

A PROJECT REPORT (IB412)
ON
ANALYSIS AND PREDICTION OF EFFECT OF VIRAL
DISEASE IN HUMAN + DEPLOYMENT

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

The degree of

BACHELOR OF TECHNOLOGY

In
COMPUTER SCIENCE AND ENGINEERING

by

Ayan Bhatnagar- 190109900 - BDA

Kartikey Garg-180109045 - BDA

Pradyumn Bahukhandi-180109025 - BDA

Yash Sharma-180109032 – BDA

Under the Guidance of

Manish Sharma Sir

Assistant Professor



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.

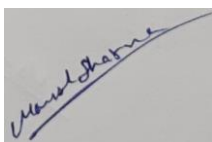
Nov,2021

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 fulfillment 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 **Aug 2021** to **Nov 2021** under the guidance of **Manish Sharma Sir**.

Date: 03/08/2021

Signature of the Instructor

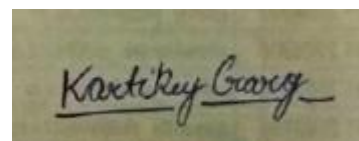


(Manish Sharma IBM Trainer)

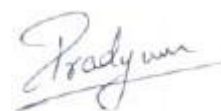
Signature of the Candidates



Ayan Bhatnagar



Kartikey Garg



Pradyumn Bahukhandi



Yash Sharma

ACKNOWLEDGEMENT

This is an excellent chance for us to submit this report on the University Group Project's partial completion. We extend our heartfelt gratitude to all academic members who assisted in the development of this initiative, whether directly or indirectly. We'd also like to express our gratitude to Manish Sharma Sir, who oversaw and aided us at all times. We want to use this valuable project, as well as the professional skills we obtained throughout its development and deployment, in our professional future.

Name: Ayan Bhatnagar

Roll No.: 190109900

SAP ID: 1000013728

ABSTRACT

This project will address the potential applications of data analytics components on this viral infection, as well as describe different features of virus, disease, and present a visualization of the infection's spread. To begin, a literature review and data collection on viruses is conducted, with numerous factors such as the virus's origin, similarity to a previous disease, transmission capacity, symptoms, and so on being highlighted. Second, data analytics is applied to a dataset pertaining to data collection to determine the spread of the viral infection and model the global increase in the number of confirmed cases over time. Thirdly the application deployment will reduce our project on a platform, and it will work on real-time data.

LIST OF FIGURES

<u>FIGURE NAME</u>	<u>PAGE</u>
Figure 1: IBM Cognos	11
Figure 2: Python NumPy	12
Figure 3: Python Pandas	12
Figure 4: Python matplotlib	12
Figure 5: Python Tkinter	13
Figure 6: wxPython	13
Figure 7: Python Qt	14
Figure 8: GTK +	14
Figure 9: Power BI	15
Figure 10: Scikit Learn	15
Figure 11: Seaborn	16
Figure 12: Showing what all data columns are there	21
Figure 13: Metadata of Datasets	22
Figure 14: Data types in Dataset	22
Figure 15: Head command representing the first 5 value of dataset	23
Figure 16: representation of head plus tail	23
Figure 17: Representing no Missing Values	24
Figure 18: Distribution of people based on age and region suffering from HIV	24
Figure 19: Concept of joins used in Dataset	25
Figure 20: Year-wise representation of Gender suffered from Diarrhoea	26
Figure 21: Region-wise comparative representation of Males suffering Diarrhoea and Pneumonia	26
Figure 22: Regional Comparison of gender suffering from Diarrhoea and Pneumonia	27

Figure 23: Importing the required packages and libraries	27
Figure 24: Loading confirmed, deaths, recovered datasets using pandas library.	28
Figure 25: Data Extraction	28
Figure 26: Data transformation	29
Figure 27: Data cleaning	29
Figure28: Performing full joins on datasets	30
Figure29: Adding Month and Year as a new column	30
Figure30: Data Testing	31
Figure 31: Exporting the modified dataset to use it in Power BI	31
Figure 32: Project Dashboard	32
Figure 33: Coronavirus working model using power BI	32
Figure 34: Coronavirus working model using SVM Prediction	32
Figure 35: Coronavirus working model using Liner Regression Prediction	33
Figure 36: Coronavirus working model using Bayesian Prediction	33
Figure 37: Deaths Confirmed by Countries	34
Figure 38: Confirmed and Deaths by Countries/Regions	34
Figure 39: Confirmed by /Regions	35
Figure 40: Index.html	35
Figure 41: Code Sample 1	36
Figure 42: Code Sample 2	37
Figure 43: Code Sample 3	38
Figure 44: Code Sample 4	39
Figure 45: Code ample 5	40
Figure 46: blog.html	40
Figure 47: Code 6	41
Figure 48: CSS	42

Figure 49: Code Sample 7	42
Figure 10: Code Sample 8	43
Figure 51: Code Sample 9	43
Figure 52: Code Sample 10	44
Figure 53: Code Sample 11	45
Figure 54: Code Sample 12	46
Figure 55: Media Query	47
Figure 56: Styling	48
Figure 57: Footer	49
Figure 58: Workflow	49
Figure 59: Dataset	50
Figure 60: Sample Dataset	51
Figure 61: Naive Classifier	51
Figure 62: Divider Code	52
Figure 63: Model.py	53
Figure 65: Prediction	53
Figure 66: Front End	54
Figure 67: Flowchart	55
Figure 68: Fianal Deployment	55
Figure 69: Entered Values/Symptoms	56
Figure70: Final Deployment	57

TABLE OF CONTENT

<u>CHAPTER</u>	<u>PAGE No.</u>
Candidate's Declaration	2
Acknowledgment	4
Abstract	5
Chapter 1 -- Introduction	8
Chapter 2 –Project Description	9
2.1 Purpose	
2.2 Problem statement	
2.3 Special Features	
Chapter 3 –Tools and Technologies	10
Chapter 4- Implementation Modules and Screen Shots	20
Chapter 5 – Conclusion	54
 Bibliography	 58

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.

The issue of detecting extremely divergent or undiscovered viruses from datasets is difficult. When human samples are sequenced, a large part of the usual methods used to put them together find no relation to existing sequences. By conducting research on the dataset of viruses such as Plague, Influenza, Dengue, Ebola, Hantavirus, COVID-19, and others, we hoped to see if machine learning algorithms might improve the detection of viral sequences in metagenomic sequencing data.

The current status of Technology utilization for medical analysis is having a vast usage. Researchers from all over the globe are interested in developing, deploying, and illustrating the use of technology in HUMAN. Moreover, in this era, 3 famous terms “Artificial Intelligence, Deep Learning, Machine Learning” are boosted for Analysis motivation in researchers. We do have Several Regression models, Intelligence-based machine model functions deal a handsome amount of help in researching the field of Medical Diagnosis. The evolution of the first expert system, MYCIN, developed in 1976, marks the beginning of the relationship between health care and technology. MYCIN was designed to use 450 rules gathered from a medical expert to treat bacterial infections by recommending appropriate medications to patients. ML and AI are also utilized to help with diagnosis and screening of patients who have been identified using radio imaging technology. Studies demonstrate the promise of AI and machine learning technologies by proposing a novel model that includes speedy and accurate Diagnosis methods based on Deep Convolutional Networks. Recent research has developed an auxiliary tool to improve the accuracy of Diagnosing Algorithms using a new automatic detection-based algorithm.

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.

2.2. Problem statement

Fighting with the diseases is necessary for human Survival and Development, so a platform must be there that helps humans analyzing 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.

2.3. 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 IBM Cognos for analyzing the data and give meaningful outcomes. In IBM Cognos analysis Studio is a user-friendly interactive environment that may be used for multidimensional analysis and exploration of massive data sources, as well as answering data queries.

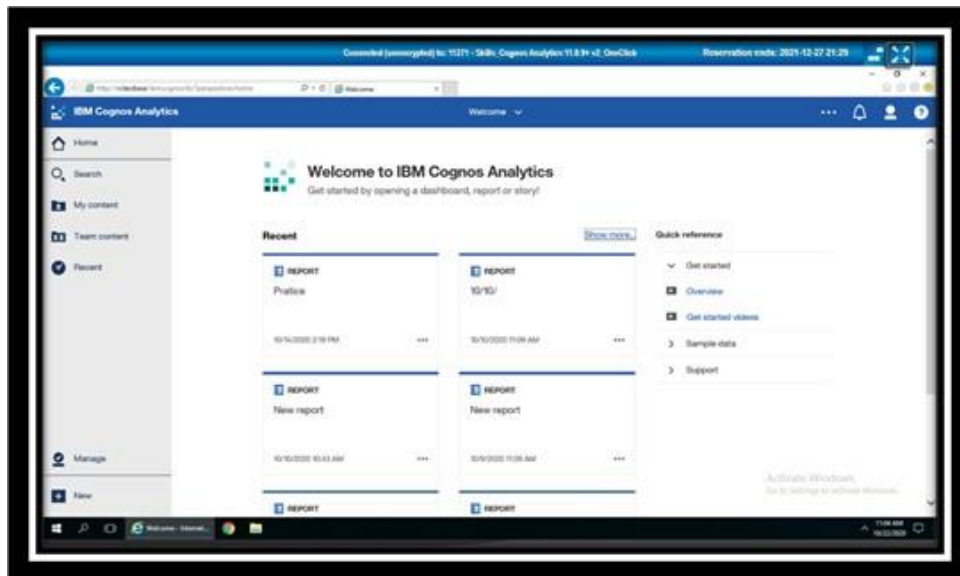


Figure 1: IBM Cognos

2. Exploration

During our Phase 2 process, we will be using Data dicing and slicing is referred to as OLAP (Online Analytical Processing) exploration. For example, we may look at the death ratio by gender from 2006 to 2008. You may notice the rise or drop in the death ratio by clicking on 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 modeling which will be corresponding to visualization.

3. Analysing Large Data

It will assist us in finding significant details while keeping summaries in mind to maintain a clear perspective of our data in Analysis Studio.

- 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 comes with a lot of handy features for working with n-arrays and matrices in Python. It facilitates the processing of arrays with the same data type values and simplifies array math operations. The vectorization of mathematical operations on the NumPy array type improves efficiency and reduces the time it takes for any data to be processed.
- **Pandas**
Pandas is a library that allows us to work with labelled and relational data while also allowing us to query in a non-intuitive way. It is built on two data structures: one-dimensional "Series" (like an item list) and two-dimensional "Data Frames" (like a table with numerous columns). It allows us to convert data structures to Data Frame objects, as well as handle missing data, add/remove columns from the Data Frame, imputing missing files, and plotting data with histogram or plot box. For data manipulation, visualisation, and wrangling, it's a must-have.



Figure 3: Python Pandas

- **Matplotlib**

Matplotlib is a library for the Python programming language that is also used for NumPy's numerical and mathematical extensions. It also includes an object-oriented API for embedding charts into data sets, as well as an application for working with a general-purpose GUI toolkit



Figure 4: Python matplotlib

- Tkinter



Figure 5: Python Tkinter

- wxPython

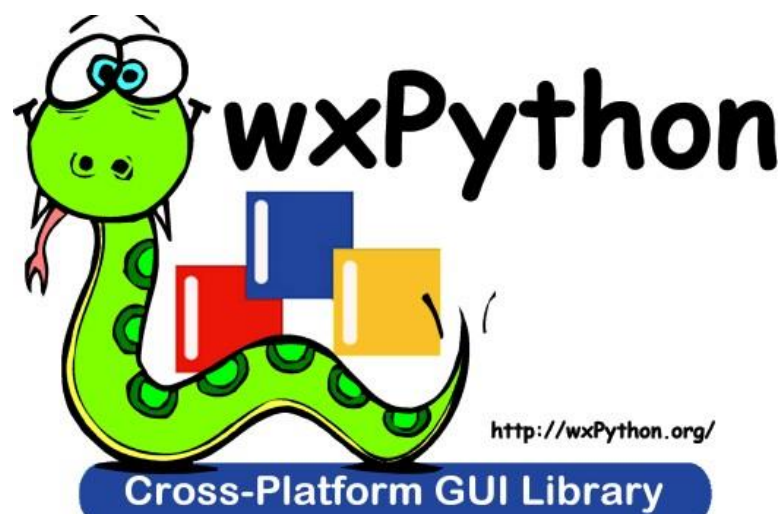


Figure 6: wxPython

- Qt



Figure 7: Python Qt

- GTK+.

