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

GROUP 10

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.

WORK-FLOW

KNOW THE REASON

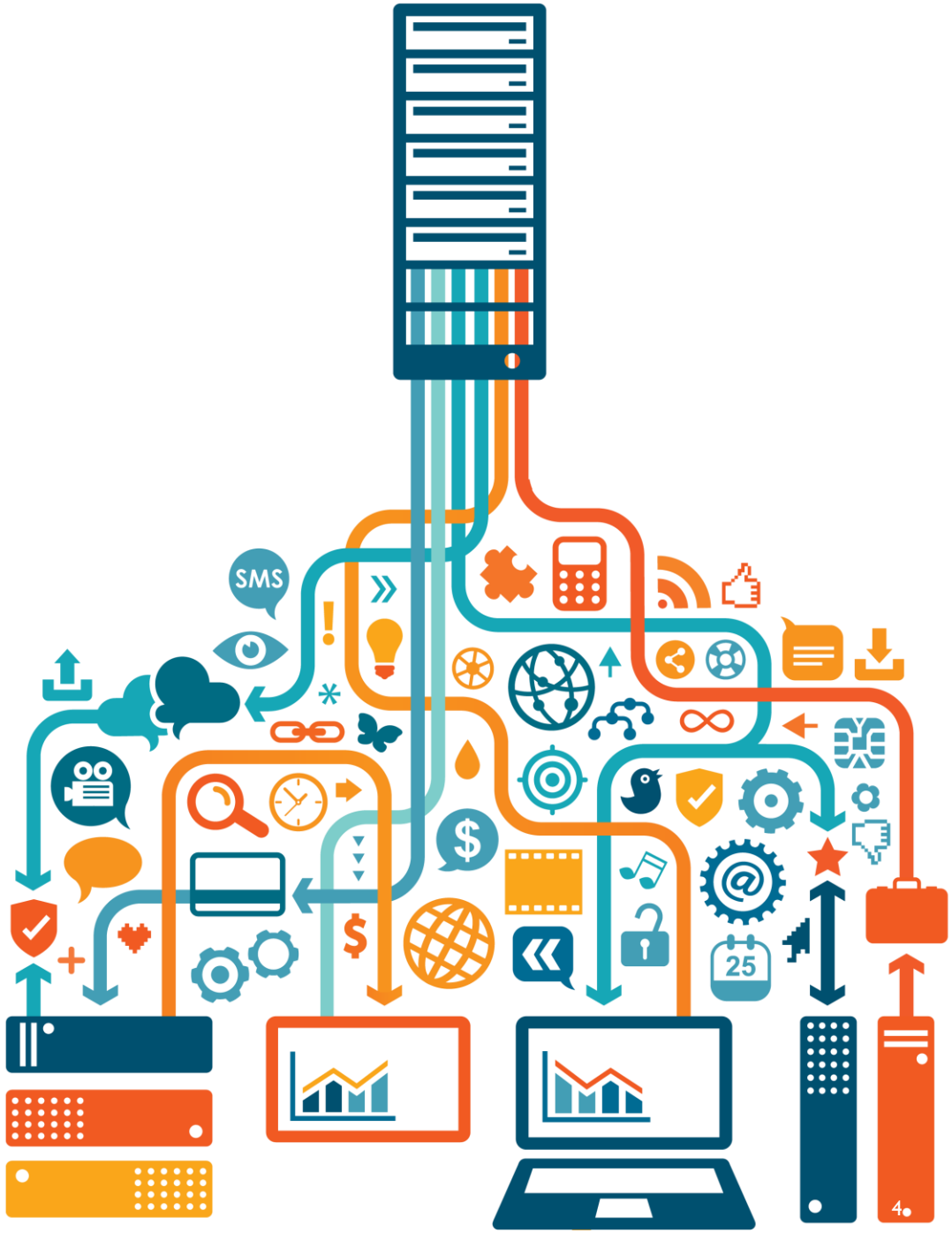
- Analysis on this dataset can help us to understand human symptoms.
- It helps us figuring out the future scope of viral diseases and their symptoms too.
- Not only this for an accuracy of 80% this limit bounded analysis can be helpful for the further future researches where in they would have future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

COLLECTING

We have reached to our dataset long after browsing through web:

- The data we have collected are in format of xlsx, csv.
- We have gathered our data form sources like CERN, Kaggle, UNICEF, Data World etc.
- The data that we have collected is very incompatible to our systems
- So our next step is to prepare it for our understanding.
- The datasets we have till date are of :

COVID , Dengue , HIV-AIDS , Hanta Fever , Pneumonia , Ebola , Influenza , Zika Virus.



DATA PREPARATION

BUILDING FROM MEANING LESS TO MEANINGFUL

Since our data contain too many missing and null values so we have decided to filter out some of these values as well as filling some of them. For that
For that we are using some libraries and packages from python, R

Now we are ready to understand this data based on some constraints that are required for our analysis.

CONSTRAINTS

- At each visualization we will focus on the relationship between human and virus.
- More specifically we will study the effect on human based on gender, age, time, intensity, place.
- We will also look for symptoms reflected and their commonness.
- We will neglect the analysis above and below a specified age.
- We will relate the age with disease categorized by place.

METHODS AND MATERIALS

After collecting the datasets from various sources they include the data that is incorrect, incomplete, irrelevant, duplicated or formatted in different data type so this data is not helpful in terms of analyzing because it may provide incorrect results so in this current project phase, we check whether our dataset is cleaned or not. Most of our data sets are accurate and does not require cleaning of our data , if it is required, we will do data cleaning with the help of python using it's libraries in order to fill empty fields and to identify duplicate data types.

The procedures that we acquire in order to clean the datasets are shown in the slides below:

PEOPLE OF ALL AGE-GROUP EFFECTED BY COVID-19

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

	Sno	AgeGroup	TotalCases	Percentage
0	1	0-9	22	3.18%
1	2	10-19	27	3.90%
2	3	20-29	172	24.86%
3	4	30-39	146	21.10%
4	5	40-49	112	16.18%

TOTAL CASES EFFECTING THE AGE-GROUP

```
M↓
df.describe()
```

	Sno	TotalCases
count	10.00000	10.000000
mean	5.50000	69.200000
std	3.02765	59.241127
min	1.00000	9.000000
25%	3.25000	23.250000
50%	5.50000	52.500000
75%	7.75000	106.250000
max	10.00000	172.000000

```
M↓
df.dtypes
```

```
Sno             int64
AgeGroup       object
TotalCases      int64
Percentage     object
dtype: object
```

```
M↓
df.replace('?',np.nan, inplace=True)
M↓
df.isnull().sum()
```

```
Sno            0
AgeGroup       0
TotalCases     0
Percentage     0
dtype: int64
```

DEATHS OF PERSON DIAGNOSED WITH HIV-AIDS

```
M↓
df = pd.read_csv(r'E:\Training Second Year\deaths-of-persons-diagnosed-with-hiv-aids.csv')
df.head()
```

	Year	Category	Group	Count
0	2011	Age at Death	0-11	0
1	2011	Age at Death	12-14	0
2	2011	Age at Death	15-17	0
3	2011	Age at Death	18-19	0
4	2011	Age at Death	20-24	19

AVERAGE DEATH DUE TO HIV-AIDS

```
df.describe()
```

	Year
count	287.000000
mean	2013.996516
std	2.006107
min	2011.000000
25%	2012.000000
50%	2014.000000
75%	2016.000000

```
M↓
df.isnull().sum()
```

```
Year        0
Category    0
Group       0
Count       0
dtype: int64
```

CONCLUSION

- In our current project phase, we have collected datasets from various sources like CERN, Kaggle, UNICEF, Data World etc. The datasets that we have collected are in the form of xlsx, csv so the datasets we are having till today are of COVID, Dengue, Influenza, Pneumonia, Ebola, Hanta virus, Zika virus, HIV Aids.
- After collecting the datasets, we have performed data cleaning on these datasets. Data cleaning help us to remove improper data type which will make sure that our data is within the correct range not only that we are also able to do prediction with 80% accuracy.
- By performing data cleaning we have removed the inconsistencies which are caused due to user entry errors, or by corruption in transmission or storage so our datasets can now be used to generate graphs, charts, lists and other types of visualization.
- Visualizations will help us to understand various aspects like which virus is more effective during which years, what are the chances of its returning in the upcoming seasons, what are the symptoms of a particular virus in humans and many more.
- In this phase the data collection and data cleaning of the data has been completed now in phase-2 we have a plan to implement various machine learning models on the collected data and also, decided to use various tools like IBM Cognos, R, Orange etc in order to generate different types of visualizations.

FUTURE SCOPE

- In the future the study may be improved by including following points:
- At each visualization we will focus on the relationship between human and virus. More specifically we will study the effect on human based on gender, age, time, intensity, place. We will also look for symptoms reflected and their commonness. We will neglect the analysis above and below a specified age. We will relate the age with disease categorized by place.
- We will be studying more virus datasets in order to extend the usage of our model and to generate visualizations with more details.
- After data cleaning we have decided to do data processing in which we will filter out the missing values for this we will use some packages and libraries from python like:Numpy,Scipy,Scikit-learn,Theano,TensorFolw,Keras,PyTorch,Pandas,Seaborn,Matplotlib etc. We will be adding more libraries (if required) in the future for further advancement of our algorithms.
- When we are done with our data processing, we will understand the data flow with the help of visualizations and then we will predict the data according to certain constraints.
- After performing all these operations on the dataset's deployment phase will come in which we will be creating either the web or an android application(whatever suits best for our model) and then deploy our model on that platform which will help the user to communicate with our data easily.

ROAD MAP OF PHASE-2

The data will be analyzed on various platform like IBM Cognos, Python, R studio, Orange.

Mainly the implementation will be analyzing the data.

The flow-diagram showing the process phase 2.

FLOW CHART

FILTER DATA SETS/API FROM PHASE-I

IMPLEMENTATION OF NUMPY, PANDAS & MATPLOTLIB

USING OF IBM COGNOS AND ORANGE FOR ANALYSIS

PHASE-3

ABOUT IBM COGNOS

Analytics tool

IBM COGNOS CONT....

- We will be using the IBM congos for the analyzing the data and give the meaningful outcome. In IBM congnos 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.
- **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.

- **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.

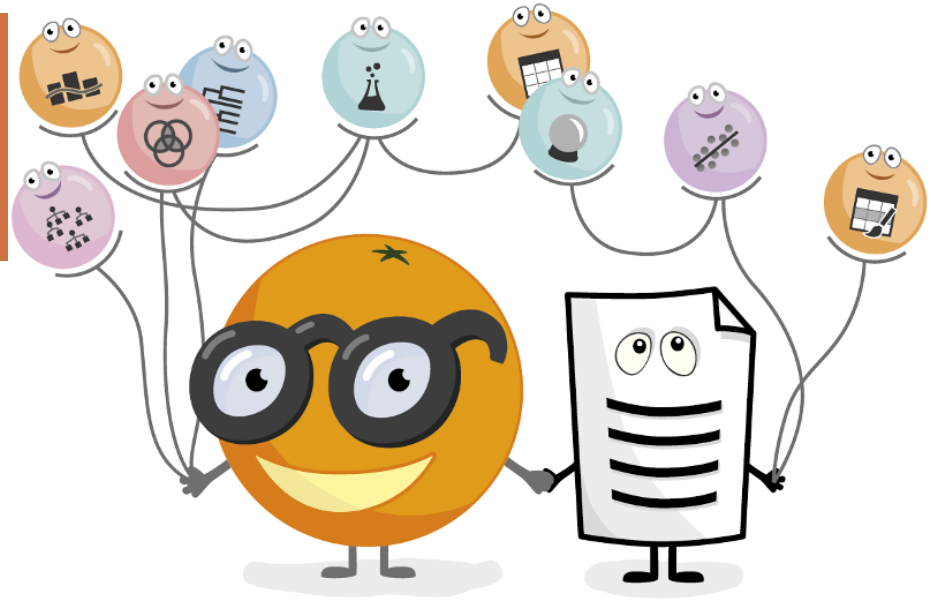
PYTHON LIBRARAIES USED FOR ANALYZING AND VISUALIZATION OF DATASETS

- NumPy: NumPy full form is Numerical Python is a perfect tool for performing basic and scientific computing and advanced array operations. It offers many useful features for performing many operations on n-arrays and matrices in Python.

- Pandas: Pandas is a library will helped us to work with labelled and relational data and query counterintuitive. It's a must-have for data manipulation, visualization, and wrangling.

- Matplotlib: Matplotlib is a library used in python programming language and it is also used for the numerical, mathematical extension of the NumPy

ORANGE



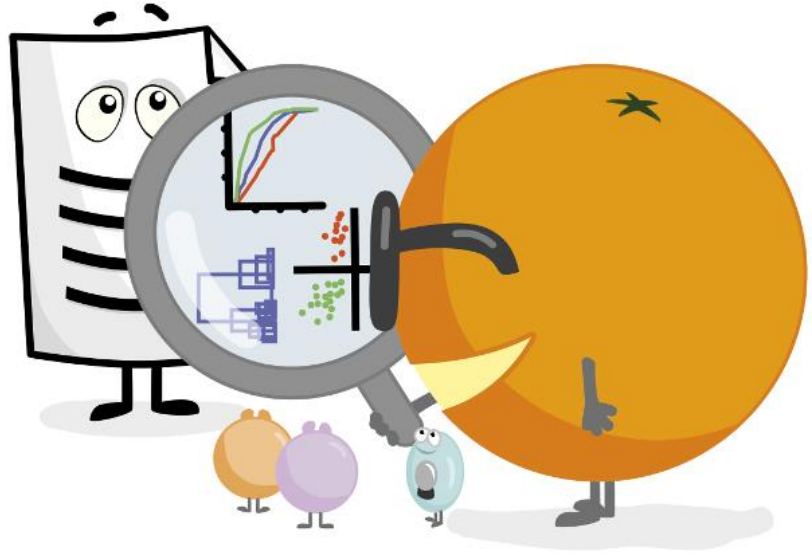
- Open source machine learning and data visualization toolkit.
- Build data analysis workflows visually, with a large, diverse toolbox.

Interactive Data Visualization

- Perform simple data analysis with clever data visualization.
- Explore statistical distributions, box plots and scatter plots, or dive deeper with decision trees, hierarchical clustering, heatmaps, MDS and linear projections.

Visual Programming

- Interactive data exploration for rapid qualitative analysis with clean visualizations
- Place widgets on the canvas, connect them, load your datasets and harvest the insight!



WORKING ON DEMO-DISTRIBUTIONS

ON HIV-API

DEMO DISTRIBUTIONS

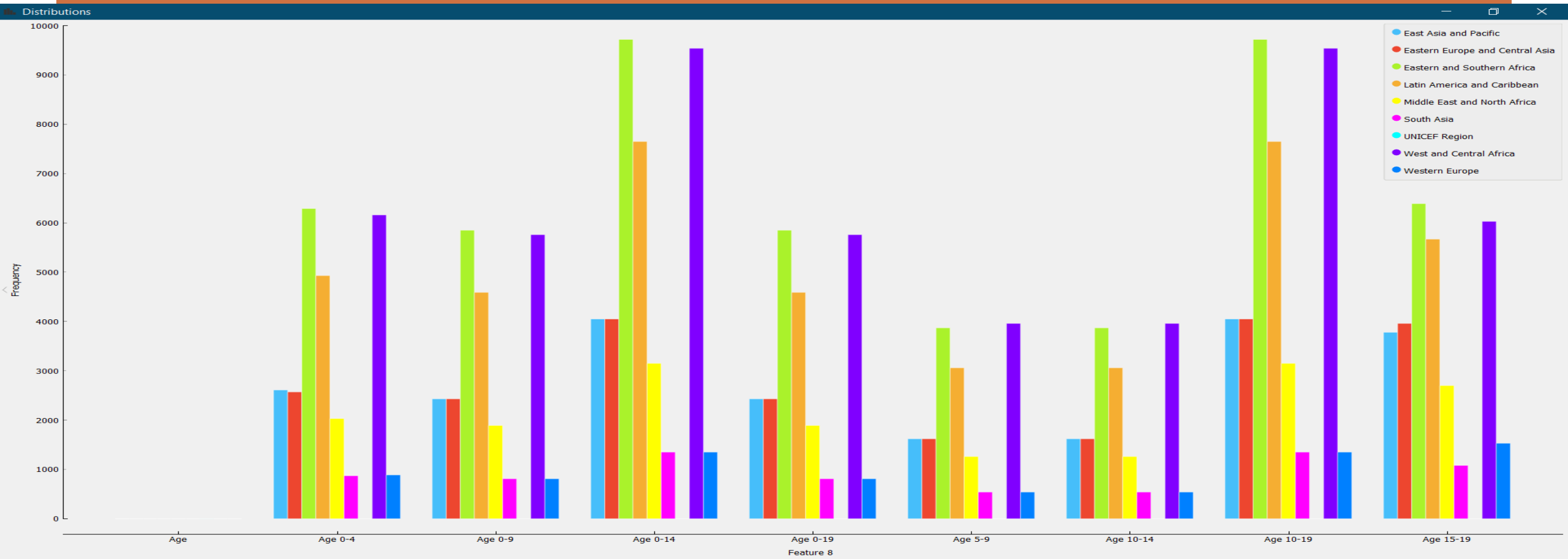


Diagram Shows the Distribution of Age group suffering from HIV Infections w.r.t. Region.

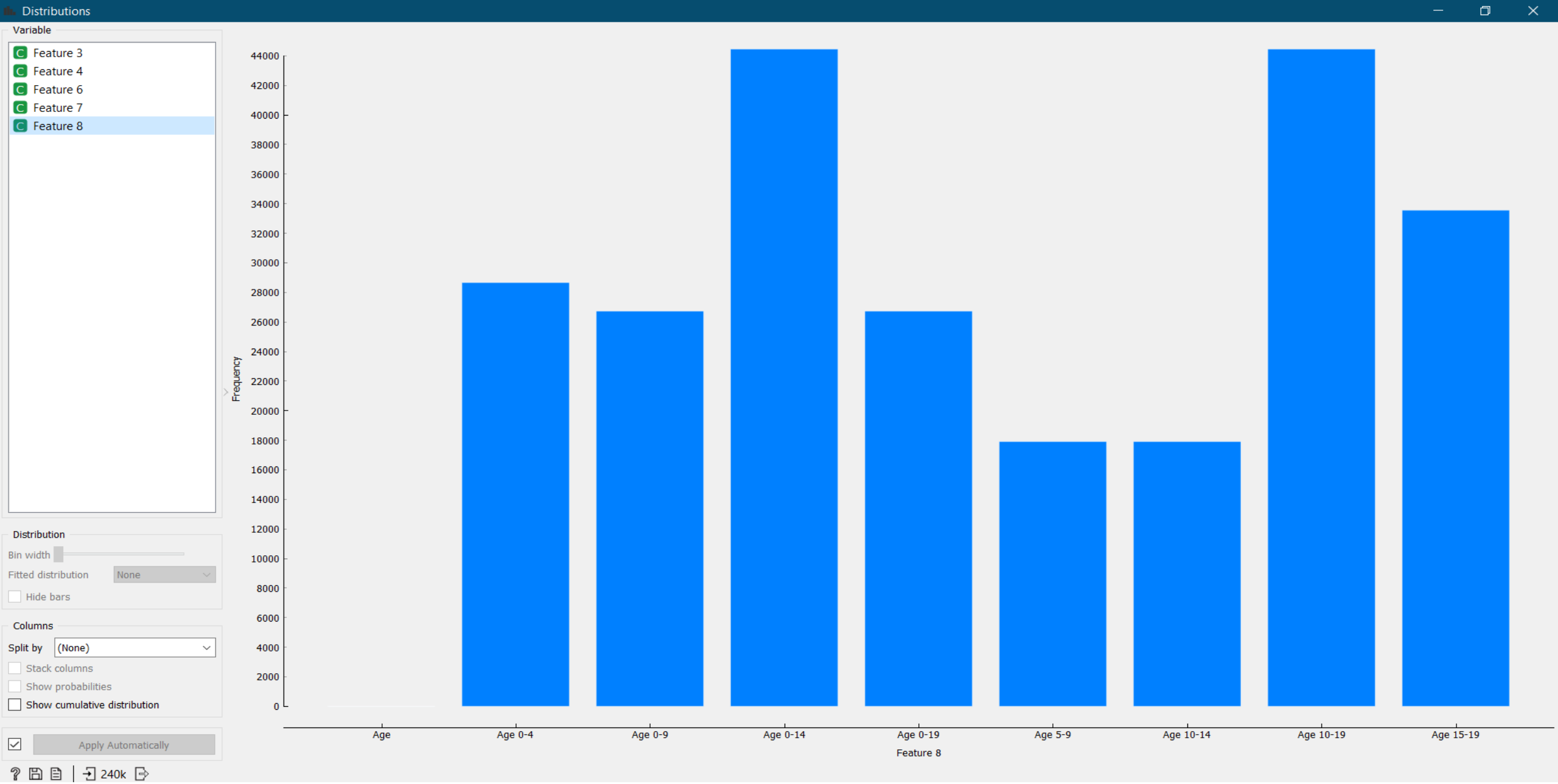
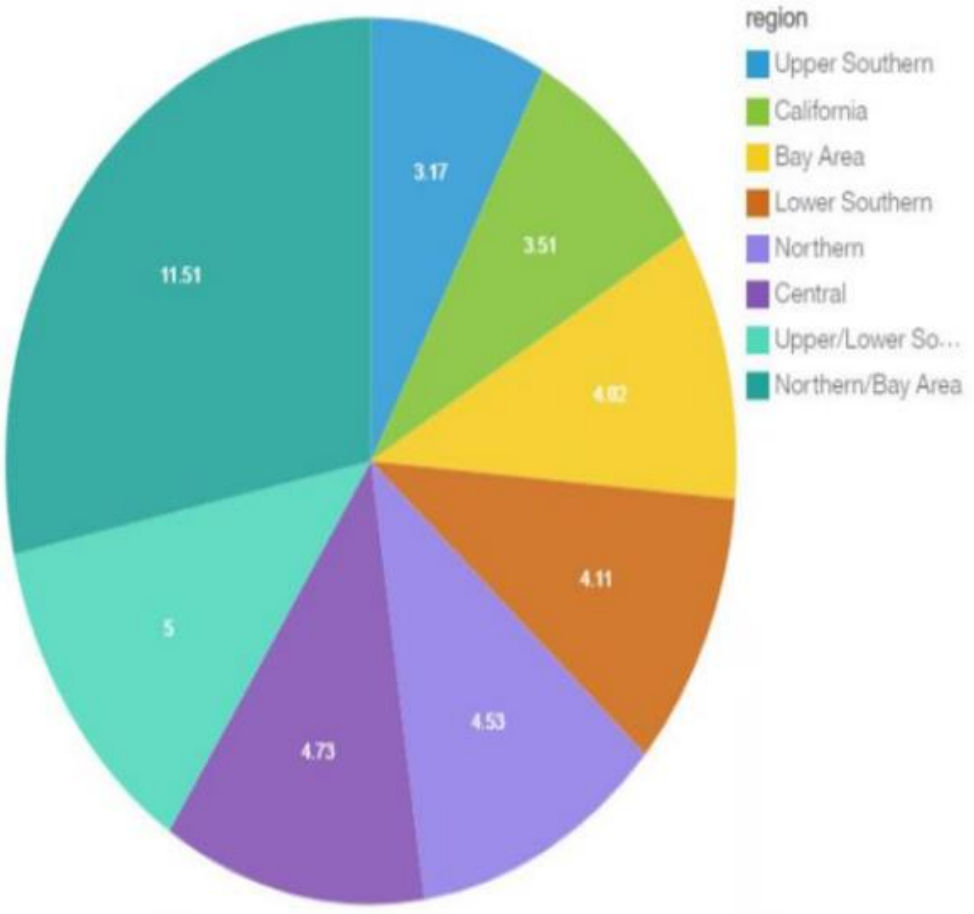


