Final Semester Project :

Covid-19 Predictions (A Machine Learning Model)

Supervised by

Assistant Professor Abdul Qudus

Institute of Information Technology

QAU, Islamabad

Submitted By:

Bisma Ishfaq

Registration No:01161911006

MSc-IT. 4th Semester

Statement Of Submission

This is to certify that, I am**Bisma Ishfaq** S/o**Muhammad Ishfaq**, underregistration no **01161911006** at Information Technology Department, Quaid-e-Azam University Islamabad. I declare that I have completed my FYP project under the supervision of Mr. Abdul Qudus Abbasi in May 2021.

Date: 22-04-2021 Signature: _____________

Internal Examiner **External Examiner**

Mr. Abdul Qudus Abbasi
Assistant Professor
Institute of Information Technology,
Quaid-Azam University Islamabad

بسم الله الرحمن الرحيم

In the name of Allah the Most Gracious and the most Merciful

ACKNOWLEDGEMENT

First of all I would like toexpress my sincere and humble gratitude to Almighty Allah whose blessings and guidance has been a real source of all the achievements in my life. I am thankful to my parents, sisters and friends fortheir prayers and moral and financial support during my educational career.

Special thanks to my supervisor **Abdul QudusAbbasi** whose timely suggestions and guidance made this achievements on time. Thanks are extended to my Seniors **Mr. Hamza Riaz ,Mr. BaseerAhmed** for their guidance and then for their kind help and guidance. I am also thankful to my class fellows for their enjoyable company , care and concerns their persistent support and encouragement in my studies.

Bisma Ishfaq ,April 2021

Abstract

The current destructive pandemic of coronavirus disease 2019 caused by severe acute respiratory syndrome coronavirus 2 was first reported in Wuhan china, in December 2019. COVID-19 has affected more than 100 countries in a matter of no time. In such a situation , forecasting and proper study of pattern of disease spread can inspire design better strategies to make more efficient decision.

Technology advancement have a rapid effect on every field of life. Artificial intelligence has shown the promising results in healthcare through its decision making by analysis the data. Machine learning requires a huge amount of data for classifying or predicting diseases . Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Covid-19 Prediction(A Machine Learning Model) is basically designed to facilitate nation. People can get knowledge through this system about the current disease. They can predict cases by easily login to the website.

This system consists of two parts. In First part we get dataset from Kaggle website and after preprocessing of data I did different analysis.Data analysis allows analysts, researchers, and business users to make better and faster decisions using data that was previously inaccessible or unusable. Data Science is an interdisciplinary field that combines machine learning, statistics, advanced analysis, and programming. There are two type of analytics. Predictive analytics and Descriptive analytics. Prescriptive analytics is an advanced analytics technology that can provide recommendations to decision makers and help them achieve business goals by solving complicated optimization problems. Predictive analytics brings together advanced analytics capabilities spanning ad-hoc statistical analysis, predictive modeling, data mining, text analytics, optimization, real-time scoring and machine learning. These tools help organizations discover patterns in data and go beyond knowing what has happened to anticipating what is likely to happen next.

In Analysis I use different graphs which helps user to get information in just one look. After This I tried different models for my data.

In second part, I get data from the trained model and store in pickle file. Then this data is used to predict cases. User get (seen) this data on their screen through Flask. Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

I develop a website with the help of Javascript and Bootstrap which consists of different pages. User can browse to these pages and get their desired data about Covid_19.

I have also used MYSQL for storing data. I have attach database to my website which store the data entered by user into database for future use.

Table of Contents

Chapter No1

Problem Definition

Corona Virus are a large family of viruses that cause illness ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV).

Common symptoms of infection include respiratory difficulties, fever, cough, shortness of breath and breathing difficulties. In more complicated cases, infection may lead to pneumonia, severe acute respiratory syndrome, kidney failure and even death. The continuous rise in the spread of the virus has continue to pose great setback to Italy and the entire globe. The difficulties in containing the spread of the virus across localities is challenging and required drastic measures.

1.1 Background:

If we take a look at history comprising of 100 years, the Flu virus has been affecting mankind over a period of the past many decades resulting in a heavy death toll globally. The Flu virus strain(a genetic variant of the virus) which we have come across today has a specific name **Novel Corona Virus or COVID-19**.

Since the outbreak of the Coronaviruses (COVID_19) continues to spread across countries like Pakistan, defiling all the measures put in places by government and individuals to curb it. Corona Virus are a large family of viruses that cause illness ranging from the common cold to more severe diseases such as Middle East Respiratory Syndrome (MERS-COV) and Severe Acute Respiratory Syndrome (SARS-COV). Common symptoms of infection include respiratory difficulties, fever, cough, shortness of breath and breathing difficulties. In more complicated cases, infection may lead to pneumonia, severe acute respiratory syndrome, kidney failure and even death. The continuous rise in the spread of the virus has continue to pose great setback to Pakistan and the entire globe. The difficulties in containing the spread of the virus across localities is challenging and required drastic measures.

1.2 Problem statement:

As the Covid_19 disease is increasing day by day, and vaccine of this disease is not proved to be very effective. It cause several deaths day by day. Therefore it is mandatory to provide a System that is used to predict the new arrived cases.

Technology advancement have a rapid effect on every field of life. Artificial intelligence has shown the promising results in healthcare through its decision making by analysis the data. Machine learning can play important role to forecast the nature of virus across the globe.

1.3Purpose:

Prevention is better that Cure.Just as vaccination is given to a child to prevent disease, now a days with such higher COVID cases happenings, it has become necessary to have prediction systems that prevent from such disease. We use Machine Learning techniques for development of my system. Machine Learning has various tools that can be used for visualization and prediction of Active cases..

1.4Objectives:

Following are the objectives of this project:

- Collect historical COVID_19 disease data
- Choose a Machine Learning algorithm best for regression problems
- Train the prediction model on historical data
- Test the model on test data
- Model should predict with best accuracy

1.5Proposed Solution:

In order to predict this COVID_19 disease, we will make a prediction model and train it on historical COVID data and will test it on test data in order to get higher accuracy..The end user can easily predict the COVID 19 Activecases on providing required criteria for prediction.

Chapter No 2

System Requirements

2.1 Functional Requirements:

Functional requirements are those functionalities that software/project should perform. Functional requirements specify the work of system that it performs. These requirements are necessary to run a system smoothly and make them functional.

The new system will be web base application which will allow all users to enter specific date andcity and province) to get the predict results. It should be possible to generate various types of reports from system.

2.2 Non-Functional Requirements

Software requirements specifications included in the non-functional requirements of Covid_19 predictions, which contains various process, namely Security, Performance, Maintainability, and Reliability.

2.2.1 Security:

System should be such that it is secure and does not contain any intruder action or virus that will affect our system.

System should be secure so that no any disallowed person can modify and change any system data.

2.2.2 Modifications:

As the data of Covid -19 is updated every month , so system should be such that it accommodate any modification in data that the admin want to change.

2.2.3 Performance:

- **Response Time:** The system should provide acknowledgment in just one second once the information is checked.
- **Capacity:** The system should needs to support at least 1000 people at once.
- .**User-Interface:** The user interface should acknowledge within five seconds.
- **Conformity:** The system should need to ensure that the guidelines of the Microsoft accessibilities are followed.

2.2.4Maintainability:

- **Back-Up**: The system should offer the efficiency for data backup.

2.2.5 Reliability:

- **Availability**: The system should be available all the time.

2.3 Hardware Requirements

Following are the resources which are used to complete the project:

2.3.1 Hardware Resources

- Personal Computer
- 8GB RAM (or more)
- 565GB Hard Disk or more
- Intel® Core™ m3-7Y30 @1.00GHz 1.61 GHz Processor

2.4 Software Requirements

Following are the resources which are used to complete the project:

2.4.1 Software Resources

Resource TypeResource Name

Operating System Windows (10)

Languages Python (3.8.5)

Javascript

Bootstrap

ComplierJupyterNotebook(6.1.4)
Spyder(4.1.5)

Environment Anaconda (4.9.2)
Visual Studiocode(1.53.2) user setup

2.5 Communications Interfaces

The Customer must connect to the Internet to access the Website:

- Dialup Modem of 52 kbps
- Broadband Internet
- Dialup or Broadband Connection with Internet Provider.

2.6 User Requirements

The user requirements for this system are to make the system flat, flexible, less prone to error, reduce expenses and save time.

- Time can be saved by online prediction that can be performed by click of button.
- A facility to see run time graphs which are changing with time.

2.7 User Interface

Application will be accessed through a Browser Interface. The interface would be viewed best using 1024 x 768 and 800 x 600 pixels resolution setting. The software would be fully compatible with Microsoft Internet Explorer for version 6 and above. No user would be able to access any part of the application without logging on to the system.

Chapter 3

Tools and Technologies

This project has wide areas of concern, so it requires a bunch of tools and technologies. In this chapter, we will discuss all the respective technologies used for accomplishment of this project.

3.1 Programming Languages:

This project requires a number of languages to work on. Like some of the libraries are written in a separate language and may be used in some other language. These are described below:

3.1.1 Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

Some distinguishable characteristics of python are:

3.1.1.1 Advantages of Python

- **Python is Interpreted**

  Python is processed at runtime by the interpreter. User did not require to compile program before execution. Python executes code line by line. In case of any error it stops the execution and report back the error which has occurred.

- **Python is Interactive**

You can sit at a Python prompt and interact with the interpreter directly to write program.

- **Python is Object-Oriented**

Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

- **Python is a Beginner's Language**

Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to gaems.

- **Python is fast and popular**

It includes an internal standard library that is called “BETTERIES INSIDE” among Python lovers. It provides all facilities that are needed for programming from the basic operations to advanced functions.The third party tools make everything possible in Python.For example, by writing just 3 lines of codes you can create a web server.

- **Python support other Technologies**

It can support COM, .Net etc objects. Also, some alternatives and complements were created for python that make it easier to work with these objects in an integrated mode.

- **Python is portable**

Python scripts can be used on different operating system such as Windows,Linux,UNIX ,Amigo Mac OS etc.Also, some versions were released to work on .Net, Java Virtual Machine and Nokia S60. It is interesting to know that output of programs is similar on all these platforms.

- **Python is Simple**

It is a very high-level language that has many sources for learning. Also, wide variety of third party tools makes this language sweet and easy to use and motivates the users to continue with. 5.2.5 Python Is Open Source Even though all rights of this program are reserved for the Python institute, but it is open source and there is no limitation in using, changing and distributing.

3.1.1.2 Disadvantages of Python

Following are some disadvantages of python language:

- **Get Slow in speed**

Python executes with the help of an interpreter instead of compiler, which causes it to slow down because it executes code line by line which is a slow process.

- **Run-time Errors**

The python language is dynamically typed so it has many design restrictions that are reported by some python developers. It requires more testing time, and the error show up when the application are finally run.

- **Weak in Mobile Computing:**

Python has made its presence on many desktop ad server platforms, but it is seen as a weak language for mobile computing.

- **Python has high memory consumption**

For more intensive tasks, python is not always the best choice. The memory consumption of python is high due to flexibility of datatype.

For large and long-running systems developed using python, dealing with memory management is difficult.

3.1.2 HTML:

HTML is a markup language that defines the structure of your content.HTML consists of a series of elements which we use to enclose ,or wrap different parts of the content to make it appear a certain way ,or act a certain way.HTML elements tells the browser how to display the content.

3.1.3 Javascript

Javascript is the programming language for the web. Javascript can update and change both HTML and CSS. Javascript is a text based programming language use both on client side and server side that allows you to make web pages interactive.

3.1.4 CSS

Cascading Style Sheets (**CSS**) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

3.1.5 Bootstrap

Bootstrap is the most popular **CSS Framework** for developing responsive and mobile-first websites.**Bootstrap 4** is the newest version of Bootstrap.

3.2 Libraries:

Libraries are sets of routines and functions that are written in a given language. A robust set of libraries can make it easier for developers to perform complex tasks without rewriting many lines of code. Machine learning is largely based upon mathematics. Specifically, mathematical optimization, statistics and probability. Python libraries help to easily "do machine learning".

Following are the libraries which we will use in our project:

3.2.1 Numpy:

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

3.2.2 Pandas:

Pandas is a fast, powerful, flexible, and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

3.2.3 Scipy

Scipy is an open source python library that is used for both scientific and technical computation.it is a free python library and suitable for machine learning.it is also very popular for image manipulation as well.

3.2.4 Matplotlib:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension Numpy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

3.2.5XGBoost:

XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solvemany data science problems in a fast and accurate way.

3.2.6 Seaborn

Seaborn is a data visualization library built on the top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of seaborn which helps in exploration and understanding of data.

3.2.7Scikit-learn:

Scikit-learn is an open-source Python machine learning library which provides numerous classification, regression and clustering algorithms. This library was used in this project to perform the actual task of model building and prediction. It provides a variety of evaluation metrics to validate the performance of the model, which makes it a valuable tool.

3.3 Open-Source Distribution:

A useful software, which we will use in our project, is:

3.3.1 Anaconda:

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. Package versions are managed by the package management system conda.

3.4 Integrated Development Environments (IDEs):

An integrated development environment is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger.

IDEs, which we will use in our project, are as follows:

3.4.1Jupyter Notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

3.4.3 Visual Studio Code:

Visual Studio Code is a free source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

3.4.4 Tableau:

Tableau is a powerful data analytics tool which is used for building interactive dashboards. Tableau was mainly used in the project to generate interactive graphs and observe patterns in the data. This information proved to be useful in determining the features that could contribute well to the actual model building. It also provides a rich map interface for geographical data.