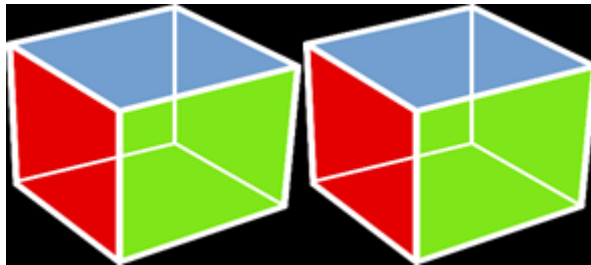


Figure 8: GTK

- **Power BI**

We can use Power BI to perform forecasting, prediction, and hindcasting while visualising and exploring data. Forecasting in Power Bi makes use of built-in predictive statement models with exponential smoothing to detect seasonality in data and provide prediction results from a set of data. We can also investigate forecast results by altering the given confidence interval or outlier knowledge to see how these influence outcomes. We can also look back in time to see if Power Bi would have predicted this and whether recent evidence supports older understanding.

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

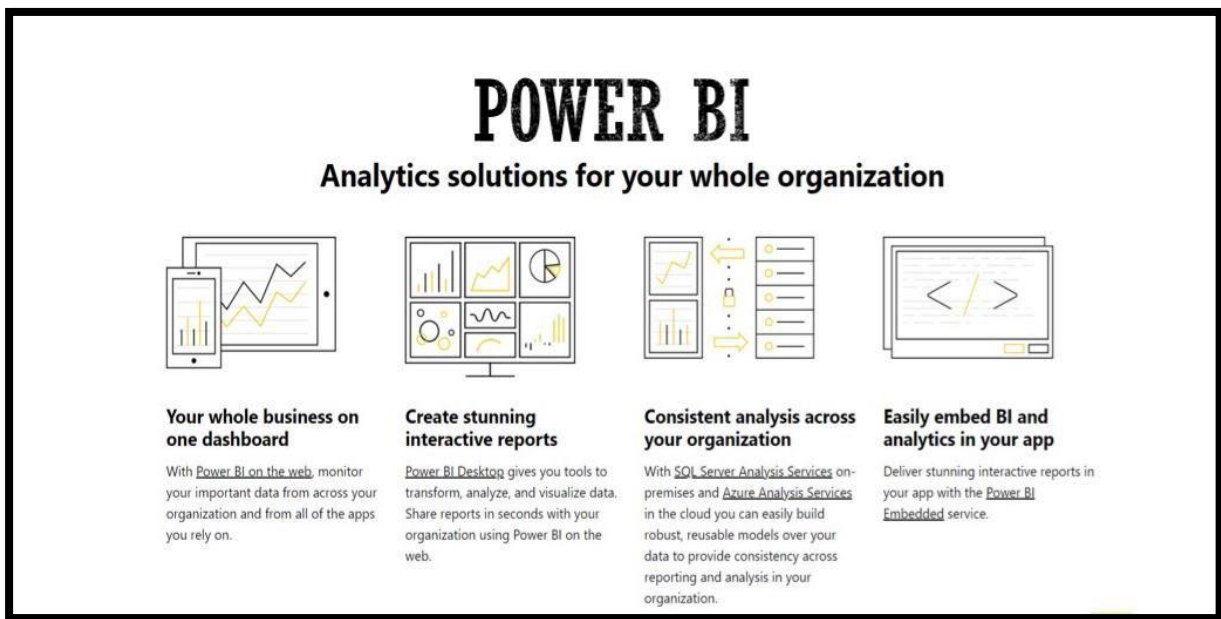


Figure 9: Power BI-

- **Scikit-Learn (Sklearn)**

Scikit-learn (Sklearn) is Python's most useful and reliable machine learning library. Through a consistent Python interface, it delivers a number of cost-effective tools for machine learning and applied math modelling, as well as classification, regression, clustering, and spatial property reduction. NumPy, SciPy, and Matplotlib are used to build this package, which is primarily developed in Python.



Figure 10: Scikit – Learn

- **Seaborn Library**

Seaborn is a Python module for making applied math graphics. It is based on matplotlib and tightly integrates with pandas knowledge structures.

Seaborn aids in the exploration and perception of your information. Its charting routines work with data frames and arrays including entire datasets, performing the necessary linguistics mapping and applied math aggregation internally to produce useful charts. Its

declarative, dataset-oriented API helps you to focus on what the various pieces of your plots represent, rather than on the little instructions or how to create them.

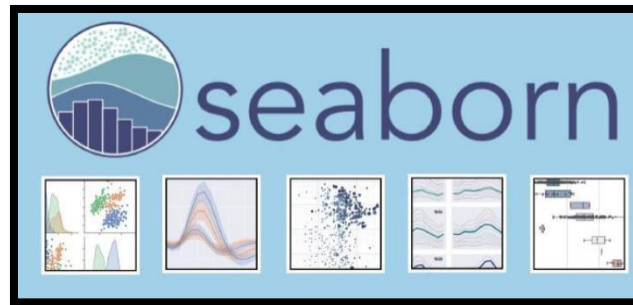


Figure 11: Seaborn

The methods and materials required to make this project are described in this section.

3.1 LOGISTIC REGRESSION

Logistic regression is a statistical approach that Machine Learning has inherited. Classification problems can be solved using this method. Logistic regression and linear regression are very similar to each other as they attempt to predict a response variable Y given a set of X input. It's a type of supervised learning that seeks to predict the responses of unlabeled, unseen data by first training with labelled data, which includes both independent (X) and dependent (Y) variables. It's a method for modelling and analyzing the relationship between variables, as well as how they contribute to and are related to achieving a specific outcome.

1. Logistic regression is used to model the likelihood that Y , the response variable, belongs to a specific group. In many circumstances, the response variable will be binary, therefore logistic regression will want to model a function $y=f(x)$ that returns a normalized value ranging from 0 to 1 for practically all values of X corresponding to two possible Y values.

1. **Binary Logistic Regression** - There are just two possible results for the categorical response. Spam or not, for example

1. **Multinomial Logistic Regression**- It has more than three categories, none of which are ordered. Predicting which dish is preferred more is an example.

2. **Ordinal Logistic Regression**- It has more than three, and it is ordered with order.

3. For instance, a movie's rating ranges from 1 to 5

4. **Logistic Functions-**

As demonstrated in equation 2, the Sigmoid Function is another term for the Logistic Function. The shape is an S-curve, which can take any real-valued number and convert it to a value

between 0 and 1, but never exactly between those bounds.

$1/(1+e^{-value})$ (2) Where e is the base of natural logarithms and value denotes the actual numerical value to be transformed..

Predictions using Logistic Regressions:

Input values (x) are linearly blended using weights or coefficients to predict an output value (y).

The output value being modelled is binary values (0 or 1) rather than a numeric number, which is a fundamental difference from linear regression.

As illustrated in equation 3, below is an example of logistic regression:

$$y = e^{(b_0 + b_1 * x)} / (1 + e^{(b_0 + b_1 * x)}) \quad (3)$$

Where y is the projected output, b_0 is the bias term, and b_1 is the single unit value coefficient (x).

There is a related b coefficient in each column of input data that is understood from training the data.

Applications of Logistic Regression:

1. (To classify mail as spam or not.
2. It helps us to determine the presence or absence of certain diseases like cancer-based on symptoms and other medical data.
3. We can classify images based on pixel data.

3.2. SUPPORT VECTOR MACHINE (SVM)

SVM is an approach that is machine learning and is used for regression and classification problems. It is capable of taking continuous as well as categorical data. In the SVM algorithm, a support vector clustering algorithm was developed to categorize unlabeled data because supervised learning is impossible when data is unlabeled, an unsupervised learning strategy is necessary, which tries to uncover natural clustering of data into groups. In industrial applications, it is the most widely used clustering technique.

Maximal-Margin Classifier:

SVM denotes that data points are represented in hyperplane space and can be divided into two groups. A hyperplane is a line that divides the input variable space into two parts. A hyperplane is chosen to split the points in the input variable space by their class, either class 0 or class 1. Similar points are grouped together. The dataset is represented as a p -dimensional vector in SVM with linear kernel, which can be split by $p - 1$ planes called hyper-planes. The boundaries between data groupings are defined by these planes. A suitable hyper-plane is chosen based on the boundaries between the two classes. We can suppose that a perfect separation between all of our input locations is possible.

Example is shown in equation 4:

$$B_0 + (B_1 * X_1) + (B_2 * X_2) = 0 \quad (4)$$

Where, coefficients B_1 and B_2 determine the slope of the line, and the intercept (B_0) is found by the learning algorithm, and X_1 and X_2 are the two input variables.

Soft Margin Classifier:

Hyperplane cannot properly separate messy real data in the real world. The Soft Margin Classifier allows the maximum margin of the line that divides the classes to be relaxed. Some locations in the training data are allowed to cross the splitting lines.

Additional coefficients are introduced, giving each dimension more wiggle room. These variables are known as slack variables, and they raise the model's complexity by adding more parameters for the model to fit the data and give complexity.

The magnitude of the twitch allowed across all dimensions is determined by C parameters, which dictate the amount of violation of the margins allowed. We're back to the inflexible Maximal - Margin Classifier because $C=0$ isn't a violation. The higher the value of C , the more hyperplane breaches are allowed.

Kernels of SVM:

The SVM method is implemented using a kernel. The hyperplane is learned in linear SVM by changing the issue with some linear algebra. The multiplication total of each input value pair is the inner product of any two vectors.

The equation which is used for making prediction involving a new input using dot product between input(x) and each support vector (x_i) is calculated as shown in equation 5:

$$f(x) = B_0 + \sum(a_i * (x, x_i)) \quad (5)$$

This equation requires calculating the inner products input vector(x), which is a new feature with all support vectors in training data. The learning algorithm must estimate the coefficients B_0 and a_i from the training data.

Linear Kernel SVM- The dot product called kernel is written in equation 6 :

$$K(x, x_i) = \sum(x * x_i) \quad (6)$$

It specifies the distance or similarity between new data and the support vector. Because the distance involves a linear combination of inputs, the dot product can be called a similarity measure and is utilised for linear SVM.

Polynomial Kernel SVM - We can use the polynomial kernel svm instead of using the dot product since it allows for curved lines in the input space, as illustrated in equation 7:

$$K(x, x_i) = 1 + \sum(x x_i)^d \quad K(x, x_i) = 1 + \sum(x x_i)^d \quad (7)$$

Where the polynomial's degree must be given. This is the same as the linear kernel when $d=1$.

Radial Kernel SVM- It's local, and it can be used to make complex regions within the feature space, similar to closed polygons in two-dimensional spaces.

$$K(x, x_i) = \exp(-\gamma \|x - x_i\|^2) = \exp(-\gamma \sum_{j=1}^d (x_j - x_{ij})^2) = \exp(-\gamma \sum_{j=1}^d (x_j^2 - 2x_j x_{ij} + x_{ij}^2)) = \exp(-\gamma \sum_{j=1}^d x_j^2) \exp(2\gamma \sum_{j=1}^d x_j x_{ij}) \exp(-\gamma \sum_{j=1}^d x_{ij}^2)$$

Where γ is a learning method parameter that must be given. 0.1 is a suitable number for γ by default, and γ is frequently 0.1 .

Data Preparation for SVM

Some suggestions for how to prepare for training data when learning an SVM model.

1 Numerical Inputs- SVM assumes that the inputs are numeric. If we have categorical inputs then we may need to convert them to binary dummy variables.

1 Binary Classification- Basic SVM is described for binary classification problems. Although, extensions were developed for multi-class classification and regression.

Properties:

- Flexibility in choosing a similarity function.
- By the Soft Margin Approach overfitting can be controlled.
- Feature selection.
- Large feature spaces are handled i.e. complexity is not dependent on the dimensionality feature space.
- The sparseness of solution is required when dealing with the large data sets

3.3.1 Data Collection

Data Collection is the process of measuring information and gathering it. This information can then help us to evaluate answers and outcomes. The data can be collected in two ways to create a **Training Data Set**. They are:

1. Primary Data Collection: When the data is collected for the first time by person and is

therefore original. The training data set is then freshly created. This type of data collection takes a bit more time but generates quite accurate answers.

2. Secondary Data Collection: When the data is retrieved from other sources, i.e. already collected data is used to construct a data set. This type of data set is time-saving but does not guarantee an accurate answer.

Data Collection is a vital step towards any project, research, or model as it guarantees an optimal solution or output. When the data is appropriately collected, then only we can hope to achieve the results we want.

