea

File Data Server Window Help

Pheuomonea-Child-Health-Coverage-Database-Aug...

Connections Add

Pheuomonea-...020-updated
Microsoft Excel

Sheets

☑ Cleaned with Data Interpreter

Review the results. (To undo changes, clear the check box.)

- DiarCare
- Global and Re...al Aggregates
- Global and Regional Trends
- Index
- IPTP
- ITN
- ITN2
- ITNOwn
- ITNPreg
- Long Form
- MLRACT
- MLRCare
- MLRDiag
- ORS
- ORSZinc
- ORTCF
- Pivot
- PneuCare
- Zinc
- New Union

Zinc — PneuCare

Edit Relationship

How do relationships differ from joins? Learn more

Zinc		PneuCare
Countries and areas	=	Countries and areas (PneuCare)

Add more fields

Performance Options

Sort fields Data source or

ISO	Countries and ...	Yea		
MLI	Mali			
UGA	Uganda			
UGA	Uganda	2011	2.1000	1.7000
EGY	Egypt	2005	0.8000	0.8000
GMB	Gambia	2010	0.2000	0.3000
PHL	Philippines	2017	26.9000	21.6000
ALB	Albania	2018	13.3000	17.2000
PER	Peru	2009	1.5000	1.5000
SEN	Senegal	2016	7.8000	5.8000
IRQ	Iraq	2018	7.6000	7.1000
MDG	Madagascar	2018	11.3000	9.9000

Data Source | Sheet 1 | Sheet 2 | Sheet 3 | Sheet 5 | Sheet 6 | Sheet 7 | Sheet 8 | Sheet 9

Figure 12: Concept of joins used in Dataset

- Below is the Representation of a Comparative study between Males and females suffering from Diarrhoea.

- As seen in figure 1 the number of Cases increases Heavily in the Year 2013-14 and similarly in 2015-16 and 2017-18. Also, there's a gradual decrease in these cases. But the thing that remains the same is the comparison of males to females.

- Conclusion: There’s not a very large difference between the Ratio of Males and Females.

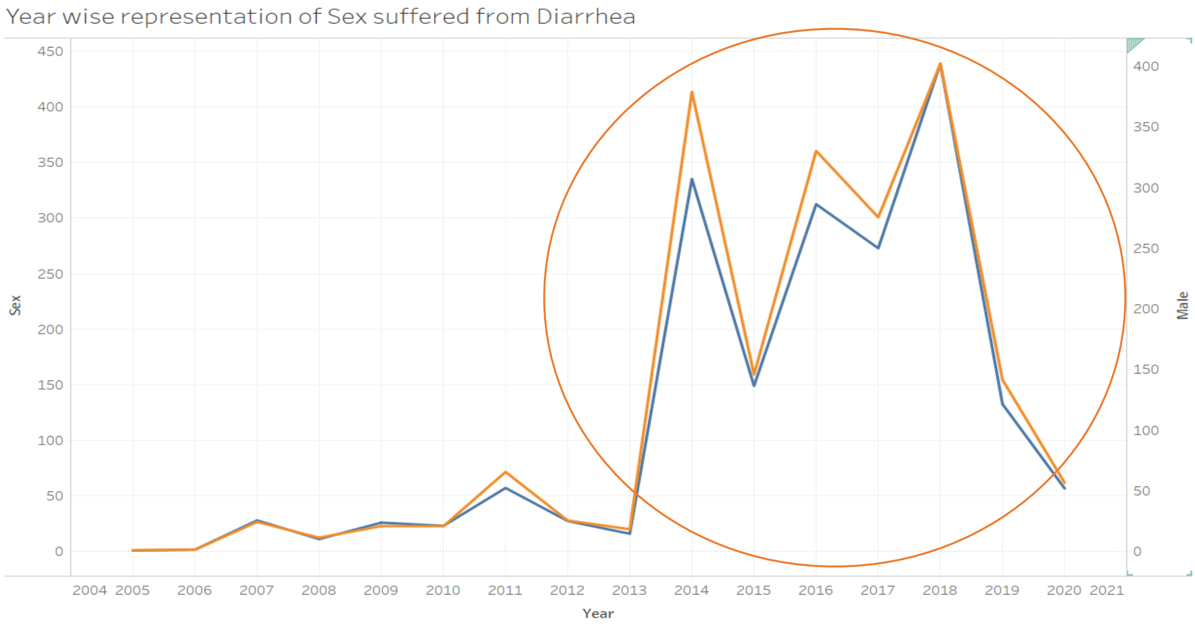


Figure 13: Year-wise representation of Gender suffered from Diarrhoea

- Another representation of a regional comparison between males suffering from Diarrhoea and Pneumonia. We can see here that the places like Guyana, Peru, Bhutan, Comoros have low patients suffering from diarrhoea rather than Pneumonia. A place like Peru has an extremely large number of Cases of Pneumonia and vice versa for Diarrhoea.

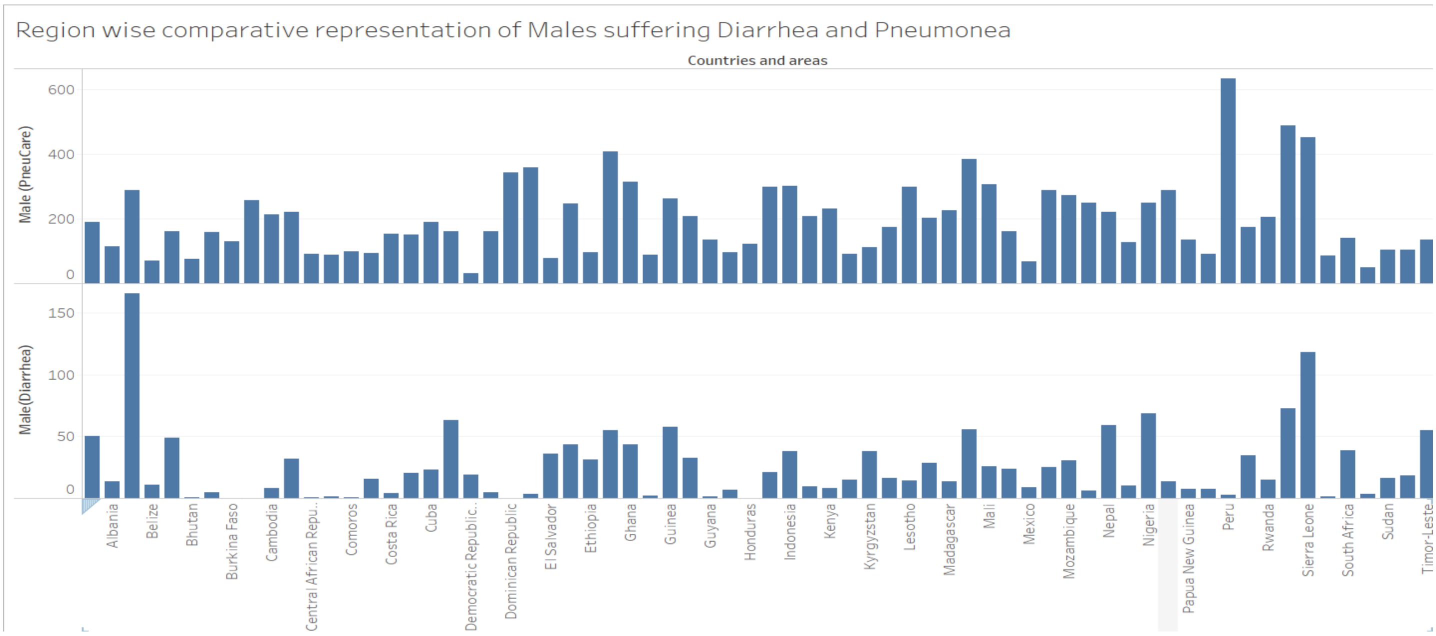


Figure 14: Region-wise comparative representation of Males suffering Diarrhoea and Pneumonia

Below is the Comparative representation of gender suffering from Diarrhoea and pneumonia we can see that the ratio of female to that female pneumonia to that of diarrhoea is very high and in some cases their proportion to that of males suffering from Diarrhoea and pneumonia.