3.5 Web Framework

3.5.1 Flask

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

CHAPTER NO 4

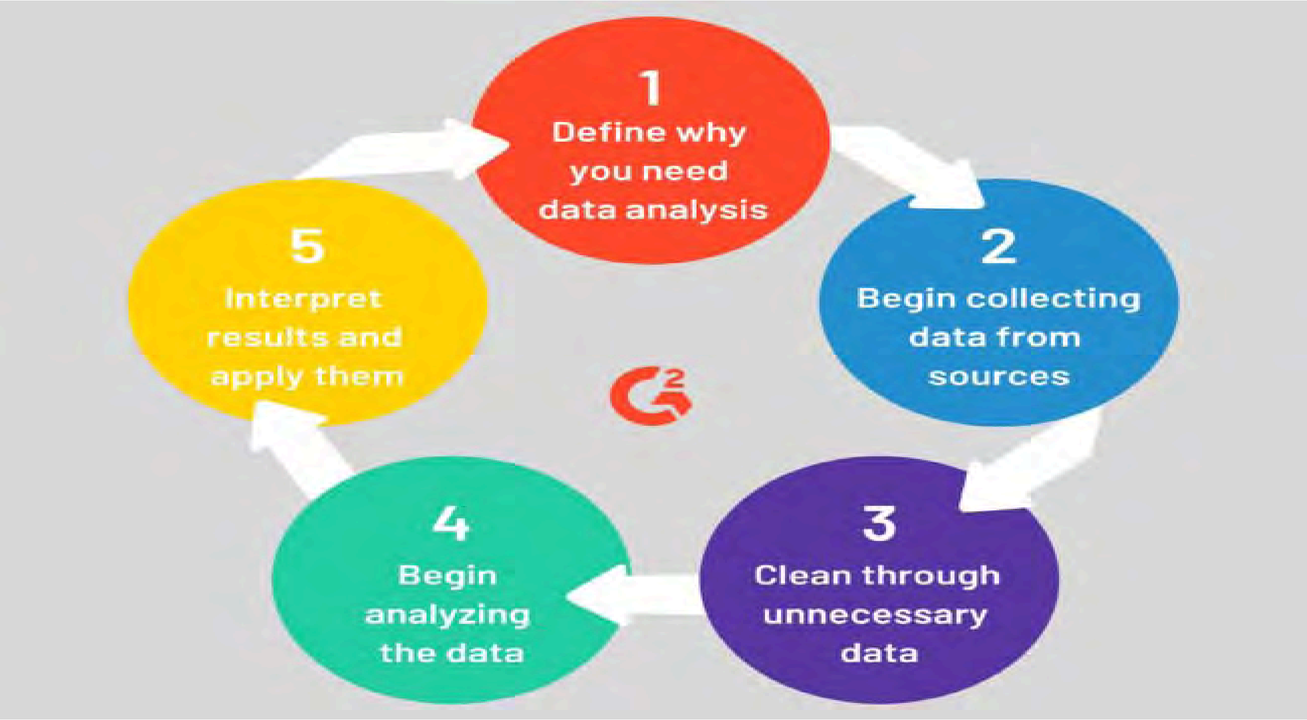
Data analysis

4.1 Data Analysis

4.1.1 What is Data Analysis

Data analysis is defined as a process of cleaning, transforming, and modeling data to discover useful information for business decision-making. The purpose of Data Analysis is to extract useful information from data and taking the decision based upon the data analysis.

A simple example of Data analysis is whenever we take any decision in our day-to-day life is by thinking about what happened last time or what will happen by choosing that particular decision. This is nothing but analyzing our past or future and making decisions based on it. For that, we gather memories of our past or dreams of our future. So that is nothing but data analysis. Now same thing analyst does for business purposes, is called Data Analysis.


1
Define why you need data analysis
2
Begin collecting data from sources
3
Clean through unnecessary data
4
Begin analyzing the data
5
Interpret results and apply them

4.1.2 Why we analyze data

Data analysis is an internal arrangement function done by data analysts through presenting numbers and figures to management. It involves a more detailed approach in recording, analyzing, disseminating, and presenting data findings in a way that is easy to interpret and make decisions for the business.

With data analysis you will be able to make decisions on customer trends and behavior prediction, increasing business profitability, and drive effective decision-making.

4.1.3 Techniques and Methods

There are several **types of Data Analysis** techniques that exist based on business and technology. However, the major Data Analysis methods are:

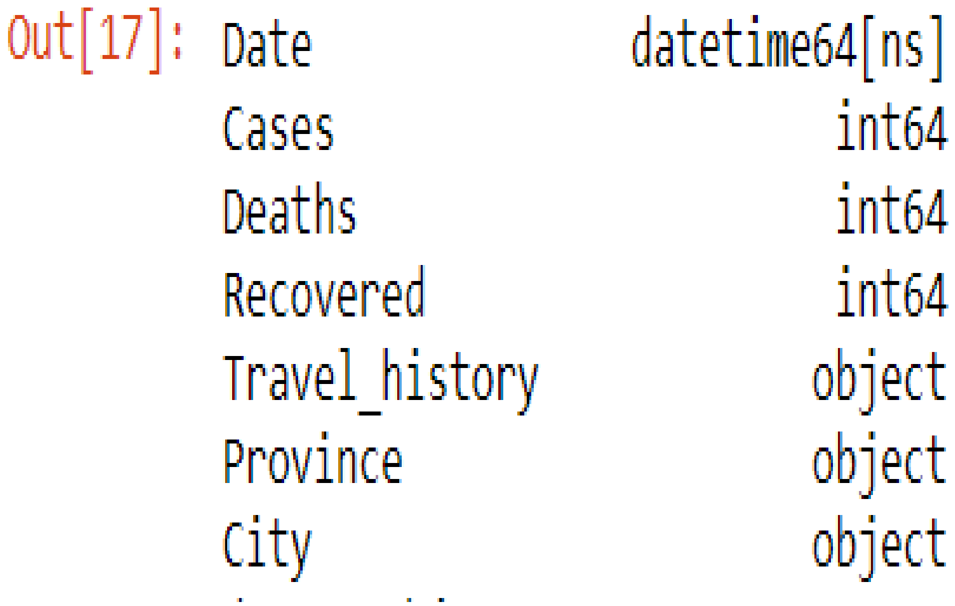
- Text Analysis
- Statistical Analysis
- Diagnostic Analysis
- Predictive Analysis
- Prescriptive Analysis

4.2 Dataset:

The data used in this research project is the COVID_19 dataset made available by the Kaggle website which is a part of the open data initiative. This dataset contains cases of COVID_19 which include confirmed ,Active,Recovered and Deaths cases of three months. The data ranges from 2/26/2020 to 6/3/2020. The dataset consists of the following attributes:

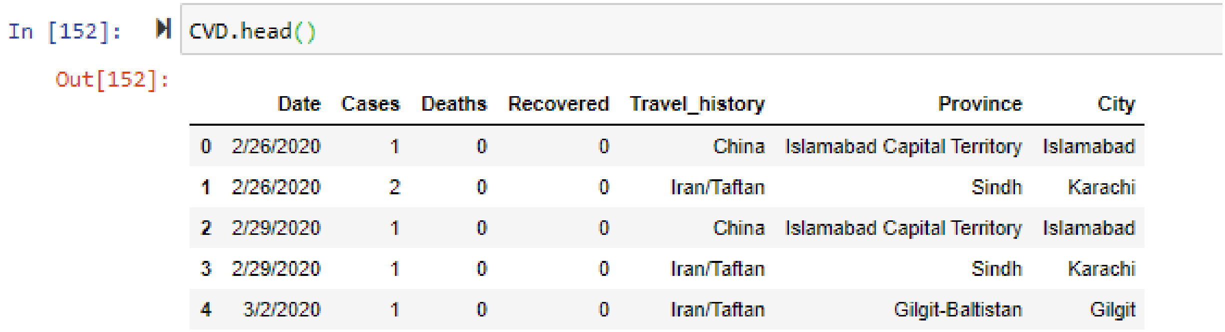
- Date: It is a numerical field. Specifies the date when the cases occurred.
- Province: It is a text field. It represent all provinces of Pakistan whether cases occurred in all provinces or not. It mention total 7 Provinces.
- City: It is a text field. It represents all the cities if Pakistan where COVID_19 cases happened.
- Confirmed cases: It represent confirmed cases which occur in particular date and in particular Province and city.
- Active: It represent Active cases which occur in particular date and in particular Province and city in all over country.
- Recovered cases: It represent total Recovered cases that has been recovered on particular date.
- Deaths cases: It represent Deaths cases which occur in particular date and in particular Province and city.
- Travel_history: It is a text field. It indicates whether the person who is infected has any travel history or not..

There are about 2800 rows in the dataset and the size of the dataset is approximately 5MB. It contains data from the month (FEB) to (MAY) 2020. A snapshot of the column’s is shown in following figure.



```
Out[17]: Date                datetime64[ns]
         Cases                        int64
         Deaths                       int64
         Recovered                    int64
         Travel_history              object
         Province                    object
         City                        object
```

DisplayFirst 5 Records of Dataset

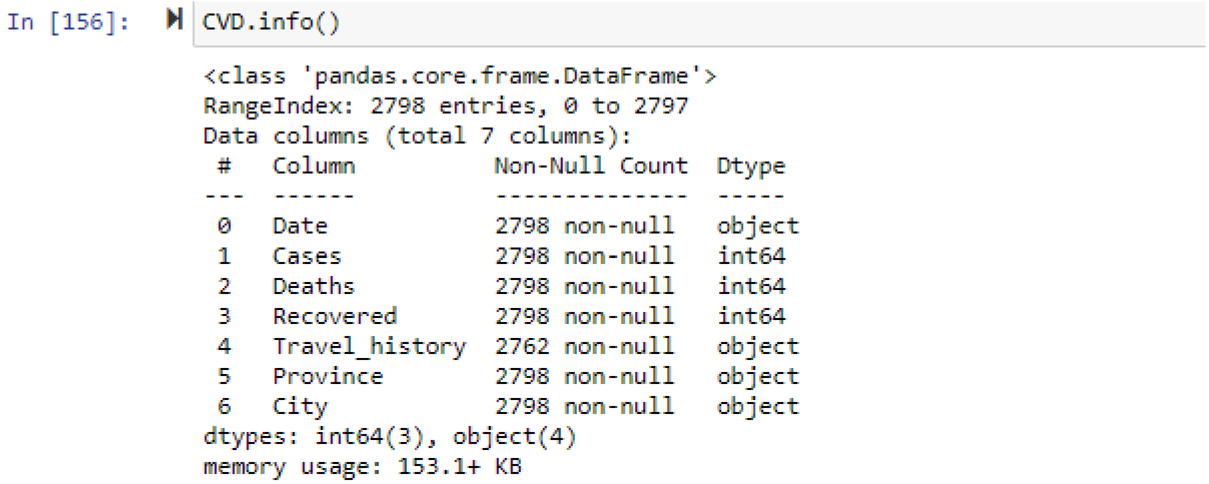


```
In [152]:  CVD.head()
```

Out[152]:

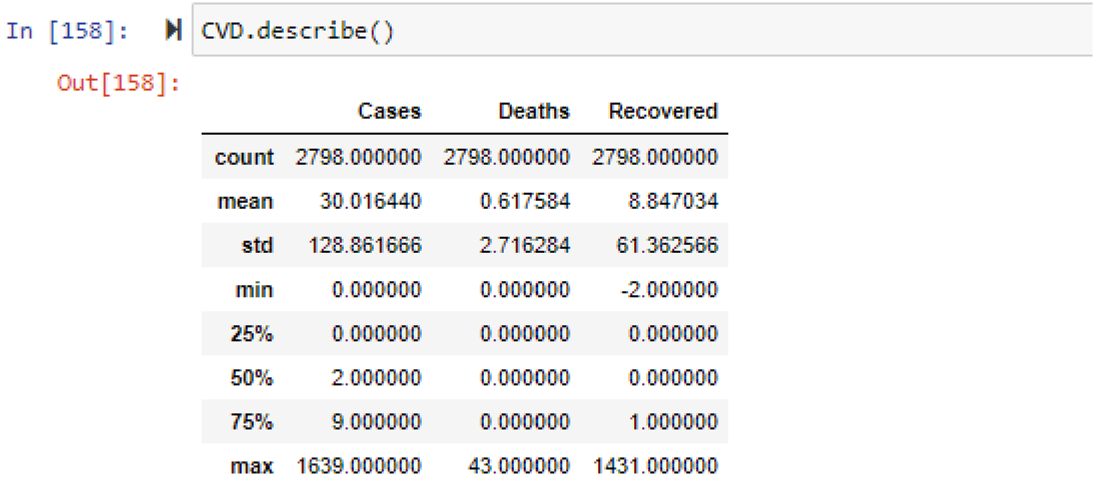
	Date	Cases	Deaths	Recovered	Travel_history	Province	City
0	2/26/2020	1	0	0	China	Islamabad Capital Territory	Islamabad
1	2/26/2020	2	0	0	Iran/Taftan	Sindh	Karachi
2	2/29/2020	1	0	0	China	Islamabad Capital Territory	Islamabad
3	2/29/2020	1	0	0	Iran/Taftan	Sindh	Karachi
4	3/2/2020	1	0	0	Iran/Taftan	Gilgit-Baltistan	Gilgit

Statistics of Dataset



```
In [156]:  CVD.info()

           <class 'pandas.core.frame.DataFrame'>
           RangeIndex: 2798 entries, 0 to 2797
           Data columns (total 7 columns):
            #   Column          Non-Null Count  Dtype 
           ---  ------          --------------  ----- 
            0   Date            2798 non-null   object
            1   Cases           2798 non-null   int64 
            2   Deaths          2798 non-null   int64 
            3   Recovered       2798 non-null   int64 
            4   Travel_history  2762 non-null   object
            5   Province        2798 non-null   object
            6   City            2798 non-null   object
           dtypes: int64(3), object(4)
           memory usage: 153.1+ KB
```

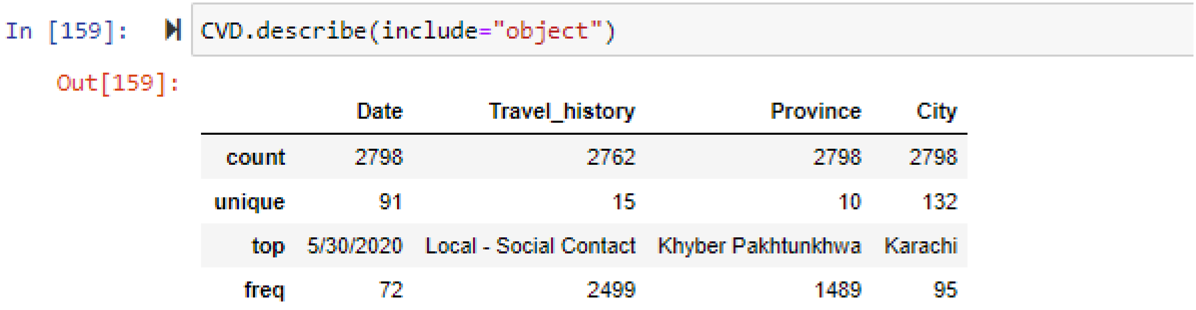
Statistics of Numerical columns



```
In [158]:  CVD.describe()
Out[158]:
```

	Cases	Deaths	Recovered
count	2798.000000	2798.000000	2798.000000
mean	30.016440	0.617584	8.847034
std	128.861666	2.716284	61.362566
min	0.000000	0.000000	-2.000000
25%	0.000000	0.000000	0.000000
50%	2.000000	0.000000	0.000000
75%	9.000000	0.000000	1.000000
max	1639.000000	43.000000	1431.000000

Statistics of Categorical columns



```
In [159]:  CVD.describe(include="object")
Out[159]:
```

	Date	Travel_history	Province	City
count	2798	2762	2798	2798
unique	91	15	10	132
top	5/30/2020	Local - Social Contact	Khyber Pakhtunkhwa	Karachi
freq	72	2499	1489	95

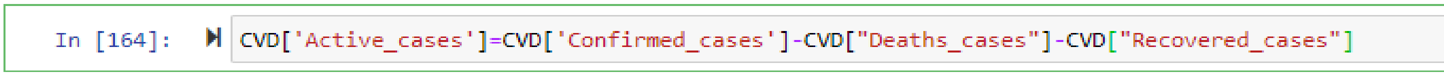
4.3 Data Preprocessing(Cleaning):

The first step to develop machine learning model is Data Preprocessing (Data Cleaning).In data preprocessing we perform following steps.