3.3.2 Data Description

Describing and documenting data is very essential as it confirms that the person who will deal with the collected data can understand it. The researcher should be able to understand the data as well as the processes and analyses that were done to create it. We will evaluate mean, median, mode, standard

deviation, etc. to summarize the data. The data we deal with is a large one, which makes it rather mandatory to summarize the respective data. For this “summary of data” we most commonly used measures of central tendency. This includes mean- averaging the values, median finding the middle value from data, and mode- most frequently occurring data values. These values of central tendency help in determining the distribution of the data as well as to characterize the large data sets.

3.3.3 Data Pre-processing

Pre-Processing generally means converting data into a much more understandable format. Generally, the data which we need to handle is raw data which is then preprocessed by – normalization, cleaning, instance selection, feature extraction, and selection. The ultimate result of data preprocessing is the Training Set.

For our model, we preprocessed our data in two steps:

1. Manual Pre-Processing – The pre-processing which involved manual efforts are kept under this category. The following steps were done manually, without the aid of any programming language or such technologies. The steps can be listed down as follows:

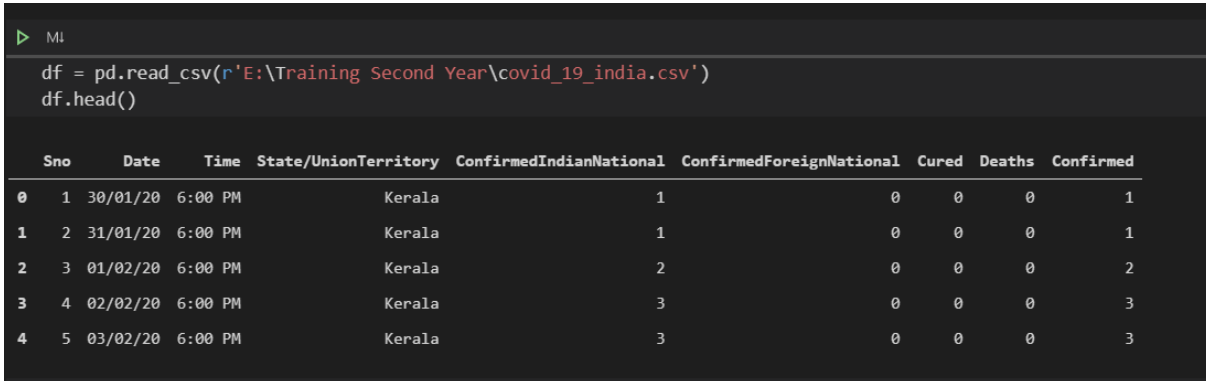
Deletion of un-required Columns

All the text columns like Name, Gender, Age, Program, Year of Study, etc. were removed from the complete data set. These fields are not required for construction of Training Data Set. So, unrequired data fields are removed.

Chapter 4- Implementation Modules and Screen Shots

Data Pre-Processing

```
df = pd.read_csv(r'E:\Training Second Year\covid_19_india.csv')
df.head()
```



	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0	1	1
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0	1	1
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0	2	2
3	4	02/02/20	6:00 PM	Kerala	3	0	0	0	3	3
4	5	03/02/20	6:00 PM	Kerala	3	0	0	0	3	3

Figure 12: Showing what all data columns are there

Pre-Processing through Python Programming –

In this second step of preprocessing, we used **Python**- a programming language, to work on our complete data set, and preprocess it. The steps can be listed as follows:

a) Import libraries

In Python, various in-built libraries are available using which we can perform various operations.

For our model, we have imported the following libraries shown in figure 10- pandas, matplotlib and sklearn, preprocessing, Imputer.

- **Pandas** is used for various tasks such as :

- Creation of data frames and
- Navigation of data frames.

- **matplotlib** is used for plotting target variables in the form of a bar chart.

- **Imputer** is used for handling missing values in the dataset.

b) Load Dataset

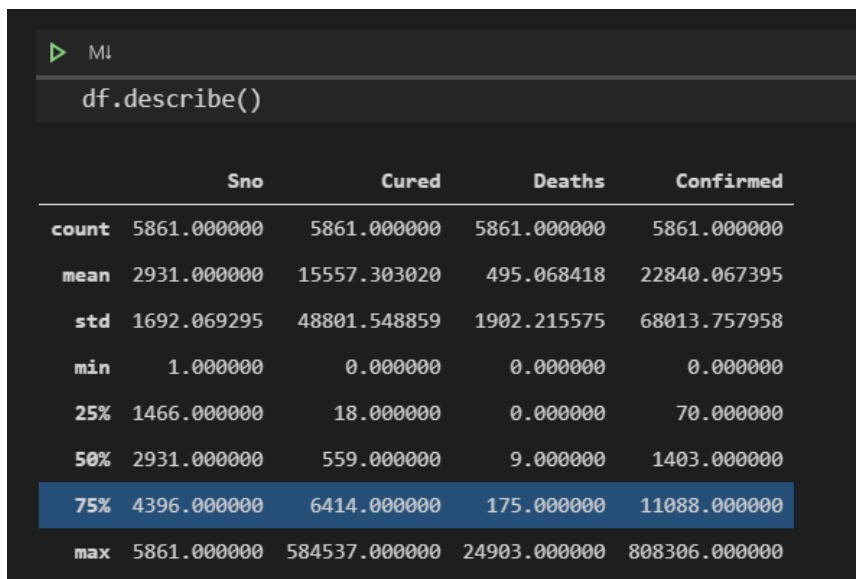
- Before starting our project we had loaded our dataset which was stored in the form of a CSV file shown in figure 11.

- The pandas package was then used to convert CSV files to data frames using the `pandas.read_csv()` function. This function returns a pandas. Data Frame.

c) Verify Dataset

The following commands are shown in figure 12 and functions are used to verify the performance of the dataset values.

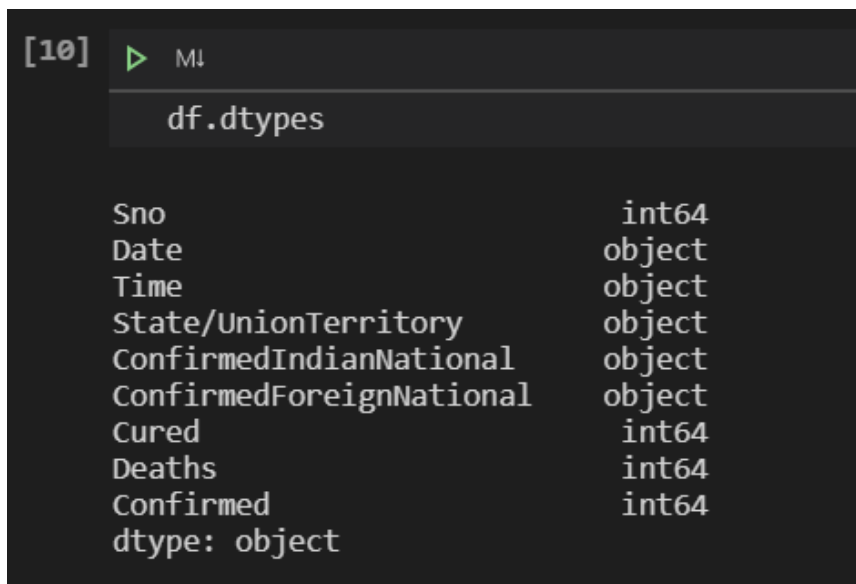
- **describe()** function is used to see some of the core statistics of the dataset, such as - mean, min, max, etc.



```
df.describe()
```

	Sno	Cured	Deaths	Confirmed
count	5861.000000	5861.000000	5861.000000	5861.000000
mean	2931.000000	15557.303020	495.068418	22840.067395
std	1692.069295	48801.548859	1902.215575	68013.757958
min	1.000000	0.000000	0.000000	0.000000
25%	1466.000000	18.000000	0.000000	70.000000
50%	2931.000000	559.000000	9.000000	1403.000000
75%	4396.000000	6414.000000	175.000000	11088.000000
max	5861.000000	584537.000000	24903.000000	808306.000000

Figure 13: Metadata of Datasets




```
df.dtypes
```

Sno	int64
Date	object
Time	object
State/UnionTerritory	object
ConfirmedIndianNational	object
ConfirmedForeignNational	object
Cured	int64
Deaths	int64
Confirmed	int64
dtype:	object

Figure 14: Datatypes in Dataset

- **shape** command is used to get dataset size. It returns a tuple with several rows and the number of columns for the data in the Data Frame.
- **head()** function is used to show the top 5 rows of data in the Data Frame.

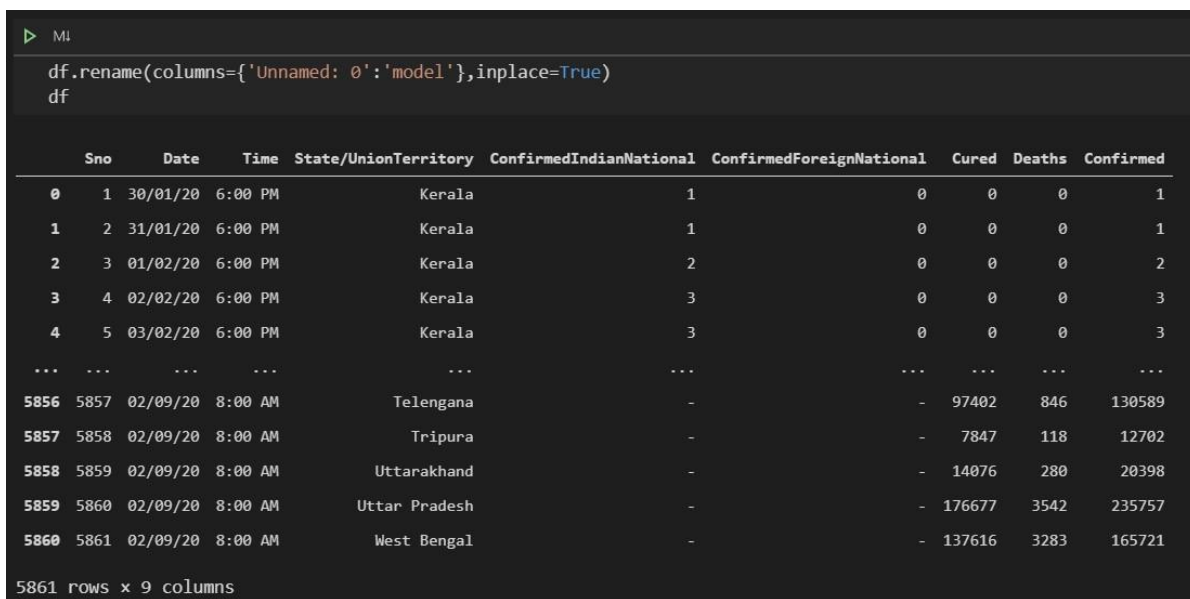


```
MI
df = pd.read_csv(r'E:\Training Second Year\covid_19_india.csv')
df.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0	1
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0	1
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0	2
3	4	02/02/20	6:00 PM	Kerala	3	0	0	0	3
4	5	03/02/20	6:00 PM	Kerala	3	0	0	0	3

Figure 15: Head command representing the first 5 values of a dataset

- **groupby()** function is used on the output variable of the dataset to find out the total number of 0 and 1 categories.



```
MI
df.rename(columns={'Unnamed: 0': 'model'}, inplace=True)
df
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	30/01/20	6:00 PM	Kerala	1	0	0	0	1
1	2	31/01/20	6:00 PM	Kerala	1	0	0	0	1
2	3	01/02/20	6:00 PM	Kerala	2	0	0	0	2
3	4	02/02/20	6:00 PM	Kerala	3	0	0	0	3
4	5	03/02/20	6:00 PM	Kerala	3	0	0	0	3
...
5856	5857	02/09/20	8:00 AM	Telengana	-	-	97402	846	130589
5857	5858	02/09/20	8:00 AM	Tripura	-	-	7847	118	12702
5858	5859	02/09/20	8:00 AM	Uttarakhand	-	-	14076	280	20398
5859	5860	02/09/20	8:00 AM	Uttar Pradesh	-	-	176677	3542	235757
5860	5861	02/09/20	8:00 AM	West Bengal	-	-	137616	3283	165721

5861 rows x 9 columns

Figure 16: representation of head plus tail

Checking of Columns with missing values

Using IsNull() and sum() functions we found the columns with the number of missing values.