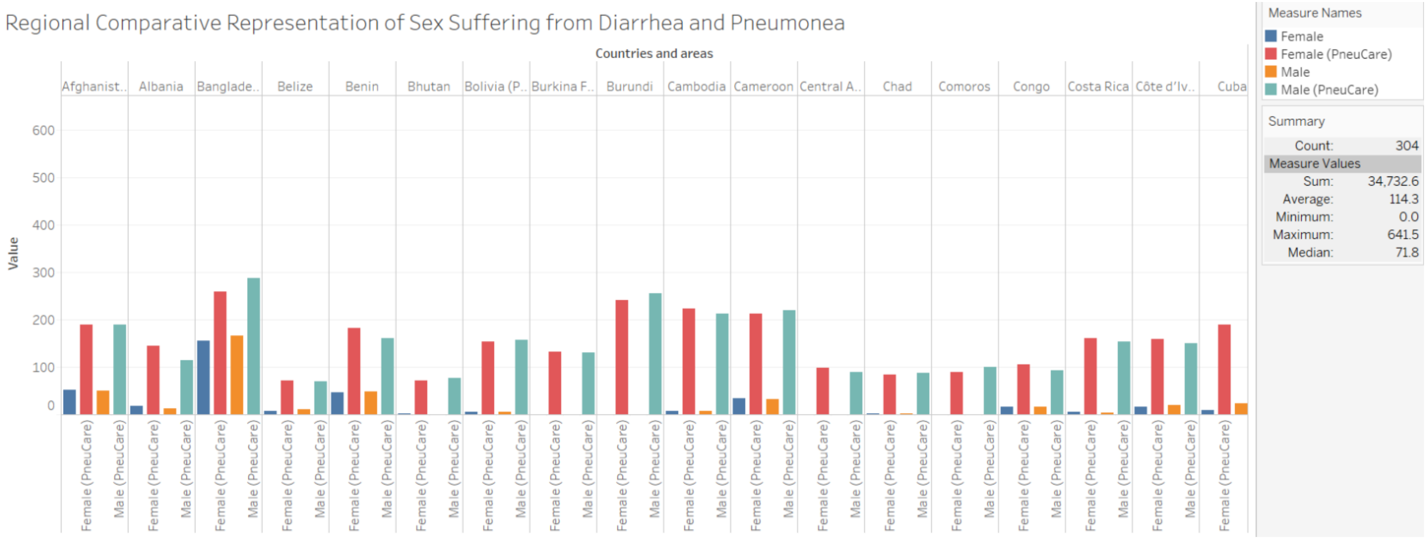
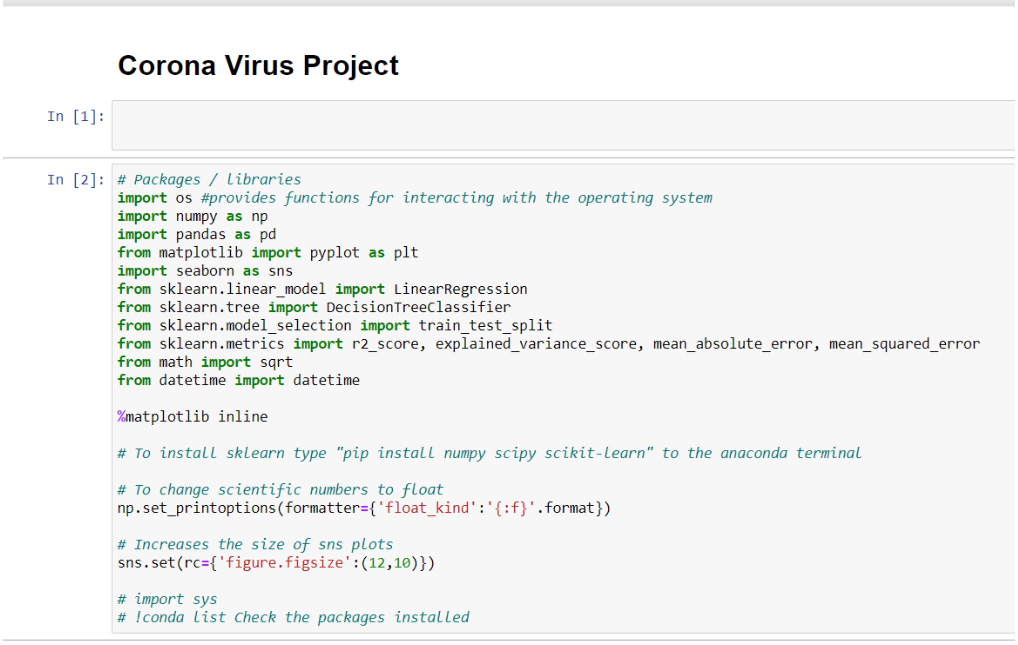


Figure 15: Regional Comparison of gender suffering from Diarrhoea and Pneumonia

CORONA VIRUS

Here we have imported the required packages and libraries, mostly we have used pandas for data manipulation and analysis, NumPy for working with array, and datetime for manipulating date and time.

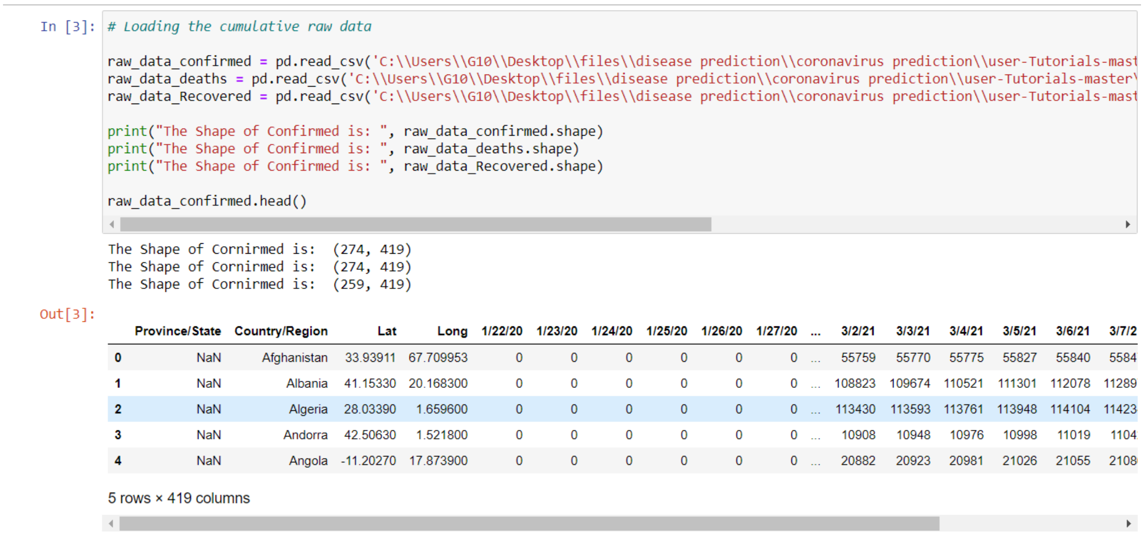


Corona Virus Project

```
In [1]:

In [2]: # Packages / libraries
        import os #provides functions for interacting with the operating system
        import numpy as np
        import pandas as pd
        from matplotlib import pyplot as plt
        import seaborn as sns
        from sklearn.linear_model import LinearRegression
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import r2_score, explained_variance_score, mean_absolute_error, mean_squared_error
        from math import sqrt
        from datetime import datetime

        %matplotlib inline

        # To install sklearn type "pip install numpy scipy scikit-learn" to the anaconda terminal

        # To change scientific numbers to float
        np.set_printoptions(formatter={'float_kind':'{:f}'.format})

        # Increases the size of sns plots
        sns.set(rc={'figure.figsize':(12,10)})

        # import sys
        # !conda list Check the packages installed
```

Figure 16: Importing the required packages and libraries

Here we have imported three datasets namely confirmed, death, and recovered using pandas library to extract the useful information, all the datasets are from 22nd January 2020 till 7th March 2020. The shape function is used to know the exact number of rows and columns present in the datasets and the head function is used to fetch and display the first 5 rows of the dataset.



```
In [3]: # Loading the cumulative raw data

        raw_data_confirmed = pd.read_csv('C:\\Users\\G10\\Desktop\\files\\disease prediction\\coronavirus prediction\\user-Tutorials-mast
        raw_data_deaths = pd.read_csv('C:\\Users\\G10\\Desktop\\files\\disease prediction\\coronavirus prediction\\user-Tutorials-master\
        raw_data_Recovered = pd.read_csv('C:\\Users\\G10\\Desktop\\files\\disease prediction\\coronavirus prediction\\user-Tutorials-mast

        print("The Shape of Confirmed is: ", raw_data_confirmed.shape)
        print("The Shape of Confirmed is: ", raw_data_deaths.shape)
        print("The Shape of Confirmed is: ", raw_data_Recovered.shape)

        raw_data_confirmed.head()

        The Shape of Cornirmed is:  (274, 419)
        The Shape of Cornirmed is:  (274, 419)
        The Shape of Cornirmed is:  (259, 419)

Out[3]:
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	3/2/21	3/3/21	3/4/21	3/5/21	3/6/21	3/7/2
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	55759	55770	55775	55827	55840	5584
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	108823	109674	110521	111301	112078	11289
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	113430	113593	113761	113948	114104	11423
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	10908	10948	10976	10998	11019	1104
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	20882	20923	20981	21026	21055	2108

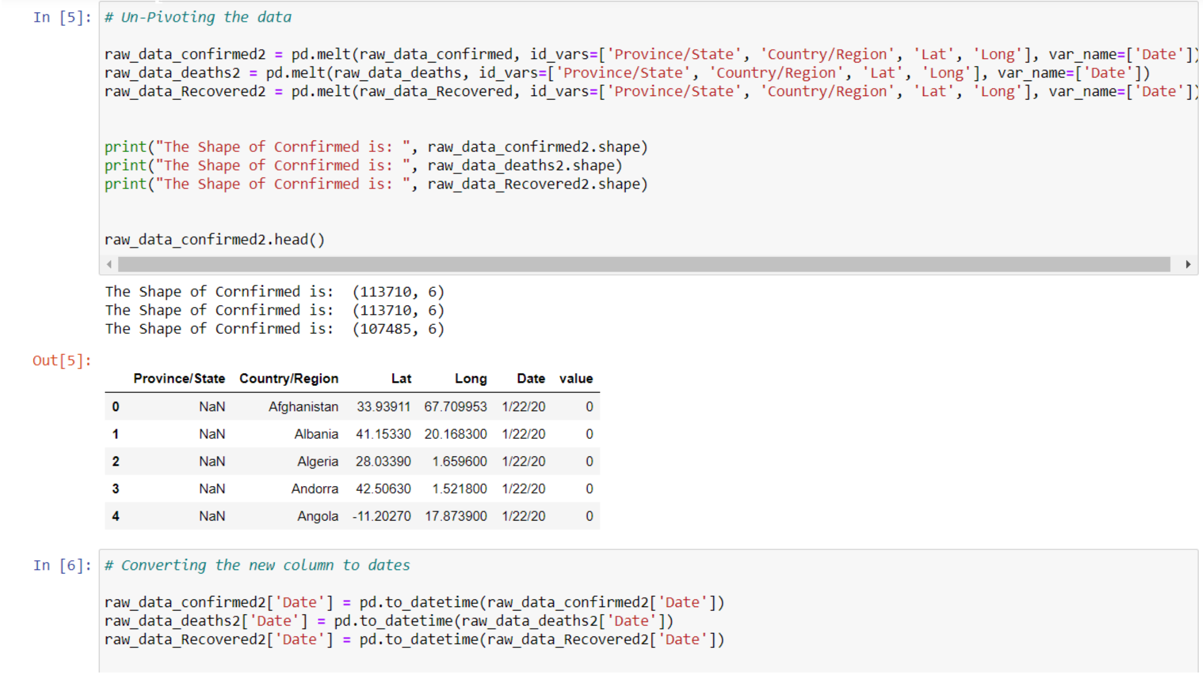
5 rows × 419 columns

Figure 17: Loading confirmed, deaths, recovered datasets using pandas library.

Data transformation means we are converting the data from one format to another.

1. Unpivoting of data

Since initially, the dataset contains dates in columns so we try to unpivot the data, hereby unpivoting the data means we will take all the dates in columns and put them as new rows. Melt function is useful to convert a Data Frame into a format where one or more columns are identifier variables.