Diagram represents the analysis of Frequency of people suffering from HIV Infections under certain Age groups, as shows this data is the representation of 240k values

VISUALIZATIONS

- Visualization is a key technology for understanding large datasets.
- It is useful throughout the analysis process, for exploratory descriptive analysis, to aid in model building; and for presenting the analysis results.
- Our approach to visualizing abstract, non-geometric data involves domain-specific representations, multiple linked views, color, and a highly-interactive user interface using filtering and focusing to reduce visual clutter.
- By using visual elements like charts, graphs and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

Research on influenza virus



The pie chart visualization is showing how many specimens are tested in a particular region based on the virus and year that we have selected from the list

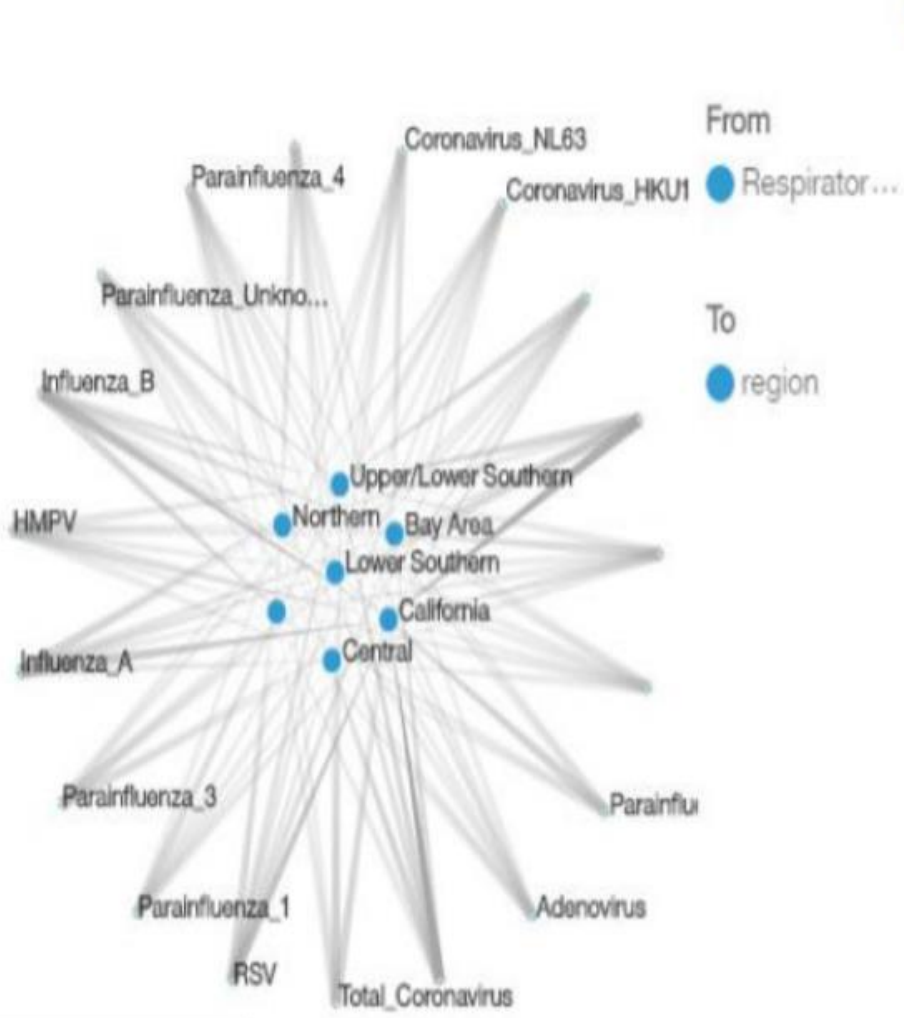
Adenovirus

- 2013-2014
- 2014-2015
- 2015-2016
- 2016-2017
- 2017-2018

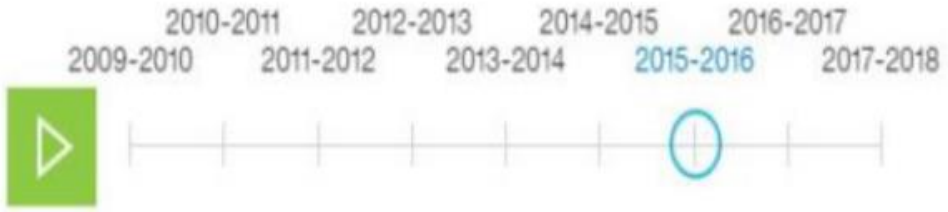
Coronavirus_229E

- 2015-2016
- 2016-2017
- 2017-2018

Research on Influenza Virus

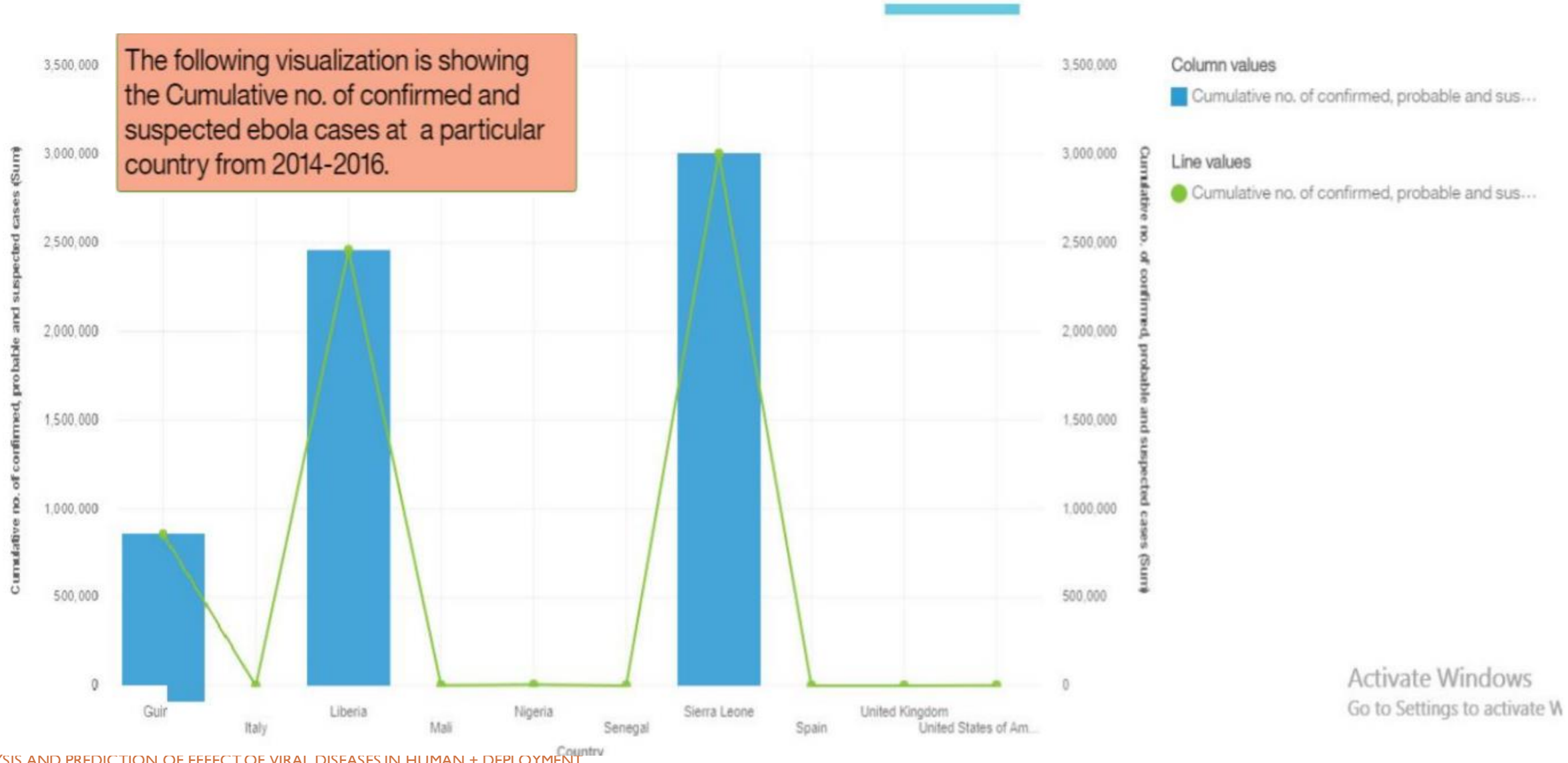


The following network visualization is showing which influenza virus is more effective in which region and the data is changing based on the year duration provided by the data player.



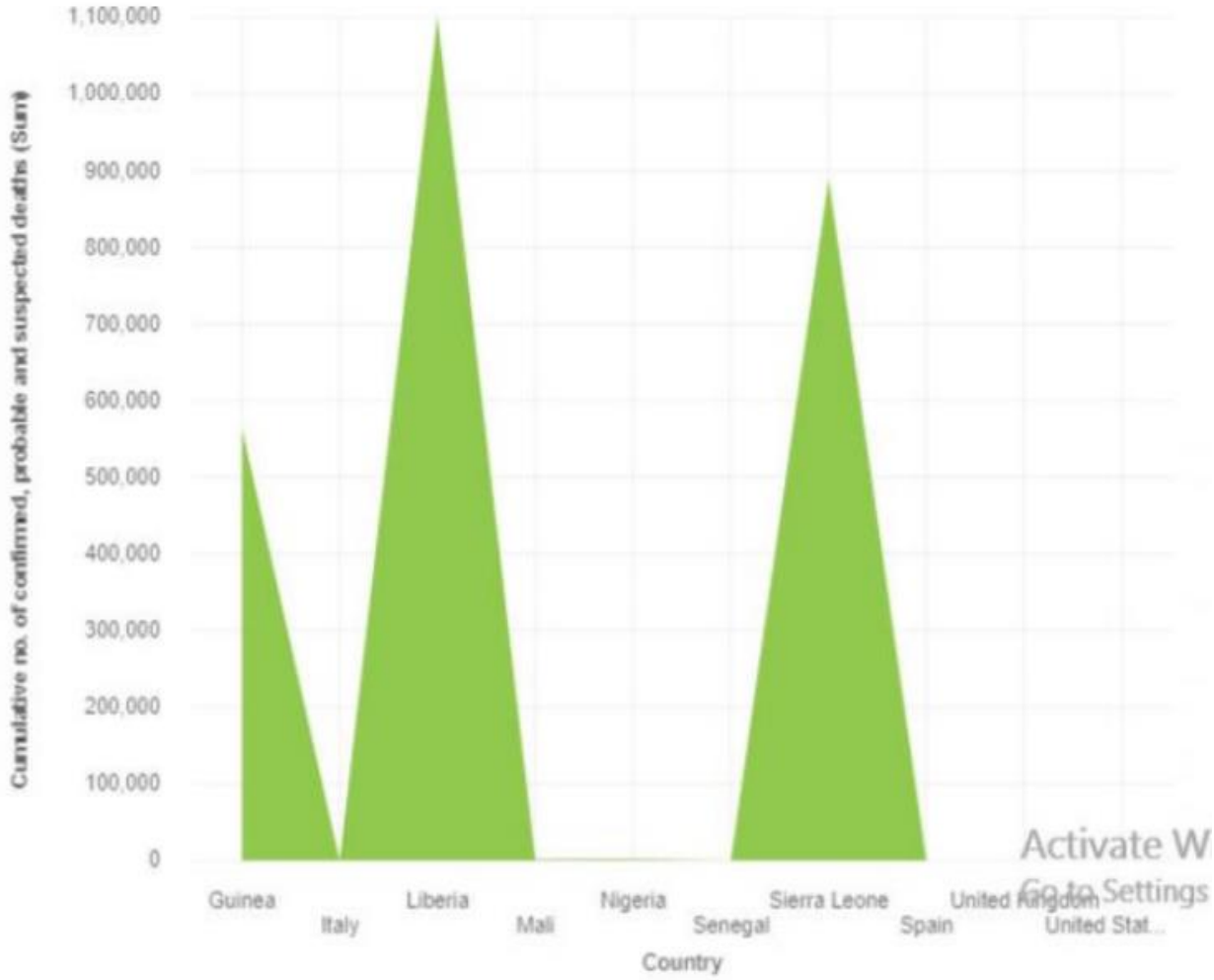
Research on Ebola Virus

The following visualization is showing the Cumulative no. of confirmed and suspected ebola cases at a particular country from 2014-2016.



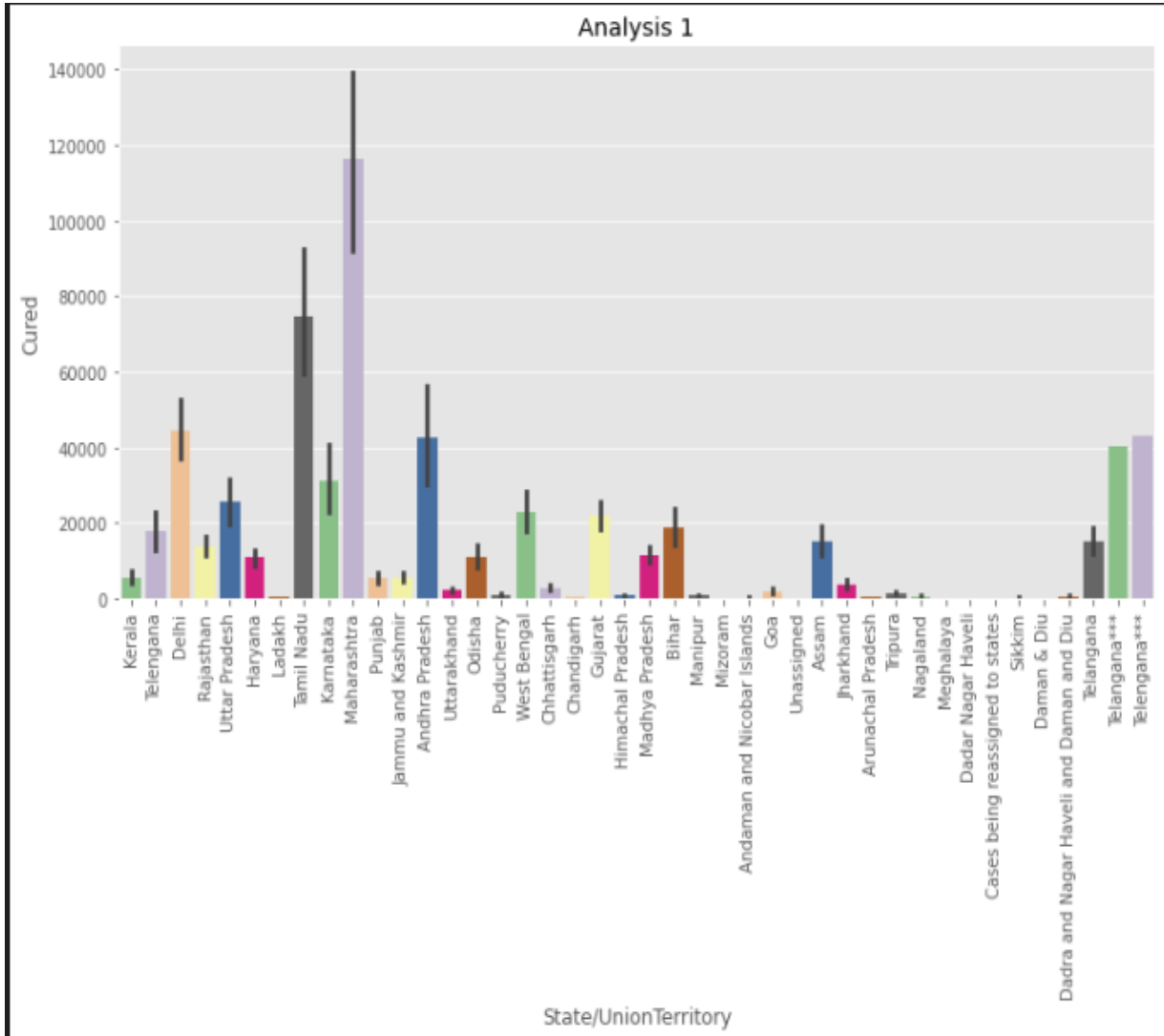
Research on Ebola Virus

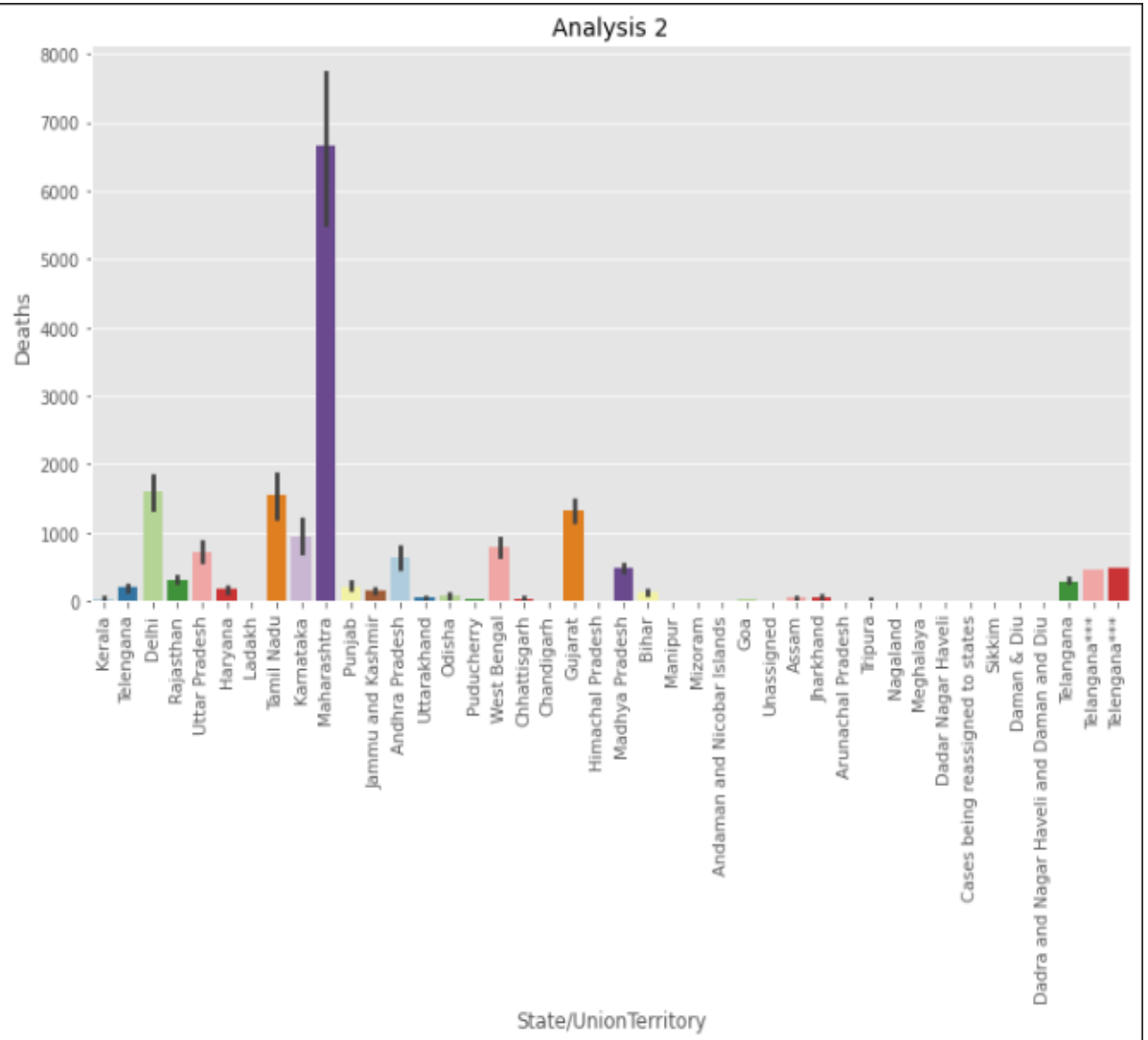
The following area visualization is showing probable deaths in countries during 2014-2016.