It is required so that we can apply imputer for filling those missing values

```

▶ MI
df.replace('?', np.nan, inplace=True)

▶ MI
df.isnull().sum()

Sno                0
Date               0
Time              0
State/UnionTerritory 0
ConfirmedIndianNational 0
ConfirmedForeignNational 0
Cured             0
Deaths           0
Confirmed         0
dtype: int64

```

Figure 17: Representing no Missing Values

Understanding the content

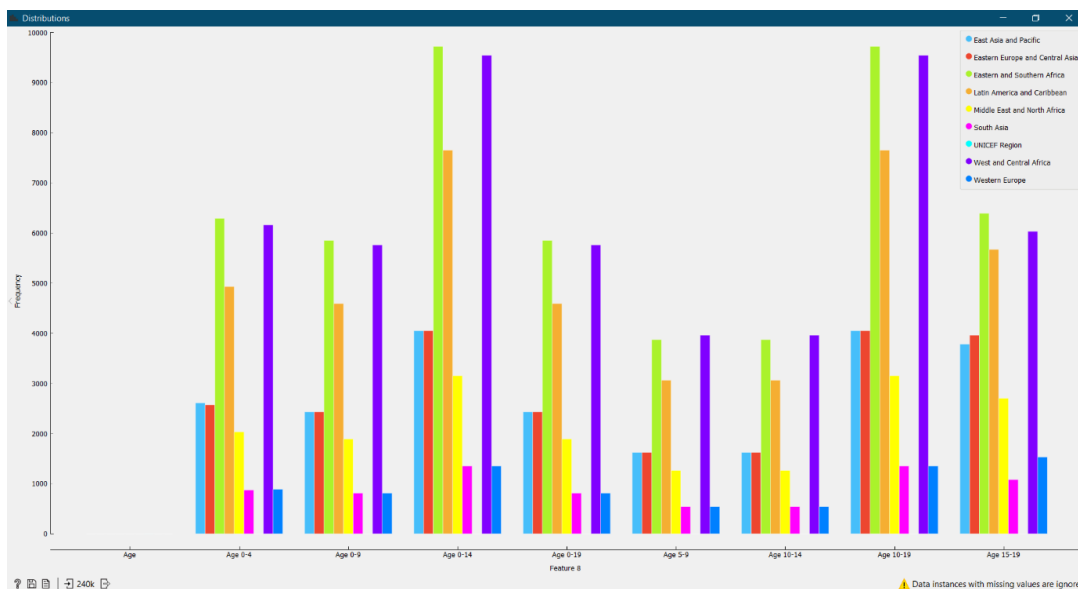


Figure 18: Distribution of people based on age and region suffering from HIV

After successful completion of the data preparation and cleaning part, we headed towards Data Understanding Part. As shown in the figure above the figure represents the distribution of people age based on frequency suffering from HIV and split by their region. As a warning is shown that data instances missing are ignored are not missing values but not applicable so this is the representation of 240K values of dataset done in the ORANGE software. IN the upper right corner the regions are mentioned. From this distribution, we understand that the age group from 0 to 14 and 10m to 19 have a very large number of HIV Infections especially in East Asia and the Pacific, and West and Central Africa. We also concluded that Africa Suffers a large

number of HIV Infections. Whereas South Asia and western Europe suffer the minimum number of HIV Infections.

Pneumonia and diarrhea

PneuCare 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.

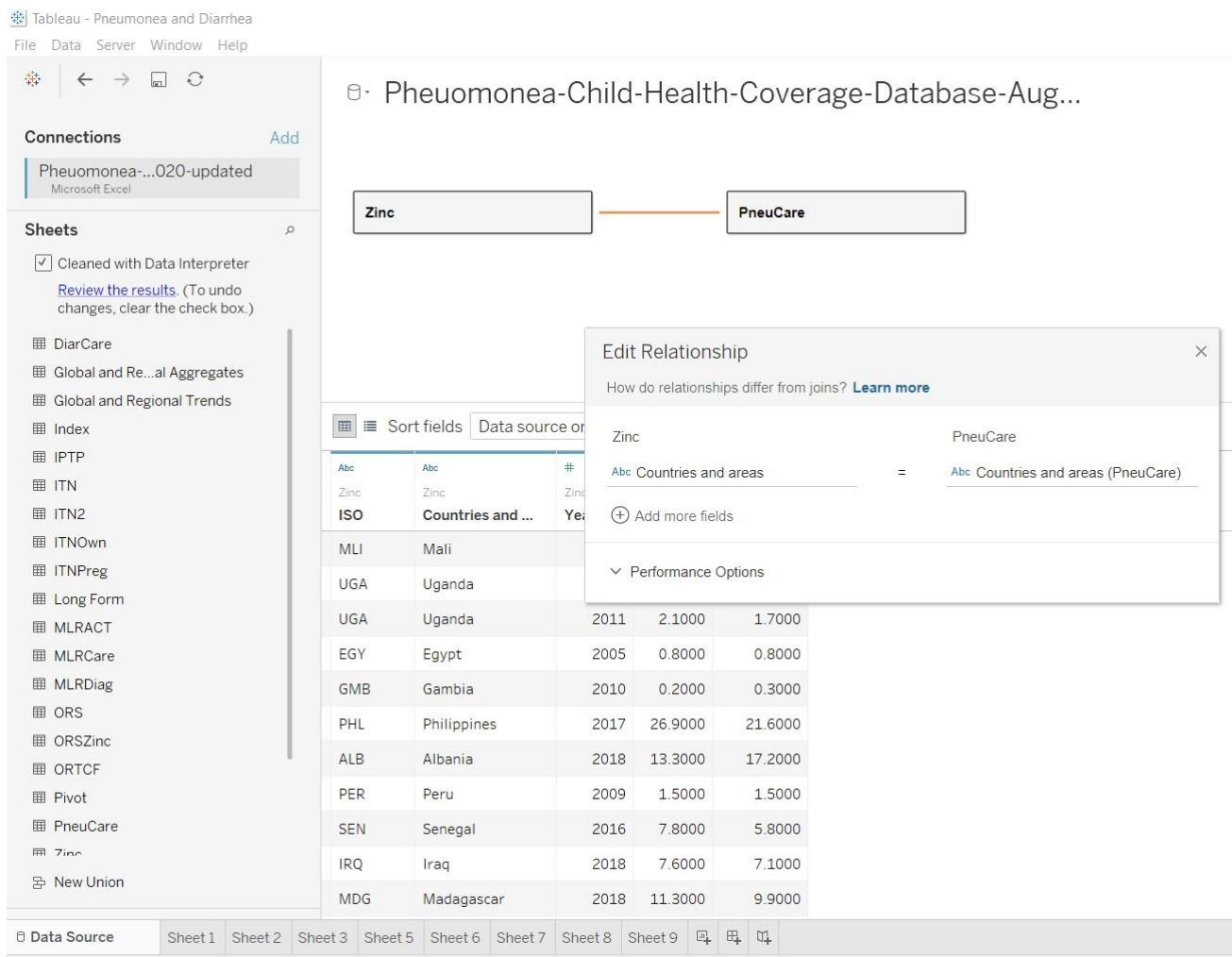


Figure 119: Concept of joins used in Dataset

- 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.

Year wise representation of Sex suffered from Diarrhea

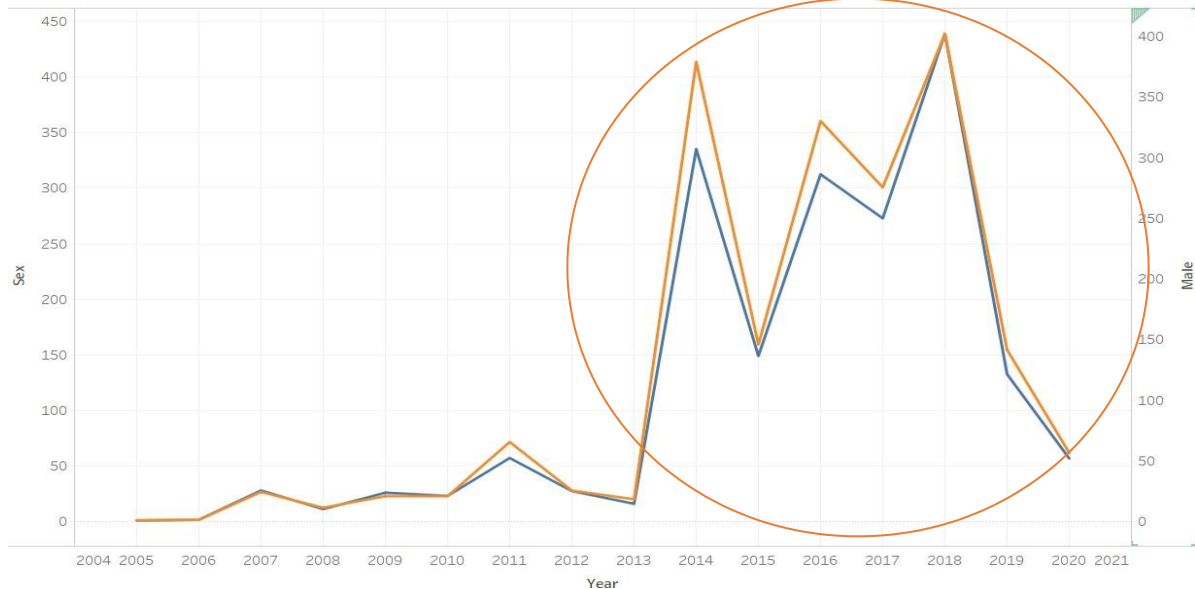


Figure 20: Year-wise representation of Gender suffered from Diarrhoea

- 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 diarrhea rather than Pneumonia. A place like Peru has an extremely large number of Cases of Pneumonia and vice versa for Diarrhoea.

Region wise comparative representation of Males suffering Diarrhea and Pneumonia

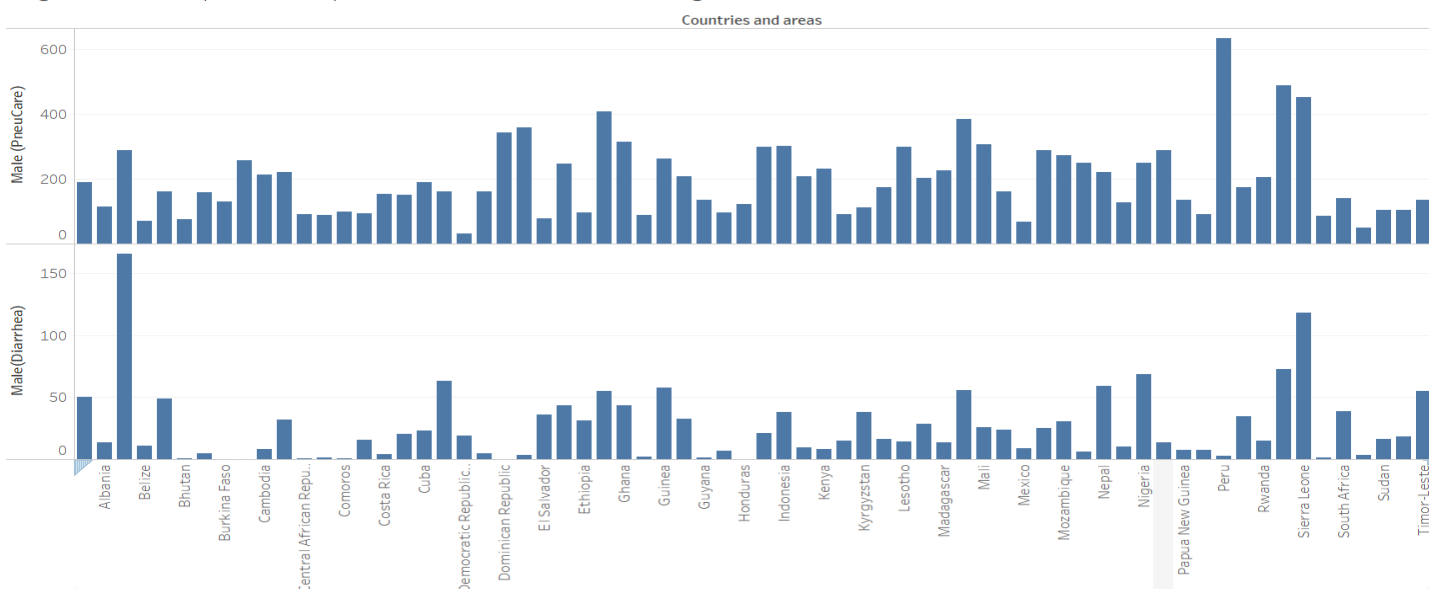


Figure 21: Region-wise comparative representation of Males suffering Diarrhoea and Pneumonia

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 diarrhea is very high and in some cases their proportion to that of males suffering from Diarrhoea and pneumonia.