```
In [5]: # Un-Pivoting the data

        raw_data_confirmed2 = pd.melt(raw_data_confirmed, id_vars=['Province/State', 'Country/Region', 'Lat', 'Long'], var_name=['Date'])
        raw_data_deaths2 = pd.melt(raw_data_deaths, id_vars=['Province/State', 'Country/Region', 'Lat', 'Long'], var_name=['Date'])
        raw_data_Recovered2 = pd.melt(raw_data_Recovered, id_vars=['Province/State', 'Country/Region', 'Lat', 'Long'], var_name=['Date'])


        print("The Shape of Cornfirmed is: ", raw_data_confirmed2.shape)
        print("The Shape of Cornfirmed is: ", raw_data_deaths2.shape)
        print("The Shape of Cornfirmed is: ", raw_data_Recovered2.shape)


        raw_data_confirmed2.head()

        The Shape of Cornfirmed is:  (113710, 6)
        The Shape of Cornfirmed is:  (113710, 6)
        The Shape of Cornfirmed is:  (107485, 6)
Out[5]:
```

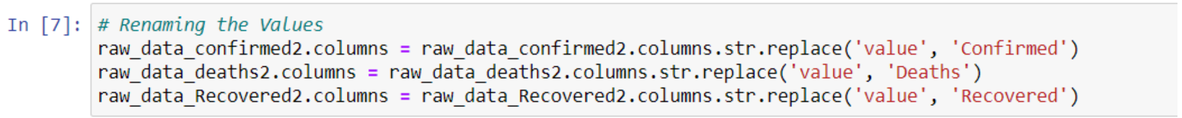
	Province/State	Country/Region	Lat	Long	Date	value
0	NaN	Afghanistan	33.93911	67.709953	1/22/20	0
1	NaN	Albania	41.15330	20.168300	1/22/20	0
2	NaN	Algeria	28.03390	1.659600	1/22/20	0
3	NaN	Andorra	42.50630	1.521800	1/22/20	0
4	NaN	Angola	-11.20270	17.873900	1/22/20	0

```
In [6]: # Converting the new column to dates

        raw_data_confirmed2['Date'] = pd.to_datetime(raw_data_confirmed2['Date'])
        raw_data_deaths2['Date'] = pd.to_datetime(raw_data_deaths2['Date'])
        raw_data_Recovered2['Date'] = pd.to_datetime(raw_data_Recovered2['Date'])
```

Figure 18: Data Extraction

2. Renaming of columns

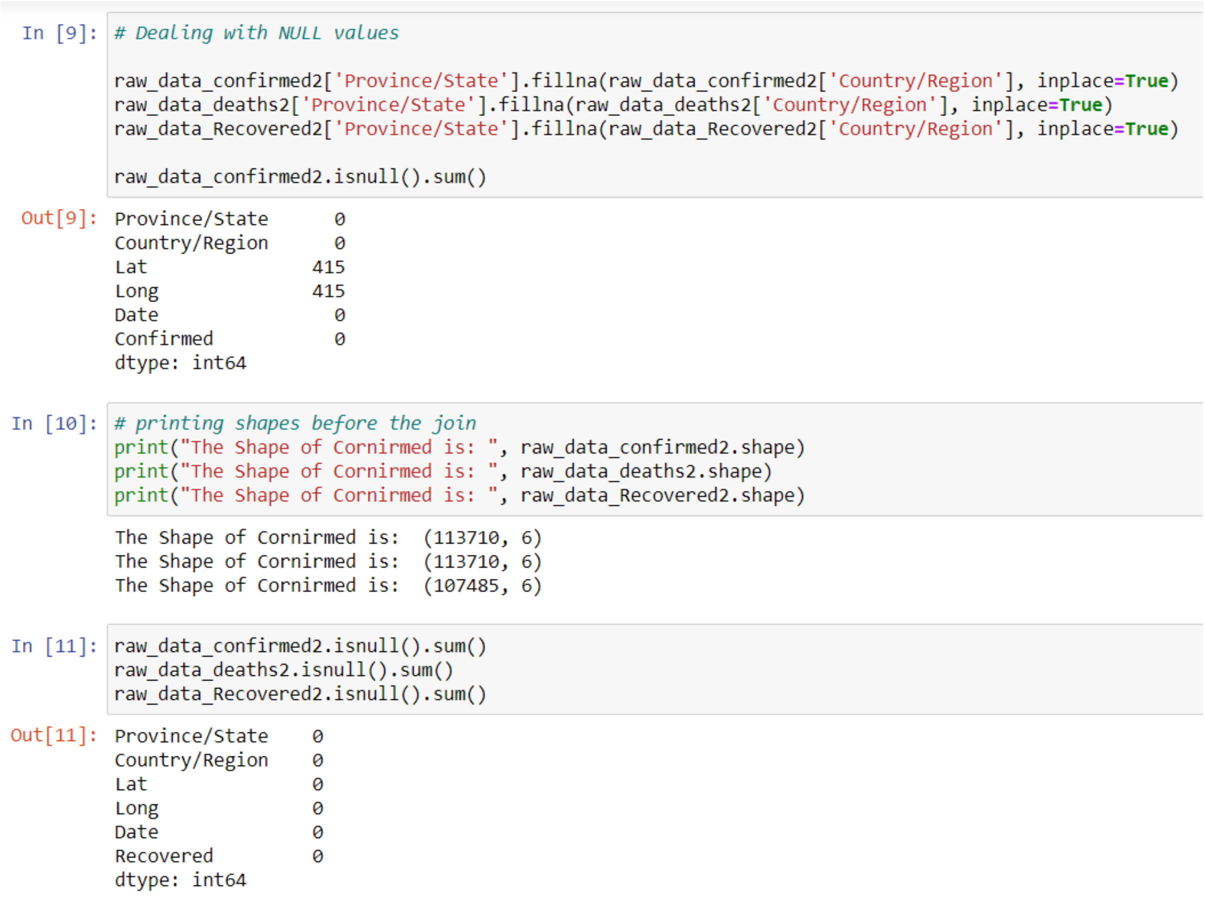
We have 3 different datasets and we have a value column in all these 3 datasets so we just have to point values to the right mapping so for the first dataset which is confirmed for this we are renaming the value column to confirm, similarly for other datasets we have replaced the value column.