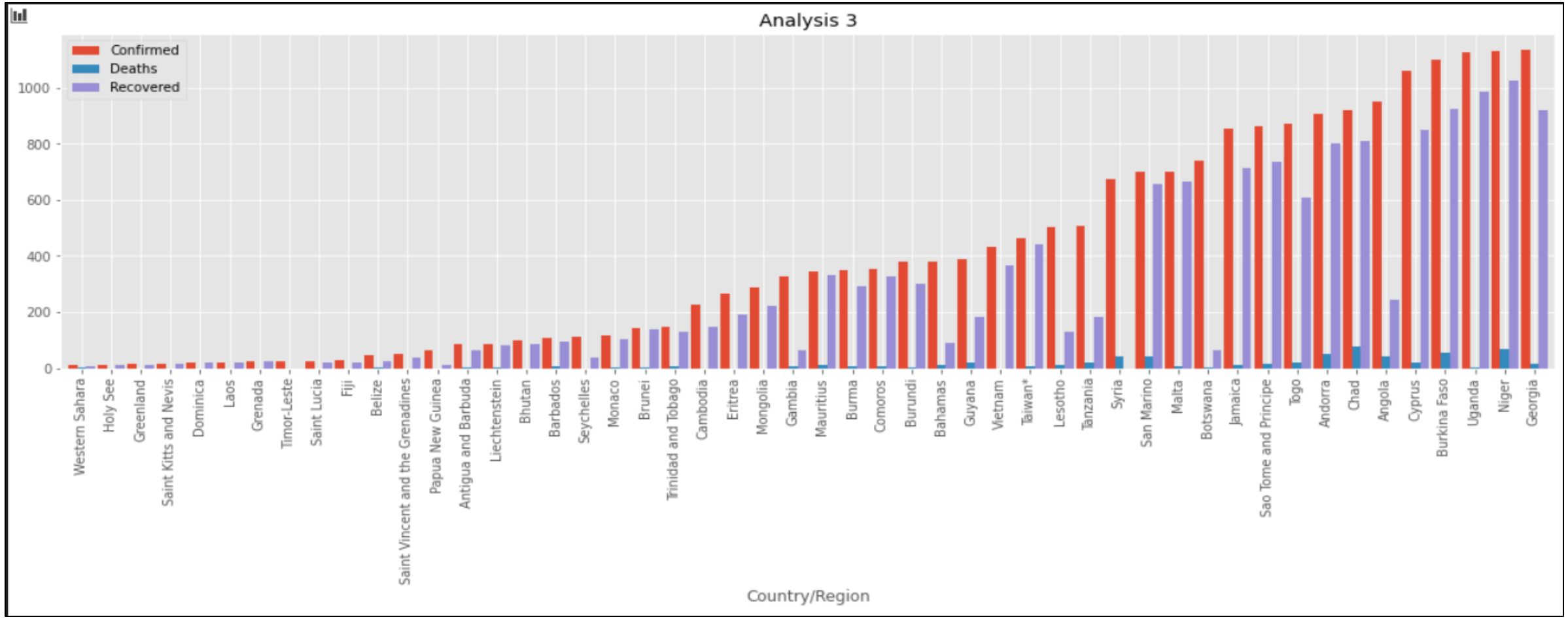
STATISTICAL RESEARCH AND ANALYSIS ON COVID-19 CASES

BASED ON DATA SOURCED FROM *KAGGLE*

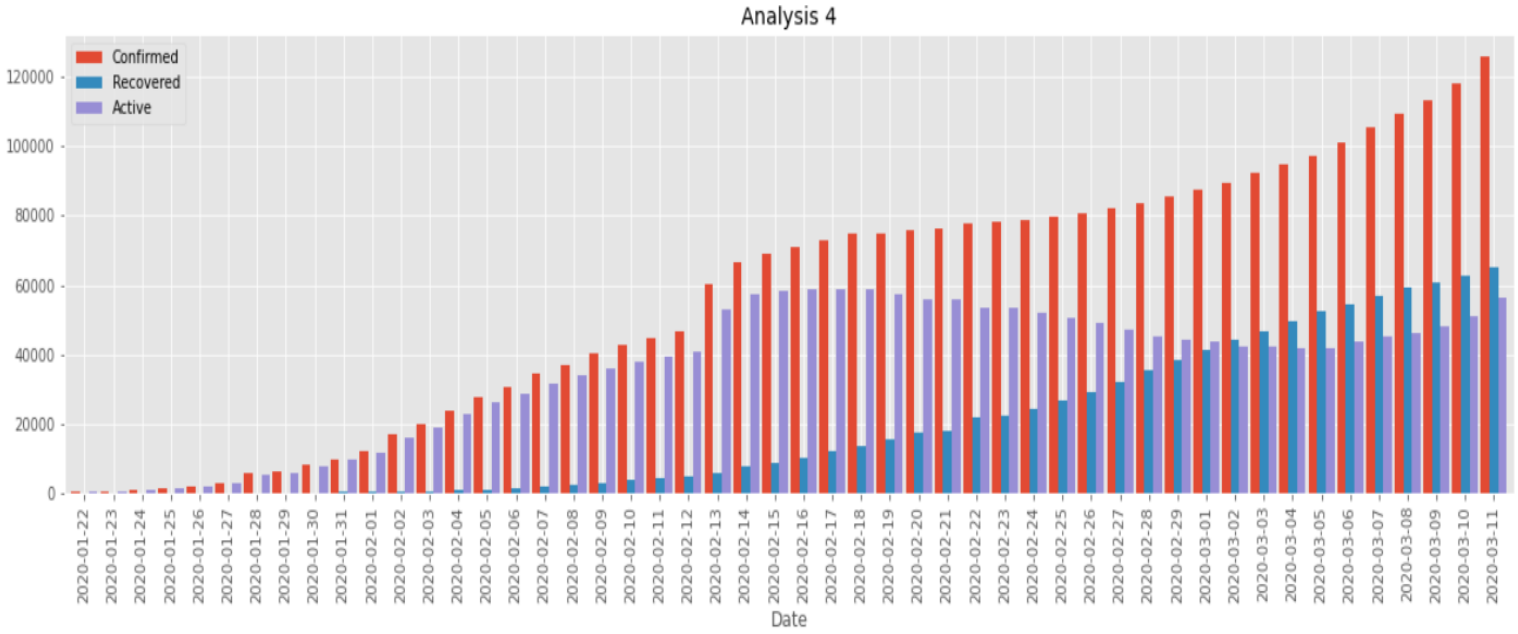


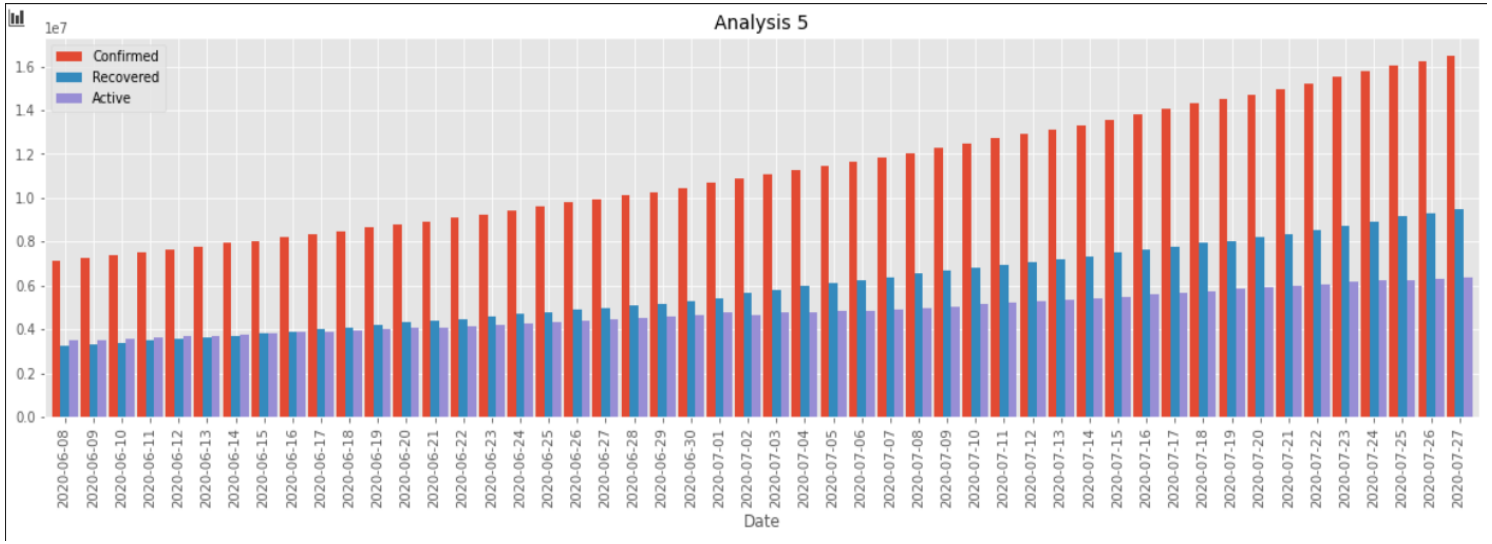


This analysis shows the correlation between the recovered/cured cases and the number of deaths in different states of India. As the number of cured cases increases, so does the number of deaths.



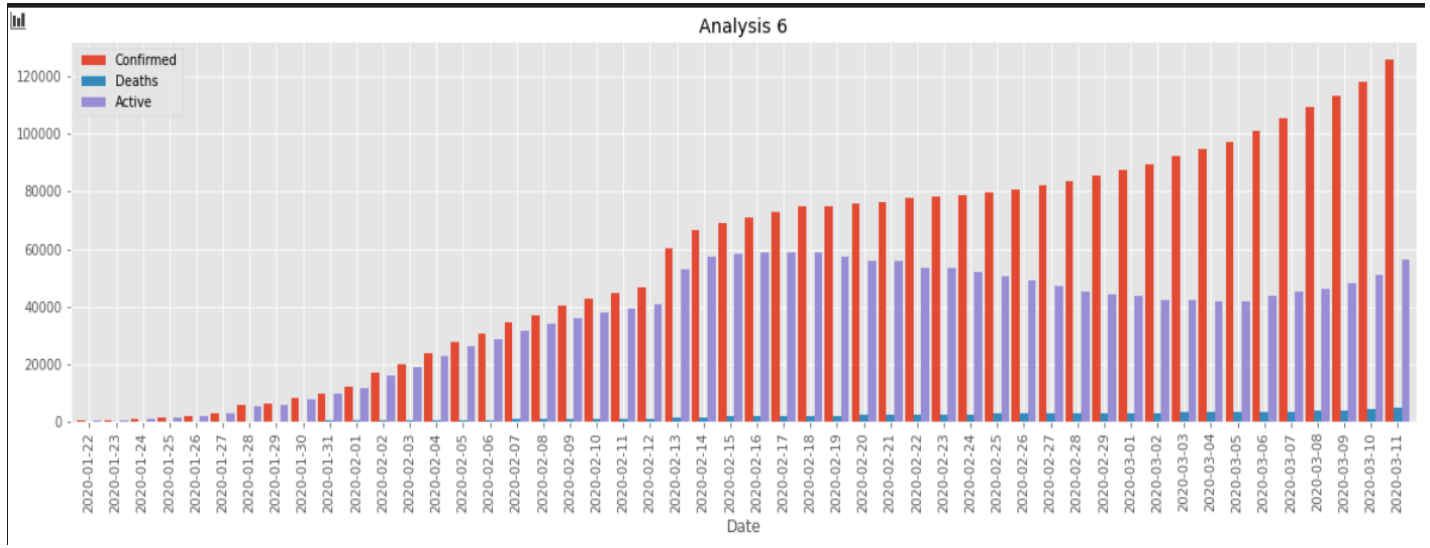
This graph here shows the division of confirmed cases and the recoveries and deaths in different countries.

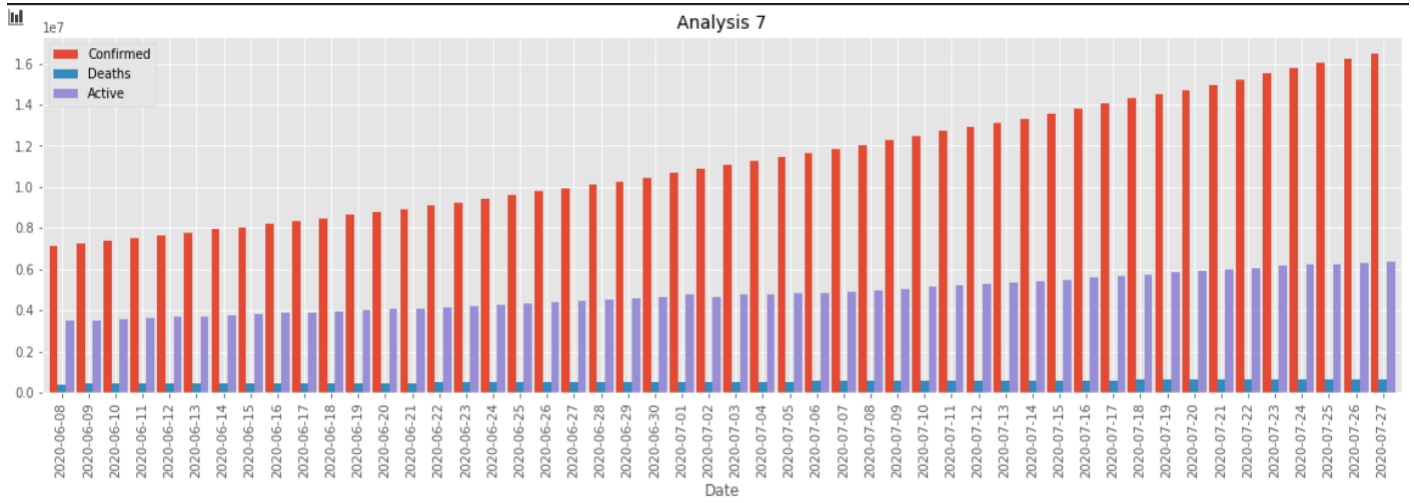




This graphical representation of the number of confirmed, recovered, and active cases in the beginning of the considered period vs the end shows the changes in the proportion of the recovered and active cases.

Initially, the active cases greatly outnumber the recovered cases but by the end the number of recovered cases is more than the active cases.



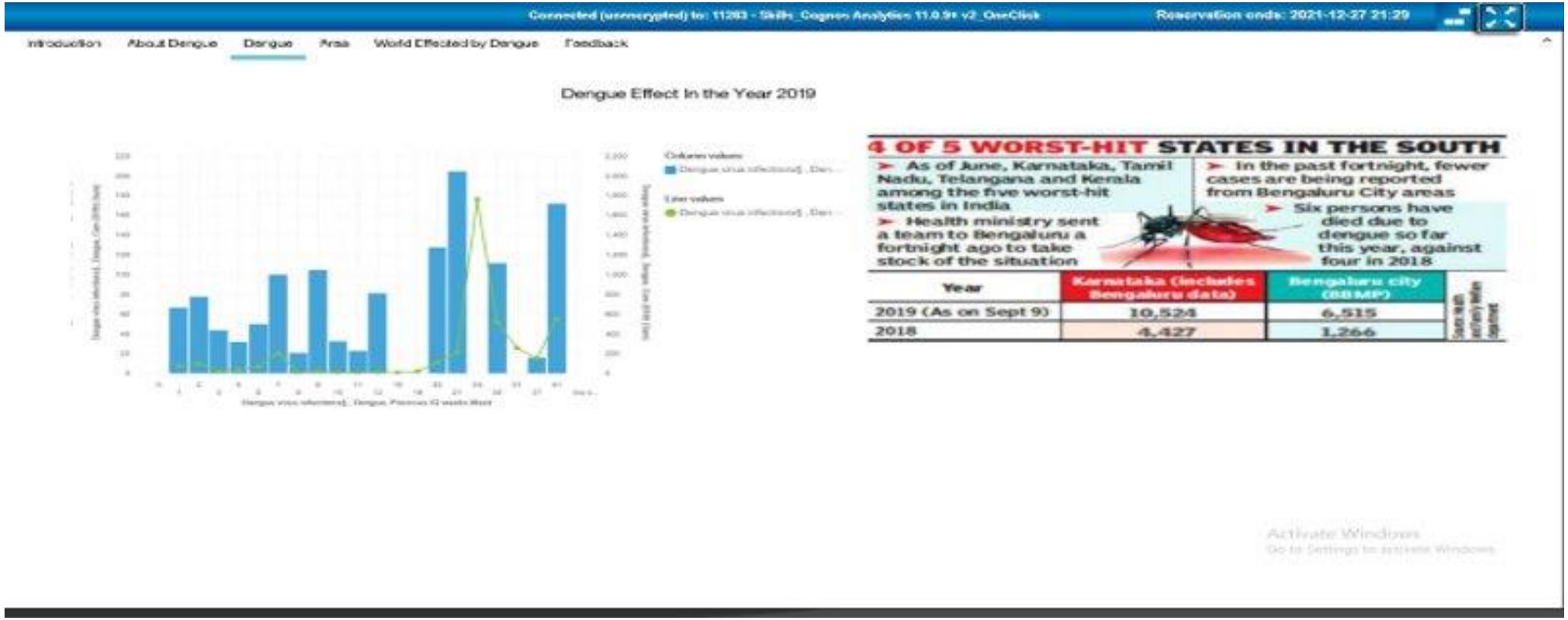


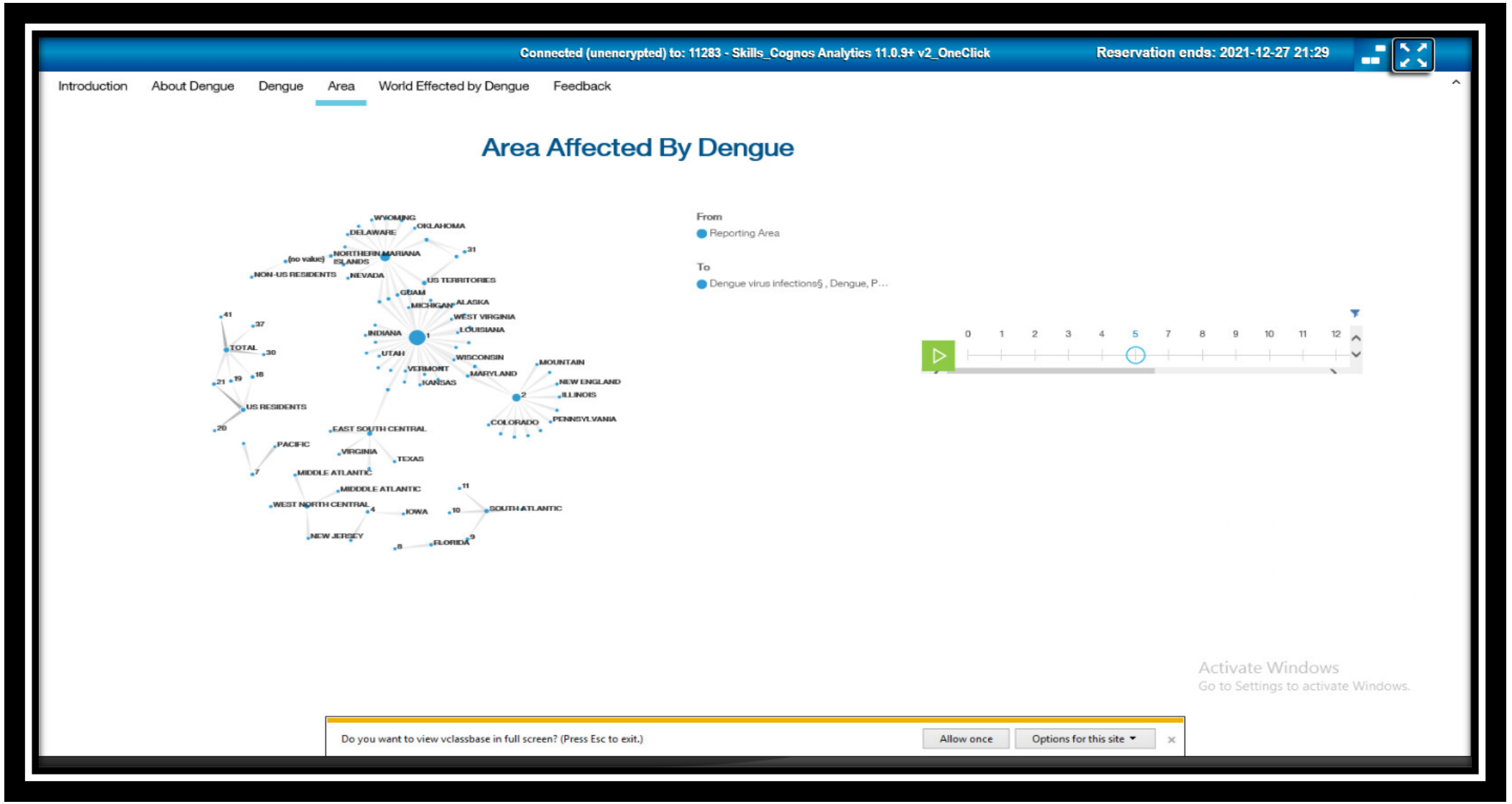
This graphical representation of the number of confirmed cases, deaths and active cases in the beginning of the considered period vs the end shows the changes in the proportion of the deaths and active cases.

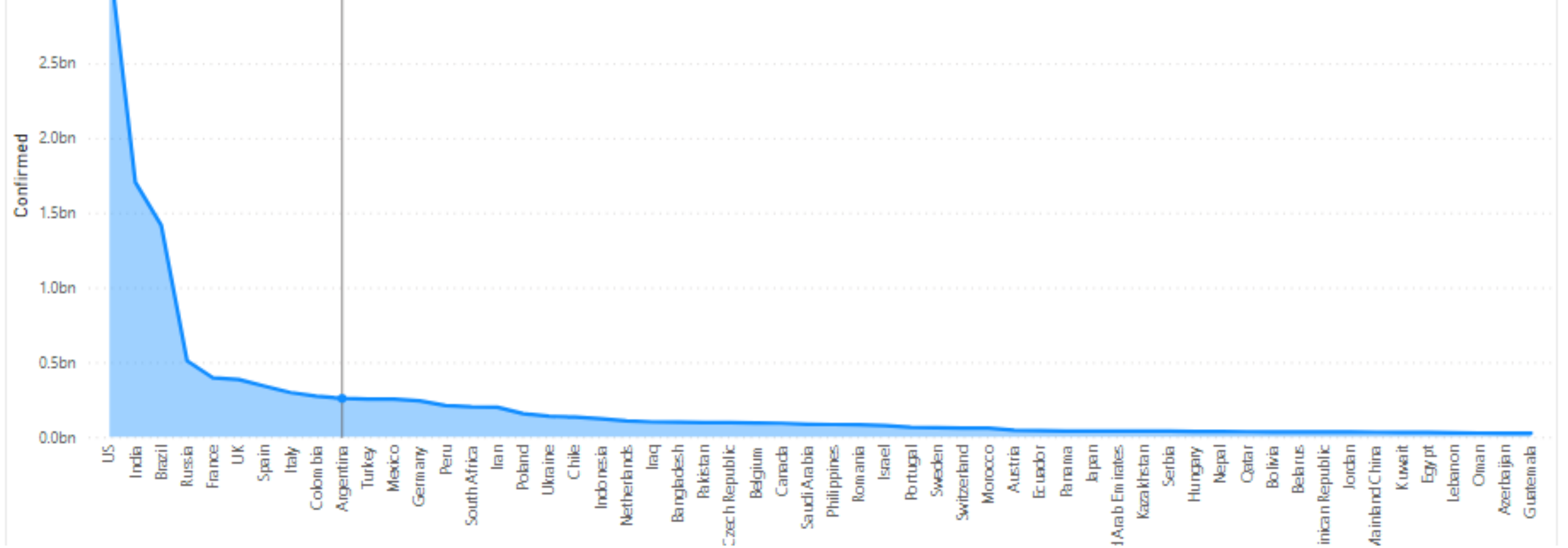
The proportion of deaths remains very small from beginning till end.

The proportion of active cases is very high initially, almost equalling the confirmed cases, but over time this proportion reduces as rate of recovery increases and by the end less than 50% of the confirmed cases are active.