Regional Comparative Representation of Sex Suffering from Diarrhea and Pneumonia

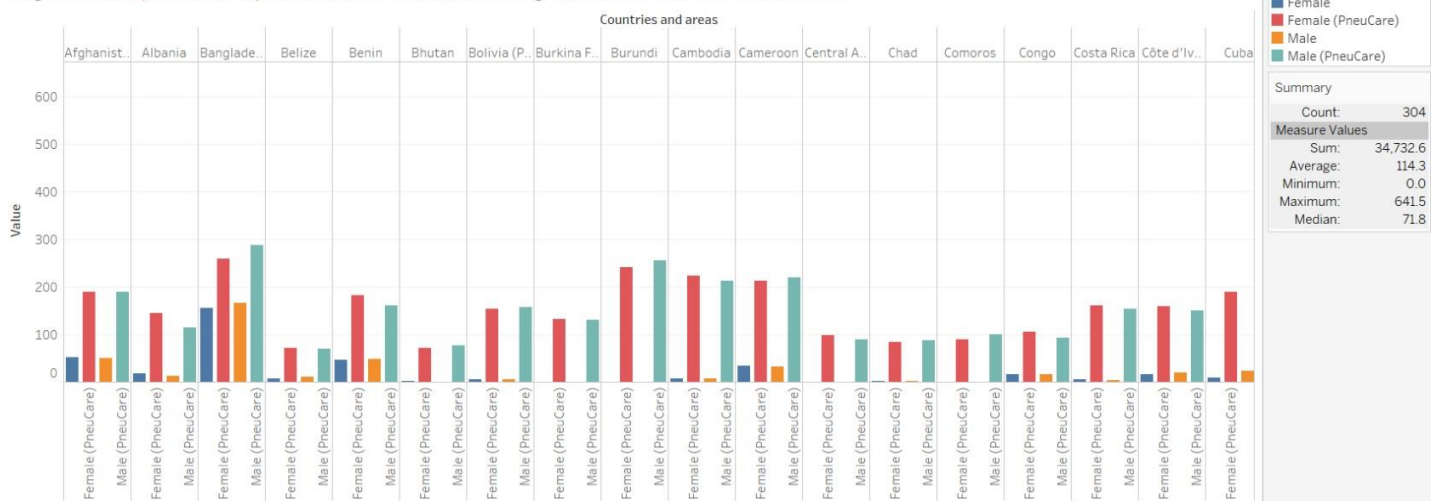


Figure 22: Regional Comparison of gender suffering from Diarrhoea and Pneumonia

CORONAVIRUS

Here we have imported the required packages and libraries, mostly we have used pandas for data analysis and manipulation, NumPy 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 23: 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. To know the exact number of rows and column in a dataset, shape function is used and to display the 1st 5 rows of dataset, head function is used.

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

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

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	3/7/21
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 x 419 columns

Figure 24: 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 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)

raw_data_confirmed2.head()
```

The Shape of Cornirmed is: (113710, 6)
The Shape of Cornirmed is: (113710, 6)
The Shape of Cornirmed 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 25: 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 26: 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 rows 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 27: 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

Figure28: Performing full joins on datasets

- Here we have used the strftime function of python 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

Figure29: 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.

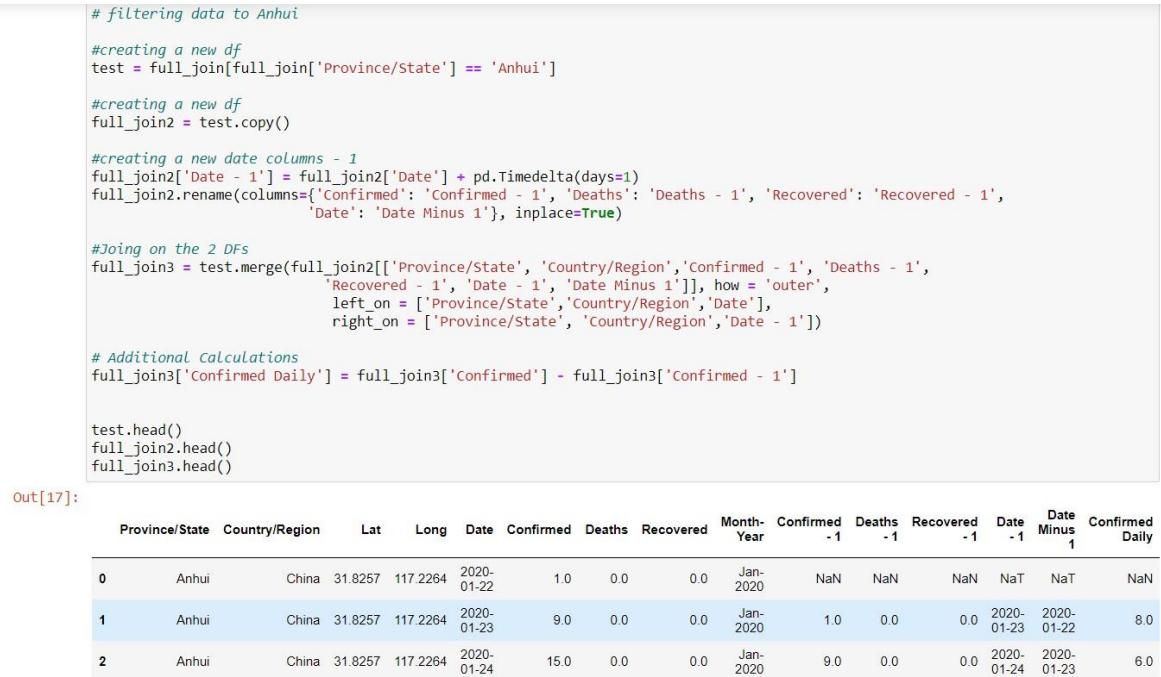


Figure30: Data Testing

- 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 31: 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.

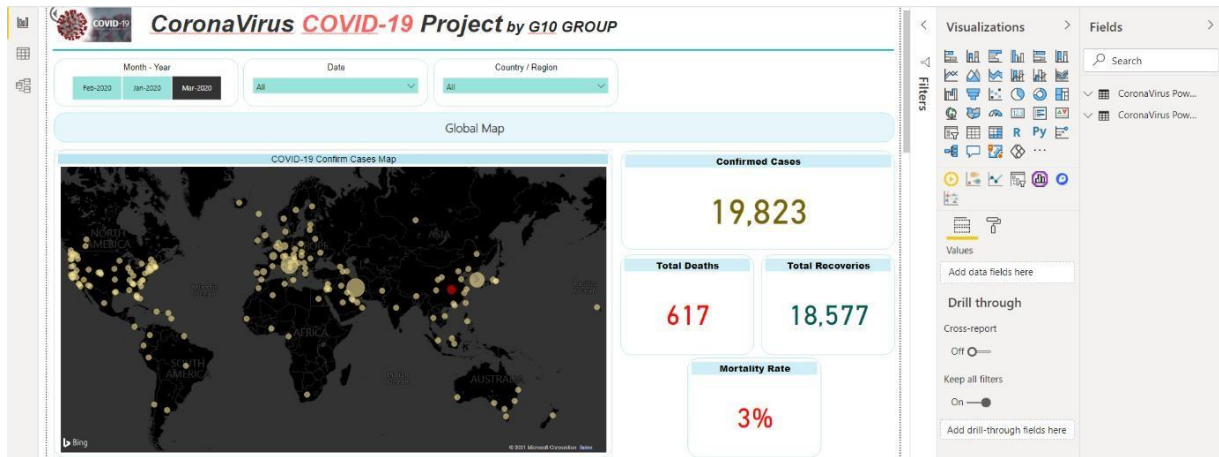


Figure 312: Project Dashboard



Figure 33: Coronavirus working model using power BI

- In this, we are showing a comparison of different prediction models of Python like the SVM prediction Model (Figure 26), Linear 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: 384.37718723946705
MSE: 113780.46033595514
```



Figure 34: 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>
```

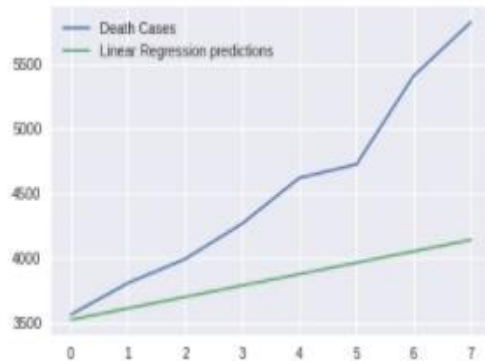


Figure 35: Coronavirus working model using Linear Regression Prediction

```
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>
```

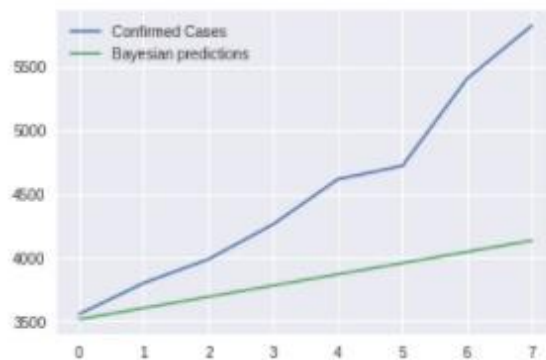


Figure 36: Coronavirus working model using Bayesian Prediction

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

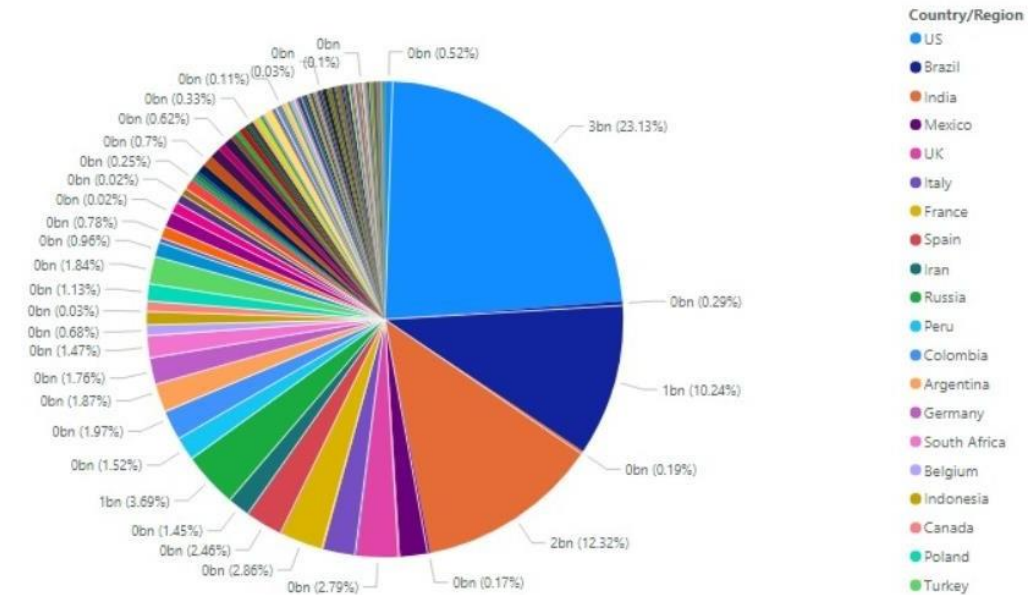


Figure 37: Deaths Confirmed by Countries

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



Figure 38: Confirmed and Deaths by Countries/Regions

- This graph shows which region of the country is most affected by dengue and how many people were dead due to this virus. Mainly the males were having a higher death rate than

the female as the dengue death rate is less than 0.2% if not treated then it could rise to 20% to 50%.

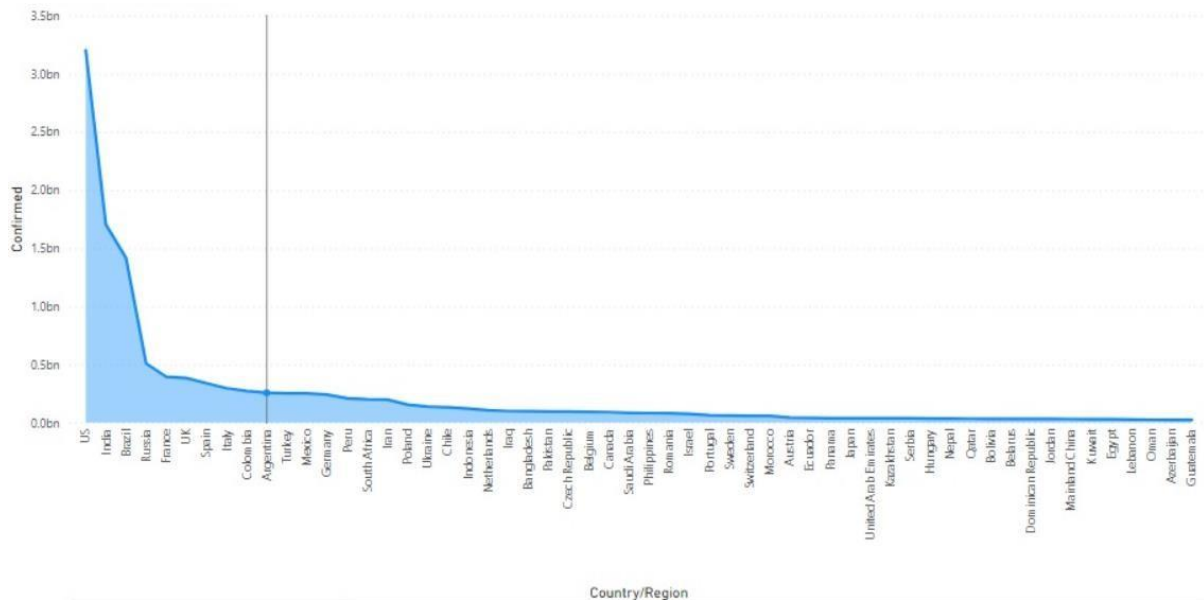


Figure 39: Confirmed by /Regions

Deployment

I have used visual studio code as the code editor to write the code for the complete website.

[index.html](#):