```
In [7]: # Renaming the Values
        raw_data_confirmed2.columns = raw_data_confirmed2.columns.str.replace('value', 'Confirmed')
        raw_data_deaths2.columns = raw_data_deaths2.columns.str.replace('value', 'Deaths')
        raw_data_Recovered2.columns = raw_data_Recovered2.columns.str.replace('value', 'Recovered')
```

Figure 19: Data transformation

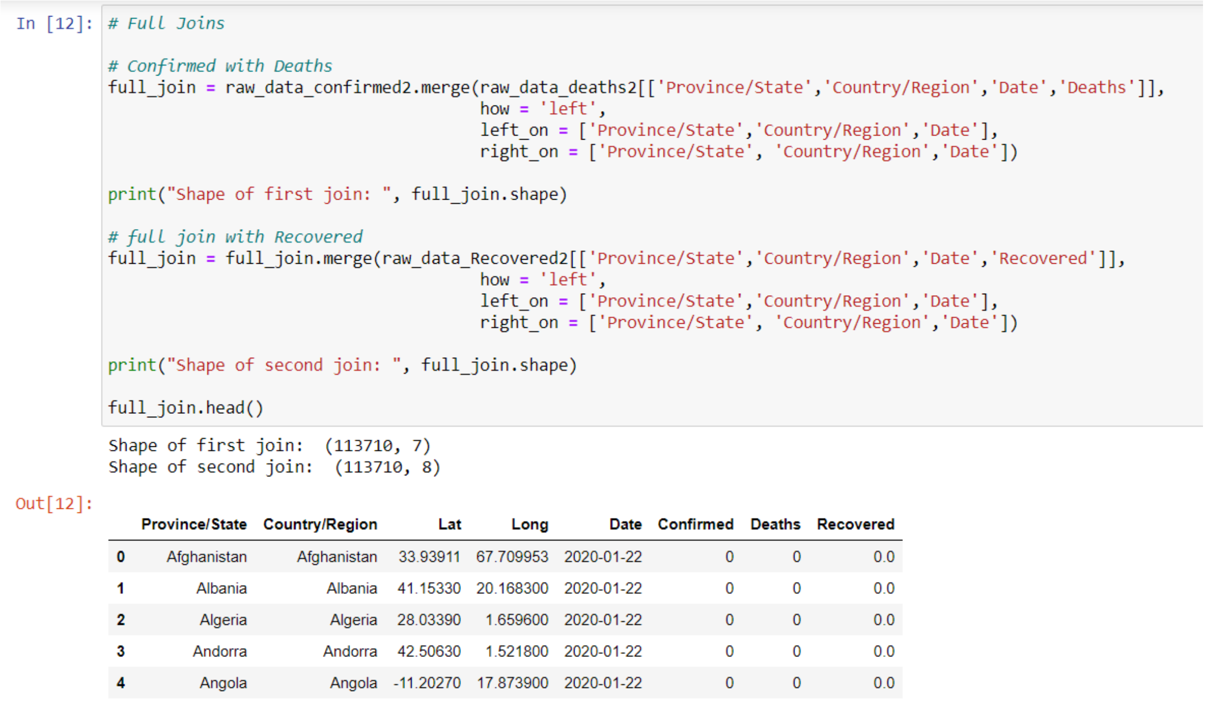
- Data cleaning: It is the process of removing inappropriate, duplicated values or fields and filling the missing values.Here we have done the data cleaning part and also check the shapes i.e row or columns before performing the joins.



```
In [9]: # Dealing with NULL values

        raw_data_confirmed2['Province/State'].fillna(raw_data_confirmed2['Country/Region'], inplace=True)
        raw_data_deaths2['Province/State'].fillna(raw_data_deaths2['Country/Region'], inplace=True)
        raw_data_Recovered2['Province/State'].fillna(raw_data_Recovered2['Country/Region'], inplace=True)

        raw_data_confirmed2.isnull().sum()

Out[9]: Province/State      0
        Country/Region      0
        Lat               415
        Long              415
        Date                0
        Confirmed           0
        dtype: int64

In [10]: # printing shapes before the join
         print("The Shape of Cornirmed is: ", raw_data_confirmed2.shape)
         print("The Shape of Cornirmed is: ", raw_data_deaths2.shape)
         print("The Shape of Cornirmed is: ", raw_data_Recovered2.shape)

         The Shape of Cornirmed is:  (113710, 6)
         The Shape of Cornirmed is:  (113710, 6)
         The Shape of Cornirmed is:  (107485, 6)

In [11]: raw_data_confirmed2.isnull().sum()
         raw_data_deaths2.isnull().sum()
         raw_data_Recovered2.isnull().sum()

Out[11]: Province/State    0
         Country/Region    0
         Lat               0
         Long              0
         Date              0
         Recovered         0
         dtype: int64
```

Figure 20: Data cleaning

- After performing data cleaning since we have 3 datasets we have joined them using the full join

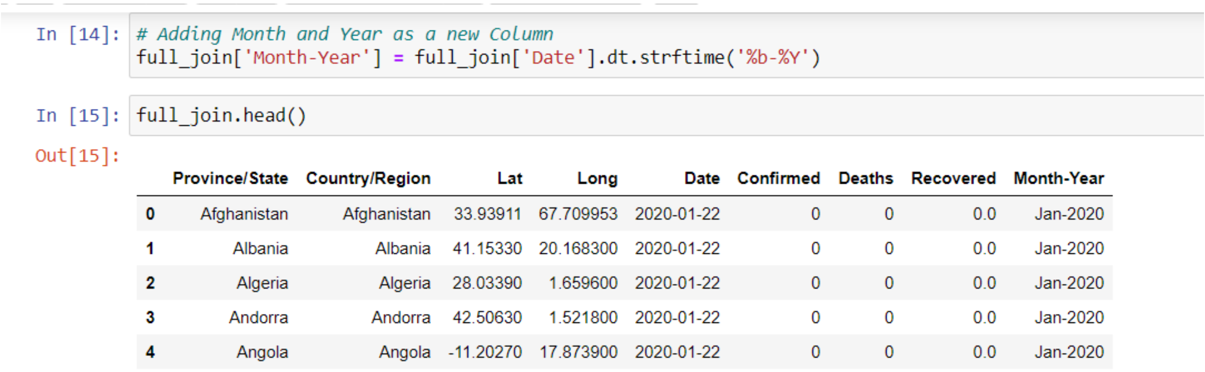


```
In [12]: # Full Joins

         # Confirmed with Deaths
         full_join = raw_data_confirmed2.merge(raw_data_deaths2[['Province/State','Country/Region','Date','Deaths']],
                                               how = 'left',
                                               left_on = ['Province/State','Country/Region','Date'],
                                               right_on = ['Province/State', 'Country/Region','Date'])

         print("Shape of first join: ", full_join.shape)

         # full join with Recovered
         full_join = full_join.merge(raw_data_Recovered2[['Province/State','Country/Region','Date','Recovered']],
                                               how = 'left',
                                               left_on = ['Province/State','Country/Region','Date'],
                                               right_on = ['Province/State', 'Country/Region','Date'])

         print("Shape of second join: ", full_join.shape)

         full_join.head()

         Shape of first join:  (113710, 7)
         Shape of second join:  (113710, 8)

Out[12]:
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered
0	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0.0
1	Albania	Albania	41.15330	20.168300	2020-01-22	0	0	0.0
2	Algeria	Algeria	28.03390	1.659600	2020-01-22	0	0	0.0
3	Andorra	Andorra	42.50630	1.521800	2020-01-22	0	0	0.0
4	Angola	Angola	-11.20270	17.873900	2020-01-22	0	0	0.0

Figure21: Performing full joins on datasets

- Here we have used the strftime function of python in order to format the date column as Month-Year