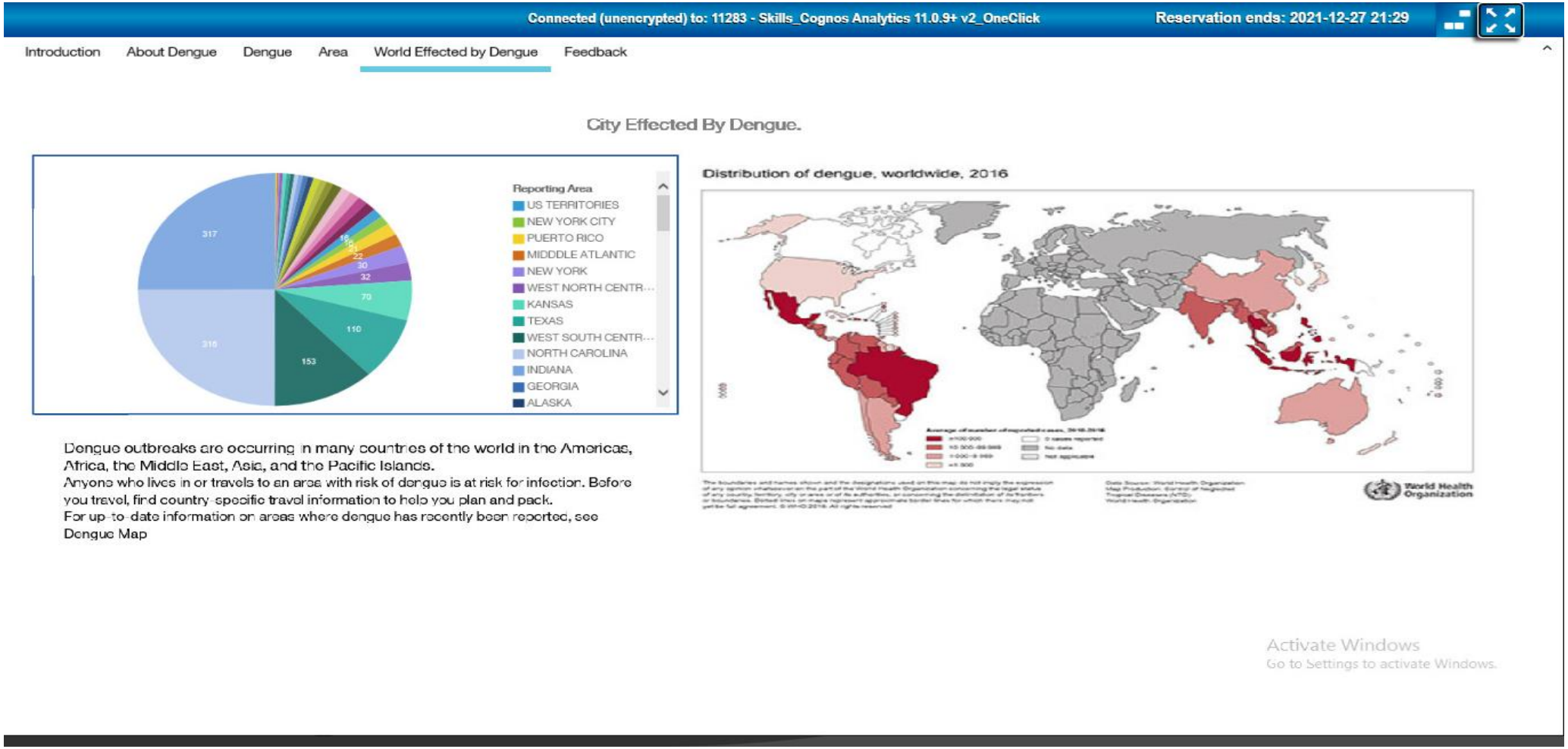
ANALYSIS AND VISUALIZATION OF DENGUE



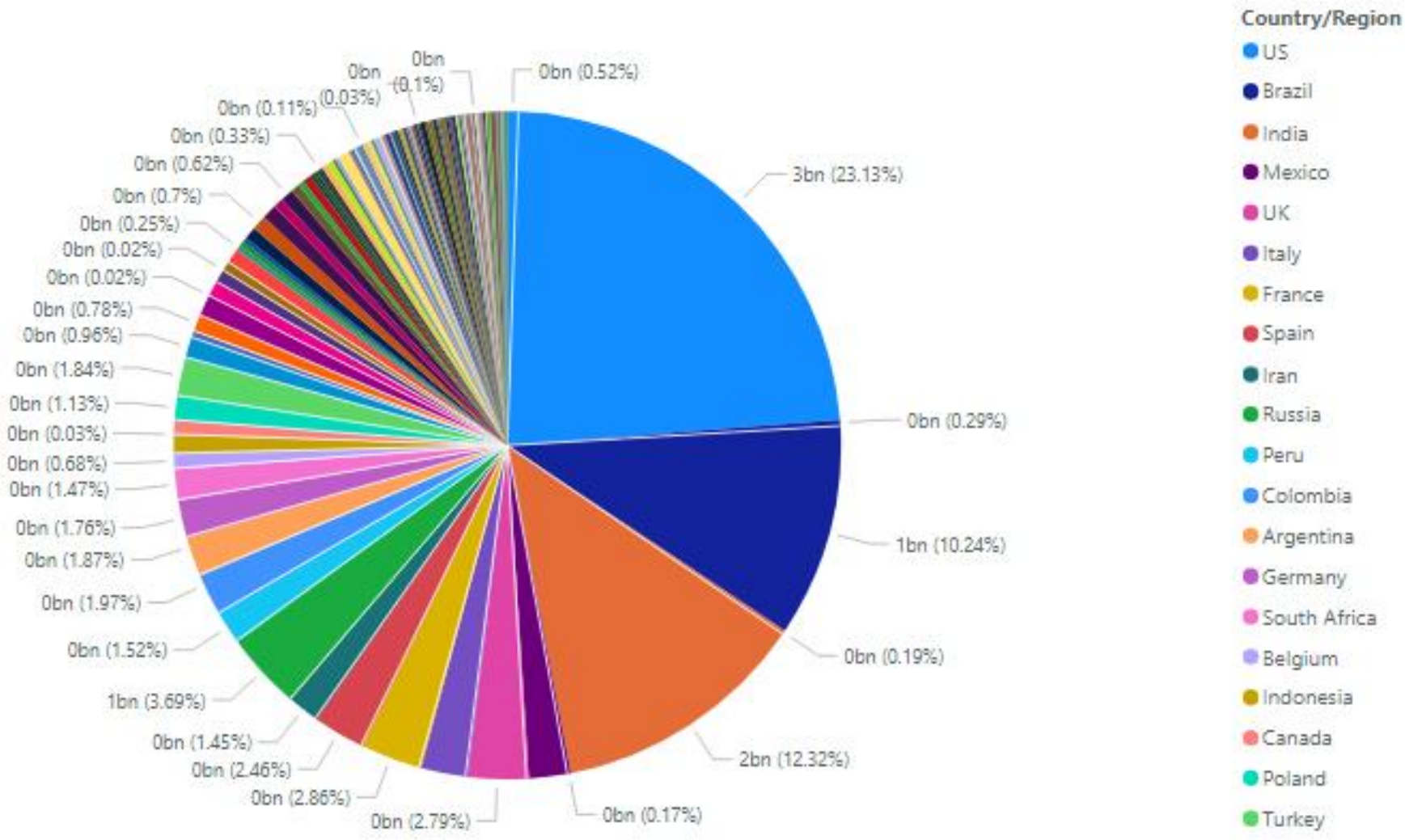

Connected (unencrypted) to: 11283 - Skills_Cognos Analytics 11.0.9+ v2_OneClick
Reservation ends: 2021-12-27 21:29
Introduction
About Dengue
Dengue
Area
World Effected by Dengue
Feedback
Area Affected By Dengue
From
Reporting Area
To
Dengue virus infections§ , Dengue, P...
0 1 2 3 4 5 7 8 9 10 11 12
WYOMING
OKLAHOMA
DELAWARE
NORTHERN MARIANA ISLANDS
(no value)
NON-US RESIDENTS
NEVADA
US TERRITORIES
GUAM
MICHIGAN
ALASKA
WEST VIRGINIA
LOUISIANA
INDIANA
TOTAL
UTAH
WISCONSIN
VERMONT
MARYLAND
KANSAS
MOUNTAIN
NEW ENGLAND
ILLINOIS
US RESIDENTS
COLORADO
PENNSYLVANIA
EAST SOUTH CENTRAL
PACIFIC
VIRGINIA
TEXAS
MIDDLE ATLANTIC
MIDDDLE ATLANTIC
WEST NORTH CENTRAL
IOWA
SOUTH ATLANTIC
NEW JERSEY
FLORIDA
Activate Windows
Go to Settings to activate Windows.
Do you want to view vclassbase in full screen? (Press Esc to exit.)
Allow once
Options for this site



CONFIRMED CASES

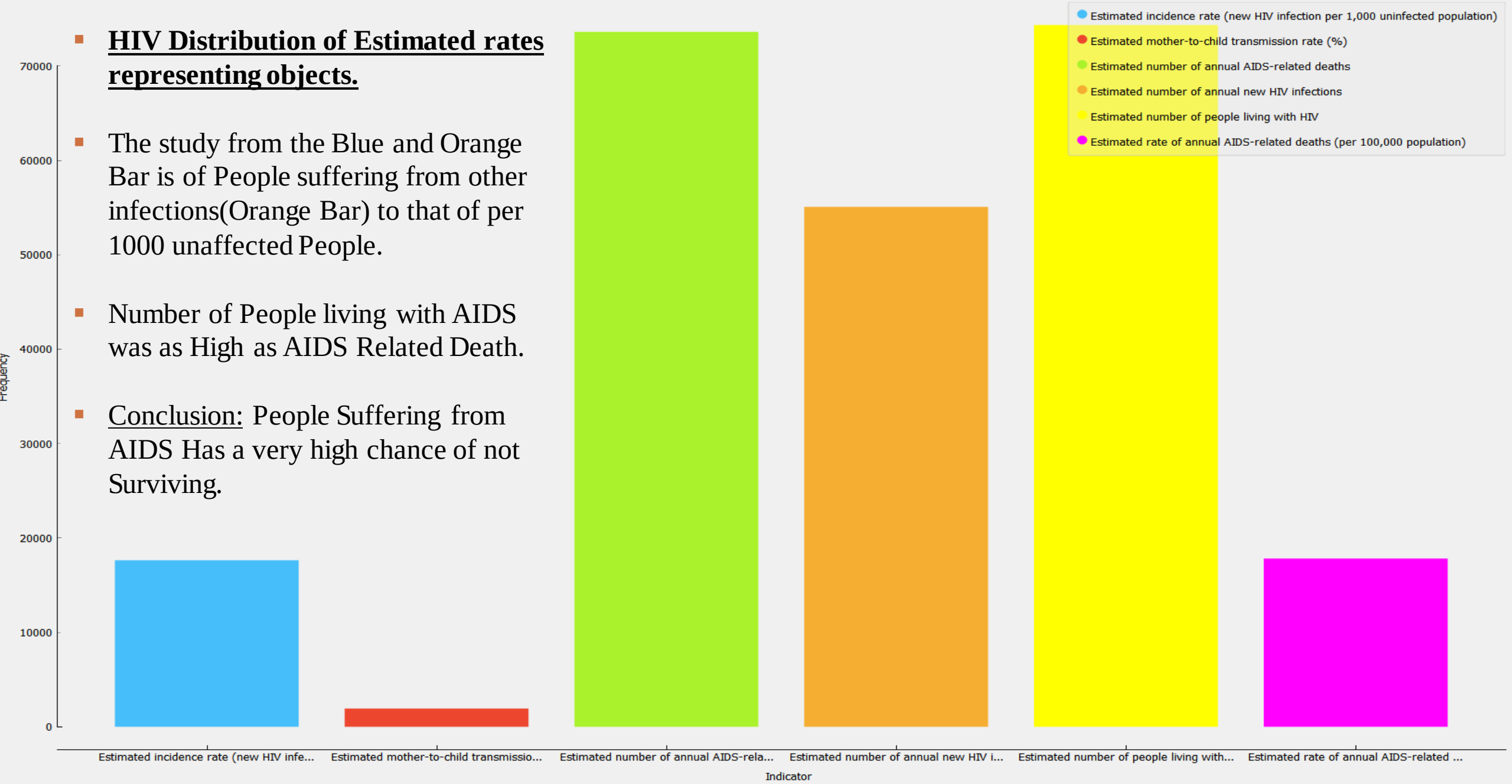

Connected (unencrypted) to: 11283 - Skills_Cognos Analytics 11.0.9+ v2_OneClick
Reservation ends: 2021-12-27 21:29
Introduction
About Dengue
Dengue
Area
World Effected by Dengue
Feedback
City Effected By Dengue.
Reporting Area
US TERRITORIES
NEW YORK CITY
PUERTO RICO
MIDDDLE ATLANTIC
NEW YORK
WEST NORTH CENTR...
KANSAS
TEXAS
WEST SOUTH CENTR...
NORTH CAROLINA
INDIANA
GEORGIA
ALASKA
317
153
110
70
32
30
22
Dengue outbreaks are occurring in many countries of the world in the Americas, Africa, the Middle East, Asia, and the Pacific Islands.
Anyone who lives in or travels to an area with risk of dengue is at risk for infection. Before you travel, find country-specific travel information to help you plan and pack.
For up-to-date information on areas where dengue has recently been reported, see Dengue Map
Distribution of dengue, worldwide, 2016
World Health Organization
Activate Windows
Go to Settings to activate Windows.

Death confirmed by country in pie

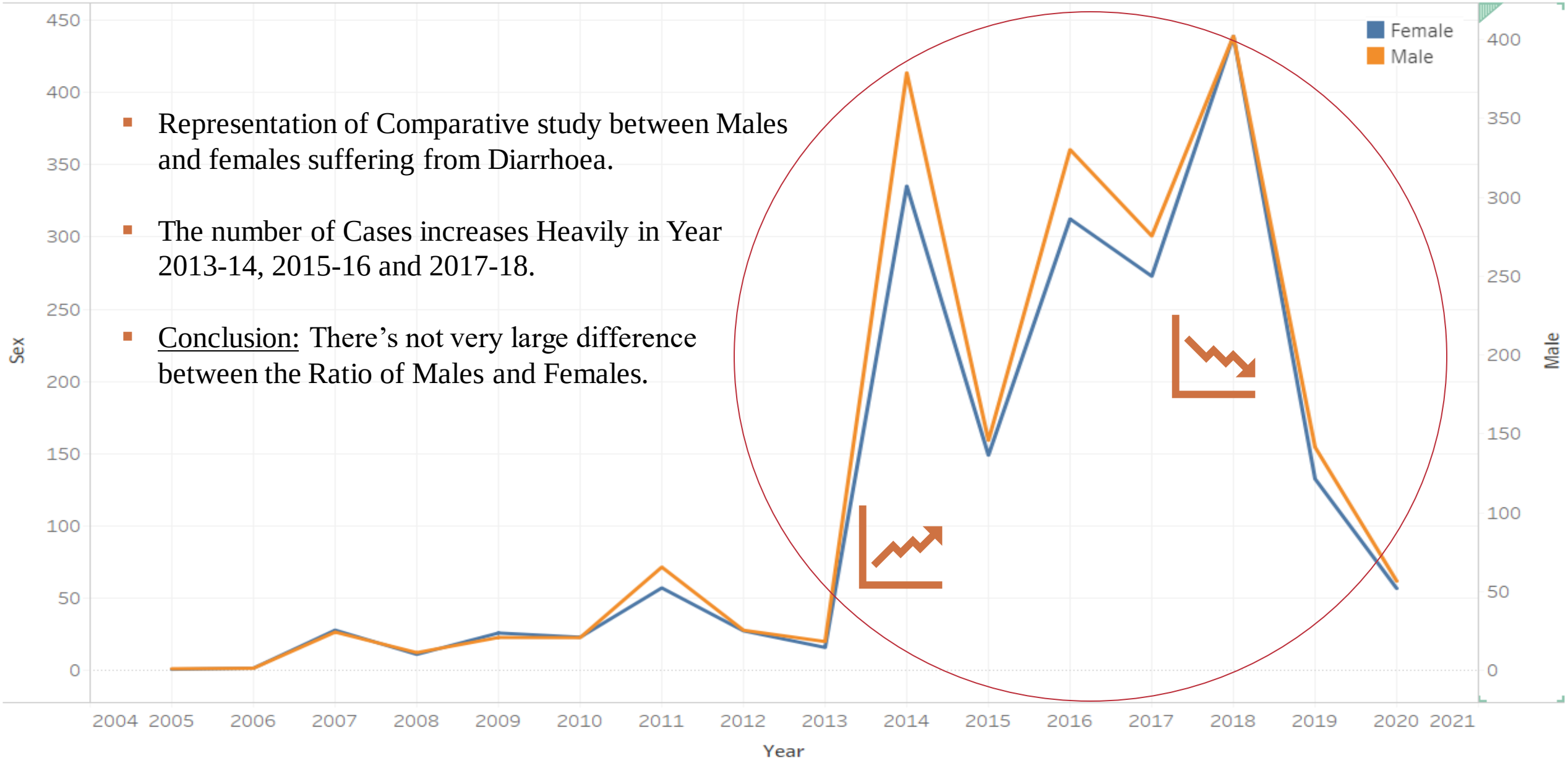


Deaths by Country/Region

- **HIV Distribution of Estimated rates representing objects.**
- The study from the Blue and Orange Bar is of People suffering from other infections(Orange Bar) to that of per 1000 unaffected People.
- Number of People living with AIDS was as High as AIDS Related Death.
- Conclusion: People Suffering from AIDS Has a very high chance of not Surviving.

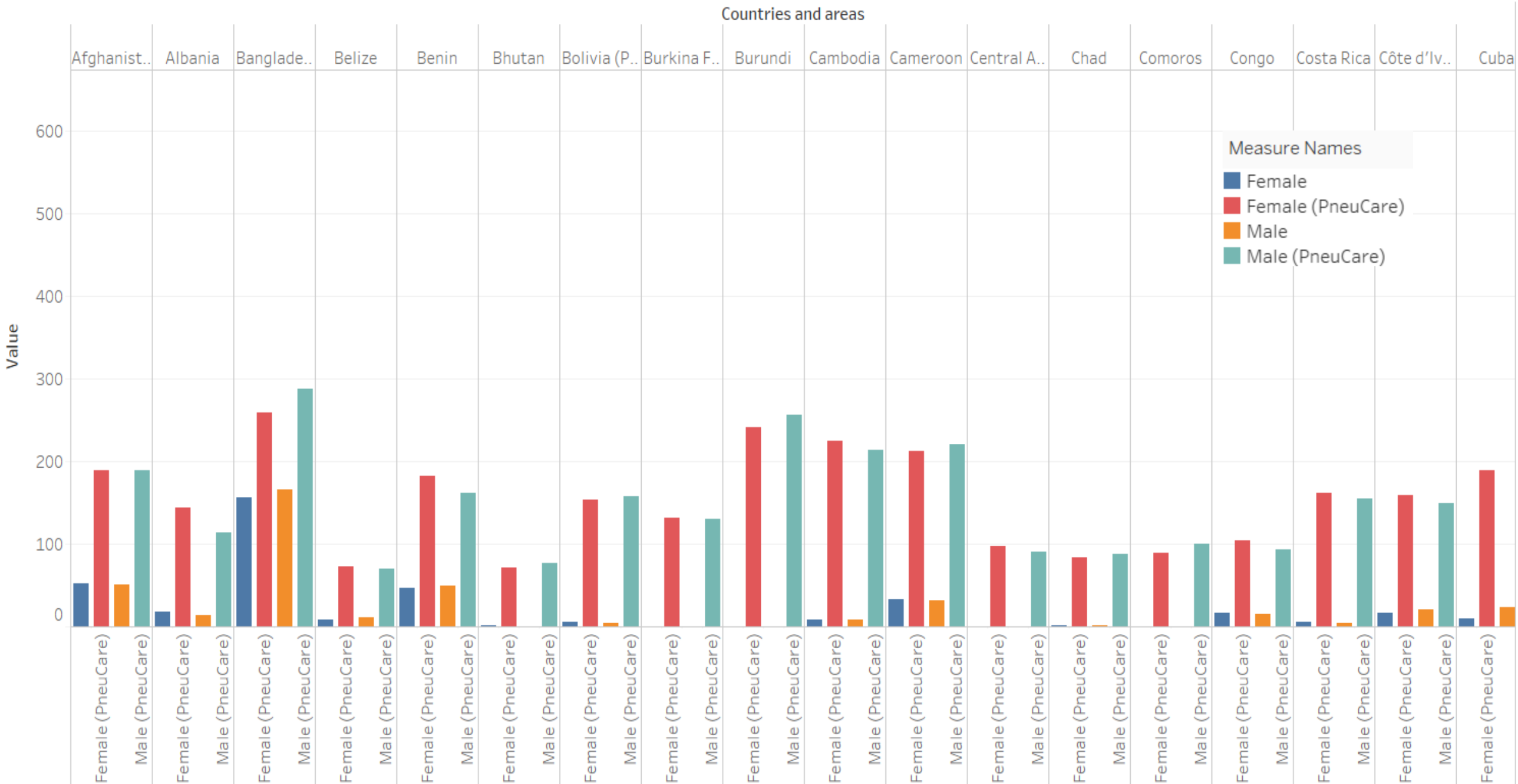


Year wise representation of Sex suffered from Diarrhea

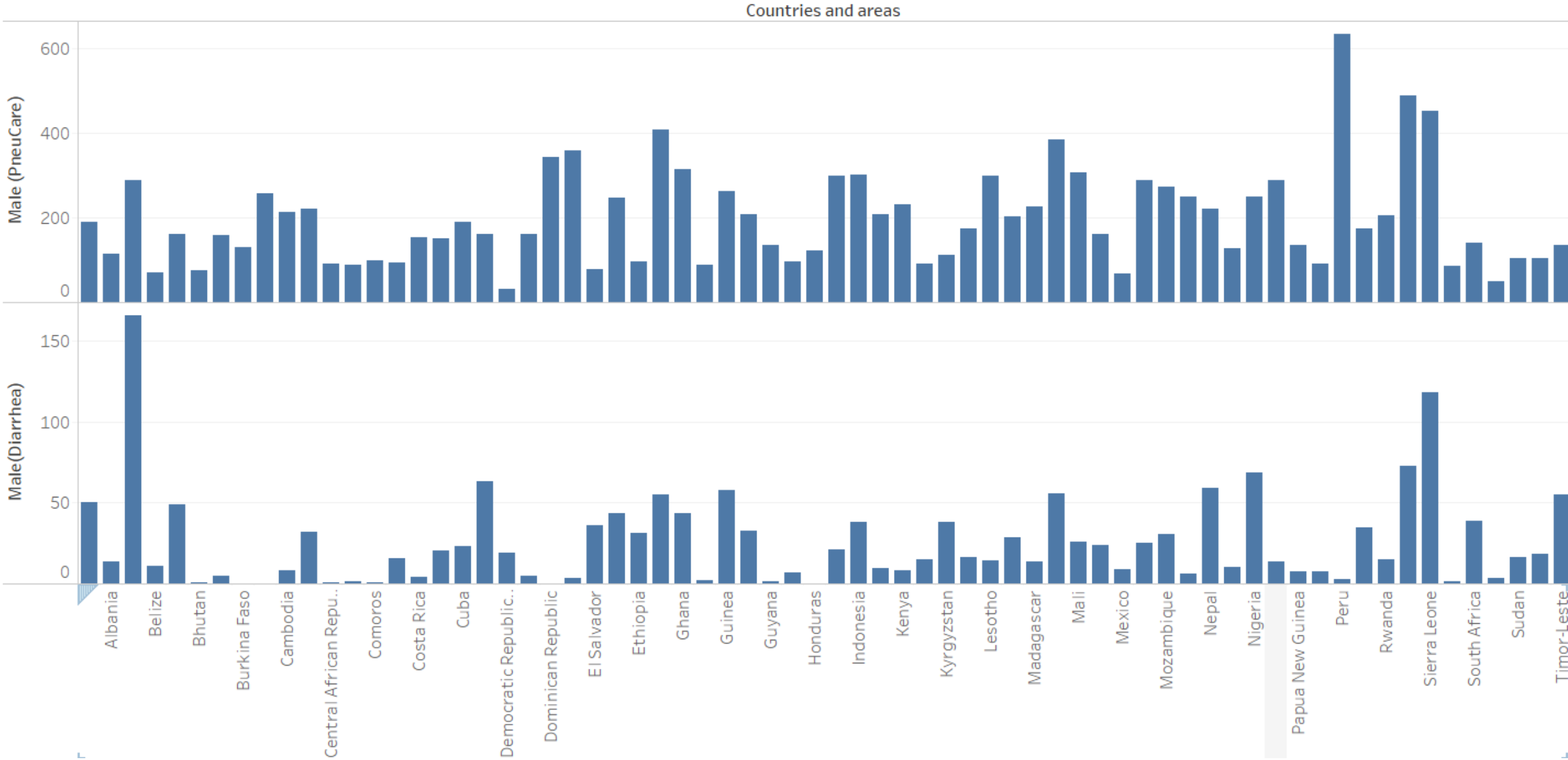


- Representation of Comparative study between Males and females suffering from Diarrhoea.
- The number of Cases increases Heavily in Year 2013-14, 2015-16 and 2017-18.
- <u>Conclusion:</u> There's not very large difference between the Ratio of Males and Females.