```

<? index.html > <? html > <? head > <? link
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Disease Prediction System</title>
8   <link rel="stylesheet" href="prediction.css">
9   <link rel="preconnect" href="https://fonts.googleapis.com">
10  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
11  <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@100;200;300;400;600;700&display=swap" rel="stylesheet">
12  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">
13 </head>

```

Figure 40: Index.html

<!DOCTYPE html> : Used to support the document for HTML5

<html>: It is the root element (1st element) for the HTML page.

<head>: Head is responsible for storing the meta-information and the Tab name of HTML page.

<title>: Title specifies the title for our HTML page

<meta> tags are always present inside the <head> tag element and are used to specify viewport(the area which is visible to the user and it varies from device to device), page description, authors for the current document, etc.

<link> tag helps us to understand the document which is linked to my current document. Here I have used the link tags to link the external CSS to index.html, to link the google fonts to the current HTML file, and also to include the bootstrap icons.

```
<body>
  <section class="header">
    <nav>
      <a href="index.html"></a>
      <div class="nav-links" id="navLinks">
        <i class="fa fa-times" onclick="hideMenu()"></i>
        <ul>
          <li><a href="index.html">HOME</a></li>
          <li><a href="blog.html">BLOG</a></li>
          <li><a href="contact us.html">CONTACT US</a></li>
        </ul>
      </div>
      <i class="fa fa-bars" onclick="showMenu()"></i>
    </nav>
    <div class="text-box">
      <h1>Disease Prediction System</h1>
      <p>
        The main objective of this research is to develop an Intelligent System using data Machine Learning<br> technique namely
      </p>
      <a href="contact us.html" class="hero-btn">Contact Us for Prediction</a>
    </div>
  </section>
```

Figure 41: Code Sample 1

<body>: This tag defines the body for the respective document. The <body> element contains the content for the HTML document like images, hyperlinks, tables ,lists, headings and we have only one <body> element present in a HTML document,

We have created a section with class="header" and in that, we have created nav tag which defines a set of navigation links, in the first link(to create a link we have used anchor tag <a>) we have used an image and linked it with index.html.

then we have created div with class="nav-links" and id="nav links" and in that, we have added bootstrap icon fa fa-times and on clicking on that hideMenu() function will execute then after that we have created an unordered list in which we have added three links for index.html,blog.html and contact us.html.We have created fa fa-bars which is a bootstrap icon and clicking on that showMenu() function will execute.

After that, we have created another div with class text-box in which we have used <h1> heading to give heading to our HTML page and a paragraph tag <p> to include paragraphs after that we have added the link contact us.html and then provided it the class="hero-btn" so that we can render this class in external CSS to provide It button shape.

```

37      <!-------disease predicted----->
38  ✓    <section class="campus">
39      <h1>Disease Predicted</h1>
40      <p>Below is the list of disease Predicted with higher accuracy:</p>
41  ✓    <div class="row">
42  ✓    <div class="campus-col">
43          
44  ✓    <div class="layer">
45          |    <h3>Allergy</h3>
46          </div>
47    </div>
48  ✓    <div class="campus-col">
49          
50  ✓    <div class="layer">
51          |    <h3>Pneumonia</h3>
52          </div>
53    </div>
54  ✓    <div class="campus-col">
55          
56  ✓    <div class="layer">
57          |    <h3>Dengue</h3>
58          </div>
59    </div>
60  </div>
61  </section>

```

Figure 42: Code Sample 2

We have created one section and assigned it to the class=" campus"

In this we have given heading and added paragraphs using the <h1> and <p> tags and inside this section, we have created two div one with class row and another with class campus col the div with a class row is the parent div and the divs with class="campus-col" can be considered as the child divs. In each child divs with campus-col, we have added images related to them and in this div with class="campus-col" we have another divs with classes layers and with <h3> type headings to style them in external CSS.

```

62     </div>
63     <section class="facilities">
64     <h1>Our facilities</h1>
65     <p>Below are some of the facilities provided by us:</p>
66
67     <div class="row">
68         <div class="facilities-col">
69             
70             <h3>Contact US</h3>
71             <p>Feel free to contact us we are there to help you.</p>
72         </div>
73         <div class="facilities-col">
74             
75             <h3>Free Of Cost</h3>
76             <p>Our services are completely free of Cost.</p>
77         </div>
78         <div class="facilities-col">
79             
80             <h3>Quick Response</h3>
81             <p>We resolve your Query within 24 hours.</p>
82         </div>
83     </div>
84     </section>

```

Figure 413: Code Sample 3

This is another section we have created with class="facilities" in which initially we have given some paragraphs and headings

Similar to the section "campus-col" we have again created one parent div with class="row" and inside them, we have created child divs with class "facilities-col" to add images, headings, and paragraphs.


```

<!--reviews-->
<section class="testimonials">
<h1>What our user says: </h1>
<p>Ratings</p>
<div class="row">
  <div class="testimonial-col">
    
    <div>
      <p>
        John Sharma
      </p>
      <h3>Excellent service</h3>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star-o"></i>
    </div>
  </div>
  <div class="testimonial-col">
    
    <div>
      <p>
        Tanya Gupta
      </p>
      <h3>Quick response</h3>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star"></i>
      <i class="fa fa-star-half-o"></i>
    </div>
  </div>
</div>
</section>

```

Figure 414: Code Sample 4

After this, we have created another section for reviews with class="testimonials" in which we have added paragraphs and headings using the <p> and <h1> tags in this also we have created one div which acts as a container with class row and inside it we have created another divs with class="testimonial-col" to add the columns now in each div which are acting as columns we have used user images and to give reviews with stars we have directly used the bootstrap star icons as shown with class"fa fa-star"

```

<!-------footer----->
<section class="footer">
  <h4>About Us</h4>
  <p>We hereby certify that the work, which is being presented, in partial fulfilment of the requirement<br> for the award of the Degree
<div class="icons">
  <i class="fa fa-facebook"></i>
  <i class="fa fa-twitter"></i>
  <i class="fa fa-instagram"></i>
  <i class="fa fa-linked-in"></i>
</div>
<p>Made with <i class="fa fa-heart-o"></i> by IBM-G10 group</p>
</section>
<script>
  var navlinks=document.getElementById("navlinks");
  function showMenu(){
    navlinks.style.right="0";
  }
  function hideMenu(){
    navlinks.style.right="-200px";
  }
</script>
</body>
</html>

```

Figure 415: Code Sample 5

Now, this is the footer part where we have created a section with class="footer" in which we have used <h4> heading tag and a <p> paragraph tag

after that we have created one div with class="icons" to add the bootstrap icons for Facebook, Twitter, Instagram, linked-in and in the paragraph tag also after this we have added one bootstrap icon with a message Made with (bootstrap icon) by IBM-G10 group

after this section we have used the <script> tags to write the javascript in which first using the document.getElementById we have fetched the id of nav links.

and we have written two functions showMenu() and hideMenu() to hide and show the menu whenever this function executes.

blog.html

```

<!-------blogs----->
<section class="about-us">
  <div class="row">
    <div class="about-col">
      <h1>Corona Virus</h1>
      <p>Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. The best way to prevent and slow down transmission is to be well informed about the COVID-19 virus, the disease it causes and how it spreads. The COVID-19 virus spreads primarily through droplets of saliva or discharge from the nose when an infected person coughs or sneezes, so it is important to practice good hygiene when around others.
    </div>
    <div class="about-col">
      
    </div>
  </div>
  <div class="row">
    <div class="about-col">
      
    </div>
    <div class="about-col">
      <h1>Ebola</h1>
      <p>Ebola is a serious and deadly virus transmitted by animals and humans. It was initially detected in 1976 in Sudan and the Democratic Republic of the Congo. Although the Ebola virus has been present for more than 35 years, the largest outbreak began in West Africa in March 2014. The virus is spread by direct contact with blood or other body fluids of a person who has the disease.
    </p>
    </div>
  </div>
</section>

```

Figure 416: blog.html

For writing the blogs we have created a section with class="about-us" and inside this we have created one parent div with class="row" in which we have created the child divs with class="about-col" which can be considered as creating columns for the current parent div.

now for writing blogs about each virus we have used <h> tag as heading for specifying the virus name and the image tag to add the particular virus image and in the paragraph tag, we have written the blog for each type of viruses as mentioned in the heading.

contact us.html

```
<!--contact us-->
<section class="location">
  <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3442.8598869458638!2d78.08112941473561!3d30.35493168177045!2m3!1s!1s!1s" />
</section>
<section class="form1">
  <div class="container">
    <form action="https://formspre.io/f/xdoylped" id="test-form" method="POST">
      <div class="form-group">
        <label for="firstName">First Name</label>
        <input type="text" id="firstName" name="firstName">
      </div>
      <div class="form-group">
        <label for="lastName">Last Name</label>
        <input type="text" id="lastName" name="lastName">
      </div>
      <div class="form-group">
        <label for="email">Email</label>
        <input type="email" id="email" name="Email">
      </div>
      <div class="form-group">
        <label for="message">5 Symptoms:</label>
        <textarea name="message" id="message" rows="10" maxlength="200"></textarea>
      </div>
      <button type="submit" id="test-form-submit">Submit</button>
    </form>
  </div>
  <div id="status"></div>
</section>
```

Figure 417: Code Sample 6

For this particular document we have created a section with class="location" and inside this section, I have <iframe> tag which specifies an inline frame to embed another document within our current HTML document and using this iframe tag we have embed map to get the current location from the map.

After this, we have created another section with class="container" for creating a form and inside this, we have one parent div and with class="form1" and child divs with class="form-group"

now the action attribute of the form specifies where the form data will be sent when we submit the form.

The “POST” method sends the form data as an HTTP-post-transaction, the label tag is used to define labels for different input types and the input type is telling us which type of input we need to take in that particular input field.

in the action attribute, we have added “Formspree API” which is an API used to send the details of the form directly to our Gmail id.

<button type=”submit”> specifies that the button is a submit button that is used to submit the form data.

prediction.css

In this CSS, we are styling our HTML elements.

```
*{
  margin: 0;
  padding: 0;
  font-family: 'Poppins', sans-serif;
}
```

Figure 418: CSS

The browser has its CSS which is applied to the HTML pages. So by using the * which is the CSS Universal Selectors we have made the default margin to 0, default padding to 0, and default font family to ‘Poppins’, sans-serif.