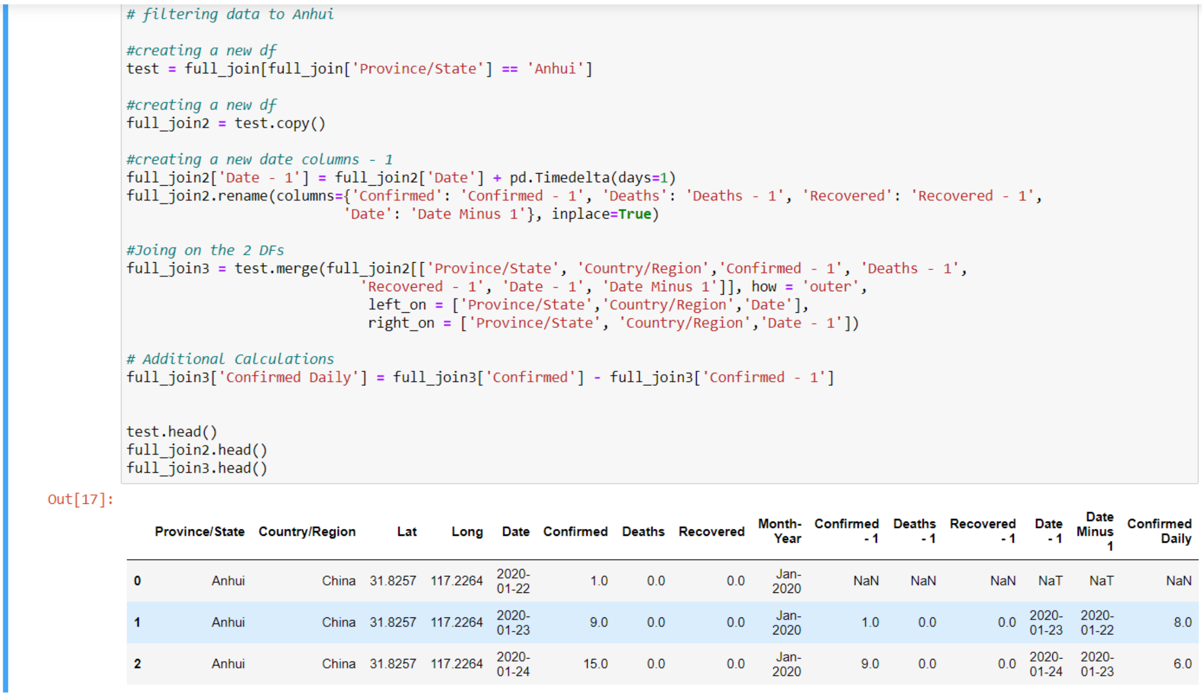


```
In [14]: # Adding Month and Year as a new Column
         full_join['Month-Year'] = full_join['Date'].dt.strftime('%b-%Y')

In [15]: full_join.head()
Out[15]:
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Month-Year
0	Afghanistan	Afghanistan	33.93911	67.709953	2020-01-22	0	0	0.0	Jan-2020
1	Albania	Albania	41.15330	20.168300	2020-01-22	0	0	0.0	Jan-2020
2	Algeria	Algeria	28.03390	1.659600	2020-01-22	0	0	0.0	Jan-2020
3	Andorra	Andorra	42.50630	1.521800	2020-01-22	0	0	0.0	Jan-2020
4	Angola	Angola	-11.20270	17.873900	2020-01-22	0	0	0.0	Jan-2020

Figure22: Adding Month and Year as a new column

Data Testing: We have performed data testing by filtering the data to Anhui state and also created extra columns like Deaths-1, Recovered-1, Date Minus 1 for testing purposes only and performed joins with these extra columns.

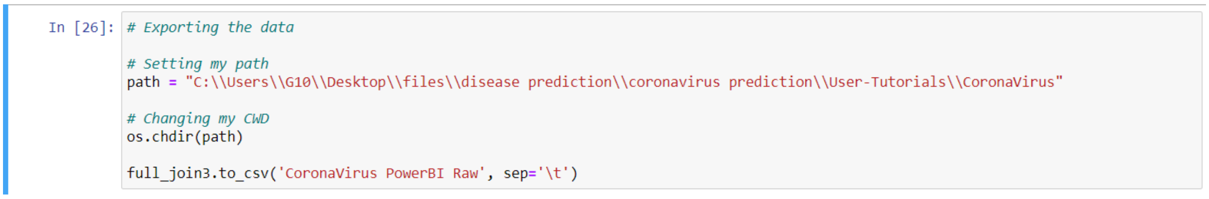


```
# filtering data to Anhui

#creating a new df
test = full_join[full_join['Province/State'] == 'Anhui']

#creating a new df
full_join2 = test.copy()

#creating a new date columns - 1
full_join2['Date - 1'] = full_join2['Date'] + pd.Timedelta(days=1)
full_join2.rename(columns={'Confirmed': 'Confirmed - 1', 'Deaths': 'Deaths - 1', 'Recovered': 'Recovered - 1',
                           'Date': 'Date Minus 1'}, inplace=True)

#Joing on the 2 DFs
full_join3 = test.merge(full_join2[['Province/State', 'Country/Region','Confirmed - 1', 'Deaths - 1',
                              'Recovered - 1', 'Date - 1', 'Date Minus 1']], how = 'outer',
                               left_on = ['Province/State','Country/Region','Date'],
                               right_on = ['Province/State', 'Country/Region','Date - 1'])

# Additional Calculations
full_join3['Confirmed Daily'] = full_join3['Confirmed'] - full_join3['Confirmed - 1']


test.head()
full_join2.head()
full_join3.head()
Out[17]:
```

	Province/State	Country/Region	Lat	Long	Date	Confirmed	Deaths	Recovered	Month-Year	Confirmed - 1	Deaths - 1	Recovered - 1	Date - 1	Date Minus 1	Confirmed Daily
0	Anhui	China	31.8257	117.2264	2020-01-22	1.0	0.0	0.0	Jan-2020	NaN	NaN	NaN	NaT	NaT	NaN
1	Anhui	China	31.8257	117.2264	2020-01-23	9.0	0.0	0.0	Jan-2020	1.0	0.0	0.0	2020-01-23	2020-01-22	8.0
2	Anhui	China	31.8257	117.2264	2020-01-24	15.0	0.0	0.0	Jan-2020	9.0	0.0	0.0	2020-01-24	2020-01-23	6.0

Figure23: Data Testing

- After performing all the modifications, we now have our final dataset ready for prediction and forecasting which we then export to use in power BI.



```
In [26]: # Exporting the data

         # Setting my path
         path = "C:\\Users\\G10\\Desktop\\files\\disease prediction\\coronavirus prediction\\User-Tutorials\\CoronaVirus"

         # Changing my CWD
         os.chdir(path)

         full_join3.to_csv('CoronaVirus PowerBI Raw', sep='\t')
```

Figure 24: Exporting the modified dataset to use it in Power BI

The frontend part is made using the Power BI which is a tool used for creating interactive dashboards. In power BI we have imported the modified dataset and used visualization features like slicers for making the Month and Year column in the form of a list, a dropdown menu for Country/Region. For analytics, we have used the forecasting feature of the Power BI in order to predict the data points and also to draw the visualization based on that.

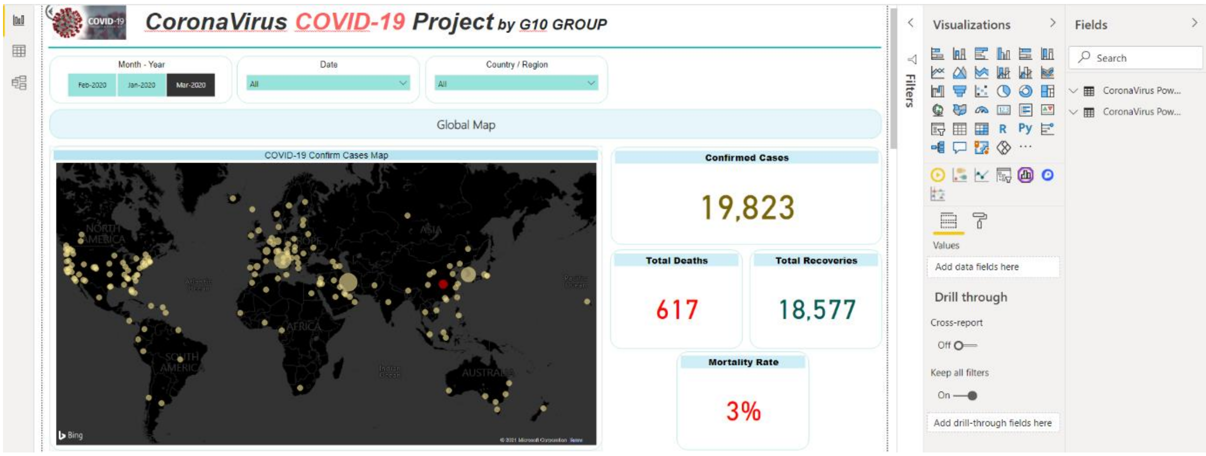


Figure 2: Project Dashboard

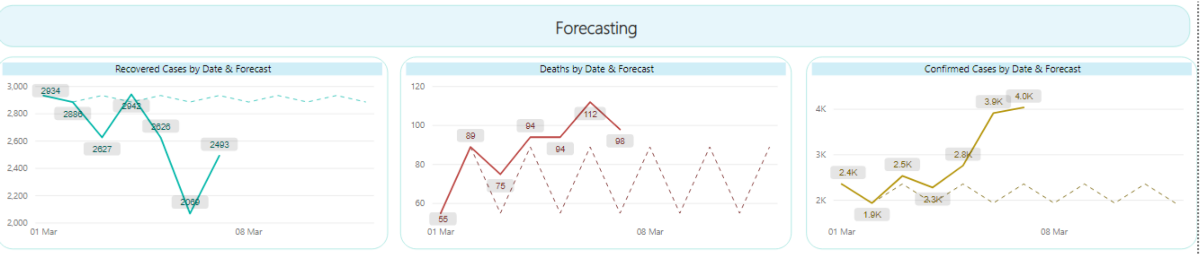
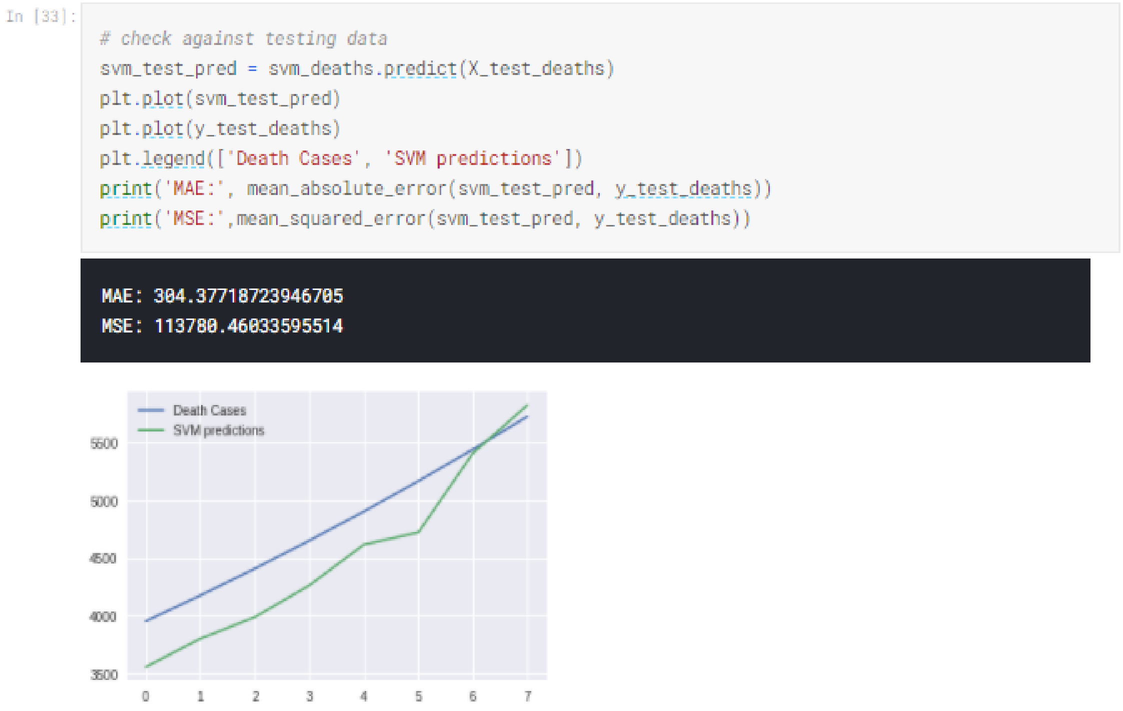


Figure 25: Coronavirus working model using power BI

- In this we are showing a comparison of different prediction models of Python like SVM prediction Model (Figure 26), Liner Regression (Figure 27) and Bayesian Prediction (Figure 28). The prediction on the same dataset and every model returns a different prediction.

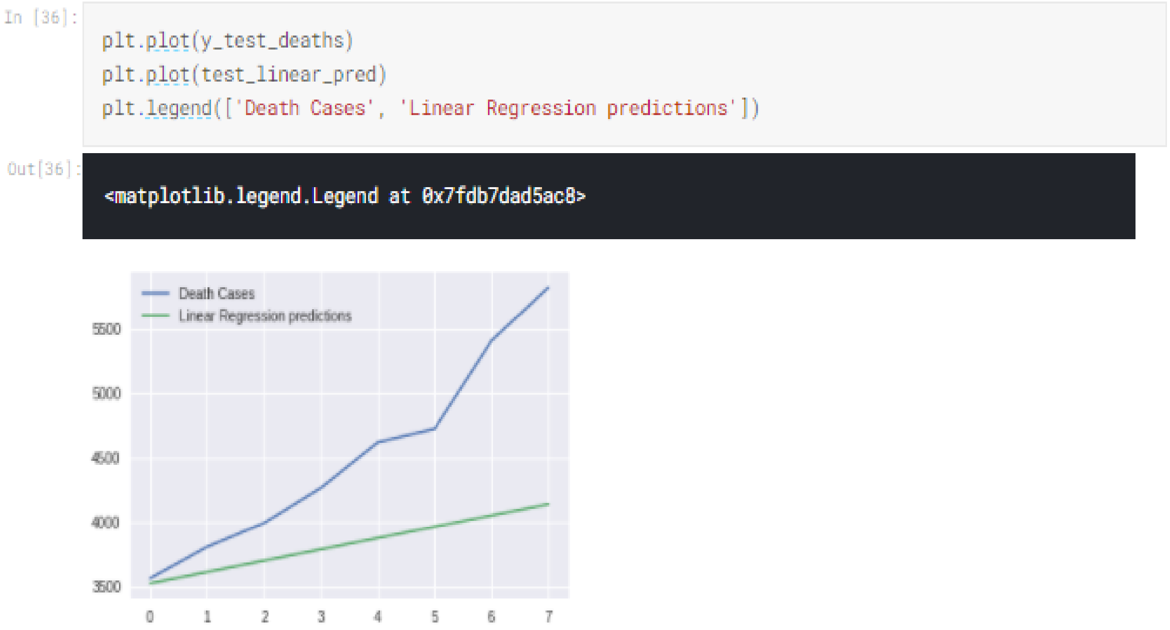


In [33]:

```
# check against testing data
svm_test_pred = svm_deaths.predict(X_test_deaths)
plt.plot(svm_test_pred)
plt.plot(y_test_deaths)
plt.legend(['Death Cases', 'SVM predictions'])
print('MAE:', mean_absolute_error(svm_test_pred, y_test_deaths))
print('MSE:',mean_squared_error(svm_test_pred, y_test_deaths))
```

```
MAE: 304.37718723946705
MSE: 113780.46033595514
```

Figure 26: Coronavirus working model using SVM Prediction



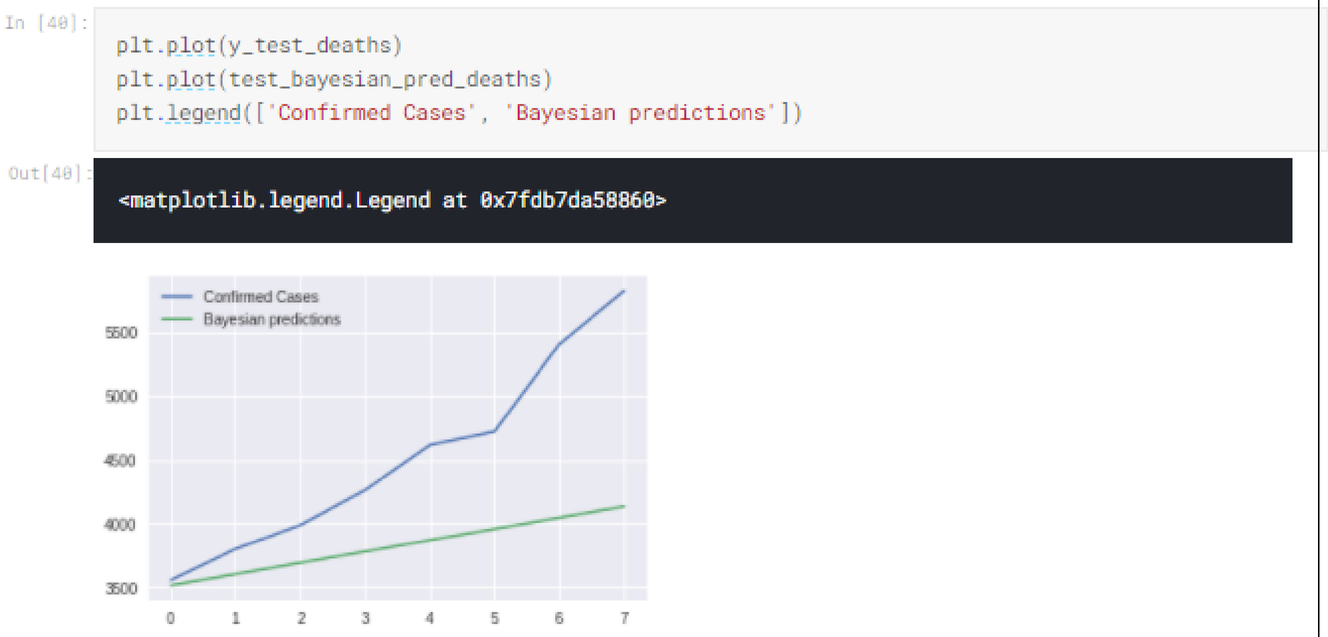
In [36]:

```
plt.plot(y_test_deaths)
plt.plot(test_linear_pred)
plt.legend(['Death Cases', 'Linear Regression predictions'])
```

Out[36]:

```
<matplotlib.legend.Legend at 0x7fdb7dad5ac8>
```

Figure 27: Coronavirus working model using Liner Regression Prediction



```
In [40]: plt.plot(y_test_deaths)
         plt.plot(test_bayesian_pred_deaths)
         plt.legend(['Confirmed Cases', 'Bayesian predictions'])
```