Addition of Active cases columns:

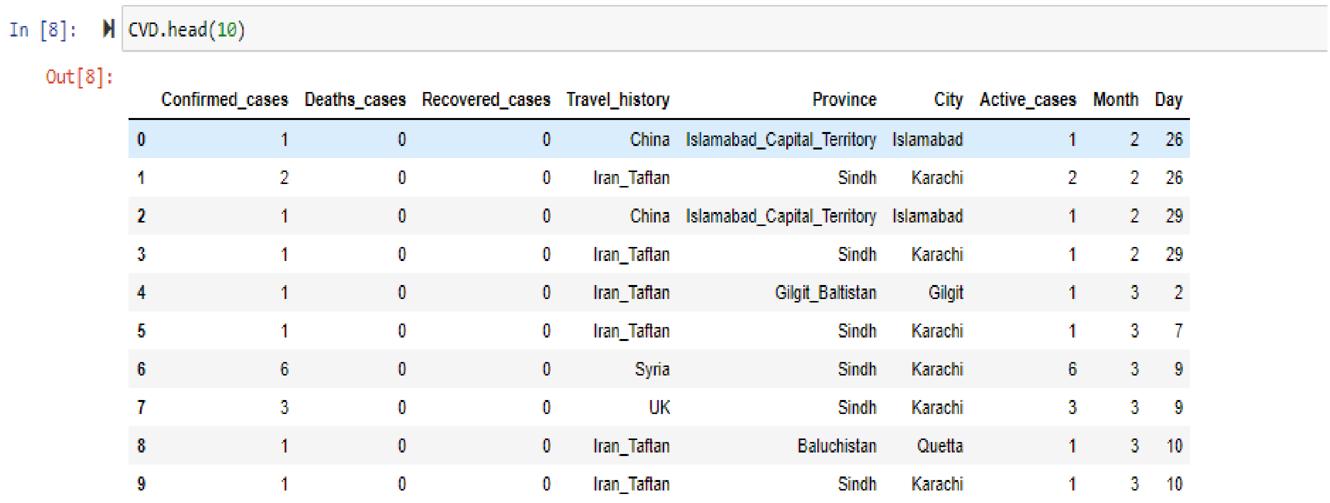
As we know that active cases also play an important role in prediction of cases so we add one columns which is derived from other 3 columns of the dataset**.**

Code to drive column (Active_cases) are as follows:



```
In [164]:  CVD['Active_cases']=CVD['Confirmed_cases']-CVD["Deaths_cases"]-CVD["Recovered_cases"]
```

Showing firt 10 Records of dataset:



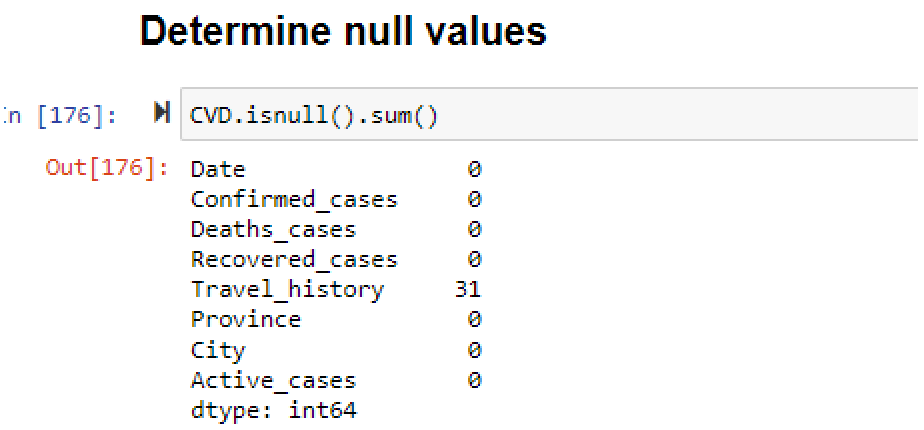
```
In [8]:  CVD.head(10)
```

Out[8]:

	Confirmed_cases	Deaths_cases	Recovered_cases	Travel_history	Province	City	Active_cases	Month	Day
0	1	0	0	China	Islamabad_Capital_Territory	Islamabad	1	2	26
1	2	0	0	Iran_Taftan	Sindh	Karachi	2	2	26
2	1	0	0	China	Islamabad_Capital_Territory	Islamabad	1	2	29
3	1	0	0	Iran_Taftan	Sindh	Karachi	1	2	29
4	1	0	0	Iran_Taftan	Gilgit_Baltistan	Gilgit	1	3	2
5	1	0	0	Iran_Taftan	Sindh	Karachi	1	3	7
6	6	0	0	Syria	Sindh	Karachi	6	3	9
7	3	0	0	UK	Sindh	Karachi	3	3	9
8	1	0	0	Iran_Taftan	Baluchistan	Quetta	1	3	10
9	1	0	0	Iran_Taftan	Sindh	Karachi	1	3	10

Removing any null values:

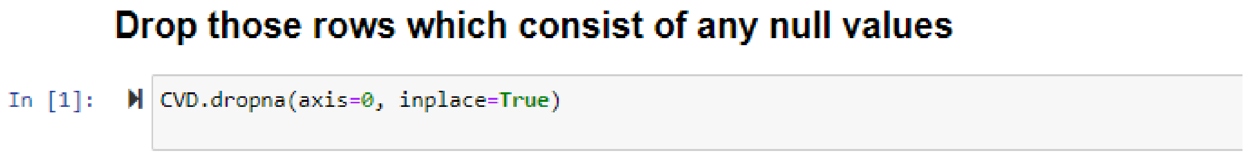
As we can see from below figure that our dataset contain null values so must be removed to avoid any mistakes in training of model.



Determine null values

```
In [176]:  CVD.isnull().sum()

Out[176]: Date                0
          Confirmed_cases     0
          Deaths_cases        0
          Recovered_cases     0
          Travel_history     31
          Province            0
          City                0
          Active_cases        0
          dtype: int64
```

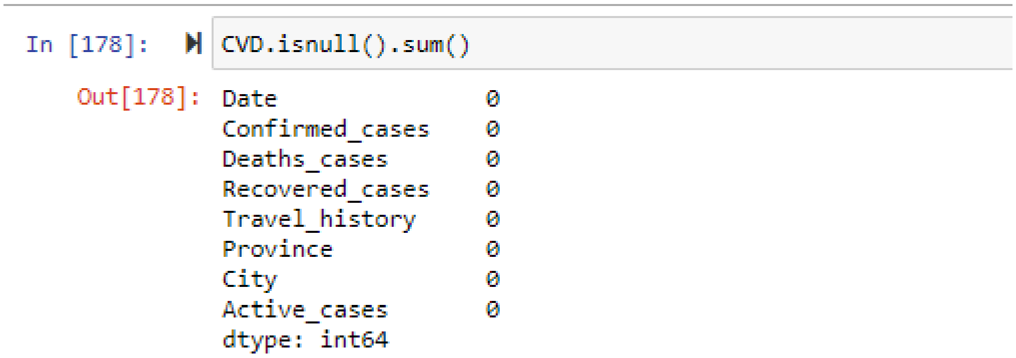
So I drop all those rows which consists of any null values.



Drop those rows which consist of any null values

```
In [1]:  CVD.dropna(axis=0, inplace=True)
```

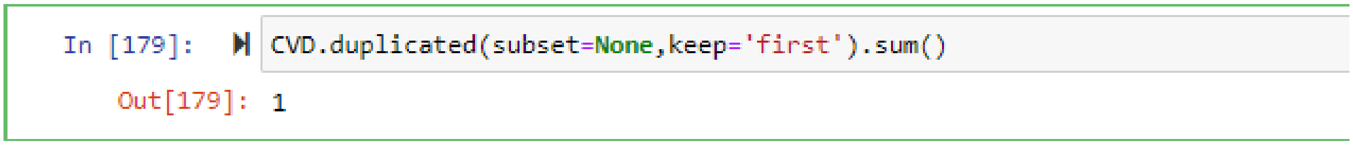
Now after removal of null values we can see our dataset does not contain any null value and it is clean now.



```
In [178]:  CVD.isnull().sum()

Out[178]: Date               0
          Confirmed_cases    0
          Deaths_cases       0
          Recovered_cases    0
          Travel_history     0
          Province           0
          City               0
          Active_cases       0
          dtype: int64
```

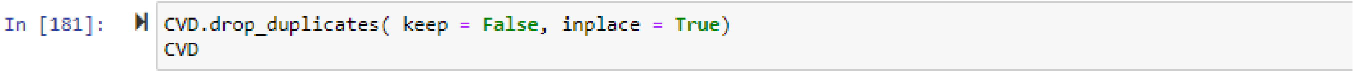
Checking and removal of Duplication of records

Now we check whether our dataset contain any duplicate record if so we must remove it**.**



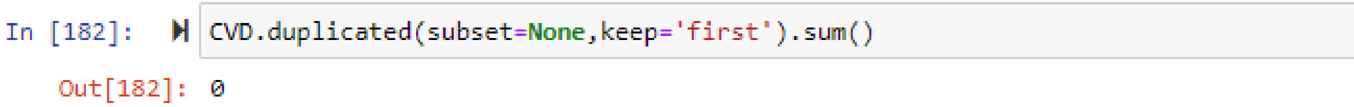
```
In [179]:  CVD.duplicated(subset=None,keep='first').sum()

Out[179]: 1
```

We can see from above figure, we have one record which is duplicated. Now we will remove it.



```
In [181]:  CVD.drop_duplicates( keep = False, inplace = True)
           CVD
```

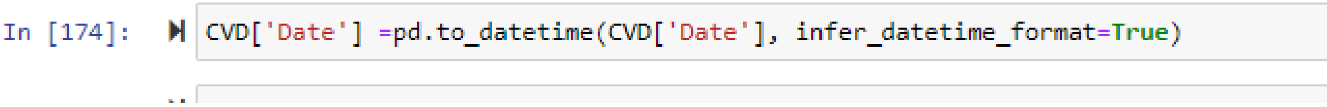
Now we can see we don't have any duplicate row.



```
In [182]:  CVD.duplicated(subset=None,keep='first').sum()

Out[182]: 0
```

Parsing date column

- Parsing the 'Date' Column because the 'Date' column type is String. It will be easier to work with by parsing it to Datetime.

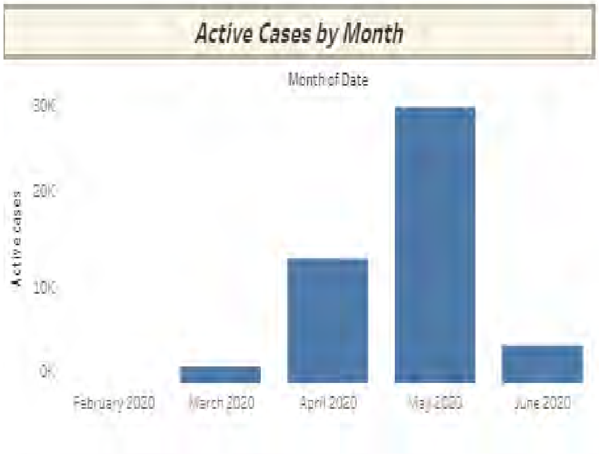


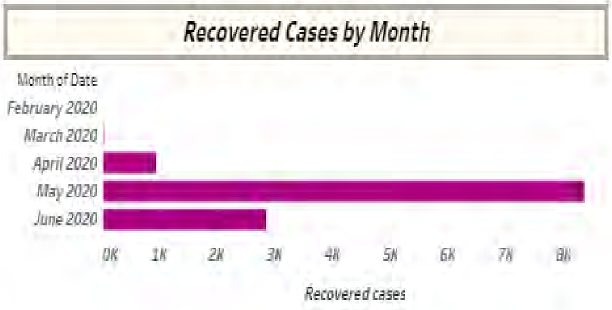
```
In [174]:  CVD['Date'] =pd.to_datetime(CVD['Date'], infer_datetime_format=True)
```

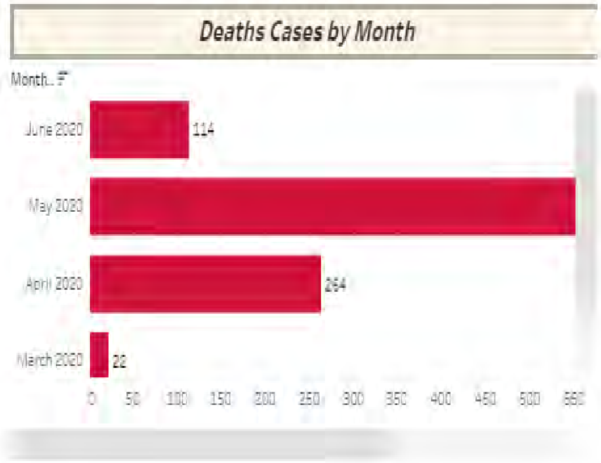
4.4 Analysis And Visualization:

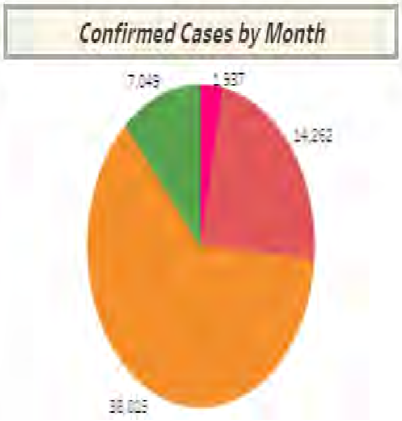
- **Cases by Month:**

Below graph show the increase and decrease of Covid-19 Cases with respect to time change.