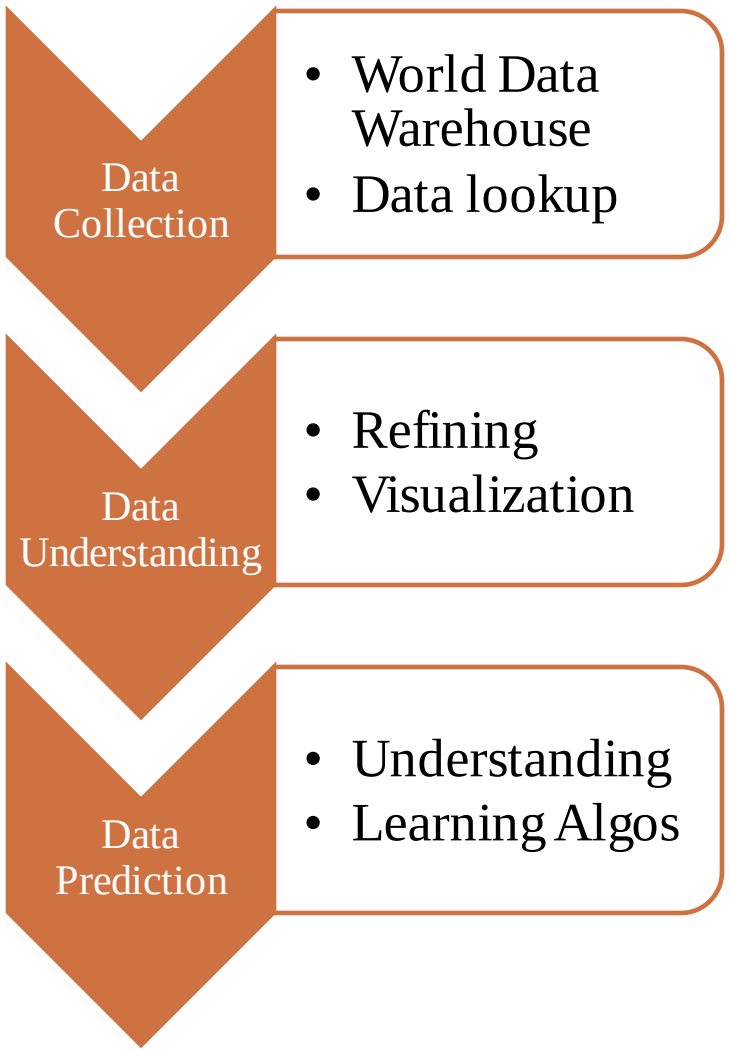
Regional Comparative Representation of Sex Suffering from Diarrhea and Pneumonea



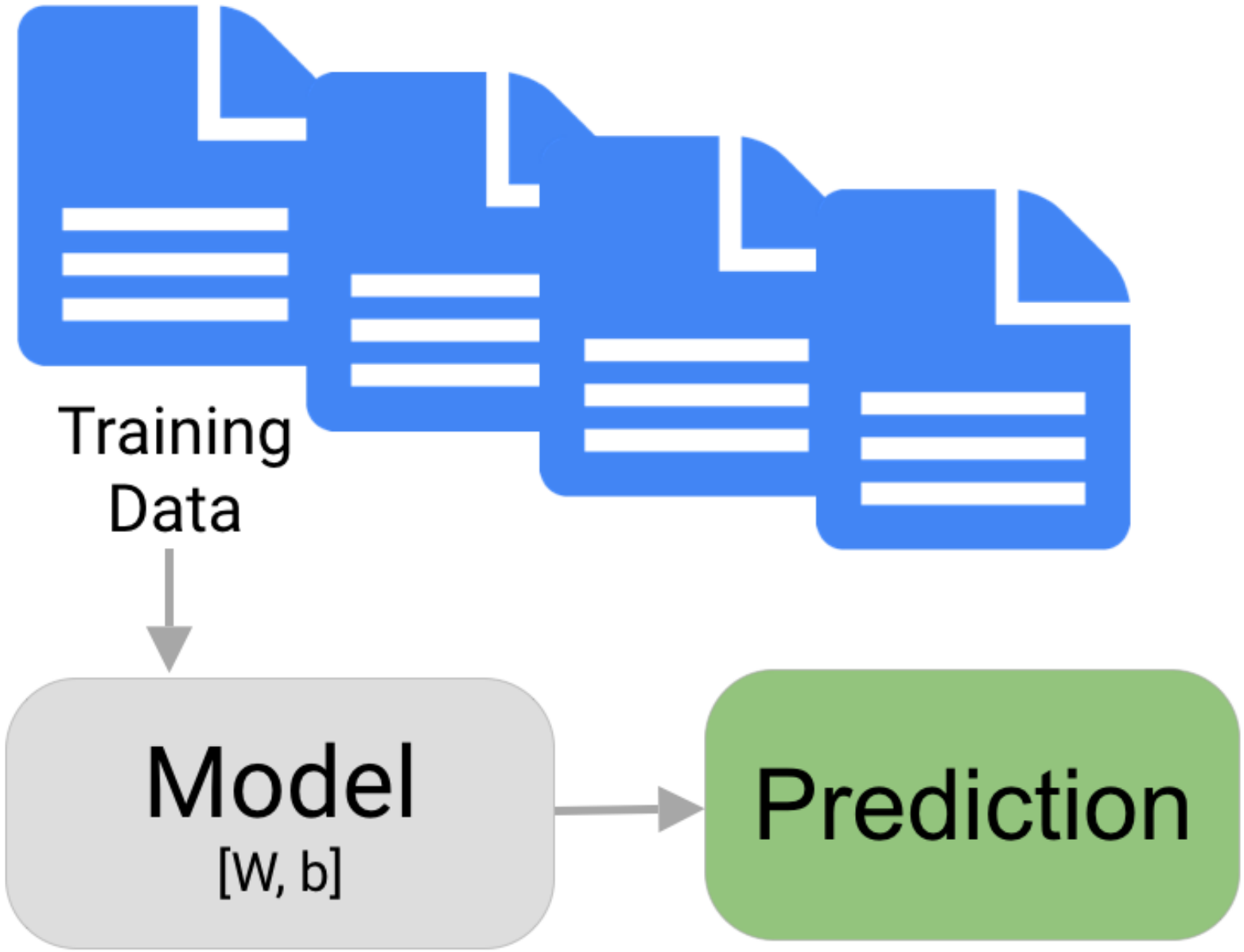
Region wise comparative representation of Males suffering Diarrhea and Pneumonea



NEXT STEPS



- We have successfully completed Phase 1 and now we are entering to Phase 2.
- So far we have completed our Data's collection, its Refining and Visualization that help us understood ill-effects of these diseases on human civilization.
- Further we will now implement Prediction models using Supervised Learning Approach.



PREDICTIONS USING DATASETS

A prediction is a statement about the future. It's a guess, sometimes based on facts or evidence

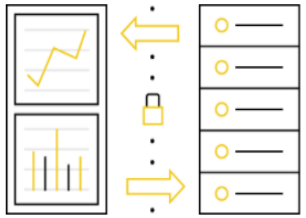
Analytics solutions for your whole organization

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.

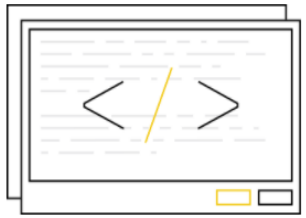
Create stunning interactive reports

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

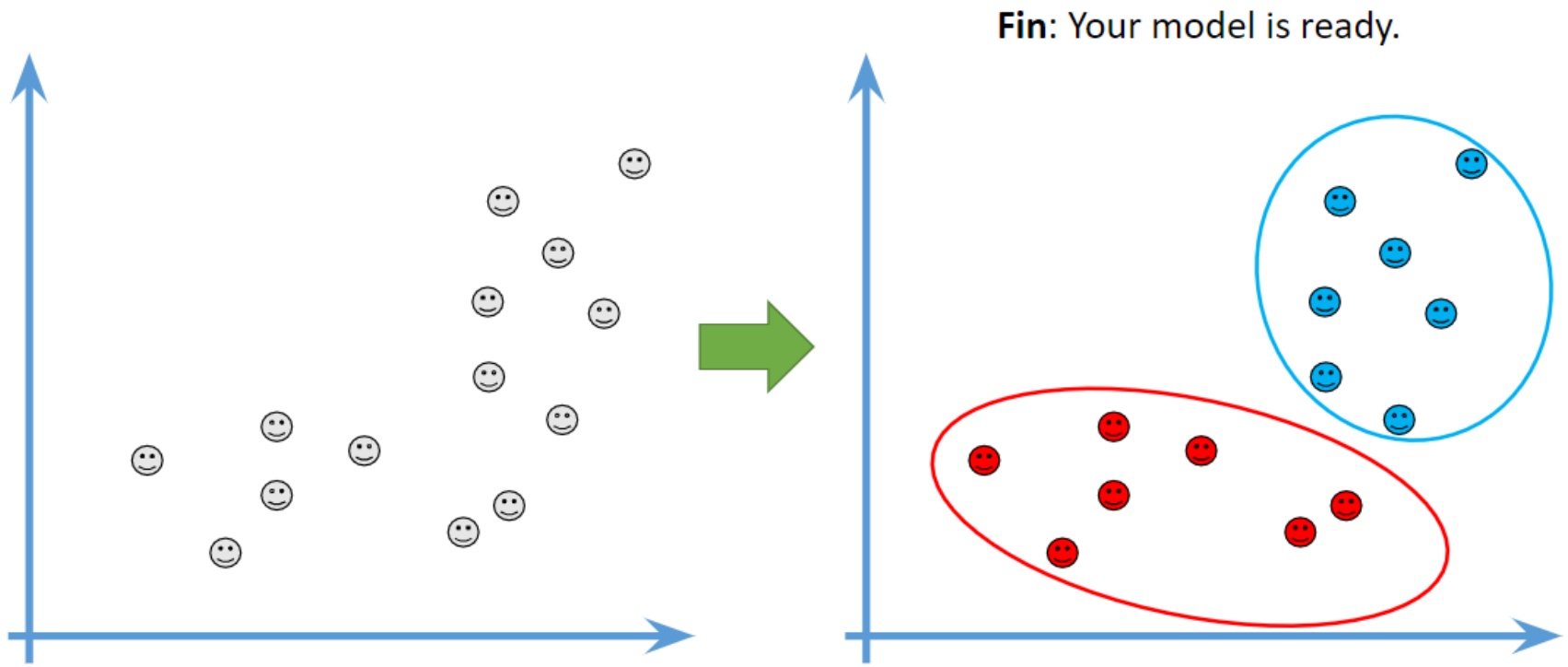
Deliver stunning interactive reports in your app with the Power BI Embedded service.

WHY POWER BI?

- In Power Bi we can apply predictive forecasting, and hindcasting, when visualizing and exploring the data. Forecasting in Power View utilizes built-in predictive forecasting models using exponential smoothing to automatically detect seasonality in the data to provide forecast results from a series of data. Explore forecast results by adjusting the desired confidence interval or by adjusting outlier data to see how they affect results. We can also hindcast to see how Power View would have predicted the present and recent past based on older data.
- The best data for forecasting is time series data or uniformly increasing whole numbers. The line chart has to have only one line. Multiple-line charts won't work, even if all but one line is filtered out

- **Data clustering** is an unsupervised classification method *(different from supervised classification where learning data is already tagged)*
- **Association Rule Learning** is to discover relationships of interest to the statistician between two or more variables stored in very large databases using association rule search algorithms.

UNDERSTANDING K-MEANS



How did it to do that?

STEP 1: Choose the number K of clusters.

STEP 2: select at random K points, the centroids (not necessarily from your dataset).

STEP 3: Assign each data point to the closest centroid → That forms K clusters.

STEP 4: Compute and place the new centroid of each cluster.

STEP 5: Reassign each data point to the new closest centroid.
If any reassignment took a place go to STEP 4. otherwise go to finish.

Your Model is Ready

WHAT IS APRIORI ?

- **Apriori** is an algorithm for frequent item set mining and association rule learning over transactional databases.
- It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets if those item sets appear sufficiently often in the dataset.

WHAT IS IT ALL ABOUT ?

APRIORI ALGORITGHM

STEP 1: Set a minimum support and confidence

STEP 2: Take all the subsets in transaction having higher support than minimum support.

STEP 3: Take all the rules of these subsets having higher confidence than minimum confidence.

STEP 4: Sort the rules by decreasing lift.

PREDICTION MODELS

Phase 2

```
df1 = pd.read_csv('Symptom-severity.csv')
df1.head()
```

	Symptom	weight
0	itching	1
1	skin_rash	3
2	nodal_skin_eruptions	4
3	continuous_sneezing	4
4	shivering	5

```
df1['Symptom'].unique()
```

```
array(['itching', 'skin_rash', 'nodal_skin_eruptions',
       'continuous_sneezing', 'shivering', 'chills', 'joint_pain',
       'stomach_pain', 'acidity', 'ulcers_on_tongue', 'muscle_wasting
       'vomiting', 'burning_micturition', 'spotting_urination', 'fati
       'weight_gain', 'anxiety', 'cold_hands_and_feets', 'mood_swings
       'weight_loss', 'restlessness', 'lethargy', 'patches_in_throat'
       'irregular_sugar_level', 'cough', 'high_fever', 'sunken_eyes',
       'breathlessness', 'sweating', 'dehydration', 'indigestion',
       'headache', 'yellowish_skin', 'dark_urine', 'nausea',
```

DISEASE PREDICTION MODEL REVIEW & ACCURACY

- Reading the weights of symptoms to identify the frequency of sample symptoms in most of the Diseases.
- Checking for symptoms present in our dataset.

GETTING STARTED WITH PREDICTION

- Getting an overview of number of diseases in dataset and accordingly assigning the disease names to labels for prediction.

- Finally Splitting the data in the ratio of

(85 : 15)

(train : test)

[17]
```
df['Disease'].unique()
```

```
array(['Fungal infection', 'Allergy', 'GERD', 'Chronic cholestasis',
       'Drug Reaction', 'Peptic ulcer diseae', 'AIDS', 'Diabetes',
       'Gastroenteritis', 'Bronchial Asthma', 'Hypertension', 'Migraine',
       'Cervical spondylosis', 'Paralysis (brain hemorrhage)', 'Jaundice',
       'Malaria', 'Chicken pox', 'Dengue', 'Typhoid', 'hepatitis A',
       'Hepatitis B', 'Hepatitis C', 'Hepatitis D', 'Hepatitis E',
       'Alcoholic hepatitis', 'Tuberculosis', 'Common Cold', 'Pneumonia',
       'Dimorphic hemmorhoids(piles)', 'Heart attack', 'Varicose veins',
       'Hypothyroidism', 'Hyperthyroidism', 'Hypoglycemia',
       'Osteoarthristis', 'Arthritis',
       '(vertigo) Paroymsal  Positional Vertigo', 'Acne',
       'Urinary tract infection', 'Psoriasis', 'Impetigo'], dtype=object)
```

[18]
```
data = df.iloc[:,1:].values
labels = df['Disease'].values
```

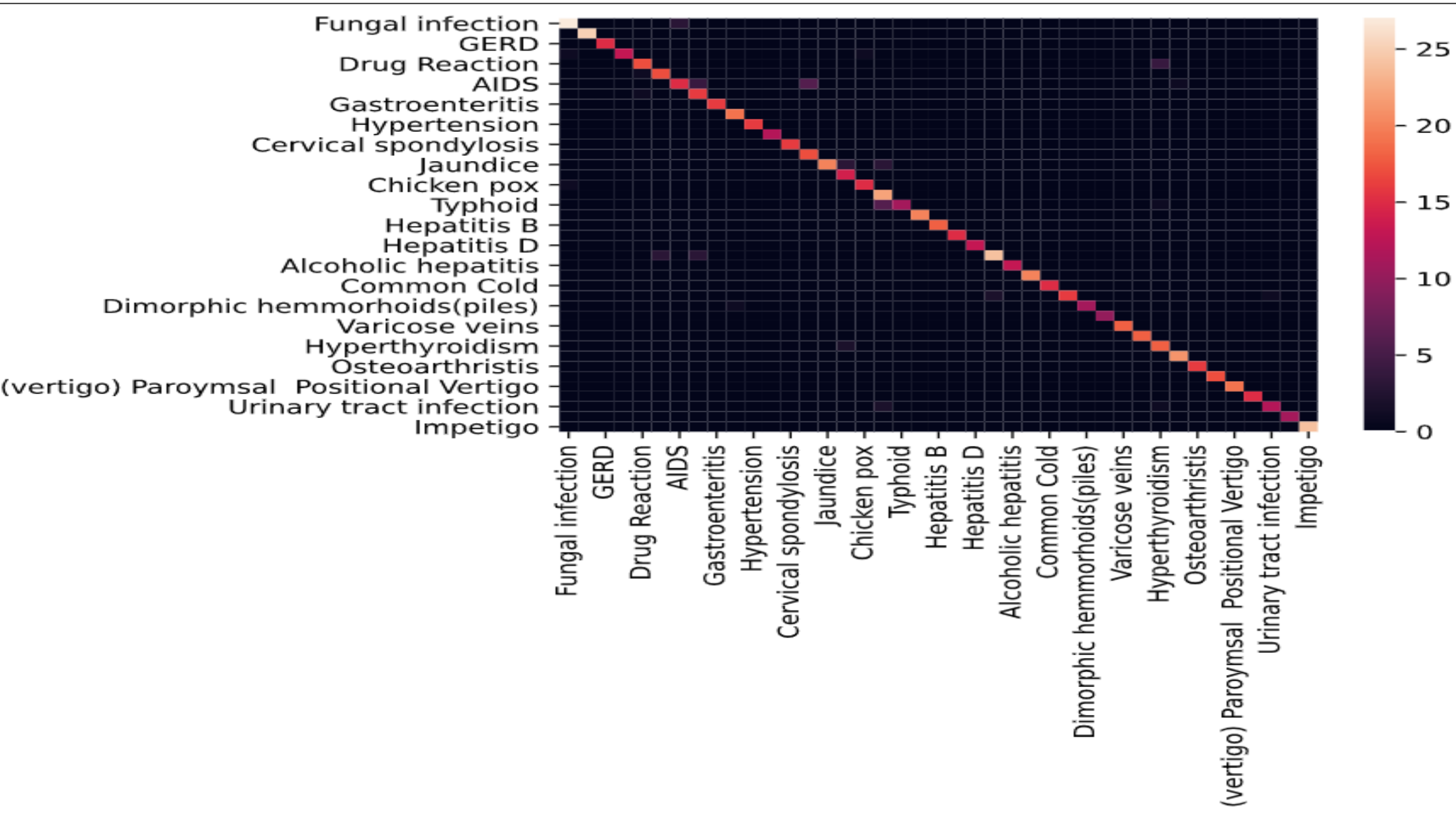
[19]
```
x_train, x_test, y_train, y_test = train_test_split(data, labels, shuffle=True, train_size = 0.85)
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

```
(4182, 17) (738, 17) (4182,) (738,)
```

MODEL'S ACCURACY

- ✓ Finally training model and predicting the Test set using SVC (Support Vector Classifier)
- ✓ Our model's accuracy comes out to be 93%.
- ✓ To understand the balance between precision and recall we calculated F1 Score that turned out to be 93.8%. The Concludes that most of the Actual True Positive Values were predicted True Positive and Similarly for the other one.