```
Out[40]: <matplotlib.legend.Legend at 0x7fdb7da58860>
```

Figure 28: Coronavirus working model using Bayesian Prediction

Dengue

- In this pie chart is the representation of both male and female death occurred by the dengue as we can see the major area affected by the dengue was India and then Brazil

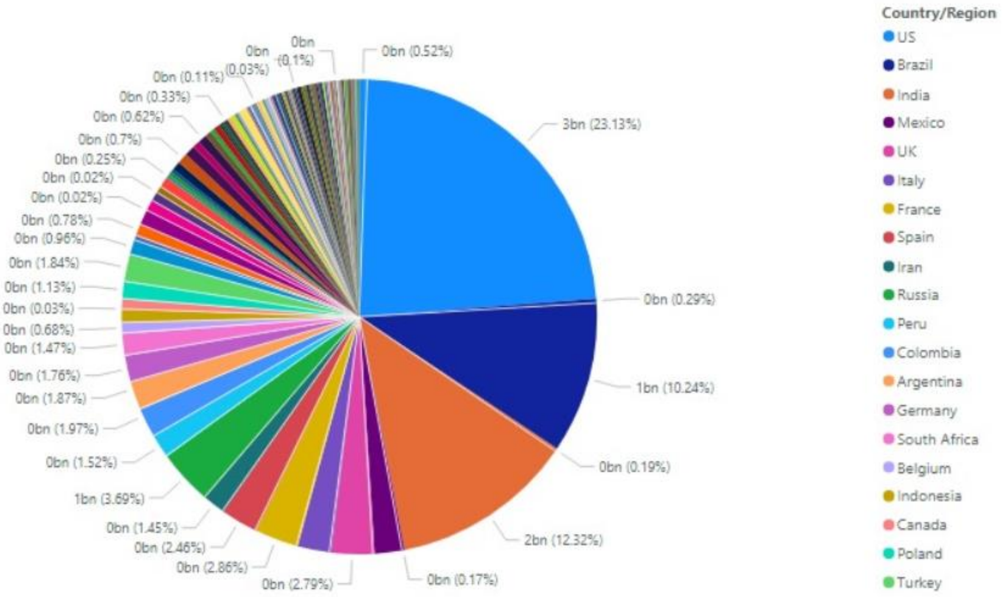


Figure 29: Deaths Confirmed by Countries

- This map graph shows the how much population is effected by the dengue in this map show major cases were affected by the children age between 12-16 year old.

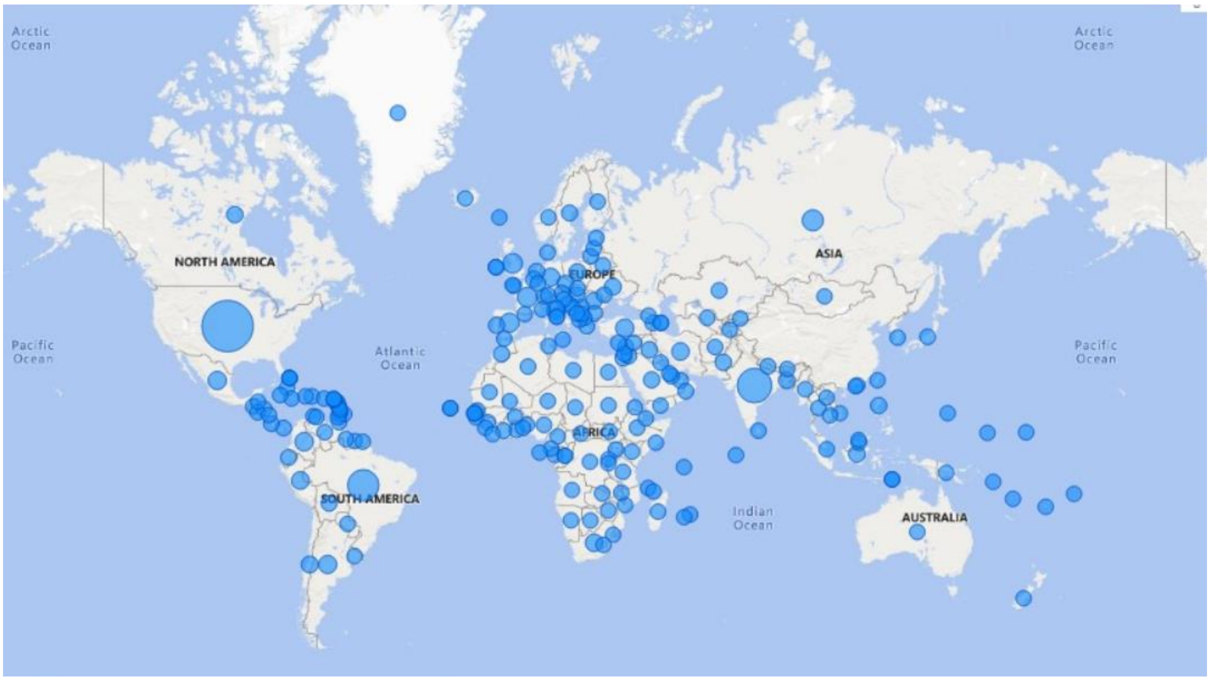


Figure 30: Confirmed and Deaths by Countries/Regions

- This graph show which region of country is most effected by dengue and the how many people were dead due to this virus. Mainly the males were having higher death rate than the female as the dengue death rate is less than 0.2% if not treated then it could rise up to 20% to 50%.

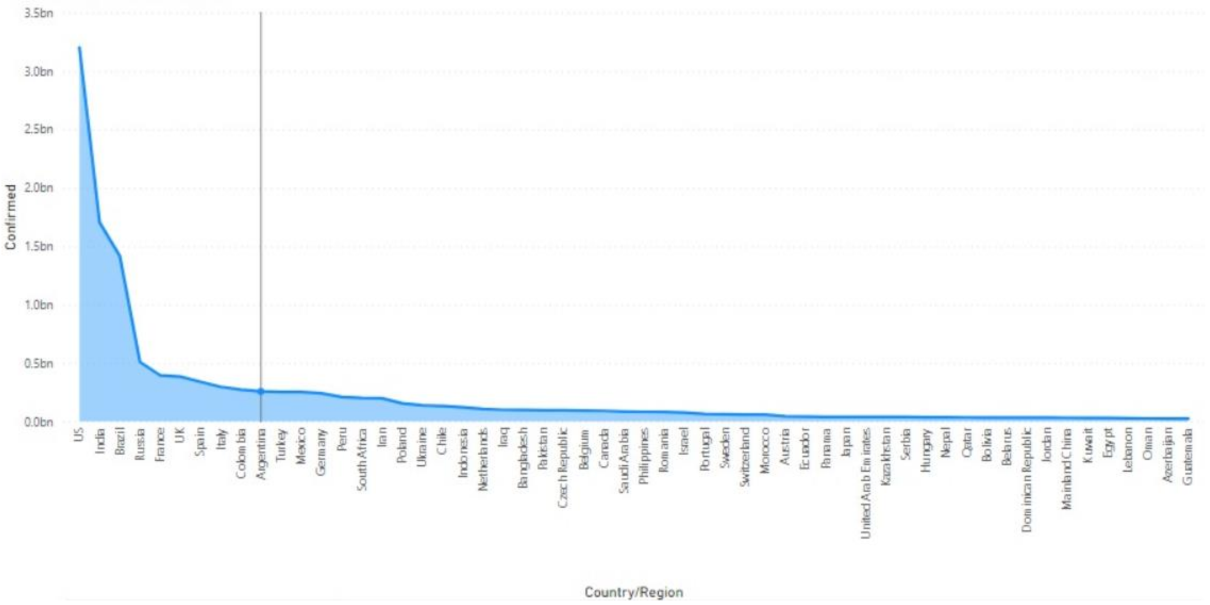
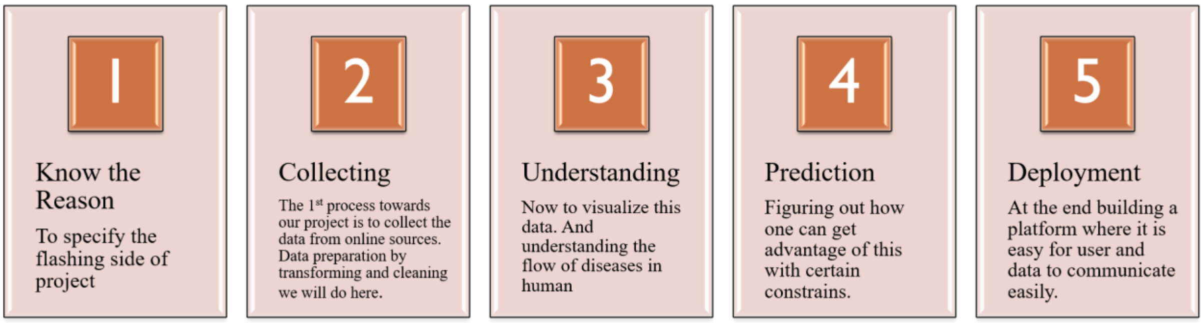


Figure 30: Confirmed by /Regions

Chapter 5 – Conclusion & Road map of Phase-3

So far, we have completed the **analysis** and are progressing in the **prediction** of diseases. We are constantly building up models, applying algorithms, and tackling new challenges to get effective new insights and better working efficiency with the lowest error possible.



1 Know the Reason

To specify the flashing side of project

2 Collecting

The 1st process towards our project is to collect the data from online sources. Data preparation by transforming and cleaning we will do here.

3 Understanding

Now to visualize this data. And understanding the flow of diseases in human

4 Prediction

Figuring out how one can get advantage of this with certain constrains.

5 Deployment

At the end building a platform where it is easy for user and data to communicate easily.

Figure 31: Flowchart

Slowly and effectively, we are progressing to phase 3 (final phase) Deployment. There have planned to build a Web Application that can help predict the person about what disease they may have based on their symptoms

Bibliography

1. https://home.cern/