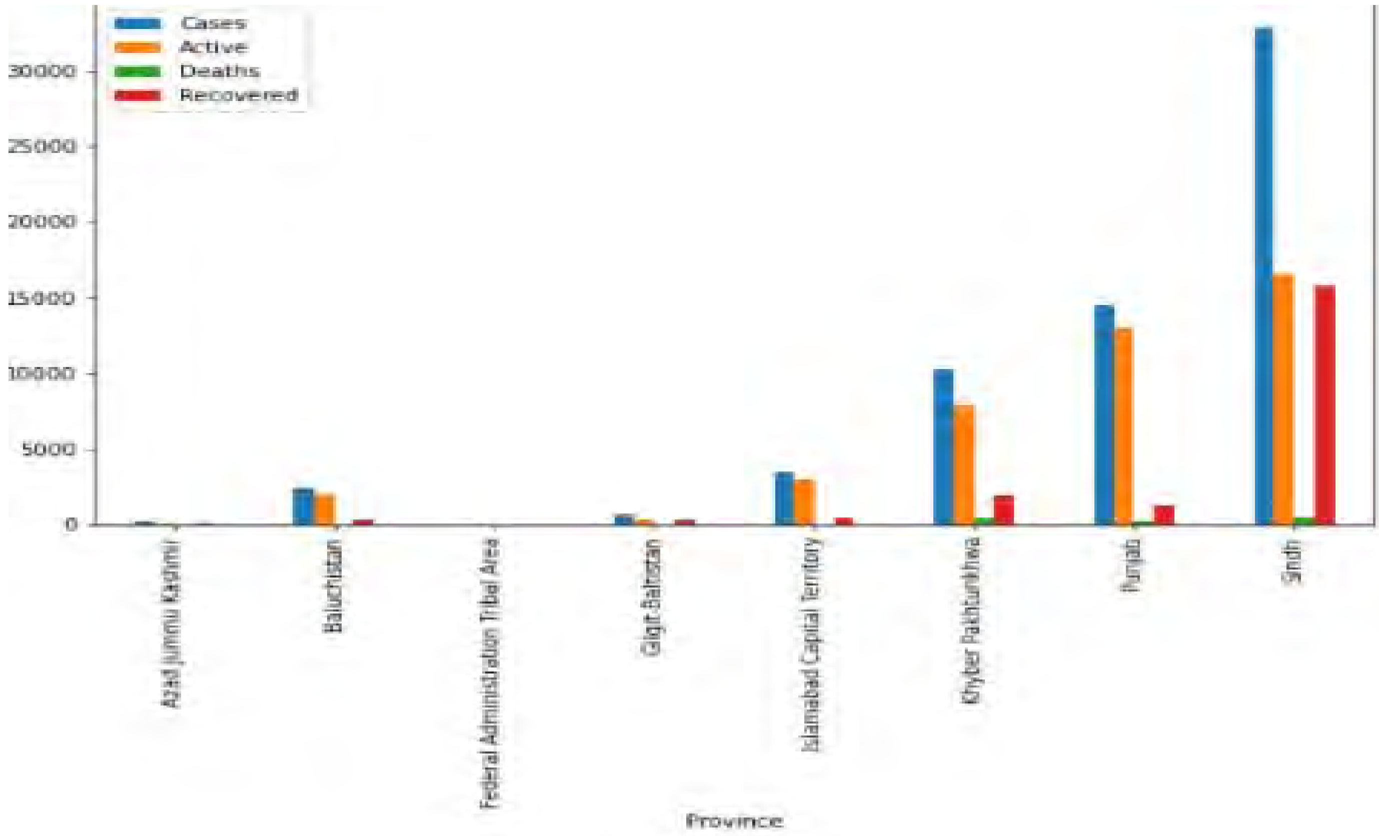




- **Province analysis:**

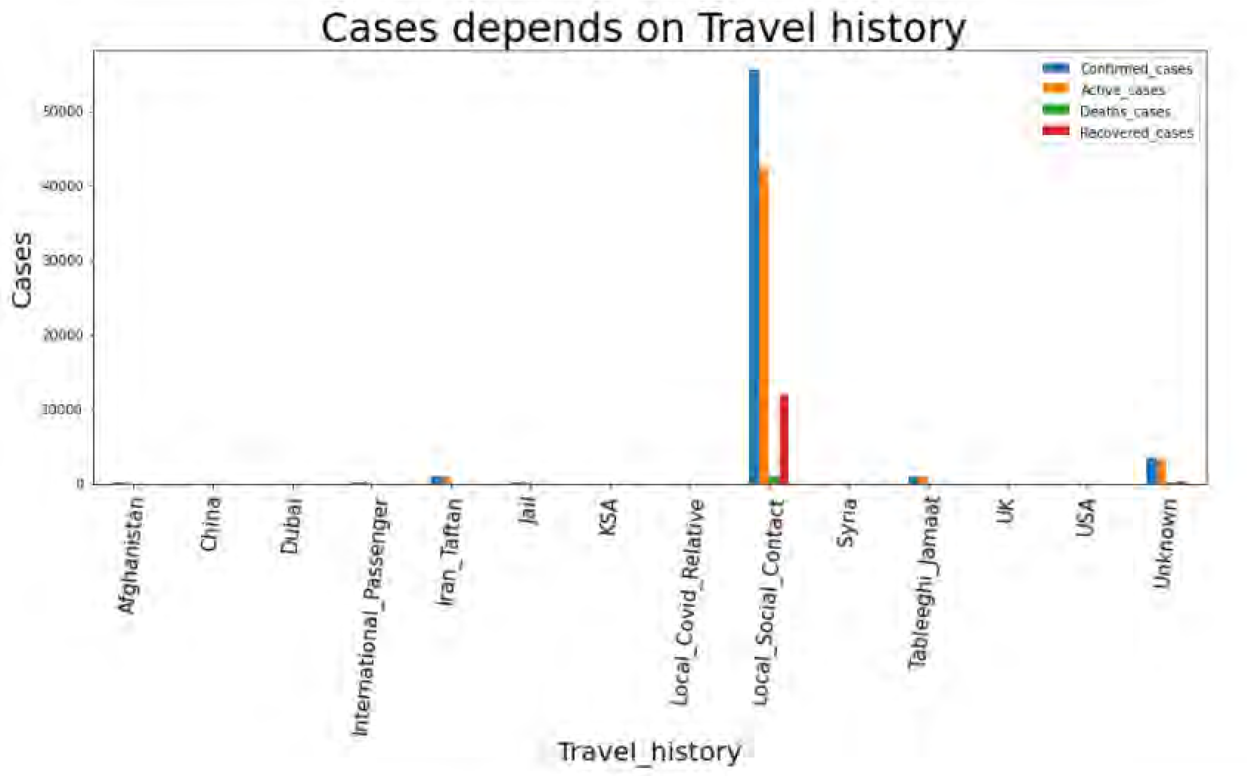
There are total 8 province mentioned in my dataset. Now we will analyze which province has most cases **(**Confirmed, Active, Deaths, Recovered cases).

As we can see from below graph Sindh is most infected province in Pakistan. And (Azad Jammu Kashmir) has least infected patient.



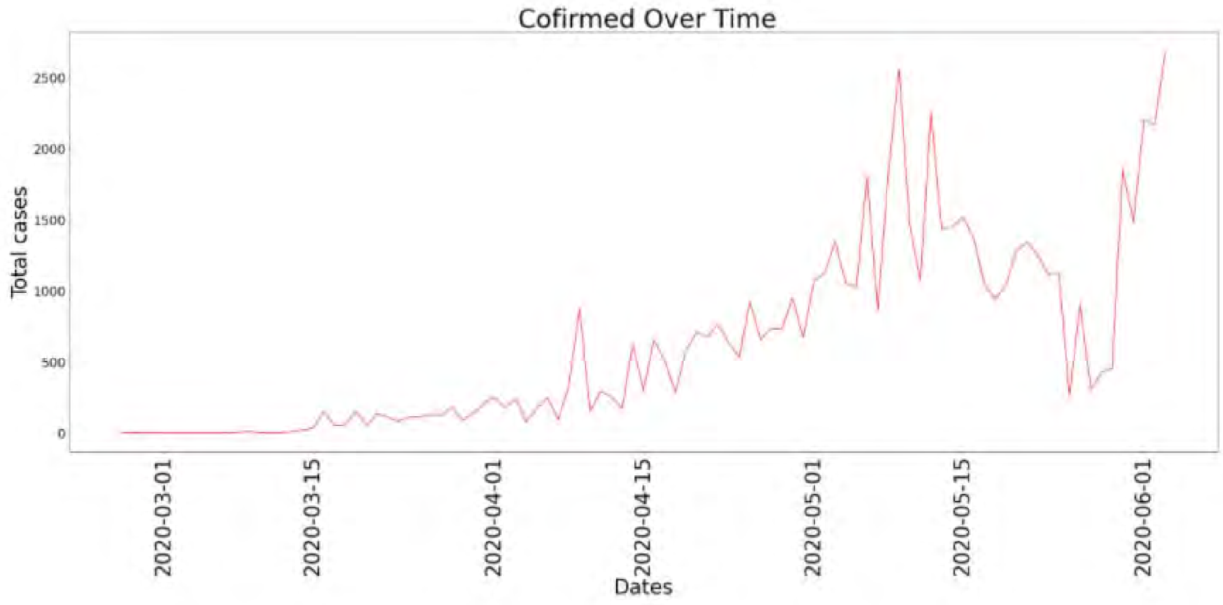
- **Travel_history analysis:**

As we know that Covid can transfer easily form one person to another. So in this case Travel-history plays a very important role in spreading the disease. Below are the graph which shows that the local-social contact is causing the severe spread of disease.

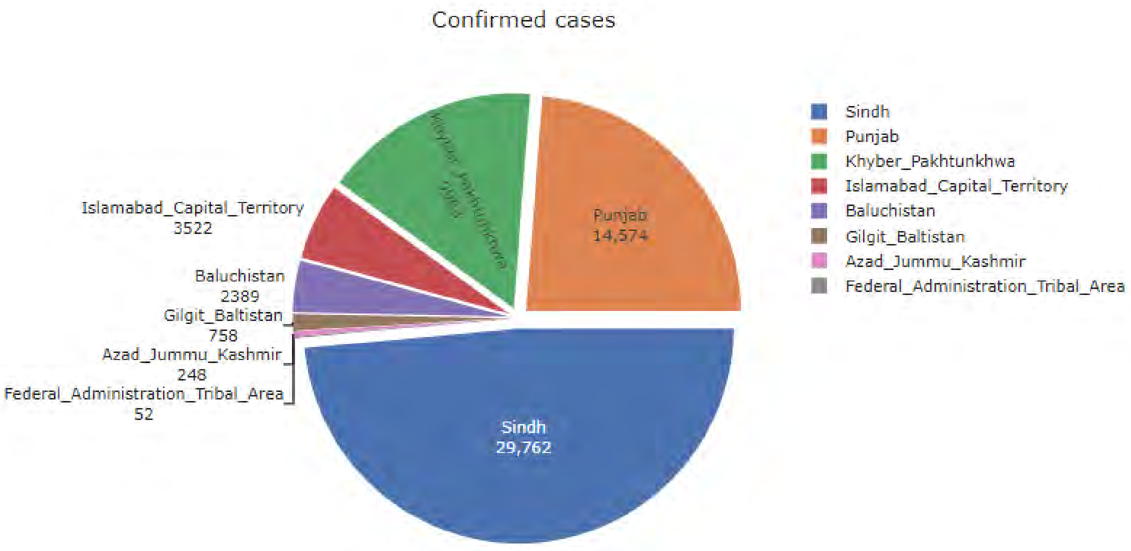


- **Datewise Plotting of Covid_19 cases**

As below graph shows that with the increase of time , the disease of covid_19 also increases.If this situation will continue it will be dangerous to the whole country.