```
[21]
preds = model.predict(x_test)

[22]
conf_mat = confusion_matrix(y_test, preds)
df_cm = pd.DataFrame(conf_mat, index=df['Disease'].unique(), columns=df['Disease'].unique())
print('F1-score% =', f1_score(y_test, preds, average='macro')*100, '|', 'Accuracy% =', accuracy_score(y_test, preds)*100)
sns.heatmap(df_cm)

F1-score% = 93.87543802830123 | Accuracy% = 93.08943089430895
```

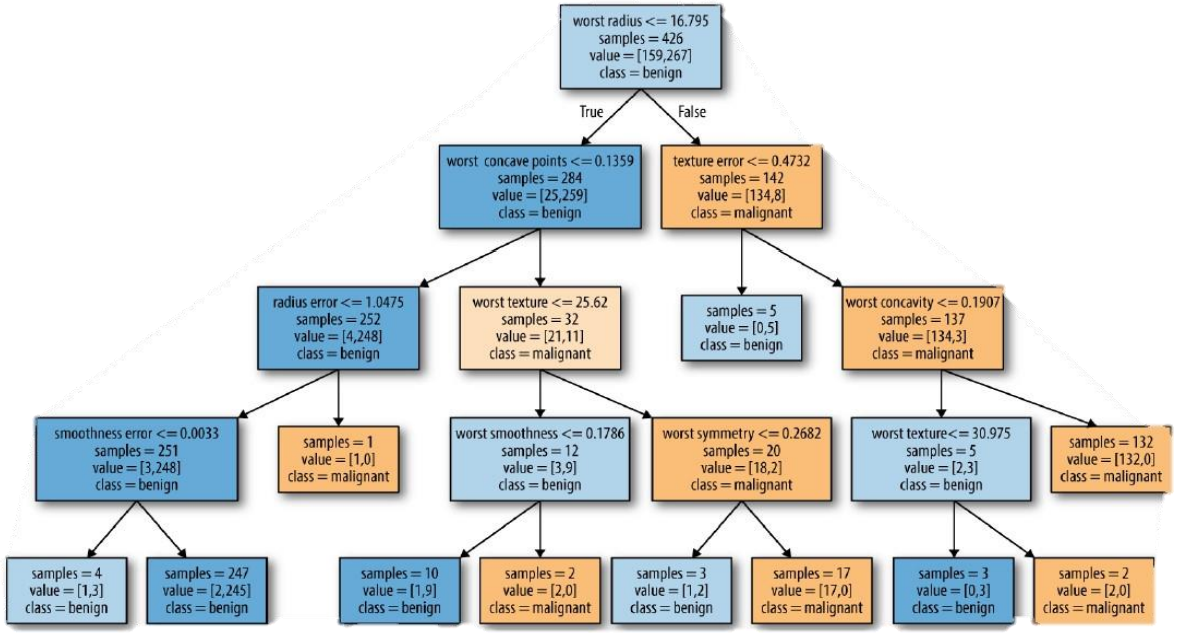

Fungal infection
GERD
Drug Reaction
AIDS
Gastroenteritis
Hypertension
Cervical spondylosis
Jaundice
Chicken pox
Typhoid
Hepatitis B
Hepatitis D
Alcoholic hepatitis
Common Cold
Dimorphic hemmorhoids(piles)
Varicose veins
Hyperthyroidism
Osteoarthristis
(vertigo) Paroymsal Positional Vertigo
Urinary tract infection
Impetigo
25
20
15
10
5
0

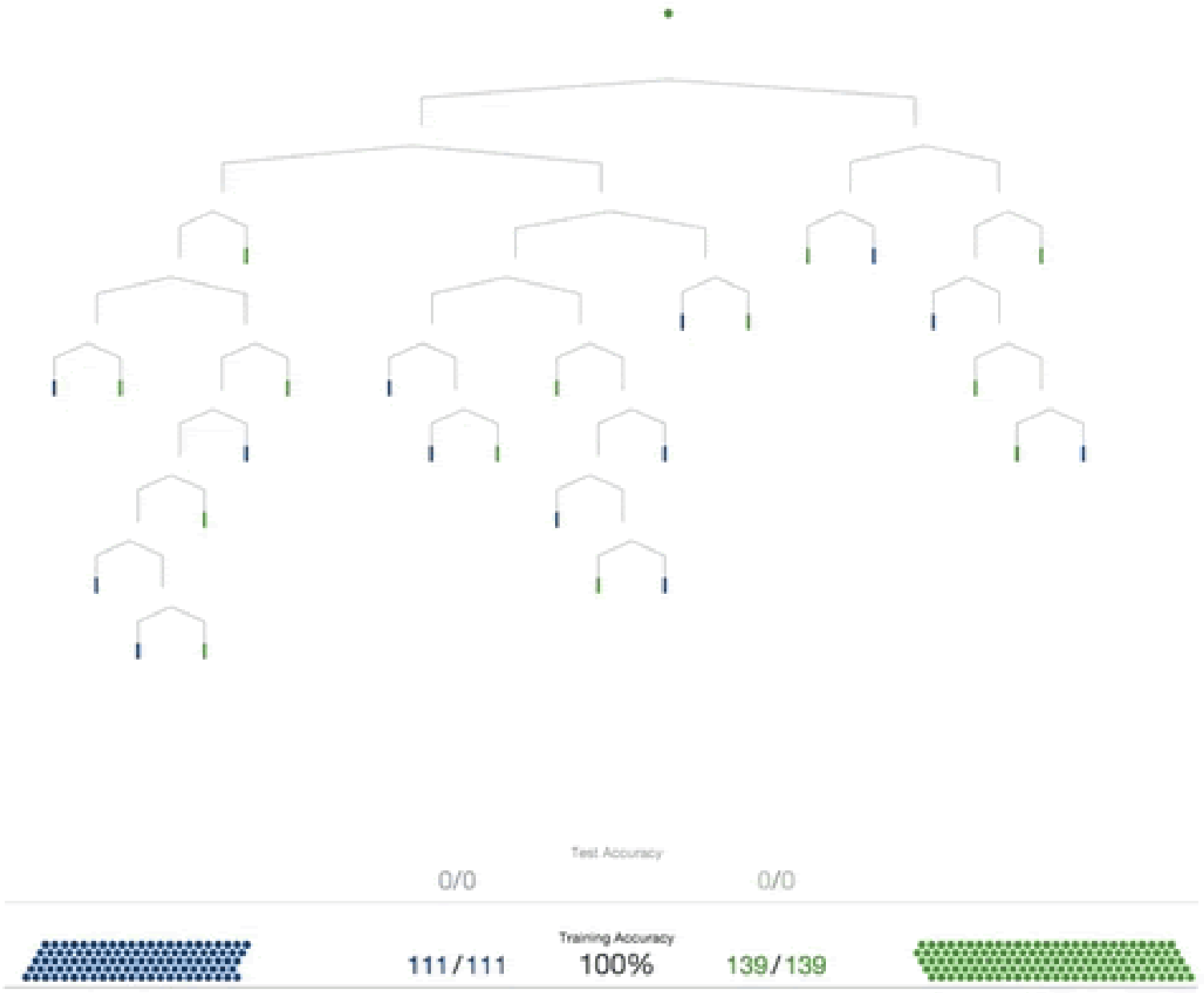
FEW NEW LIBRARIES USED

1. Folium: folium builds on the data wrangling strengths of the Python ecosystem and the mapping strengths of the leaflet.js library. Manipulate your data in Python, then visualize it in on a Leaflet map via folium.
2. Calmap: Plot Pandas time series data sampled by day in a heatmap per calendar year, like GitHub's contributions plot, using matplotlib.

DECISION TREE

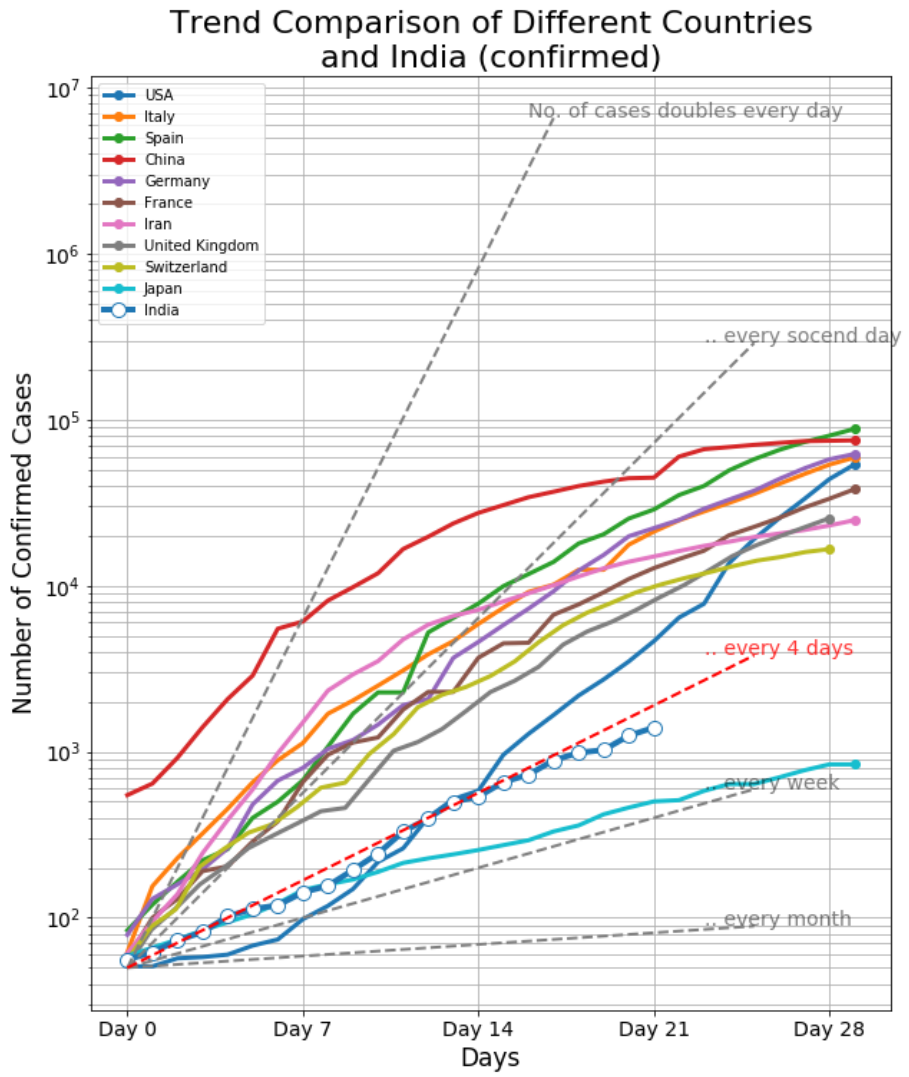
- Decision Trees are the building blocks of the supervised learning methods.
- Decision tree is a Binary tree flowchart.
- Each node splits a group of observations, according to some feature variable.
- Decision tree can be used to approximate a continuous target variable.
- Here, tree splits such that each group has the lowest mean squared error
- Decision Trees are easily interpreted.





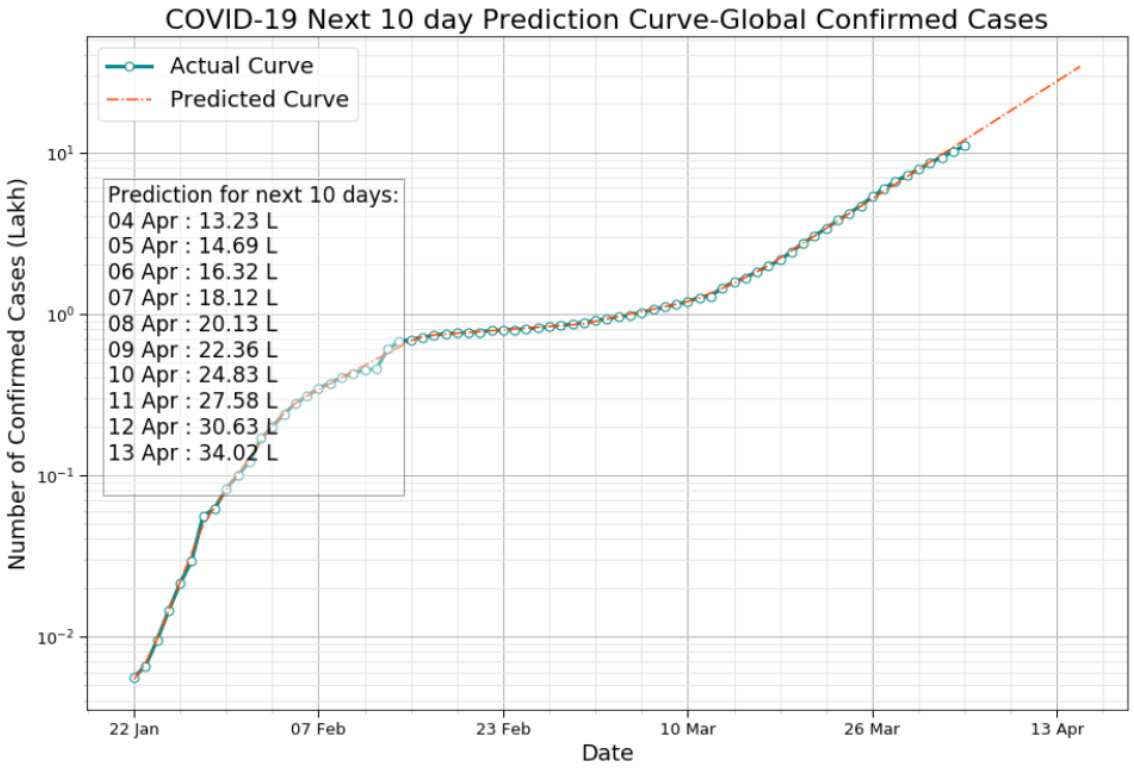
CONCLUSION

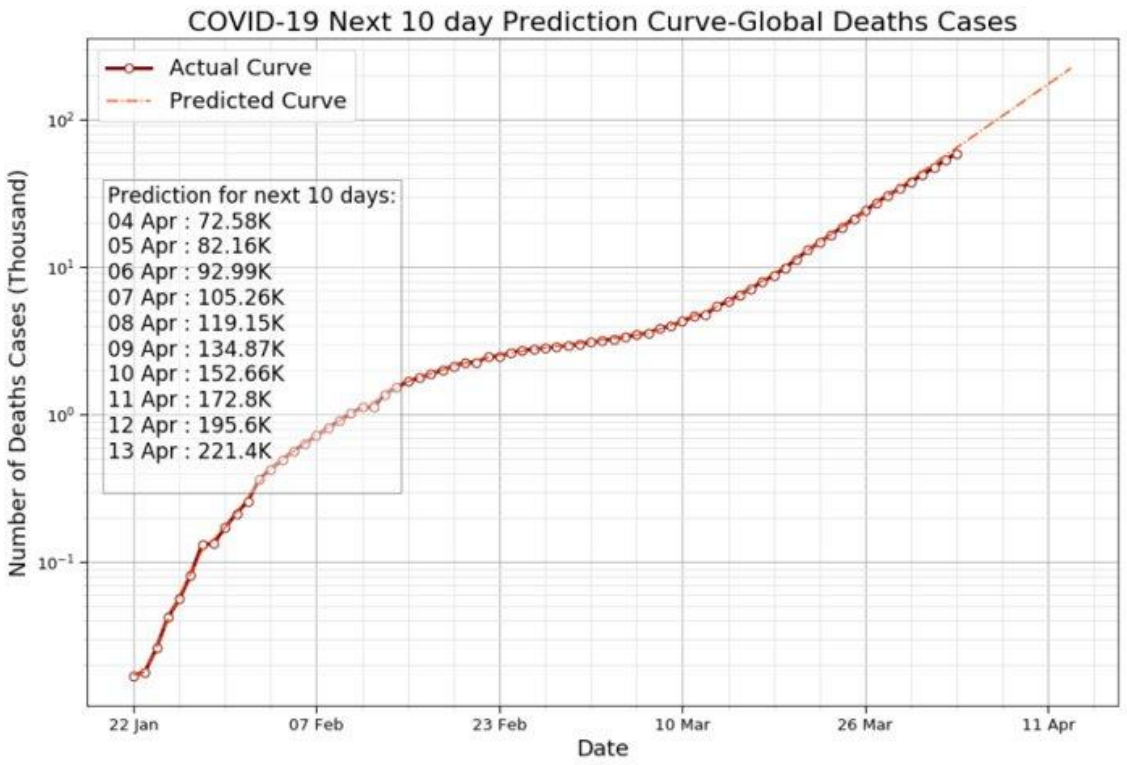
- ✓ Before **neural networks** became popular, decision trees were the state of the art algorithm in **Machine Learning**.
- ✓ Several other ensemble models like **Random Forests** are much more powerful than the **vanilla decision tree**.



THIS GRAPH HERE SHOWS THE RELATION OF HOW THE CONFIRMED CASES WERE INCREASING IN THE FEW MAJOR COUNTRIES ON THE GLOBE.

PREDICTION ON DEATHS AND CONFIRMED CASES DUE TO COVID-19





HEART DISEASE PREDICTION

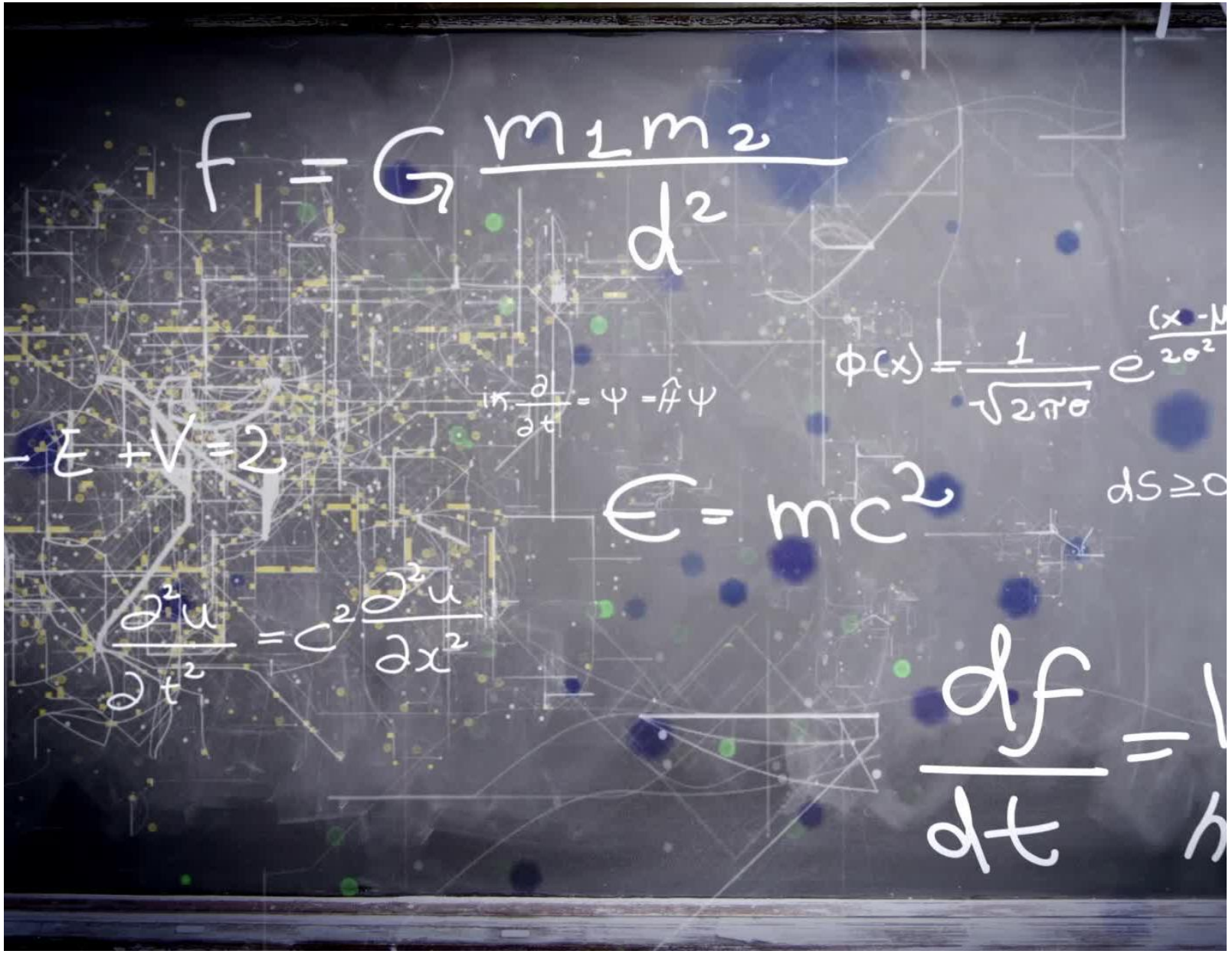
- Heart disease is the leading cause of death in the world, according to the Centers for Disease Control and Prevention (CDC)Trusted Source. In the United States, 1 in every 4 deaths in is the result of a heart disease. That's about 610,000 people who die from the condition each year.
- Heart disease doesn't discriminate. It's the leading cause of death for several populations, including white people, Hispanics, and Black people. Almost half of Americans are at risk for heart disease, and the numbers are rising.

TYPES OF HEART DISEASES

- **Coronary artery disease** is the most common type of heart disease in the US. Coronary arteries supply blood to the heart muscle and coronary artery disease occurs when there is a buildup of **cholesterol** plaque inside the artery walls. Over time, this buildup of plaque may partially block the artery and decrease blood flow through it.
- **Congenital heart defects-** A person with a congenital heart defect is born with a heart problem. There are many types of congenital heart defect.
- Arrhythmia refers to an irregular heartbeat. It occurs when the electrical impulses that coordinate the heartbeat do not work properly. As a result, the heart may beat too fast, too slowly, or erratically.
- **Dilated cardiomyopathy** the heart chambers become dilated, meaning that the heart muscle stretches and becomes thinner. The most common causes of dilated cardiomyopathy are prior heart attacks, arrhythmias, and toxins.

SYMPTOMS OF HEART DISEASE

- The symptoms of heart disease depend on the specific type a person has. Also, some heart conditions cause no symptoms at all. The following symptoms may indicate a heart problem:
- angina, or chest pain
- difficulty breathing fatigue and lightheadedness and swelling due to fluid retention, or edema
- In children, the symptoms of a congenital heart defect may include cyanosis, or a blue tinge to the skin, and an inability to exercise.
- Some signs and symptoms that could indicate heart attack include:
- Sweating, arm, jaw, back, or leg pain
- a choking sensation
- swollen ankles , fatigue
- an irregular heartbeat



METHOD USED IN THIS PREDICTION MODELS

Phase 2

K-NEAREST NEIGHBOR(KNN) ALGORITHM

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.
- It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

K-NEAREST NEIGHBOR(KNN) ALGORITHM APPLIED TO HEARTS DISEASE

In [14]:
```
y = dataset['target']
X = dataset.drop(['target'], axis = 1)
```

In [15]:
```
from sklearn.model_selection import cross_val_score
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    score=cross_val_score(knn_classifier,X,y,cv=10)
    knn_scores.append(score.mean())
```

In [16]:
```
plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
```

Out[16]:
```
Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```

FINAL PREDICTION..

```
Out[16]: Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')
```


```
In [17]: knn_classifier = KNeighborsClassifier(n_neighbors = 12)
         score=cross_val_score(knn_classifier,X,y,cv=10)

In [18]: score.mean()

Out[18]: 0.8448387096774195
```

RESULT AND CONCLUSION :

- The project is designed in such a way that the system takes symptoms from the user as input and produces output i.e. predict disease.
- The user can select a minimum of one to a maximum of five symptoms.
- Less accuracy will be attained if only one symptom is entered. More the number of symptoms, the greater is the accuracy.

SUBMITTED TO:

MANISH SHARMA SIR

SUBMITTED BY:

AYAN BHATNAGAR (LEADER)
KARTIKEY GARG
YASH SHARMA
PRADYUMN BAHUKHANDI

Phase 3

FINAL STEPS

Modelling

Deployment

Testing

DISEASE PREDICTION USING MACHINE LEARNING, WHY ?