```
.header{
  min-height: 100vh;
  width: 100%;
  background-image: linear-gradient(□rgba(4,9,30,0.7),□rgba(4,9,30,0.7)),url(images/9.png);
  background-position: center;
  background-size: cover;
  position: relative;
}
```

Figure 419: Code Sample 7

This is a Class selector that selects defined items class attribute, we can assign the same class to multiple HTML elements but each id is unique for HTML elements, and to access that id of HTML elements in CSS we use #.

The min-height is used to define the minimum height for the HTML element and only if it is smaller than the minimum height, the minimum height will be applied.

width: 100% will allow the current HTML element to occupy the complete width of the container.

linear-gradient is used to display transitions between two or more than two colors

background image is used to set the image as background and background-size and position are other properties for the background.

position: relative: here we have set the position as relative so that we can set the position of its child container as absolute which will help us to set the child content position relative to its parent.

```

}
nav{
  display: flex;
  padding: 2% 6%;
  justify-content: space-between;
  align-items: center;
}

```

Figure 20: Code Sample 8

The flex container becomes flexible by setting the display property to flex so after getting a flex container we have done The "justify-content: center;" which aligns the flex items at the center of the container and we have set align-items: center. The align-items property specifies the default alignment for items inside the flexible container.

padding: padding is the space between the content and the border.

The **justify-content** property aligns the flexible container's items when the items do not use all available **space** on the main axis (horizontally).

```

}
nav img{
  width: 120px;
  border-radius: 10px;
  height: 100px;
}

```

Figure 51: Code Sample 9

setting the width, height, and border-radius(to make the border rounded) to the nav

Here we are doing the styling for the navigation bar like text-decoration is set to none to remove the underline that we used to see in the anchor tags.

the display is set as inline-block because as compared to display: inline, the major difference is that display: inline-block allows to set a width and height on the element. The position is set

as relative which is a positioning mechanism to set the position of an element relative to its parent.

```

}
.nav-links ul li::after{/*to add line after by making its display block to assign new line to this line*/
  content: '';
  width: 0;
  height: 2px;
  background: #f44336;
  display: block;
  margin:auto;
  transition: 0.5s;
}
.nav-links ul li:hover::after{
width:100%;
}
.text-box{
  width: 90%;
  color:#fff;
  position: absolute;
  top: 50%;
  left:50%;
  transform: translate(-50%,-50%);/*bringing div to center*/
  text-align: center;
}
.text-box h1{
  font-size: 62px;
}
.text-box p{
  margin: 10px 0 40px;
  font-size: 14px;
  color: #fff;
}
}

```

Figure 52: Code Sample 10

In CSS, **::after** creates a pseudo-element that is the last child of the selected element.

the after pseudo-element is used here to add a line after the HTML element with an orange background and the transition time is 0.5 sec

and hover is used to set properties like what happens when the user hovers the mouse over the current HTML element.

margin property is used to create space around elements

other properties like color is used to set the color, text aligns to align the text, and transform to bring the div to the center.

display: Displays element as a block element. It takes the whole width while starting from new line.

```

}
@media(max-width:700px){
  .text-box h1{
    font-size: 20px;
  }
  .nav-links ul li{
    display: block;
  }
  .nav-links{
    position:fixed;
    background: ■ #f44336;
    height: 100vh;
    width: 200px;
    top: 0;
    right: -200px;
    text-align: left;
    z-index: 2;
    transition: 1s;
  }
  nav .fa{
    display :block;/*to make it visible*/
    color: ■ #fff;
    margin: 10px;
    font-size: 22px;
    cursor: pointer;
  }
  .nav-links ul{
    padding: 30px;
  }
}
}

```

Figure 53: Code Sample 11

Here we have added the media queries to apply different styles for devices of different resolutions

here @media(max-width:700) is telling the browser what are the styling we need to perform when the width of the device reaches to 700px

To specify the stack order of element z-index property is used.

An element in front have greater stack order in compare with a lower stack order that is how when the screen size reduces to 700px then the elements with class="nav-links" becomes visible since they are having greater z index;

An ‘fixed’ element position: is relative to the viewport, which means it cannot be scrolled.

display: block will allow the element to occupy the entire width of its parent container.

```

/*-----course-----*/
.course{
  width: 80%;
  margin:auto;
  text-align:center;
  padding-top: 100px;
}
h1{
  font-size: 36px;
  font-weight: 600;
}
p{
  color: #777;
  font-size: 14px;
  font-weight: 300;
  line-height: 22px;
  padding: 10px;
}
.row{
  margin-top: 5%;
  display: flex;
  justify-content: space-between;
}
.course-col{
  flex-basis: 31%;
  background: #fff3f3;
  border-radius: 10px;
  margin-bottom: 5%;
  padding: 20px 12px;
  box-sizing: border-box;
  transition:0.5s;
}
h3{
  text-align: center;
  font-weight: 600;
  margin: 10px 0;
}

```

Figure 54: Code Sample 12

In this part, we have added the CSS for the course part and the properties like: `box-sizing`: Defines the calculation width and height of an element: whether or not to include borders & padding.

`flex-basis`: A [CSS](#) property that sets the main size of a flex item. Set size of the content box set with `box-sizing`.

`justify-content: space-between`: This property will make sure that we have equal spaces between the items.

`line-height`: specifies line height.

```
}  
@media(max-width: 700px){  
  .row{  
    flex-direction: column;  
  }  
}
```

Figure 55: Media Query

Here we have used the media query for the div with class=row and sets its flex-direction: column, flex-direction is the way for assigning the flex items row-wise or column-wise by default flex-direction is row means all the items inside the flex container will be present In a row but when we do flex-direction: column it means when the width becomes 700px then the flex items will be aligned columns.

```

/*-----campus-----*/
.campus{
  width: 80%;
  margin:auto;
  text-align: center;
  padding-top: 50px;
}
.campus-col{
  flex-basis: 32%;
  border-radius: 10px;
  margin-bottom: 30px;
  position: relative;
  overflow: hidden;
}
.campus-col img{
  width: 100%;
  display: block;
}
.layer{
  background: transparent;
  height: 100%;
  width: 100%;
  position: absolute;
  top: 0;
  left: 0;
  transition: 0.5s;
}
.layer:hover{
  background: ■ rgba(226,0,0,0.7);
}
.layer h3{
  width: 100%;
  font-weight: 500;
  color: ■ #fff;
}

```

Figure 56: Styling

Here we have provided the styling for the campus section and the `overflow: hidden` property means that here the overflow is clipped, and the rest of the content will be invisible

`.layer:hover` is used to add a red layer on the divs when we hover on them.


```

}
/*-----footer-----*/
.footer{
width: 100;
text-align: center;
padding:30px 0;
}
.footer h4{
margin-bottom: 25px;
margin-top: 20px;
font-weight: 600;
}
.icons .fa{
color: #f44336;
margin: 0 13px;
cursor:pointer;
padding: 18px 0;
}
.fa-heart-o{
color: #f44336;
}
/*-----about us page-----*/

```

Figure 57: Footer

For designing the footer we have targeted different bootstrap icons and added different colors, margins to them

Disease Prediction Model:

working of the system:

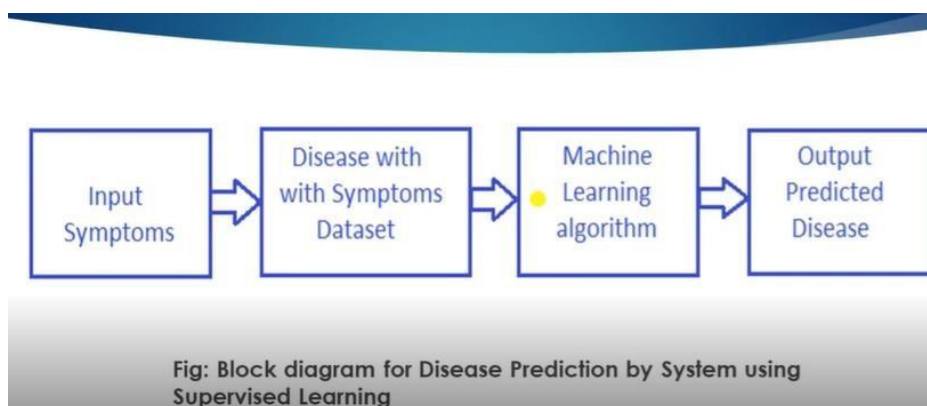
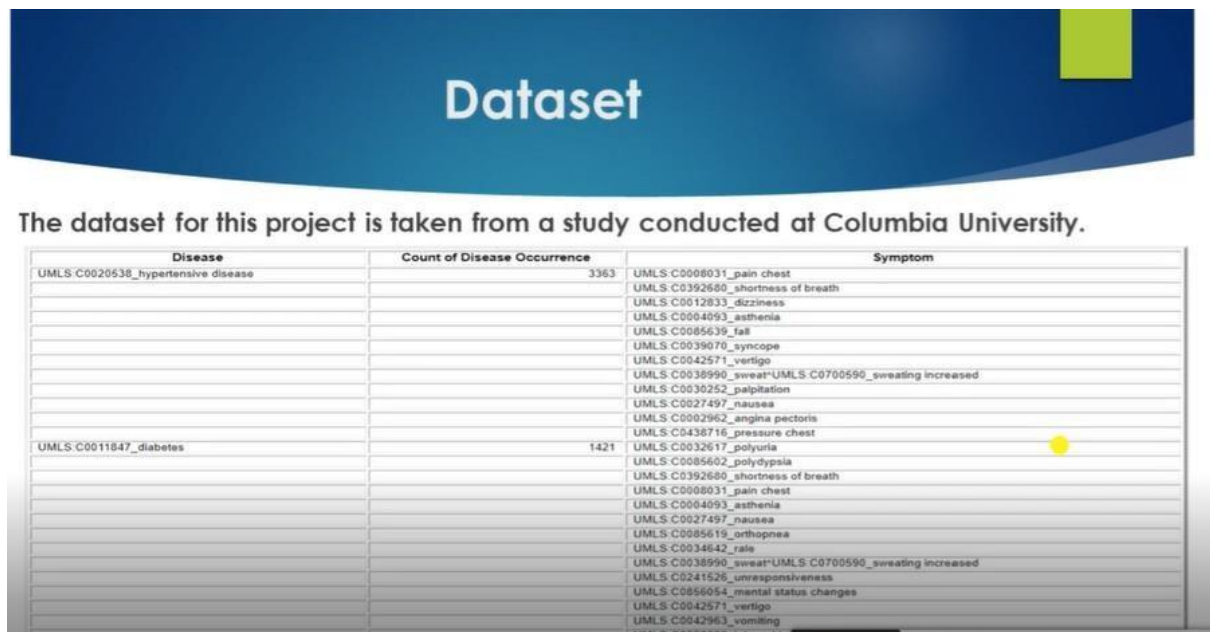


Figure 58: Workflow

we have to give symptoms as input and there will be a dataset that contains symptoms and their parallel diseases, these would be analyzed using a machine learning algorithm and the output would be predicted. Diseases will be predicted using naïve Bayes algorithm which works on probabilistic approach most specifically multinomial naïve bayesian since there can be multiple symptoms



The dataset for this project is taken from a study conducted at Columbia University.

Disease	Count of Disease Occurrence	Symptom
UMLS:C0020538_hypertensive disease	3363	UMLS:C0008031_pain chest
		UMLS:C0392680_shortness of breath
		UMLS:C0012833_dizziness
		UMLS:C0004093_asthenia
		UMLS:C0085639_fall
		UMLS:C0039070_syncope
		UMLS:C0042571_vertigo
		UMLS:C0038990_sweat-UMLS:C0700590_sweating increased
		UMLS:C0030252_palpitation
		UMLS:C0027497_nausea
		UMLS:C0002962_angina pectoris
		UMLS:C0438716_pressure chest
		UMLS:C0032617_polyuria
		UMLS:C0085602_polydipsia
		UMLS:C0392680_shortness of breath
		UMLS:C0008031_pain chest
		UMLS:C0004093_asthenia
		UMLS:C0027497_nausea
		UMLS:C0085619_orthopnea
		UMLS:C0034642_rate
		UMLS:C0038990_sweat-UMLS:C0700590_sweating increased
		UMLS:C0241526_unresponsiveness
		UMLS:C0080054_mental status changes
		UMLS:C0042571_vertigo
		UMLS:C0042963_vomiting
		UMLS:C0053568_tachycardia
UMLS:C0011847_diabetes	1421	

Figure 59: Dataset

according to a literature survey, this algorithm results in maximum accuracy for larger datasets.

The dataset contains disease as the label and for each disease, there are symptoms given, this dataset has been taken from a study conducted at Columbia University. Naïve Bayesian is a classification algorithm based on applying Baye's theorem with the assumption that there is no dependency between predictors that simply means that the presence of a feature in a class is independent of the presence of any other feature in the same class for example smartphone, if it has following features like camera, touch screen, internet, etc. so all the features are dependent somewhere or the other but still they contribute individually to the probability of the phone that is it is a smartphone. We have imported the necessary modules Tkinter is basic GUI, NumPy to perform numerical operations, and pandas which are needed to read the CSV file. in the beginning, the list contains all the symptoms that we have considered the symptoms have been written in the order in which the dataset has been made same order as the list and another list contains all the diseases 41 diseases are taken into consideration list l2[] is empty l1 and l2 has the same length and l2 will contain every element as 0.

	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED
1	coma	stomach	distention	history_of	fluid_over	blood_in	prominen	palpitatio	painful_w	pus_filled	blackhead	scarring	skin_peel	silver_like	small_der	inflamm	blister	red_sore	yellow_cr	prognosis	
2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0 Fungal infection	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Fungal infection	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0 Allergy	

Figure 60: Sample Dataset

prognosis is the opinion ie if these are the symptoms then the disease is a fungal infection. input test will have the 5 symptoms entered by a user using the .get() method. these are the 5 symptoms variables we have taken symptom 1 is a type of string var that is an object of stringVar class and initially, I have set it as none similarly for the rest of the symptoms variable