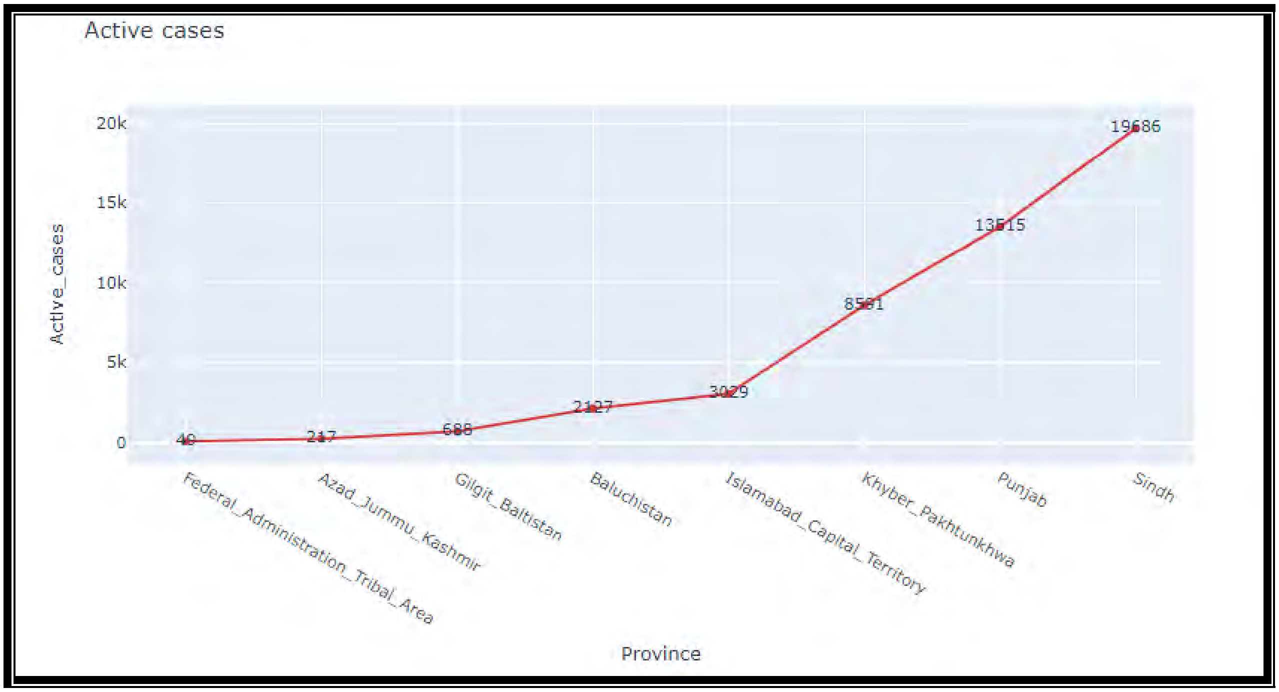


Provincewise Confirmed Cases



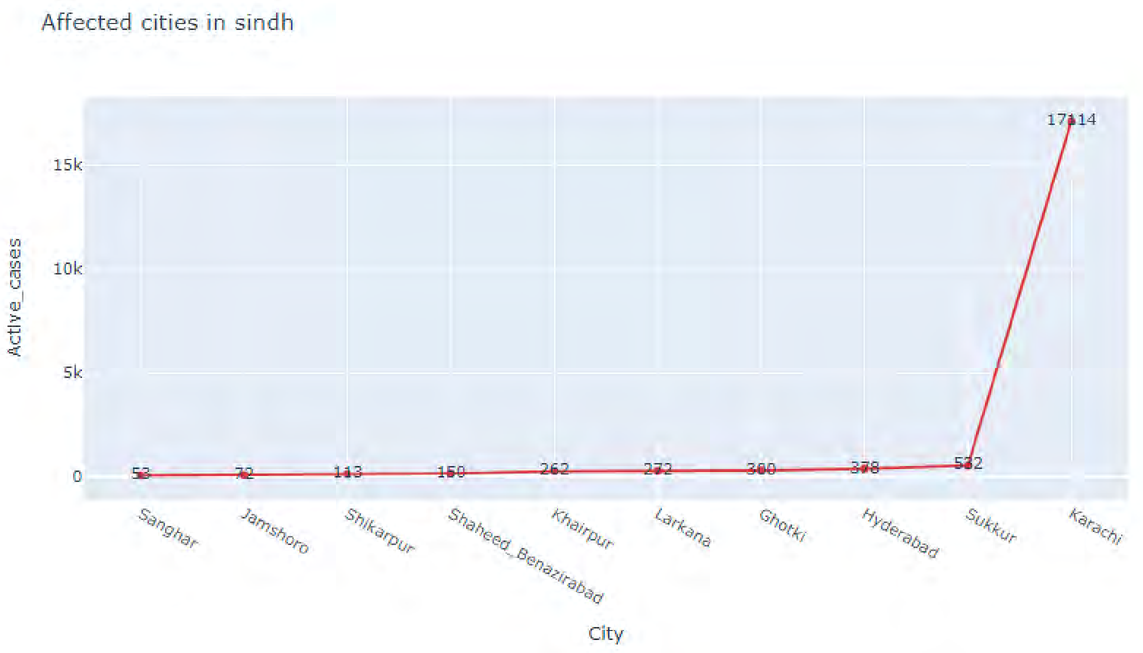
- **Provincewise Active cases:**

Below graph show the active cases in each province and helps us to determine which province has most active cases.

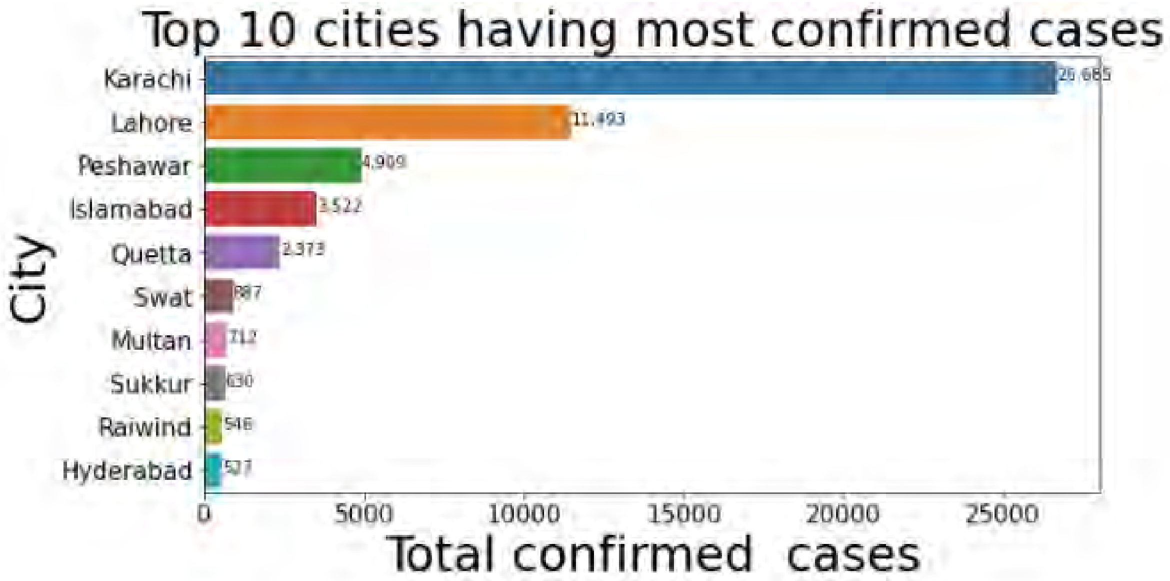


Active Cases in Sindh

As we can see from above graph that the Sindh is the most affected Province so lets examine which city in Sindh is most affected by covid.



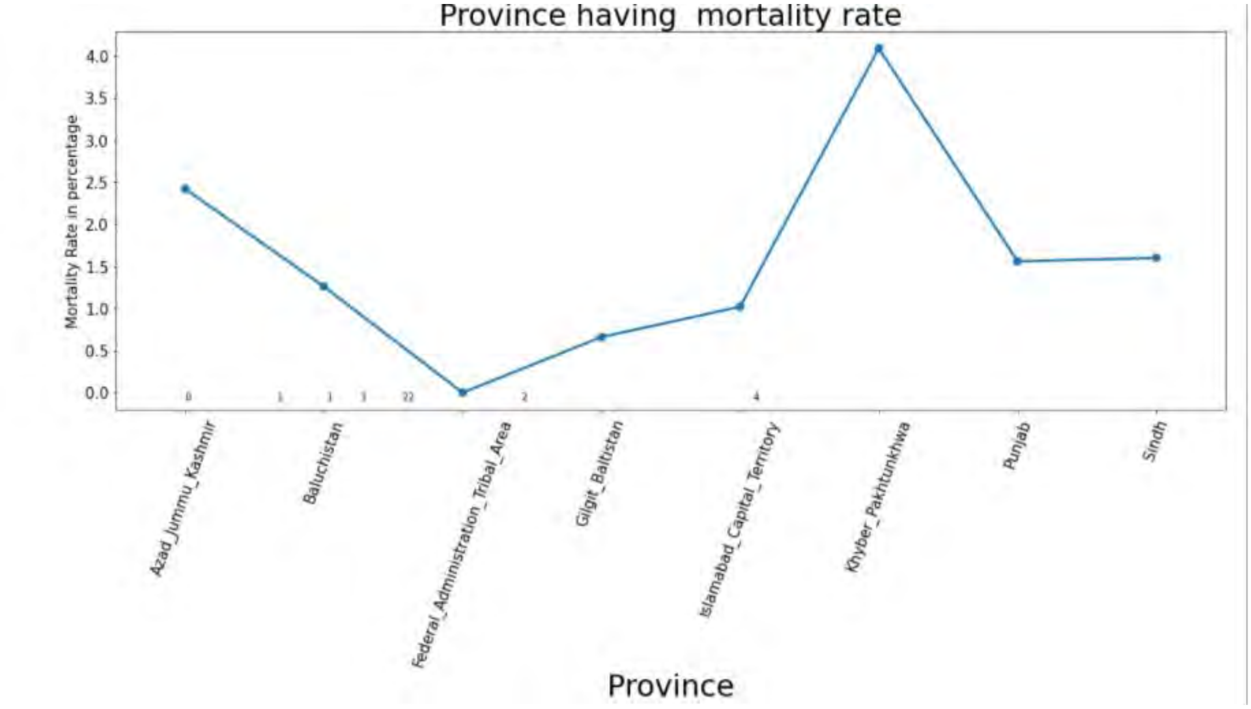
- **Top 10 cities having most confirmed cases**
- Below graph shows the top 10 cities which have most confirmed cases in the Country.



- **Province wise Mortality rate:**

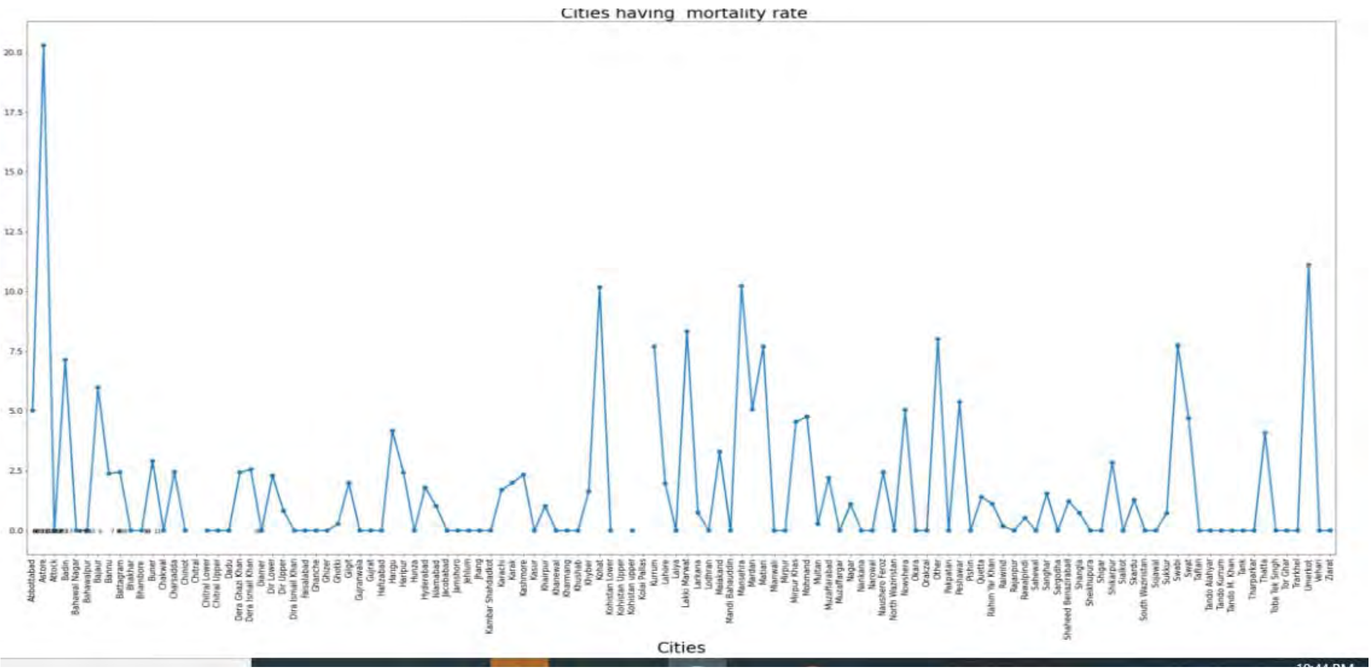
Below graph show the mortality rate of each province ,which province has least recovery rate and more deaths rate.As graph show Khyber Pakhtunkhwa has most mortality than any other province.

If we observe from our graphs then it indicates that although Sindh has most active, deaths and recovered cases but Sindh face less mortality rate and Khyber Pakhtunkhwa faced more mortality rate although it has least cases.