- The classical diagnosis method is a process where the patient has to visit a doctor, undergo various medical tests. This process is very time-consuming.
- To save time required for the initial process of diagnosisying symptoms, this project proposes an automated disease prediction system that relies on user input.
- The system takes input from the user and provides a list of “probable” diseases.
- We do not prefer user to rely on this as this is just a prediction

MODULE DESCRIPTION :

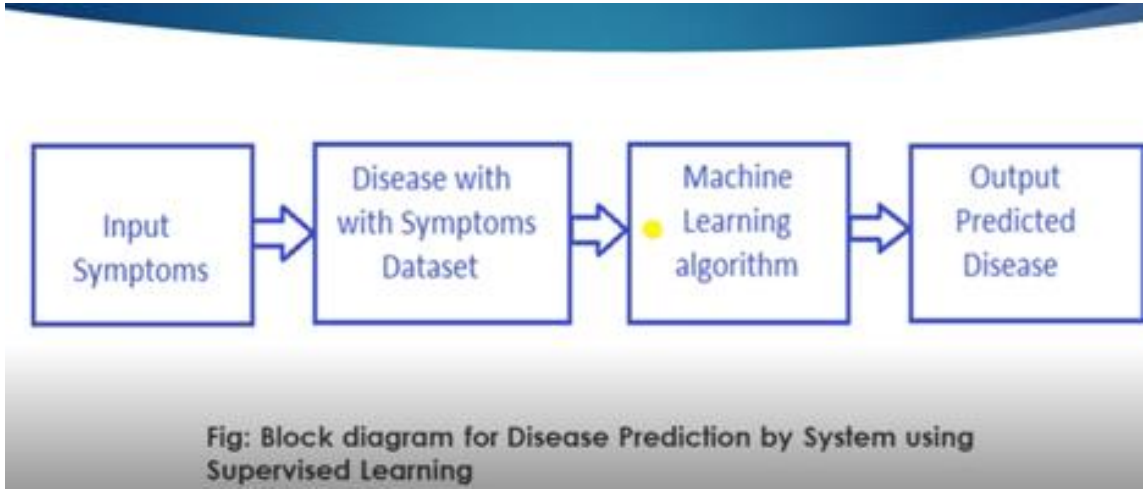
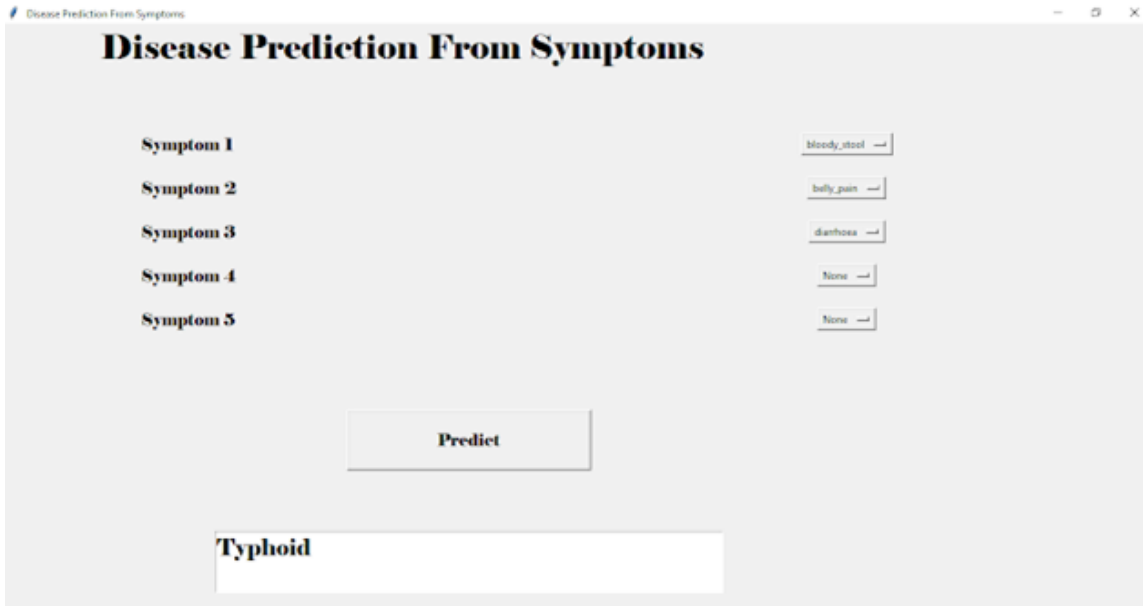


Fig: Block diagram for Disease Prediction by System using Supervised Learning

- The system will predict the disease where the symptoms are given as the input.
- The disease will be predicted using the Naive Bayesian algorithm.
- According to the literature survey, this algorithm results in the maximum accuracy for a larger dataset.
- The dataset contains disease as labels and for each disease, symptoms are given. 70% of the dataset will be used as training and 30% will be used for training data.

BASE GUI :



- Tkinter package is used for the User interface.
- Tkinter is the standard GUI library for python.
- Python, when combined with Tkinter, provides a fast and easy way to create a GUI application.
- Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

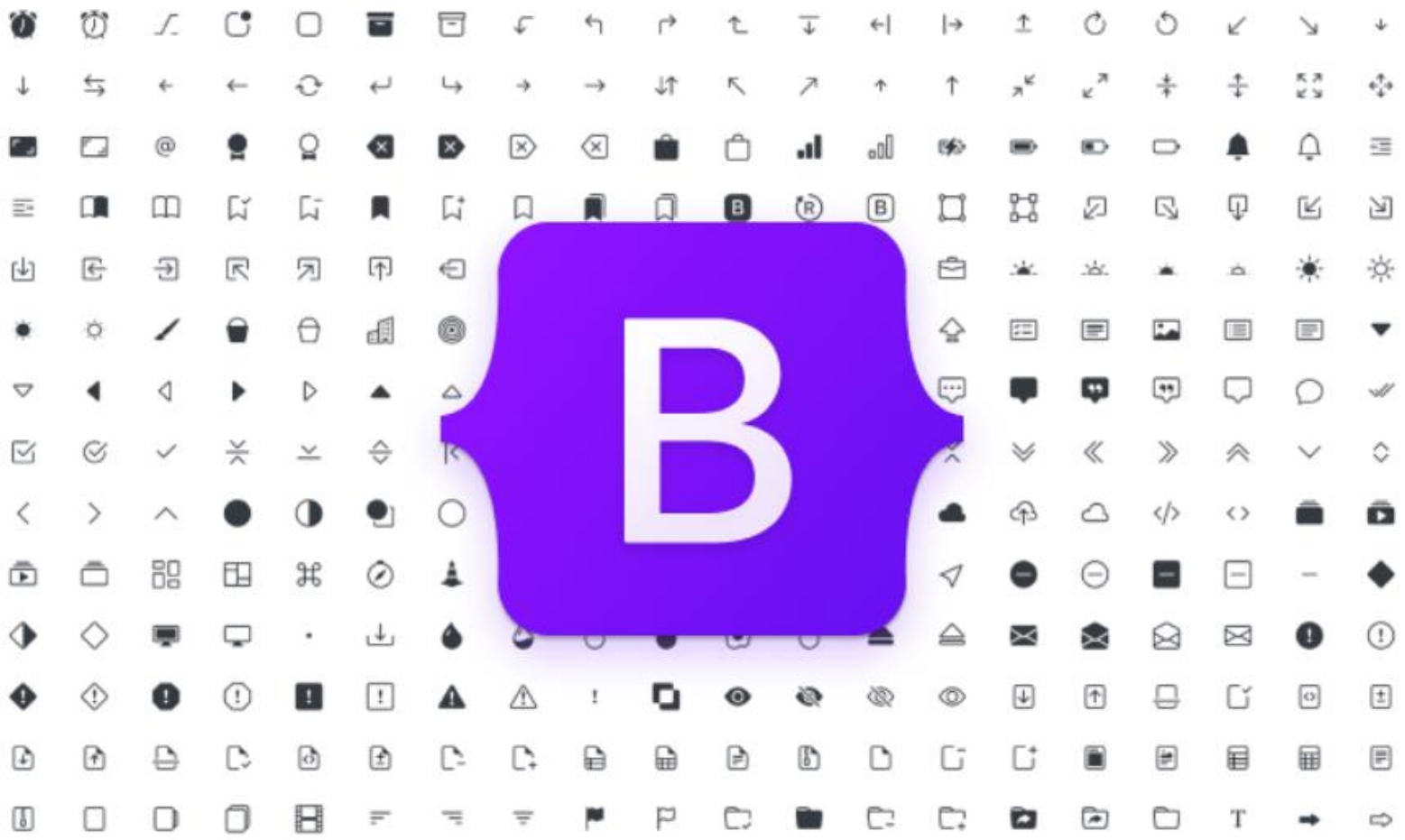
TOOLS AND TECHNOLOGIES USED FOR WEBSITE DESIGNING

Phase-3

HTML,CSS AND JAVASCRIPT

CSS3 MEDIA QUERIES

- Media queries can be used to check many things, such as:
- width and height of the viewport
- width and height of the device
- orientation (is the tablet/phone in landscape or portrait mode?)
- resolution
- Using media queries are a popular technique for delivering a tailored style sheet to desktops, laptops, tablets, and mobile phones (such as iPhone and Android phones).



BOOTSTRAP ICONS

- Bootstrap has its own open source SVG icon library, designed to work best with our components and documentation.
- Bootstrap Icons are designed to work best with Bootstrap components, but they'll work in any project. They're SVGs, so they scale quickly and easily, can be implemented in several ways, and can be styled with CSS.

FLEXBOX

- Flexbox is a one-dimensional layout method for arranging items in rows or columns.
- Items *flex* (expand) to fill additional space or shrink to fit into smaller spaces.
- Flexbox provides a property called flex direction that specifies which direction the main axis runs (which direction the flexbox children are laid out in).

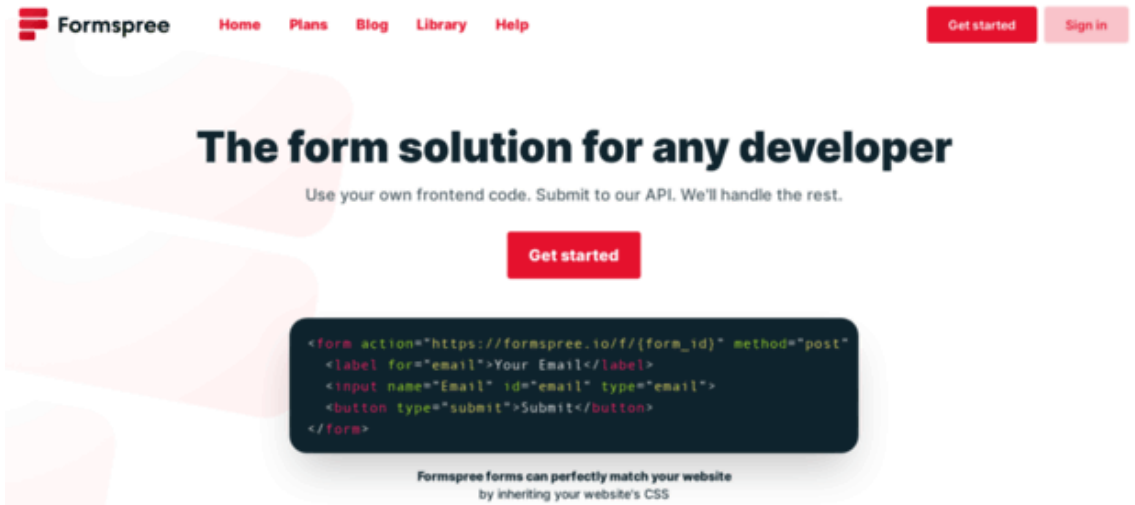
FORMSPREE API

What is Formspree?

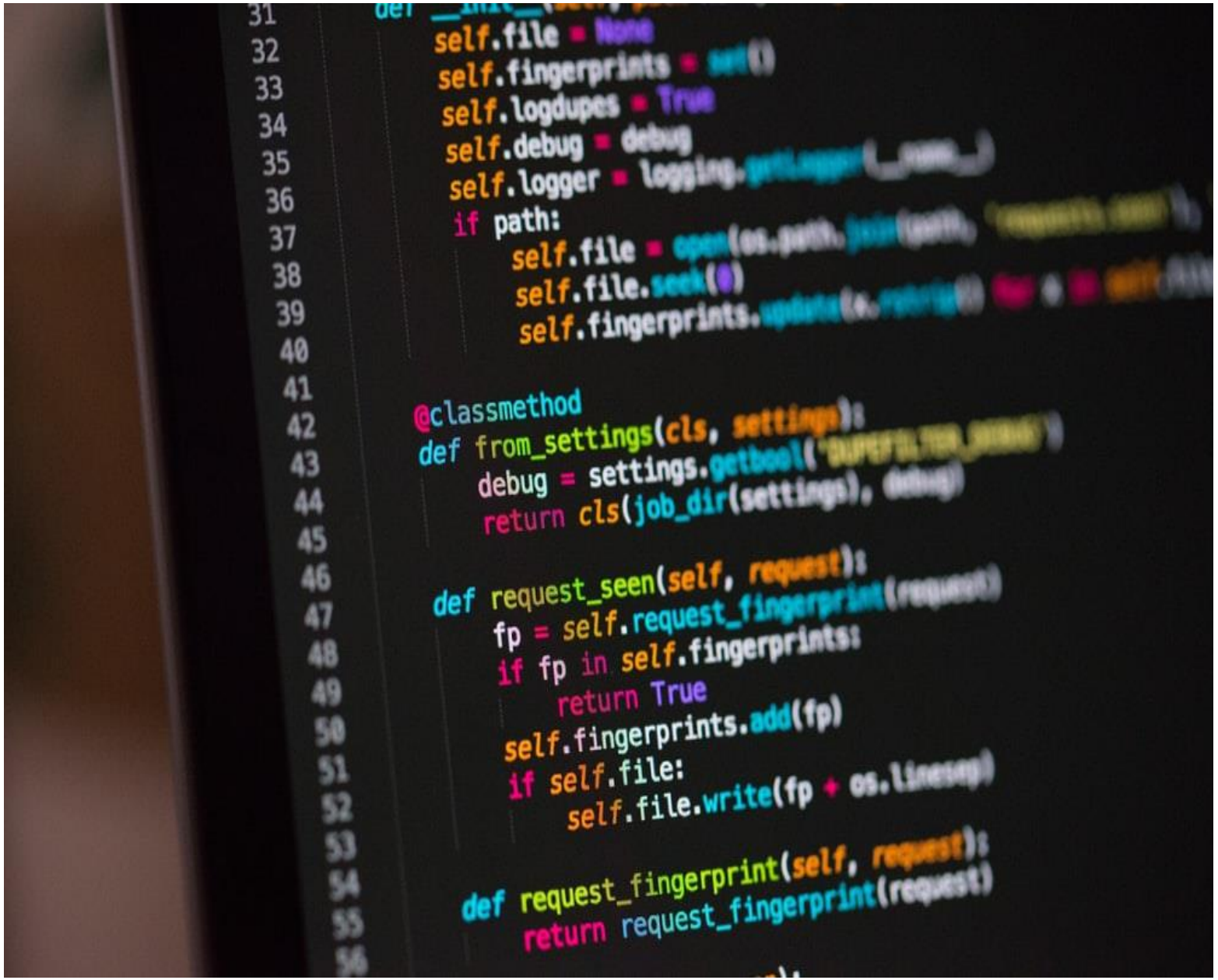
It is a form backend, API and email service for HTML forms. It is the simplest way to embed custom contact us forms, order forms, or email capture forms.

Formspree is a tool in the **Web Forms** category of a tech stack.

FUNCTIONING OF FORMSPREE API



- The Formspree API provides services to Fetch all your submissions and more. It allows you to design a form for your static site, and point the action to Formspree. Formspree provides a backend service for millions of form submissions. Formspree a form builder software that helps to create beautiful, accessible, highly customized forms using any CSS, HTML or Javascript with no server codes. It helps to enhance customer relationships and improve efficiency.
- It is the simplest way to embed custom contact us forms, order forms, or email capture forms with built-in plugins, for services like Stripe, Mailchimp, Google Sheets and more. It securely stores the form data and provides a clean, easy-to-use admin.
- Some of its features include collecting leads from simple forms, collect payments or donations with the strip elements plugin, email notifications, filter spam, and more.



CODE SAMPLES

PHASE-3

```css
/*---------------form-----------------*/
.form1{
    height: 100vh;
    width: 100%;
    background-color: aliceblue;
    display: flex;
    align-items: center;
    justify-content: center;
    flex-direction: column;
}
.container{
    width: 90%;
    max-width: 500px;
    margin:0 auto;
    padding: 20px;
    box-shadow: 0px 0px 20px #00000010;
    background-color: white;
    border-radius: 8px;
    margin-bottom: 20px;
}
.form-group{
    width: 100%;
    margin-top: 20px;
    font-size: 20px;
}
.form-group input,
.form-group textarea{
```

- This is the part where we have styled different components of the form.

- For designing the footer we have targeted different bootstrap icons and added different colours, margins on them.

```css
/*----------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--------------
```

```
def NaiveBayes():
    from sklearn.naive_bayes import MultinomialNB
    gnb = MultinomialNB()
    gnb=gnb.fit(X,np.ravel(y))
    from sklearn.metrics import accuracy_score
    y_pred = gnb.predict(X_test)
    print(accuracy_score(y_test, y_pred))
    print(accuracy_score(y_test, y_pred, normalize=False))

    psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

    for k in range(0,len(l1)):
        for z in psymptoms:
            if(z==l1[k]):
```

- Sklearn is a package which contains all the machine learning algorithm
- In line 78 it means in sklearn use naïve bayes and import the multinomial naivebayes
- In line 79 an object gnb is being created of class multinomialnb

```
inputtest = [12]
predict = gnb.predict(inputtest)
predicted=predict[0]

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

- The predict function will predict the input test (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

```
<!-----------------disease predicted----------------->
    <section class="campus">
<h1>Disease Predicted</h1>
<p>Below is the list of disease Predicted with higher accuracy:</p>
<div class="row">
<div class="campus-col">
    <img src="images/allergy.jpg" height="256">
    <div class="layer">
        <h3>Allergy</h3>
    </div>
</div>
<div class="campus-col">
    <img src="images/pneumonia.jpg" height="244">
    <div class="layer">
        <h3>Pneumonia</h3>
    </div>
</div>
<div class="campus-col">
    <img src="images/dengue.jpg">
    <div class="layer">
        <h3>Dengue</h3>
    </div>
</div>
</div>
    </section>
```

- 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 basically the div with 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 image related to them and in this div with class=”campus-col” we have another divs with classes layers and with <h3> type headings in order to sytle them in external css.

```
    <!----------reviews---------->
    <section class="testimonials">
<h1>What our user says: </h1>
<p>Ratings</p>
<div class="row">
    <div class="testimonial-col">
        <img src="images/m.png" alt="">
        <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">
        <img src="images/f.png" alt="">
        <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>
```

- 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=”testimonials-col” in order 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”

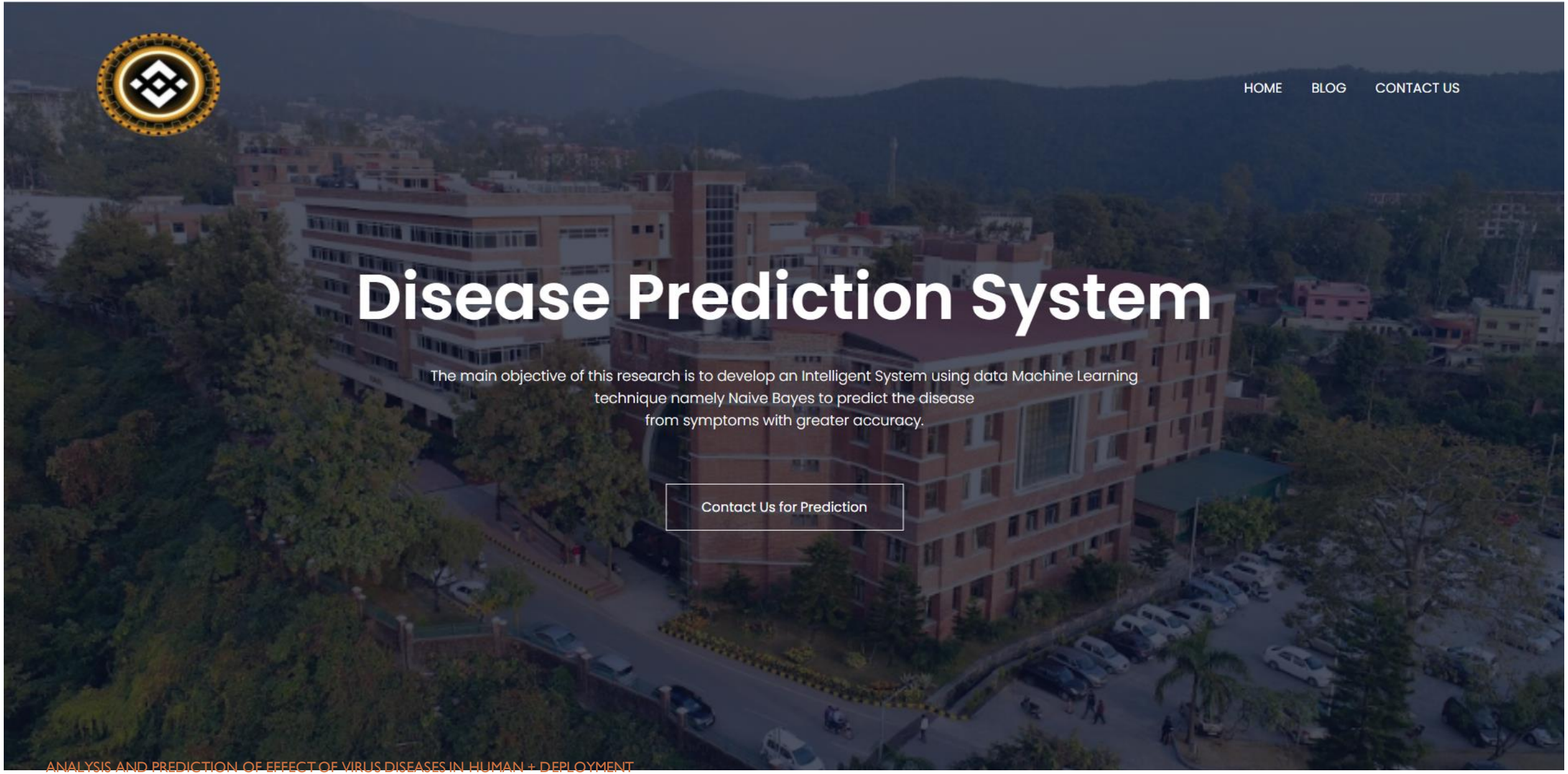
CONTACT US PAGE

```
<!------contact us---------->
<section class="location">
    <iframe src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3442.8598869458638!2d78.08112941473561!3d30.354931681770 45!2m3

</section>
<section class="form1">
    <div class="container">
      <form action="https://formspree.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="massage">S Symptoms:</label>
          <textarea name="massage" id="massage" rows="10" maxlength="200"></textarea>
        </div>