```

76
77 def NaiveBayes():
78     from sklearn.naive_bayes import MultinomialNB
79     gnb = MultinomialNB()
80     gnb=gnb.fit(X,np.ravel(y))          # [[1,2,3],[4,5,6]] -> [1,2,3,4,5,6]
81     from sklearn.metrics import accuracy_score
82     y_pred = gnb.predict(X_test)
83     print(accuracy_score(y_test, y_pred))
84     print(accuracy_score(y_test, y_pred, normalize=False))
85
86     psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]
87
88     for k in range(0,len(l1)):
89         for z in psymptoms:
90             if(z==l1[k]):

```

Figure 61: Naive Classifier

sklearn: package in python containing Machine Learning algorithm in line 78 it means in sklearn uses Naïve Bayes and import the multinomial naïve Bayes in line 79 an object and is being created of class multinomialNB in. fit this fit function is used, its basic function is to fit the naïve Bayes classifier according to x and y where x is Training vectors and y is Targeted value ravel function is used to convert the 2d arrays to a single array like so fit has x,y so here y=np. ravel(y)) from the sklearn metrics package, we are importing accuracy score to find the accuracy of the algorithm predict function performs classification of an array of the test vectors so X_test parameter will be taken into consideration and y would be predicted based on the algorithm, X_test would be my symptoms and y is the disease predicted using the algorithm and y_test is the disease which I have predefine so the accuracy score would be defined like if my predicted i.e the algorithm that predicts is equal to the one I desire then the

accuracy score would be more and maximum accuracy score will be 1. By default normalize is true ,line 83 returns a float type the algorithm is more accurate if it gives maximum value as 1 when normalize is false it returns an int type it returns the number of samples which has predicted to be true if all the samples has been predicted to be true using the algorithm and the desired output has been matched then it gives the number of samples which is found to be matched psymptoms list will then store all the symptoms now we will iterate till l2.length actually since both l1 and l2 has the same length so we will iterate on psymptoms list if this predicted symptoms matches to our list symptoms l1 then at that index I will store 1 in the l2 list since initially l2 contains only 0 so the position at which we get the symptoms i.e at kth index I will store 1 at that position in l2 list now we store the updated l2 list in the inputtest variable: the predict function will predict the inputtest (which is given by the user using get function) predict function returns an array which has only one element so here predict is the index

```

h='no'
for a in range(0,len(disease)):
    if(disease[predicted] == disease[a]):
        h='yes'
        break

if (h=='yes'):
    t3.delete("1.0", END)
    t3.insert(END, disease[a])
else:
    t3.delete("1.0", END)
    t3.insert(END, "No Disease")

root = Tk()
root.title(" Disease Prediction From Symptoms")
root.configure()

```

Figure 62: Divider Code

if predicted = 23 then I am checking if disease[23] is present in my disease list basically if the disease is there then the accuracy is 1 so t3 is the textbox so we are deleting what is present in that initially and then entering the disease.

Deployment of Machine Learning Model using Flask:

model.py

```
from sklearn.naive_bayes import MultinomialNB
gnb = MultinomialNB()
gnb=gnb.fit(X,np.ravel(y))
pickle.dump(gnb,open('model.pkl','wb'))
model=pickle.load(open('model.pkl','rb'))
```

Figure 63: Model.py

model.py is the model building file that basically means that this py file will be responsible for creating our model and this also involves feature engineering, all the data pre-processing, after doing a fit our model will be ready after that we have just used pickle. dump this pickle is coming from the pickle library so this pickle helps us to create a pre-compiled format model name which will just be like a file that will have an extension like a dot pkl.

app.py

This is one more py file known as app.py file in this what we have done is we have written our flask code in which we have created the API like how our API URL looks like, like our localhost address /predict and another API is basically /predict_api.

```
app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
```

Figure 64: Flask

in this piece of code, we have initialized our flask app and open our model.pkl file in reading mode to make sure our model is ready.

```
@app.route('/predict',methods=['POST'])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = [(x) for x in request.form.values()]
    psymptoms = [np.array(int_features)]
```

Figure 65: Prediction

When the predict function executes then the five symptoms that we have provided as user input these inputs will be read from the form using request.form.values() since it is a post request and after that, we have converted these inputs to array and store them in symptoms. We have used model.predict to predict the index of the disease based on the user inputs and if that disease at that index is present in our dataset then we will be displaying that disease on our webpage.

Front End Part For Disease Prediction Model:

```

index.html > html > body
1  <!DOCTYPE html>
2  <html >
3  <head>
4      <meta charset="UTF-8">
5      <title>ML API</title>
6      <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
7      <link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
8      <link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
9      <link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
10     <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
11
12 </head>
13
14 <body>
15     <div class="login">
16         <h1>Disease Prediction System by G16 group</h1>
17
18         <!-- Main Input For Receiving Query to our ML -->
19         <form action="{{ url_for('predict') }}" method="post">
20             <input type="text" name="experience" placeholder="Enter Symptom 1" required="required" />
21             <input type="text" name="test_score" placeholder="Enter Symptom 2" required="required" />
22             <input type="text" name="interview_score" placeholder="Enter Symptom 3" required="required" />
23             <input type="text" name="interview_score" placeholder="Enter Symptom 4" required="required" />
24             <input type="text" name="interview_score" placeholder="Enter Symptom 5" required="required" />
25
26             <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
27         </form>
28
29         <br>
30         <br>
31         {{ prediction_text }}
32     </div>
33
34 </body>
35
36 </html>

```

Figure 66: Front End

This file will act as our front-end web app so that any request that we give to our model will be in the form of API which we have hosted through flask it will interact with that, get the output from that particular API itself.

Chapter 5 – Conclusion

We are at the end of what we have started. Our special thanks to Mentor “Manish Sharma Sir”

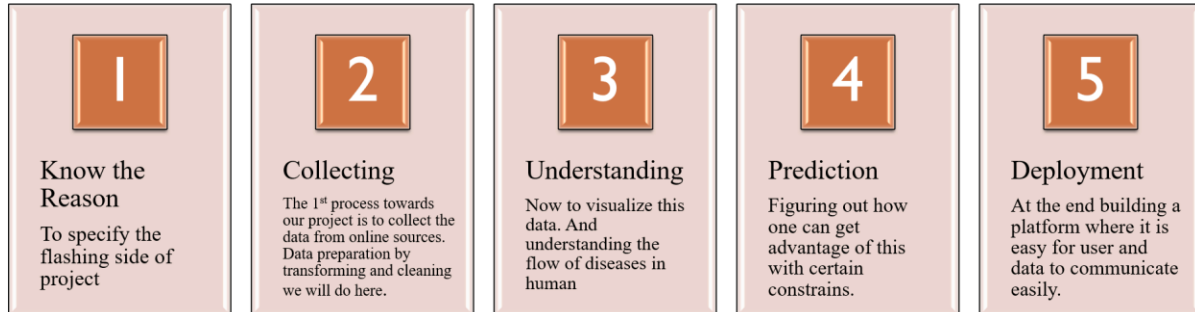


Figure 67: Flowchart

And that’s how our site looks:

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000". The page has a dark blue gradient background. At the top, it says "Disease Prediction System by G10 group". Below this, there are five input fields labeled "Enter Symptom 1" through "Enter Symptom 5". At the bottom of these fields is a blue button labeled "Predict".

Figure 68: Default View

Disease Prediction System by G10 group

Figure 69: Entered Values/Symptoms

Disease Prediction System by G10 group

Predict

You might be suffering from Fungal
infection

Figure70: Final Deployment

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/>

IBMG10 Plag-2

ORIGINALITY REPORT

15%

SIMILARITY INDEX

10%

INTERNET SOURCES

4%

PUBLICATIONS

10%

STUDENT PAPERS

PRIMARY SOURCES

1

index-of.es

Internet Source

1%

2

guide.freecodecamp.org

Internet Source

1%

3

Submitted to DIT university

Student Paper

1%

4

Submitted to Govind Ballabh Pant Engineering College, Pauri-Garhwal

Student Paper

1%

5

Submitted to Netaji Subhas Institute of Technology

Student Paper

1%

6

Submitted to Gitam University

Student Paper

1%

7

Submitted to Athlone Institute of Technology

Student Paper

1%

8

Submitted to University of Essex

Student Paper

1%

9	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1 %
10	morioh.com Internet Source	<1 %
11	Submitted to Indian School of Mines Student Paper	<1 %
12	Samuel Lalmuanawma, Jamal Hussain, Lalrinfela Chhakchhuak. "Applications of machine learning and artificial intelligence for Covid-19 (SARS-CoV-2) pandemic: A review", Chaos, Solitons & Fractals, 2020 Publication	<1 %
13	medium.com Internet Source	<1 %
14	Submitted to Universiti Teknologi MARA Student Paper	<1 %
15	redabacus.eu Internet Source	<1 %
16	Submitted to University of Northampton Student Paper	<1 %
17	www.scribd.com Internet Source	<1 %
18	Submitted to University of Strathclyde Student Paper	<1 %

19	Submitted to Brooklands College Student Paper	<1 %
20	Pudumalar S, Suriya K S, Rohini K. "chapter 8 Data Classification and Prediction", IGI Global, 2018 Publication	<1 %
21	Submitted to Universidad Del Magdalena Student Paper	<1 %
22	Submitted to Stourbridge College Student Paper	<1 %
23	www.dataquest.io Internet Source	<1 %
24	Submitted to Griffith College Dublin Student Paper	<1 %
25	www.firstour.pt Internet Source	<1 %
26	tinyfinch.com Internet Source	<1 %
27	Submitted to Melbourne Institute of Technology Student Paper	<1 %
28	Submitted to Queen Mary and Westfield College Student Paper	<1 %

29	Internet Source	<1 %
30	Submitted to Shri Guru Gobind Singhji Institute of Engineering and Technology Student Paper	<1 %
31	Submitted to Institute of Technology, Sligo Student Paper	<1 %
32	Submitted to Universidade do Porto Student Paper	<1 %
33	aistha.blogspot.com Internet Source	<1 %
34	Md. Deloar Hossain, Md. Shafiul Azam, Md Jahan Ali, Hakilo Sabit. "Drugs Rating Generation and Recommendation from Sentiment Analysis of Drug Reviews using Machine Learning", 2020 Emerging Technology in Computing, Communication and Electronics (ETCCE), 2020 Publication	<1 %
35	"Big Data Analysis and Deep Learning Applications", Springer Science and Business Media LLC, 2019 Publication	<1 %
36	Sumathi S., Indumathi S., Rajkumar S.. "chapter 9 Medical Reports Analysis Using	<1 %

Natural Language Processing for Disease Classification", IGI Global, 2020

Publication

37	Submitted to University of Kent at Canterbury Student Paper	<1 %
38	buddhikaya.wordpress.com Internet Source	<1 %
39	idoc.pub Internet Source	<1 %
40	"Intelligent Systems Design and Applications", Springer Science and Business Media LLC, 2020 Publication	<1 %
41	machinelearningmastery.com Internet Source	<1 %
42	www.slideshare.net Internet Source	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On