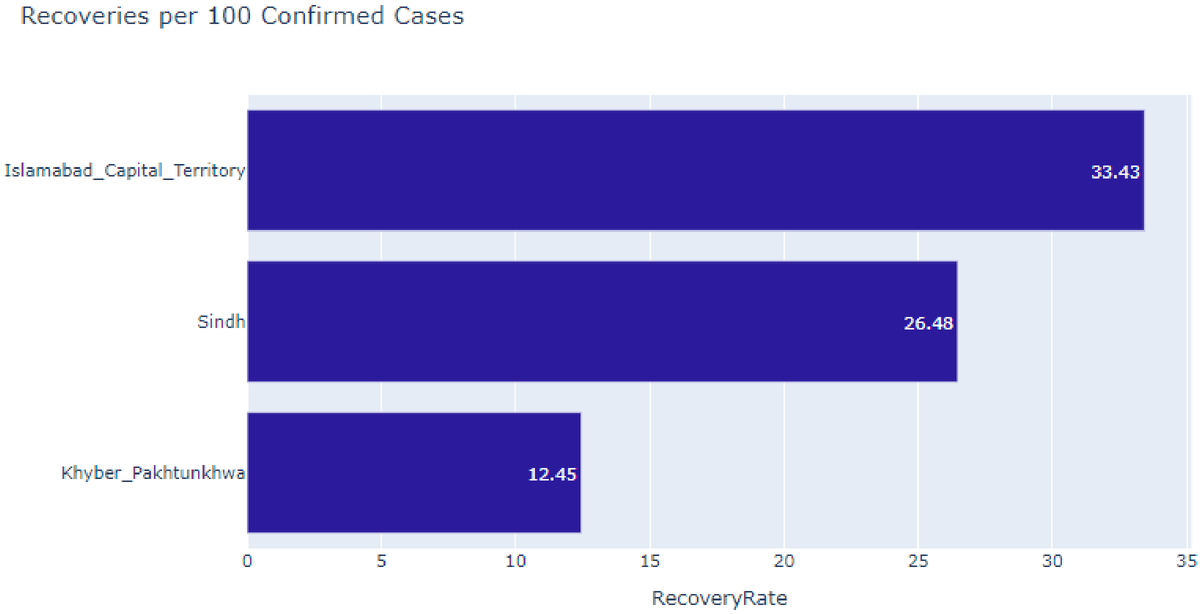


Citywise mortality rate:

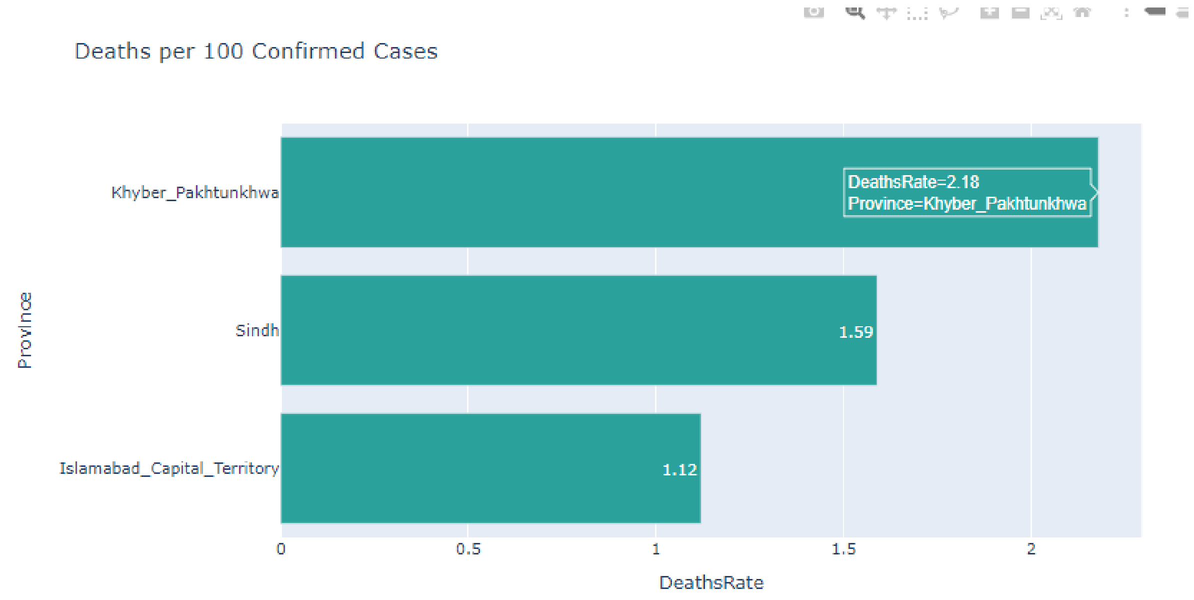
Below graph shows the citywise mortality rate of the infected people. Graph shows that ASTORE has higher mortality rate than any other city.



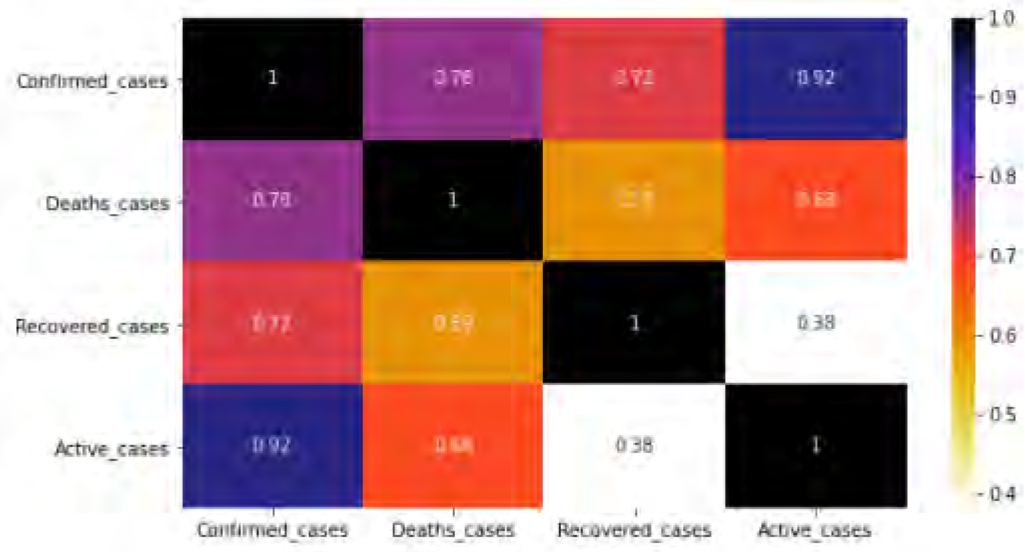
Recovery per 100 confirmed Rate



- **Deaths per 100 Confirmed cases:**



- **Correlation between numerical columns:**



4.5 Summary Results of Analysis:

- Sindh is most affected province in Pakistan.
- Kohat is most affected city in overall Country.
- Karachi is most affected city in Sindh.

- From analysis we see that People having Travel_history(Local_social Contact)are affected more than any other Travel_history record..
- From my observation we can see that Province of Islamabad has most deaths cases and less recovery rate. So it is mandatory to give attention to this Province and apply lockdown to overcome this situation.
- We can also see that with the passage of time cases are increasing rapidly so it is an alarming situation for our country.

Chapter 5

Machine Learning Project Workflow

5.1 Machine Learning Project Workflow:

We can define the machine learning projects workflow in following stages,

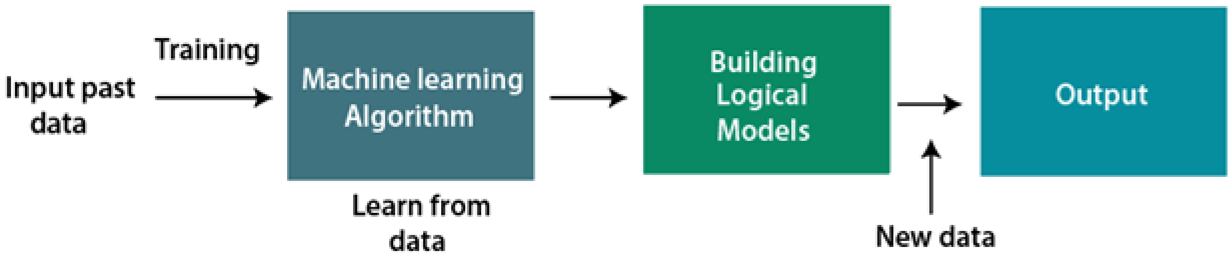
1. Data Collection
2. Data Pre-Processing
3. Researching the model that will be best for the type of data
4. Training and testing the model
5. Evaluation

5.2 What is the Machine Learning Model:

The machine learning model is nothing but a piece of code; an engineer or data scientist makes it smart through training with data. So, if you give garbage to the model, you will get garbage in return, i.e. the trained model will provide false or wrong predictions.

Machine learning centers around the improvement of PC programs that can get to information and utilize it learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide.

Now we define the steps of Machine Learning workflow.



5.3 Objectives of Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning** focuses on the development of computer programs that can access data and use it to learn for themselves.

5.4 Data Collection:

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, and then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

We can also use some free data sets that are present on the internet. Kaggle and UCI Machine learning Repository are the repositories that are used the most for making Machine-learning models. Kaggle is one of the most visited websites that is used for practicing machine-learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

5.5 Data Pre-Processing:

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.

1.What is Data Pre-Processing:

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn’t feasible for the analysis. Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

2. Why do we need it:

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

Most of the real-world data is messy, some of these types of data are:

- **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

- **Noisy data:** This type of data is also called outliers, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.
- **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

Three Types of data:

- Numeric e.g. income, age
- Categorical e.g. gender, nationality
- Ordinal e.g. low/medium/high

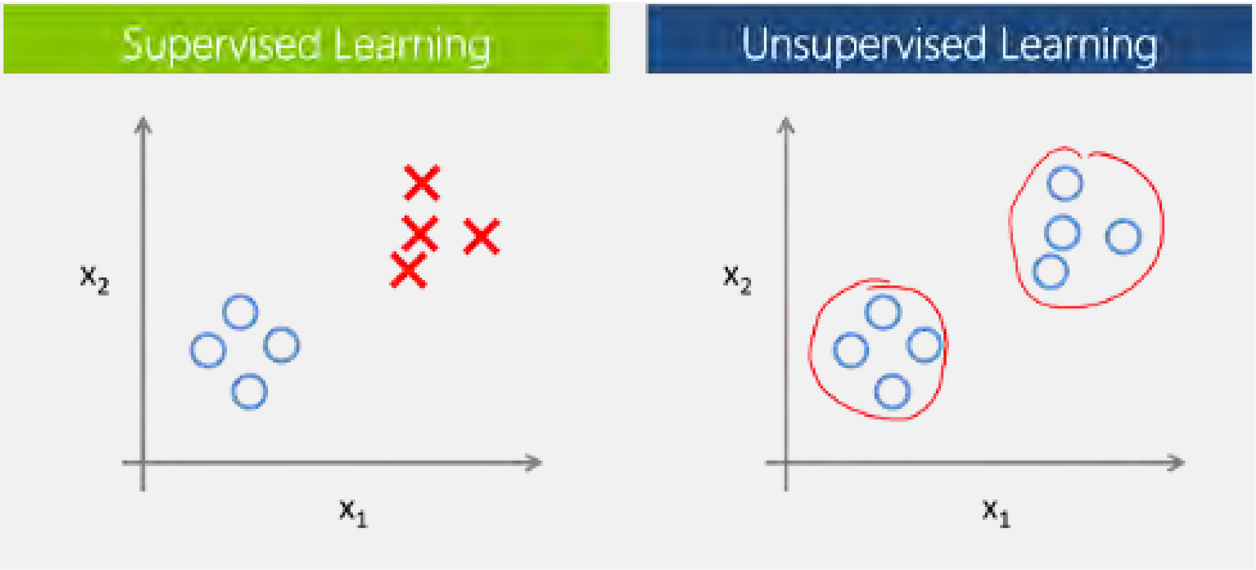
3. How can data pre-processing be performed:

These are some of the basic pre — processing techniques that can be used to convert raw data.

- **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.
- **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient, but it shouldn't be performed if there are a lot of missing values in the dataset.
- **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly, the mean, median or highest frequency value is used.
- **Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.
- **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra .

5.6. Researching the model that will be best for the type of data

Our main goal is to train the best performing model possible, using the pre-processed data.



5.6.1 Supervised Learning:

In Supervised learning, an AI system is presented with data which is labelled, which means that each data tagged with the correct label. The supervised learning is categorized into 2 other categories which are “Classification” and “Regression”.

1. Classification:

Classification is used when the target variable is categorical (i.e. the output could be classified into classes — it belongs to either Class A or B or something else). A classification problem is when the output variable is a category, such as “red” or “blue”, “disease” or “no disease” or “spam” or “not spam”.

Following are the most used classification algorithms:

- K-Nearest Neighbor
- Naive Bayes
- Decision Trees/Random Forest
- Support Vector Machine
- Logistic Regression

I have applied various classification models on my dataset but my actual work is based on regression so I had given more attention to regression instead of classification.

2. Regression:

Regression is used when the target variable is continuous (i.e. the output is numeric).

Following are the most used regression algorithms:

Linear Regression

- Support Vector Regression
- Decision Trees/ Random Forest
- Gaussian Progresses Regression
- Ensemble Methods

5.6.2 Unsupervised Learning:

In unsupervised learning, an AI system is presented with unlabeled, un-categorized data and the system's algorithms act on the data without prior training. The output is dependent upon the coded algorithms. Subjecting a system to unsupervised learning is one way of testing AI.

The unsupervised learning is categorized into 2 other categories which are "Clustering" and "Association".

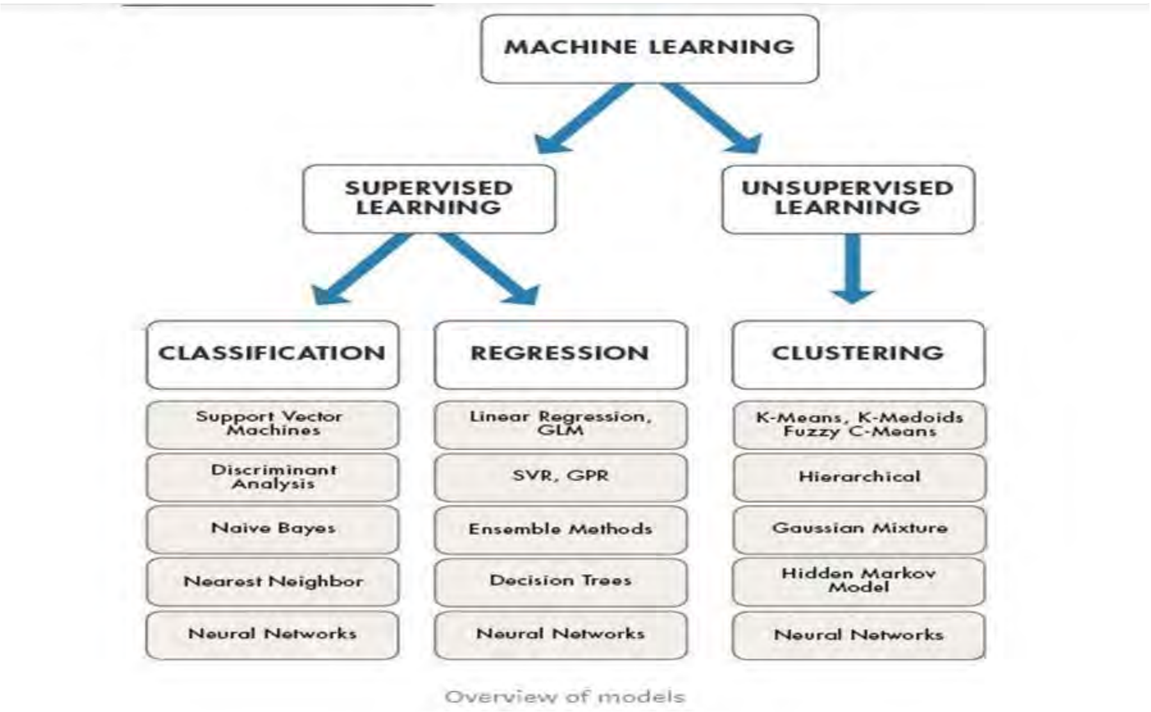
1. Clustering:

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

Methods used for clustering are:

- Gaussian mixtures
- K-Means Clustering
- Boosting
- Hierarchical Clustering
- K-Means Clustering
- Spectral Clustering

5.6.3 Overview of models under categories:



5.7 Training and testing the model on data:

For training a model we initially split the model into 3 three sections which are 'Training data' and 'Testing data'. You train the model using 'training data set and then test the performance of your classifier on unseen 'test data set'. An important point to note is that during training the classifier only the training and/or validation set is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

1. Training set:

The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier. We can give any number to our training data such as 60% , 70% , 75% or 80% any number can be used for your model training.