        <button type="submit" id="test-form-submit">Submit</button>
      </form>
    </div>
    <div id="status"></div>
  </section>
```

- 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 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.

OUR WEBSITE SAMPLES

PHASE-3


HOME
BLOG
CONTACT US
Disease Prediction System
The main objective of this research is to develop an Intelligent System using data Machine Learning technique namely Naive Bayes to predict the disease from symptoms with greater accuracy.
Contact Us for Prediction

Our facilities

Below are some of the facilities provided by us:

Contact US

Feel free to contact us we are there to help you.

Free Of Cost

Our services are completely free of Cost.

Quick Response

We resolve your Query within 24 hours.

What our user says:

Ratings

John Sharma

Excellent service

★★★★☆

Tanya Gupta

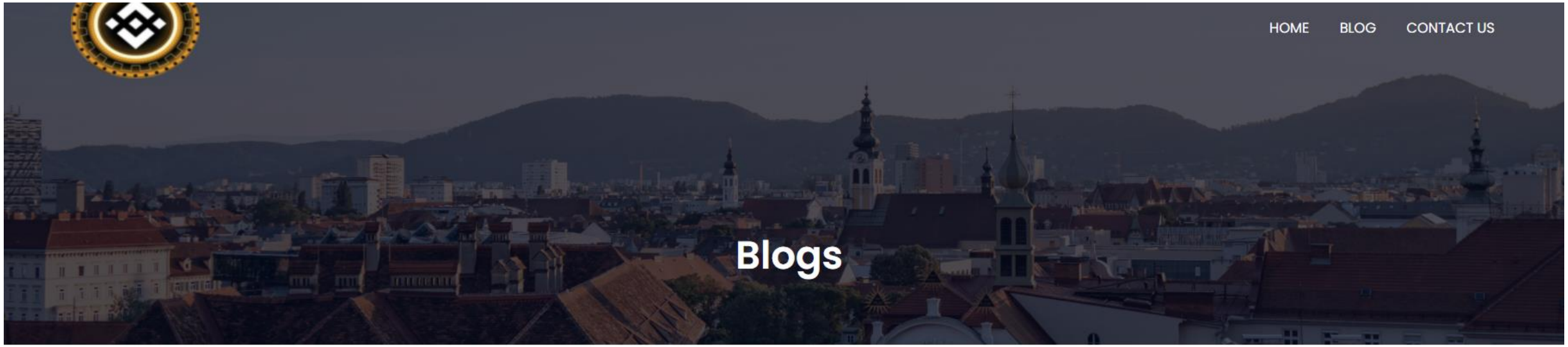
Quick response

★★★★½

About Us

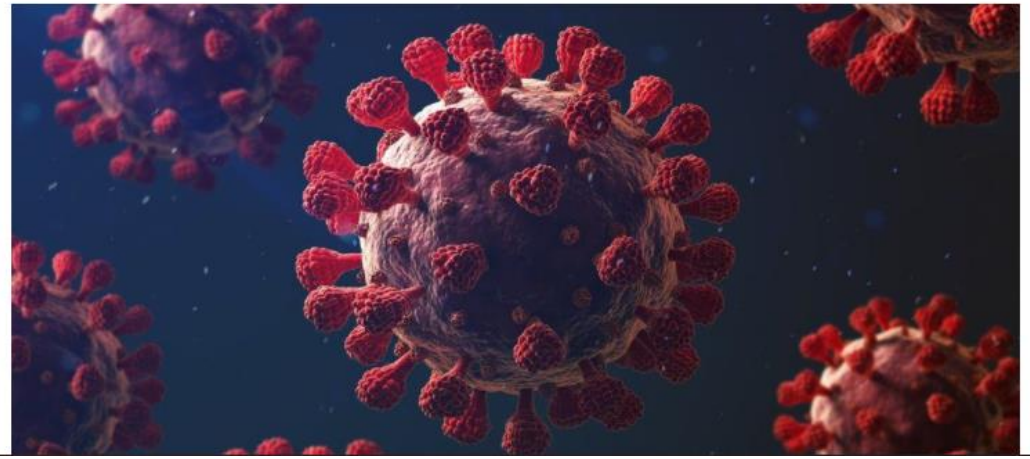
We hereby certify that the work, which is being presented, in partial fulfilment of the requirement
for the award of the Degree of Bachelor of Technology and submitted to the DIT University is an authentic record of our work under the guidance of Manish Sharma Sir.

Made with ♡ by IBM-G10 group



Corona Virus

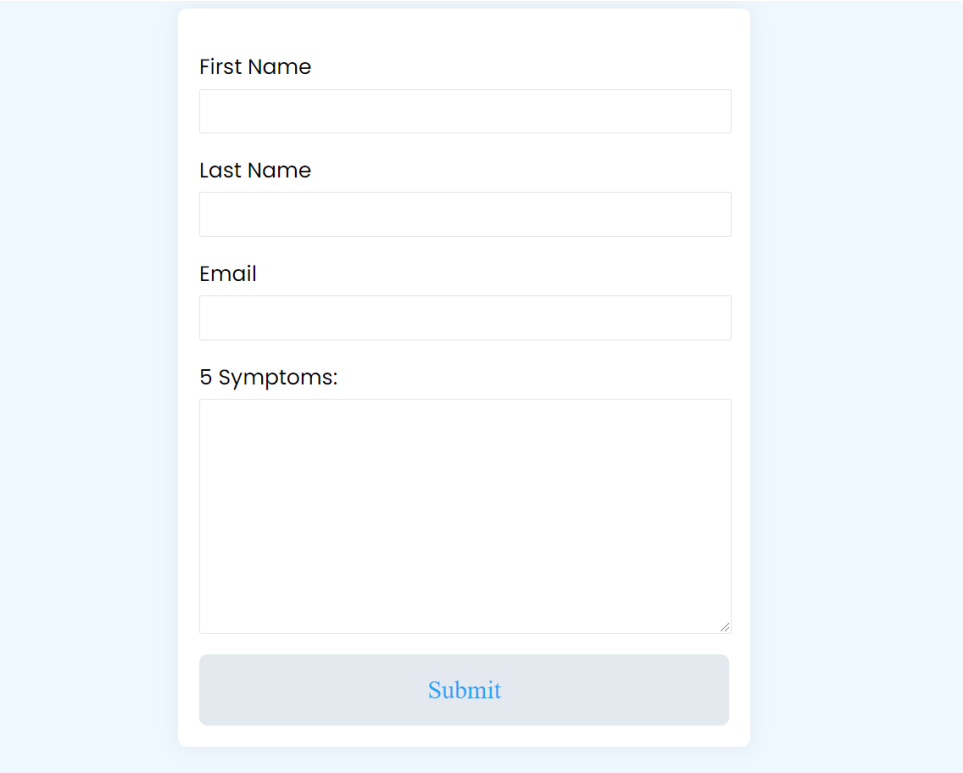
Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus. Most people infected with the COVID-19 virus will experience mild to moderate respiratory illness and recover without requiring special treatment. Older people, and those with underlying medical problems like cardiovascular disease, diabetes, chronic respiratory disease, and cancer are more likely to develop serious illness. 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. Protect yourself and others from infection by washing your hands or using an alcohol based rub frequently and not touching your face. The COVID-19 virus



Contact Us


Sahastradhara Rd
ఉత్తరాఖండ్
దిశలు
పెద్ద మ్యాప్‌ని చూడండి
Bank and ATM
కాబన అఫైర్
Cabana Affair
సుదర్శన్ విహార్, డో బాచి మార్...
Sudarshan Vihar, Do Bachi Marg...
Sai Residency
ఇన్ఫర్మేషన్ టెక్నాలజీ...
Information Technology Development Agency...
ఉత్తరాఖండ్ ఫార్మసీ...
Uttarakhand Pharmacy Council
దేవరిషి అపార్ట్‌మెంట్స్
Devrishi Apartments
దర్గా షరీఫ్ పీర్ బాబా
Dargah Shareef Peer Baba
డొరన్ గ్రీన్స్ రెసిడెన్షియల్...
Dhoran Greens Residential Complex
శివ్ మందిర్
Shiv Mandir
పోస్ట్‌మాస్టర్ పోస్ట్ ఆఫీస్...
Postmaster Post Office (Govt.)
Sahastradhara Rd
ఏ.ఎస్.ఐ.సి డిస్పెన్సరీ...
ESIC DISPENSARY, IT PARK, DEHRADUN
సుద్ధోవాలి
సున్నోవాలి
Saundhowali
రిలయన్స్ జియో, ప్లాట్ నో 32...
Reliance Jio , Plot no 32 IT park Sahastradhara...
IT PARK SAHASTRADHARA...
సిద్ధార్థ లా కాలేజ్...
Siddhartha Law College, Dehradun
పితంబెర్ మార్గ్
సిద్ధార్థ రోడ్డు
Zhen Sports
నాగరాజ మందిర్
Nagraja mandir
డీఎస్ఎల్ఆర్ కెమెరా ఆన్ రెంట్
DSLR Camera On Rent
భారతీయ స్టేట్ బ్యాంక్
State Bank Of India
ప్రార్థన మందిర్
Prarthna Mandir
బ్లూ మౌంటెన్స్ కాలేజ్ ఆఫ్...
Blue Mountains College Of Teachers Education
A2Z Pets Mania
పెంపుడు జంతువులను అమ్మే స్థలం
ఈజీడే క్లబ్
EasyDay Club
Google
టెంపుల్
Temple
కీబోర్డ్ షార్ట్‌కట్‌లు
మ్యాప్ డేటా ©2021
వినియోగ నిబంధనలు
మ్యాప్ ఎర్రర్‌ను రిపోర్ట్ చేయండి

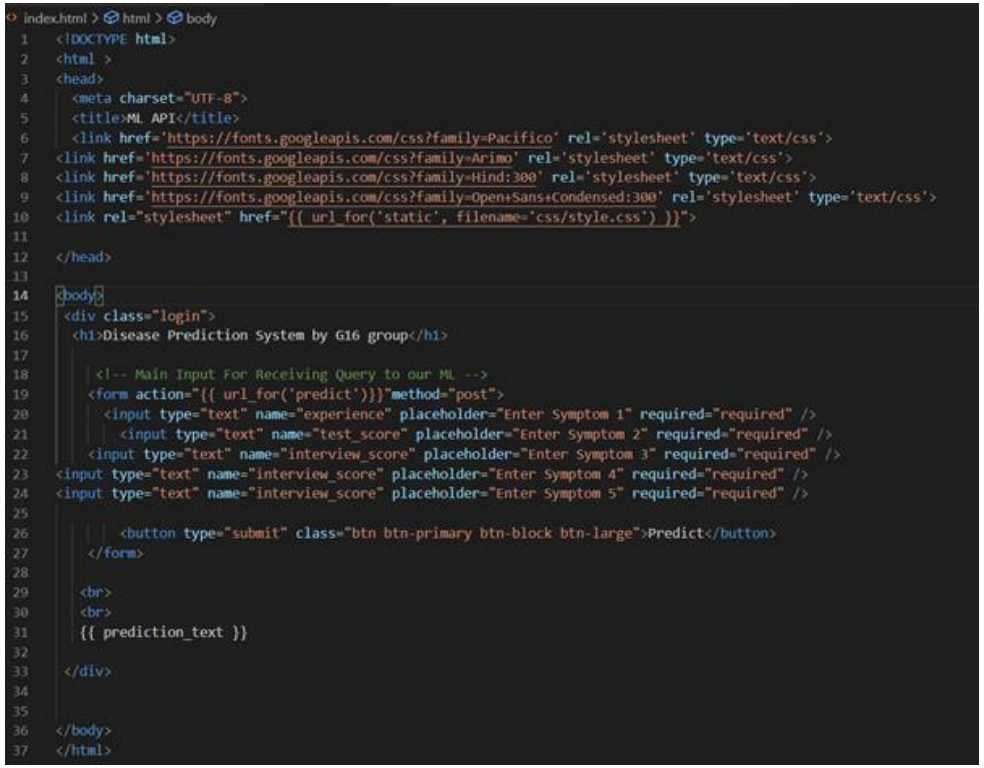
FORM FOR SUBMISSION OF USER DETAILS



FLASK

- **What is Flask?**
  Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher.
- **Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application.**
- **A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc.**
- **Flask is based on WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine.**

FRONT END PART FOR DISEASE PREDICTION MODEL:



```
index.html > html > body
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">
  <title>ML API</title>
  <link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>
<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>
<link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">

</head>

<body>
 <div class="login">
  <h1>Disease Prediction System by G16 group</h1>

    <!-- Main Input For Receiving Query to our ML -->
    <form action="{{ url_for('predict')}}"method="post">
      <input type="text" name="experience" placeholder="Enter Symptom 1" required="required" />
        <input type="text" name="test_score" placeholder="Enter Symptom 2" required="required" />
    <input type="text" name="interview_score" placeholder="Enter Symptom 3" required="required" />
<input type="text" name="interview_score" placeholder="Enter Symptom 4" required="required" />
<input type="text" name="interview_score" placeholder="Enter Symptom 5" required="required" />

        <button type="submit" class="btn btn-primary btn-block btn-large">Predict</button>
    </form>

   <br>
   <br>
   {{ prediction_text }}

 </div>


</body>
</html>
```

- This file will basically act as our front-end web app so that any request that we give to our our model which 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.

MODEL.PY FILE

```
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'))
```

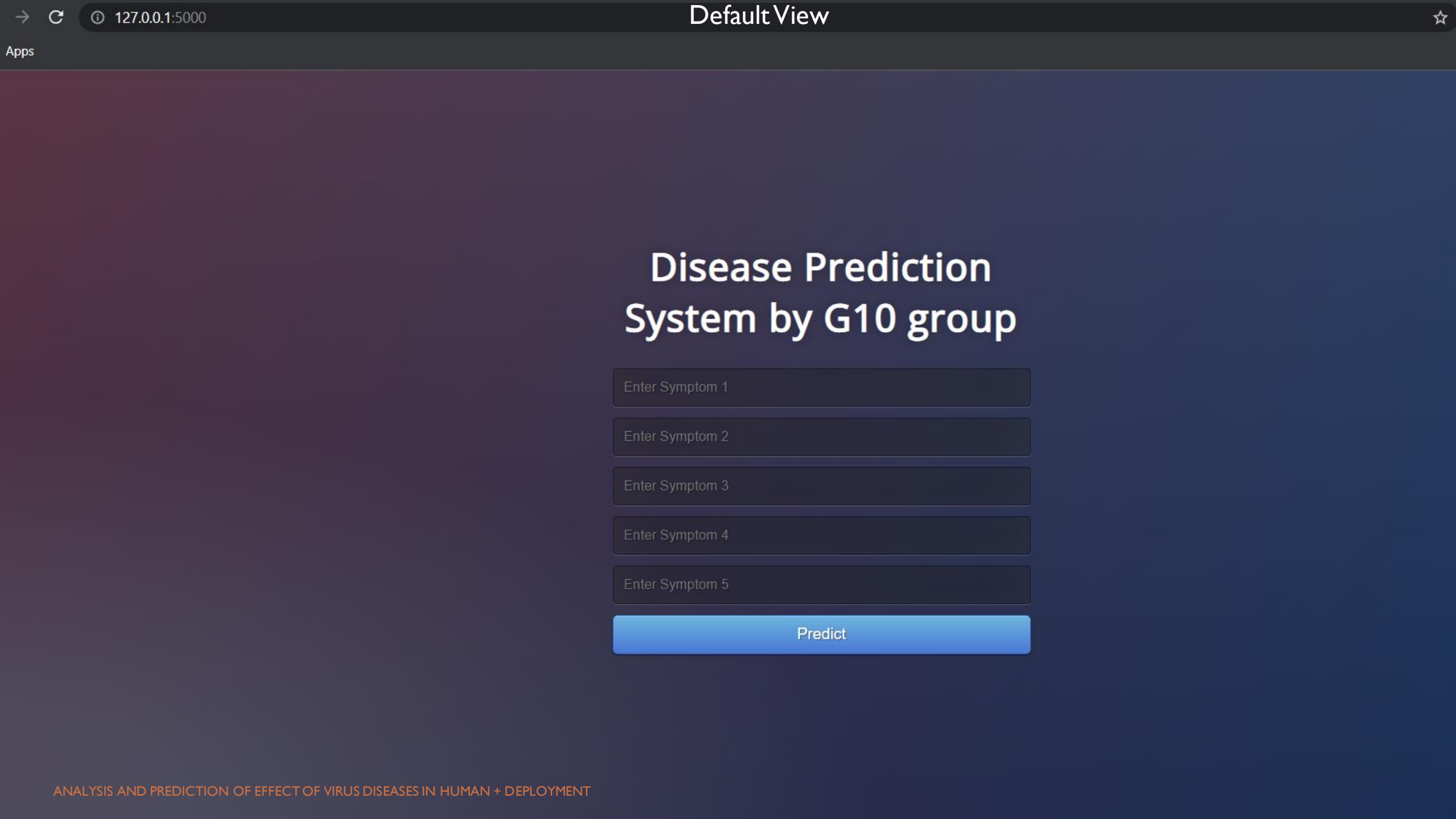
- model.py is the model building file that basically means that this py file will be responsible for creating our model.
- This also involves feature engineering, all the data pre-processing.
- After doing a fit our model will be ready after that we have just used pickle.dump this pickle is basically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

APP.PY FILE

```
@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)]
```

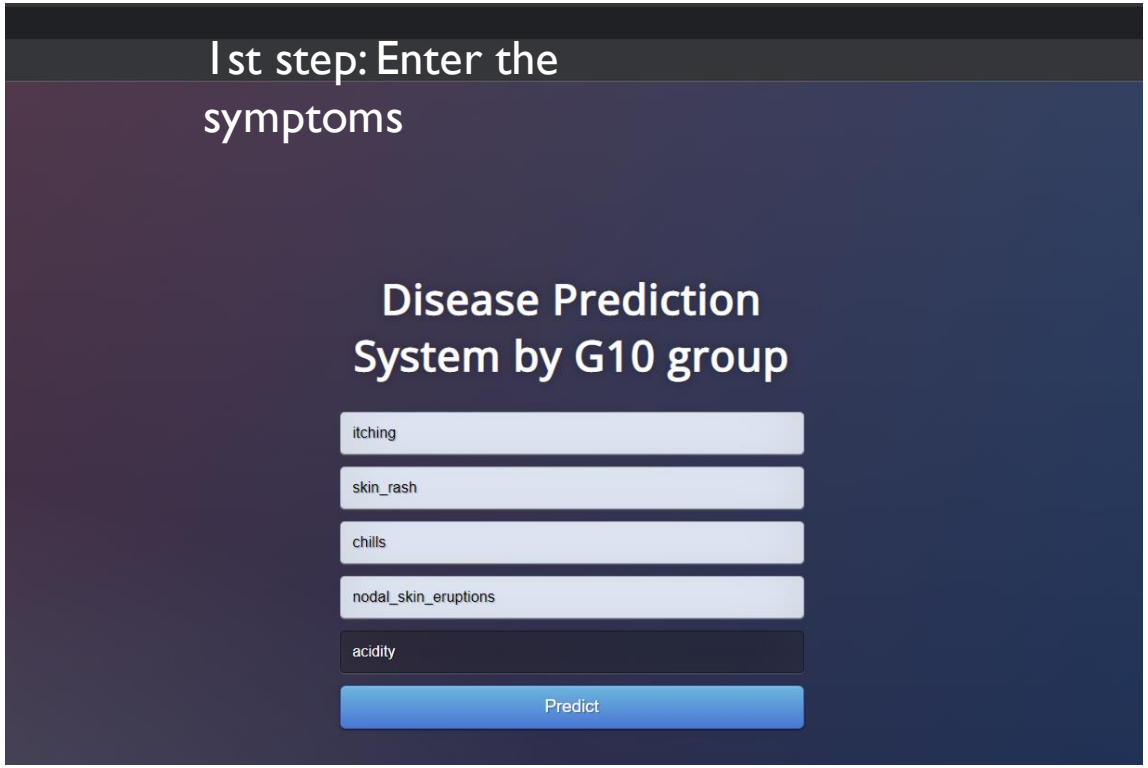
- When the predict function executes then the five symptoms that we have provided as user input these inputs will be read from the form using request.form.values().
- Since it is a post request and after that we have converted these inputs to array and store them in psymptoms.
- We have used model.predict 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 in our webpage.

Default View

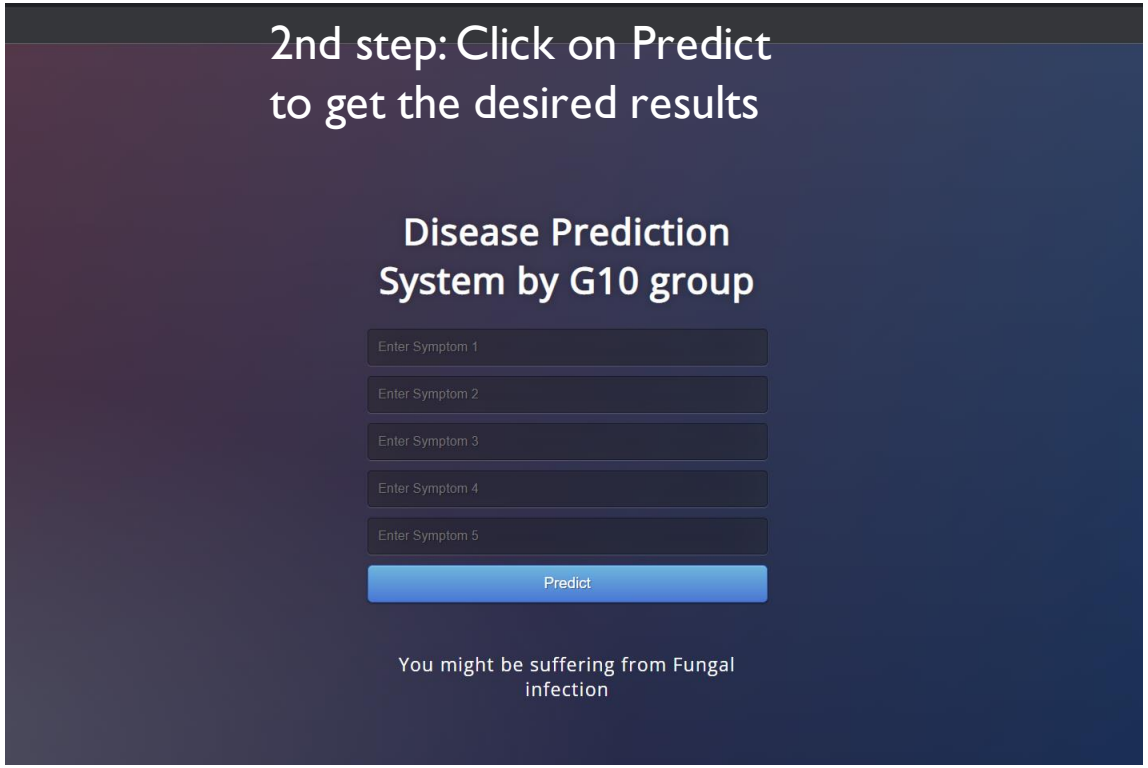


WORKING FLOW:

1st step: Enter the symptoms



2nd step: Click on Predict to get the desired results



THANKYOU