2. https://www.kaggle.com/
3. https://www.unicef.org/
4. https://data.world/
5. https://catalog.data.gov/dataset
6. https://www.kdnuggets.com/datasets/index.html
7. https://www.ibm.com/in-en/products/cognos-analytics
8. Wikipedia
9. Europa
10. NCBI-NLM(National Library of Medicine)
11. https://api.covid19india.org/

Annexures :

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5666085/

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3130384/

https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0239960

https://www.sciencedirect.com/science/article/pii/S2468042720300087

https://www.researchgate.net/publication/328746395_Reconstruction_and_prediction_of_viral_disease_epidemics

https://www.who.int/globalchange/summary/en/index5.html

https://www.frontiersin.org/articles/348571

https://jvi.asm.org/content/83/11/5296

https://www.nature.com/articles/s41467-020-16153-4

Phase2_IBM_G10_P1

by Neeraj Rathore

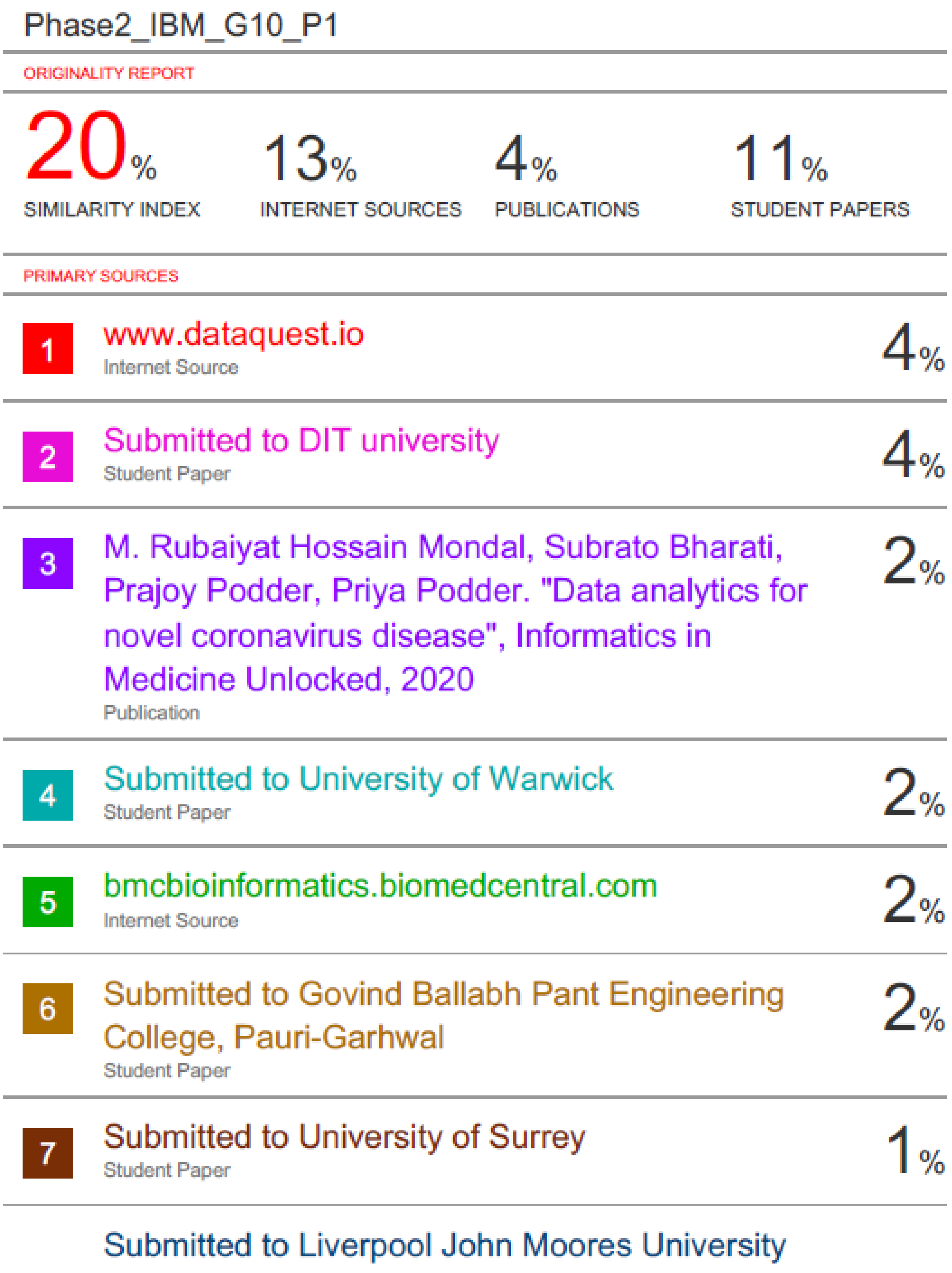
Submission date: 26-Mar-2021 10:14AM (UTC+0530)

Submission ID: 1542732100

File name: 6th_Sem-ProjectGroup-IBMG10.docx (3.71M)

Word count: 2836

Character count: 15435



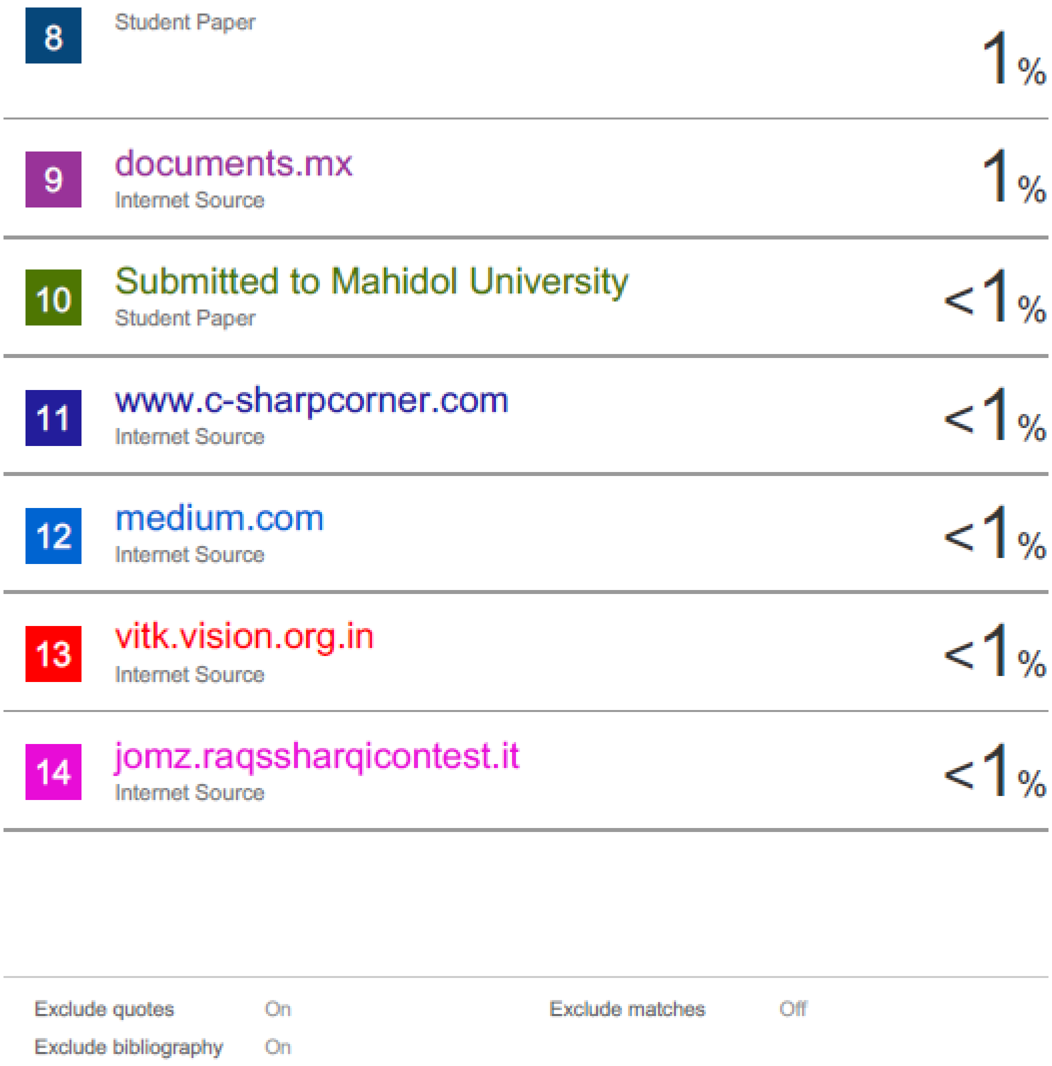
Phase2_IBM_G10_P1

ORIGINALITY REPORT

20%	13%	4%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.dataquest.io Internet Source	4%
2	Submitted to DIT university Student Paper	4%
3	M. Rubaiyat Hossain Mondal, Subrato Bharati, Prajoy Podder, Priya Podder. "Data analytics for novel coronavirus disease", Informatics in Medicine Unlocked, 2020 Publication	2%
4	Submitted to University of Warwick Student Paper	2%
5	bmcbioinformatics.biomedcentral.com Internet Source	2%
6	Submitted to Govind Ballabh Pant Engineering College, Pauri-Garhwal Student Paper	2%
7	Submitted to University of Surrey Student Paper	1%
	Submitted to Liverpool John Moores University	



8	Student Paper	1%
9	documents.mx Internet Source	1%
10	Submitted to Mahidol University Student Paper	<1%
11	www.c-sharpcorner.com Internet Source	<1%
12	medium.com Internet Source	<1%
13	vitk.vision.org.in Internet Source	<1%
14	jomz.raqssharqicontest.it Internet Source	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		