2. Validation set:

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. A set of unseen data is used from the training data to tune the parameters of a classifier.

3. Test set:

A set of unseen data used only to assess the performance of a fully specified classifier. We can set test size as 20% ….. 50% any number we wish can give to model.

5.8 Parameters tunning:

Machine Learning models are composed of two different types of parameters:

- **Hyperparameters** = are all the parameters which can be arbitrarily set by the user before starting training (e.g. number of estimators in Random Forest).
- **Model parameters =** are instead learned during the model training (e.g. weights in Neural Networks, Linear Regression).

The model parameters define how to use input data to get the desired output and are learned at training time. Instead, Hyperparameters determine how our model is structured in the first place.

Machine Learning models tuning is a type of optimization problem. We have a set of hyperparameters and we aim to find the right combination of their values which can help us to find either the minimum (e.g loss) or the maximum (e.g accuracy) of a function..

This can be particularly important when comparing how different Machine Learning models performs on a dataset. In fact, it would be unfair for example to compare an SVM model with the best Hyperparameters against a Random Forest model which has not been optimized.

Following techniques are used for hyperparameter tunning.

1. Manual Search
2. Random Search
3. Grid Search
4. Automated Hyperparameter Tuning (Bayesian Optimization, Genetic Algorithms)
5. Artificial Neural Networks (ANNs) Tuning

5.9 Model Evaluation:

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

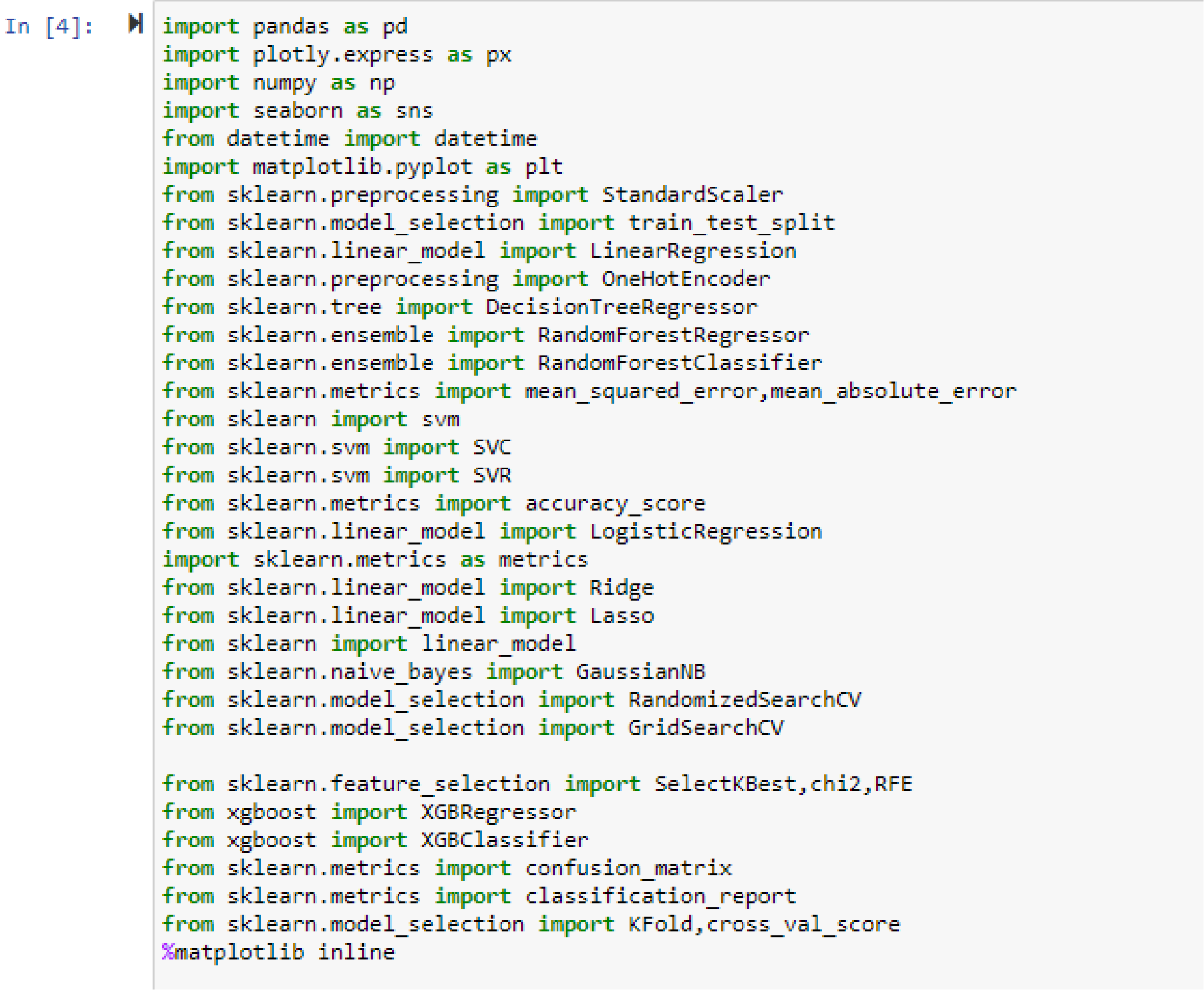
Chapter No 6

Model Implementation

6.1 Model Development

As in previous section we apply all preprocessing steps to convert the data in right format now in this section I will build machine learning models.

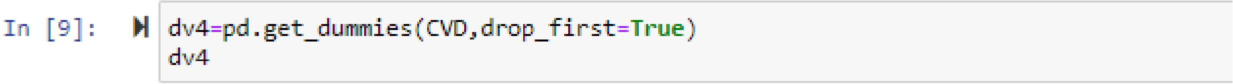
6.1.1 Import Important Libraries



```
In [4]:  import pandas as pd
         import plotly.express as px
         import numpy as np
         import seaborn as sns
         from datetime import datetime
         import matplotlib.pyplot as plt
         from sklearn.preprocessing import StandardScaler
         from sklearn.model_selection import train_test_split
         from sklearn.linear_model import LinearRegression
         from sklearn.preprocessing import OneHotEncoder
         from sklearn.tree import DecisionTreeRegressor
         from sklearn.ensemble import RandomForestRegressor
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.metrics import mean_squared_error,mean_absolute_error
         from sklearn import svm
         from sklearn.svm import SVC
         from sklearn.svm import SVR
         from sklearn.metrics import accuracy_score
         from sklearn.linear_model import LogisticRegression
         import sklearn.metrics as metrics
         from sklearn.linear_model import Ridge
         from sklearn.linear_model import Lasso
         from sklearn import linear_model
         from sklearn.naive_bayes import GaussianNB
         from sklearn.model_selection import RandomizedSearchCV
         from sklearn.model_selection import GridSearchCV

         from sklearn.feature_selection import SelectKBest,chi2,RFE
         from xgboost import XGBRegressor
         from xgboost import XGBClassifier
         from sklearn.metrics import confusion_matrix
         from sklearn.metrics import classification_report
         from sklearn.model_selection import KFold,cross_val_score
         %matplotlib inline
```

6.1.2 Create dummies values

As we can see that our dataset contain string value and Machine learning model cannot understand any string values so it is mandatory to convert string values into dummies values.



```
In [9]:  dv4=pd.get_dummies(CVD,drop_first=True)
         dv4
```

6.2 Fitting the model

We fit various models on our dataset which are described as follows:

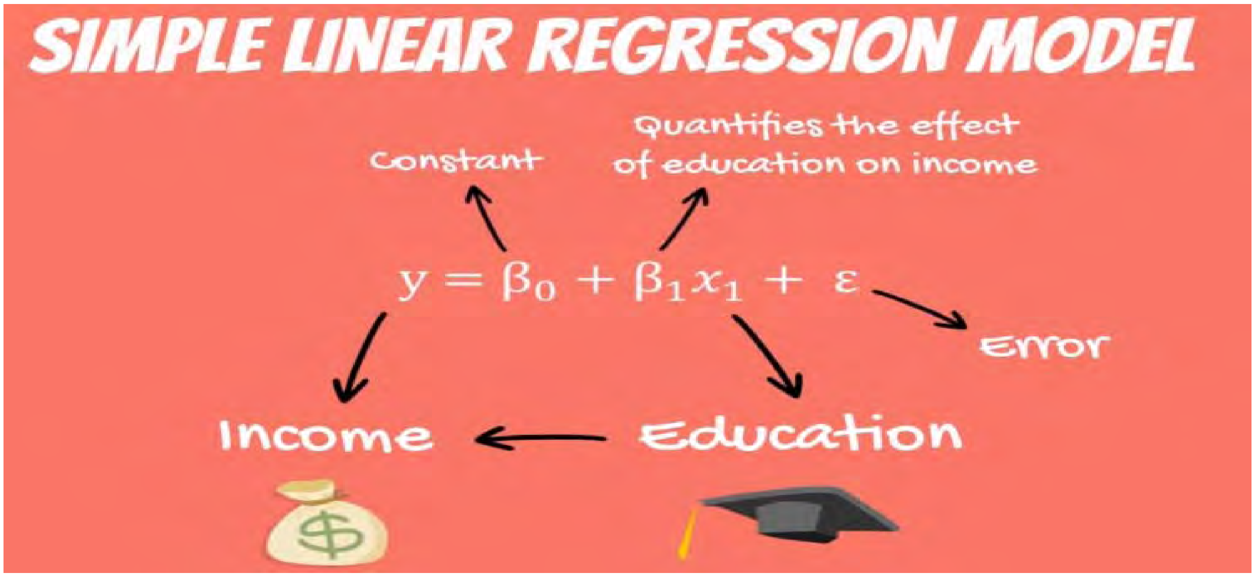
6.2.1Linear Regression

Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

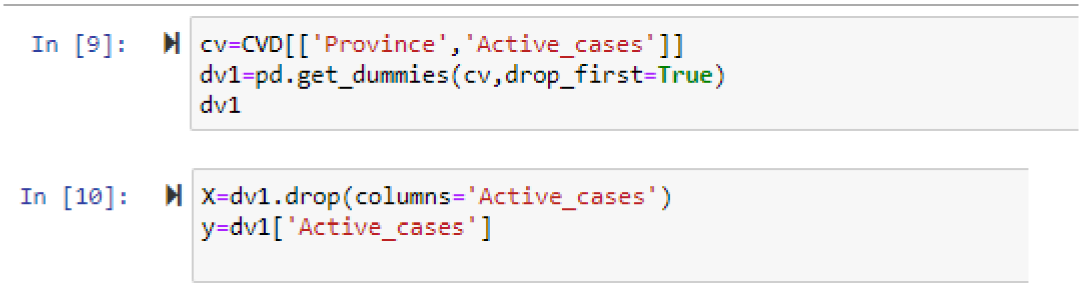
Types of regression models:

6.2.1.1 Simple linear regression

1 dependent variable (interval or ratio), 1 independent variable (interval or ratio or dichotomous)


SIMPLE LINEAR REGRESSION MODEL
Constant
Quantifies the effect of education on income
$y = \beta_0 + \beta_1 x_1 + \varepsilon$
Error
Income
Education

In this model, only one input is given and we give input as province and give back the prediction od Active cases.As we can see that Province column is a categorical column so firstly it must be converted into dummy variables.

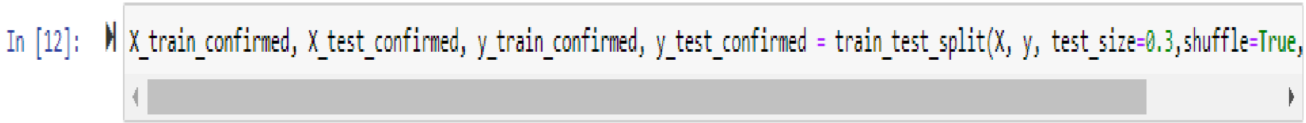


```
In [9]:   cv=CVD[['Province','Active_cases']]
          dv1=pd.get_dummies(cv,drop_first=True)
          dv1
```

```
In [10]:  X=dv1.drop(columns='Active_cases')
          y=dv1['Active_cases']
```

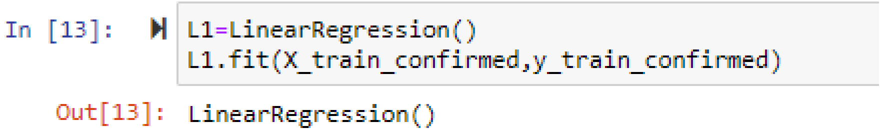
Splitting data into training and testing sets

As shown from below figure that we set test size as 30% and train size as 70%.



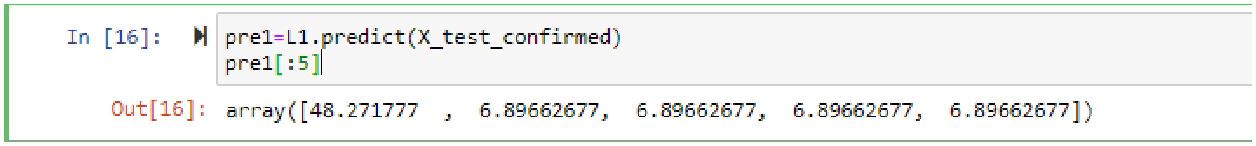
```
In [12]:  X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed = train_test_split(X, y, test_size=0.3,shuffle=True,
```

Model fitting:

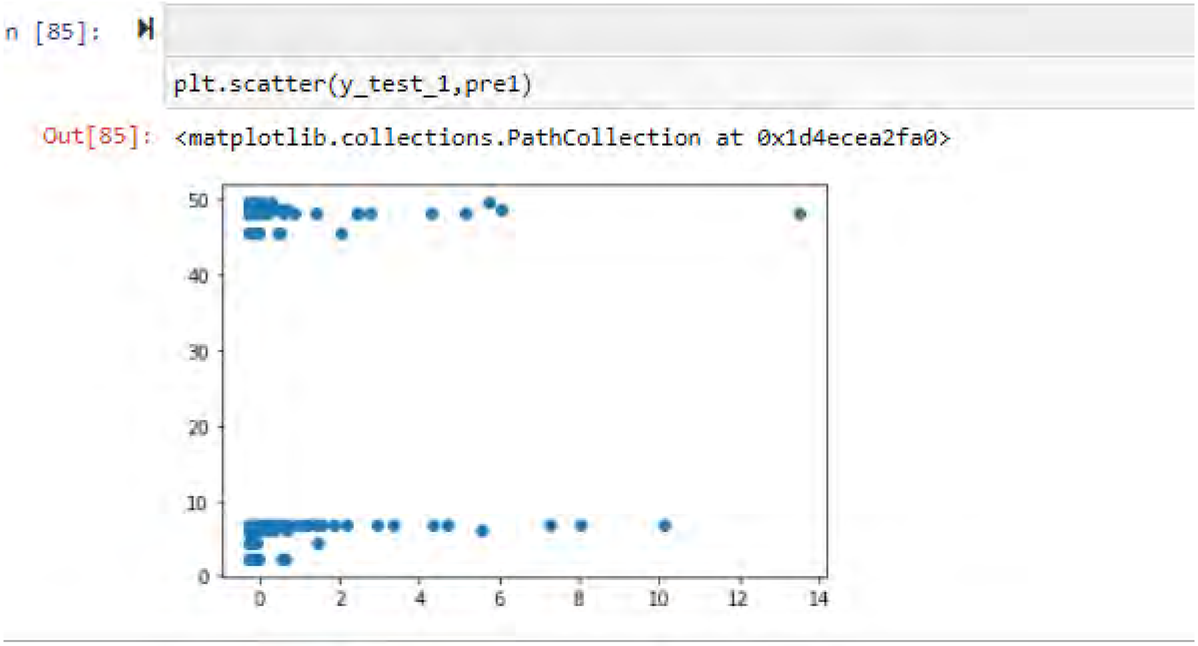


```
In [13]:  L1=LinearRegression()
           L1.fit(X_train_confirmed,y_train_confirmed)

Out[13]:  LinearRegression()
```

Prediction of cases



```
In [16]:  pre1=L1.predict(X_test_confirmed)
           pre1[:5]

Out[16]:  array([48.271777  ,  6.89662677,  6.89662677,  6.89662677,  6.89662677])
```

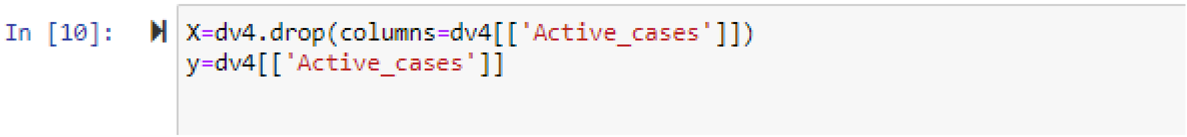


```
n [85]:

          plt.scatter(y_test_1,pre1)

Out[85]:  <matplotlib.collections.PathCollection at 0x1d4ecea2fa0>
```

6.2.1.2Multiple linear regression

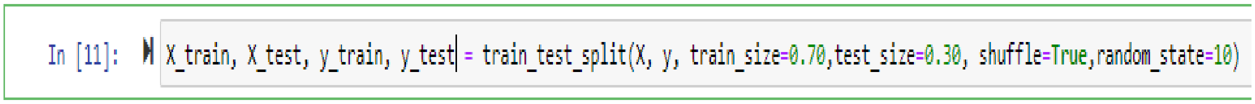
1 dependent variable (interval or ratio) , 2+ independent variables (interval or ratio or dichotomous).

In multiple linear Regression model, we give all columns except target (Active cases) as input.



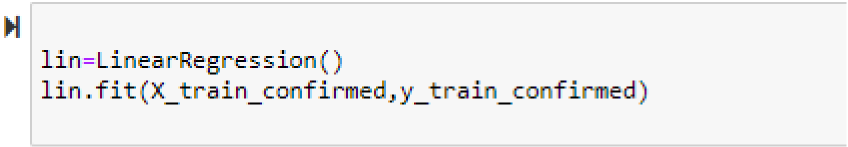
```
In [10]:  X=dv4.drop(columns=dv4[['Active_cases']])
          y=dv4[['Active_cases']]
```

Splitting data:

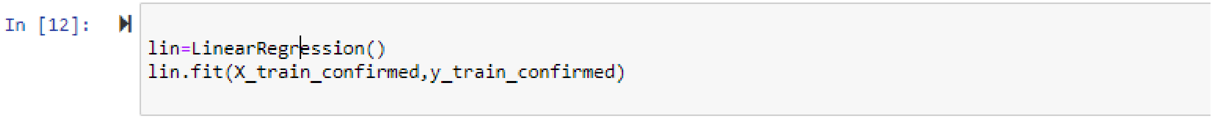


```
In [11]:  X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.70,test_size=0.30, shuffle=True,random_state=10)
```

Model fitting

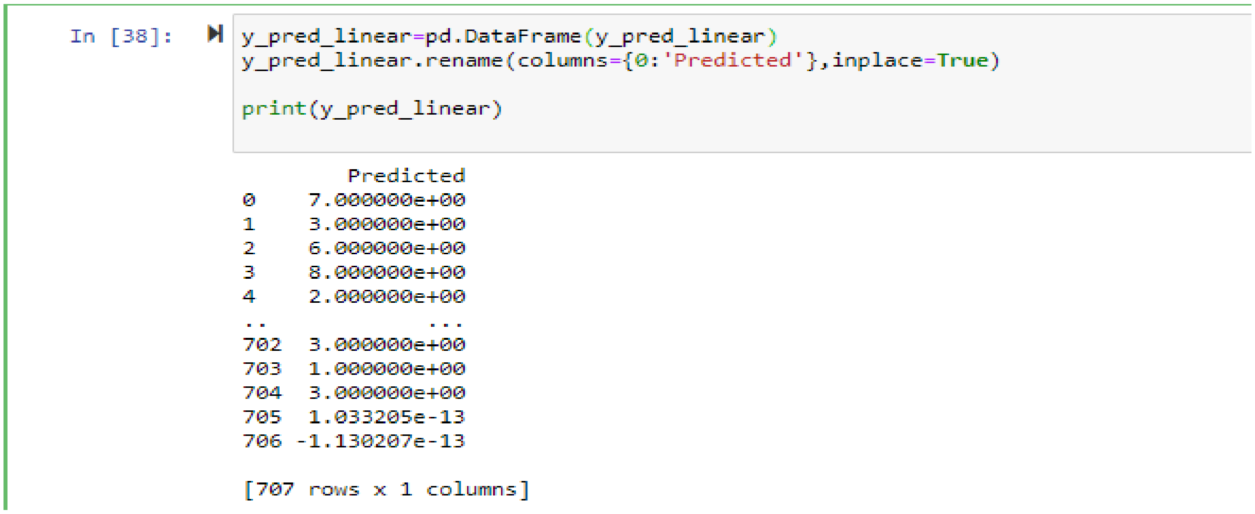


```
lin=LinearRegression()
lin.fit(X_train_confirmed,y_train_confirmed)
```

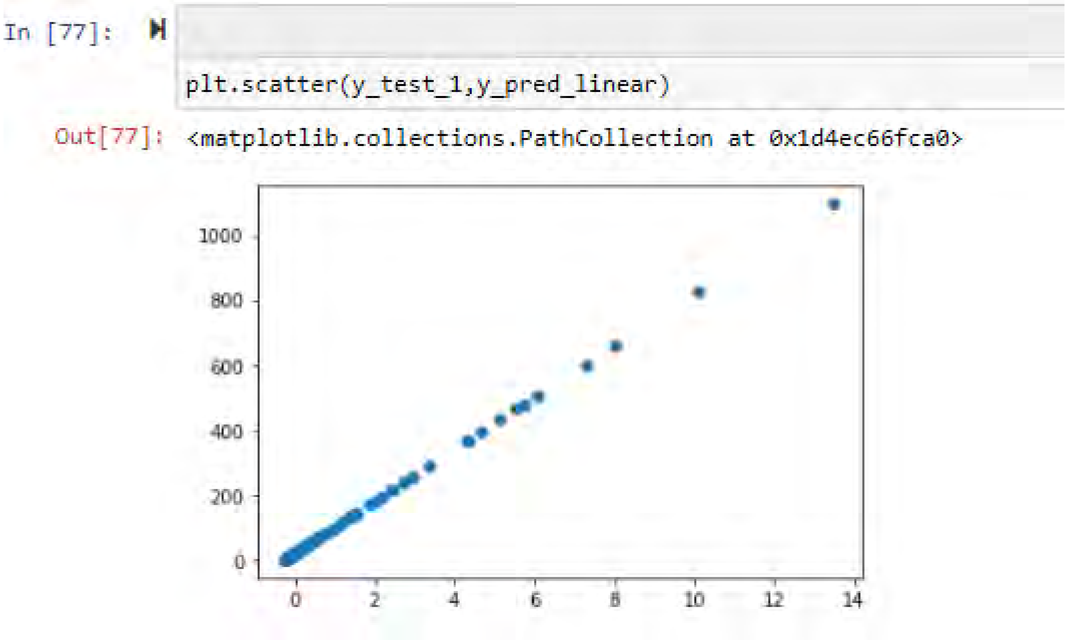


```
In [12]:  lin=LinearRegression()
          lin.fit(X_train_confirmed,y_train_confirmed)
```

Prediction of data:



```
In [38]:  y_pred_linear=pd.DataFrame(y_pred_linear)
          y_pred_linear.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_linear)

                 Predicted
          0     7.000000e+00
          1     3.000000e+00
          2     6.000000e+00
          3     8.000000e+00
          4     2.000000e+00
          ..             ...
          702   3.000000e+00
          703   1.000000e+00
          704   3.000000e+00
          705   1.033205e-13
          706  -1.130207e-13

          [707 rows x 1 columns]
```

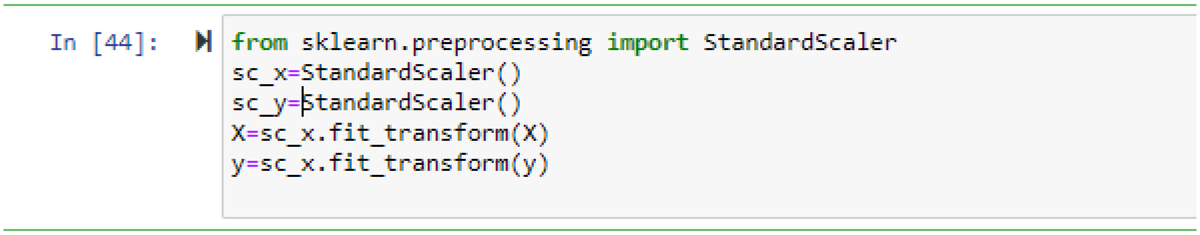


```
In [77]:  plt.scatter(y_test_1,y_pred_linear)
Out[77]: <matplotlib.collections.PathCollection at 0x1d4ec66fca0>
```

6.2.2 Support Vector Regression:

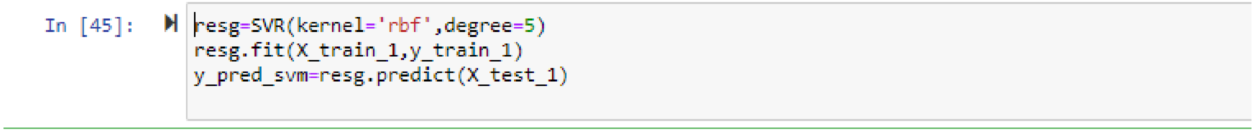
In support vector regression it is compulsory to scale the data so we first perform feature scaling and then fit the model.

Feature scaling transforms your data such that the resulting distribution has a mean of 0 and a standard deviation of 1



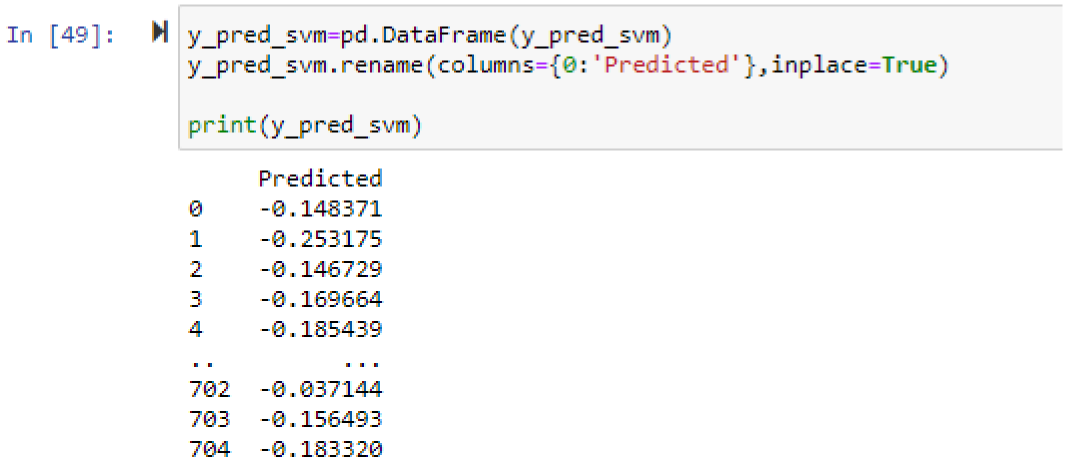
```
In [44]:  from sklearn.preprocessing import StandardScaler
           sc_x=StandardScaler()
           sc_y=StandardScaler()
           X=sc_x.fit_transform(X)
           y=sc_x.fit_transform(y)
```

Fitting model

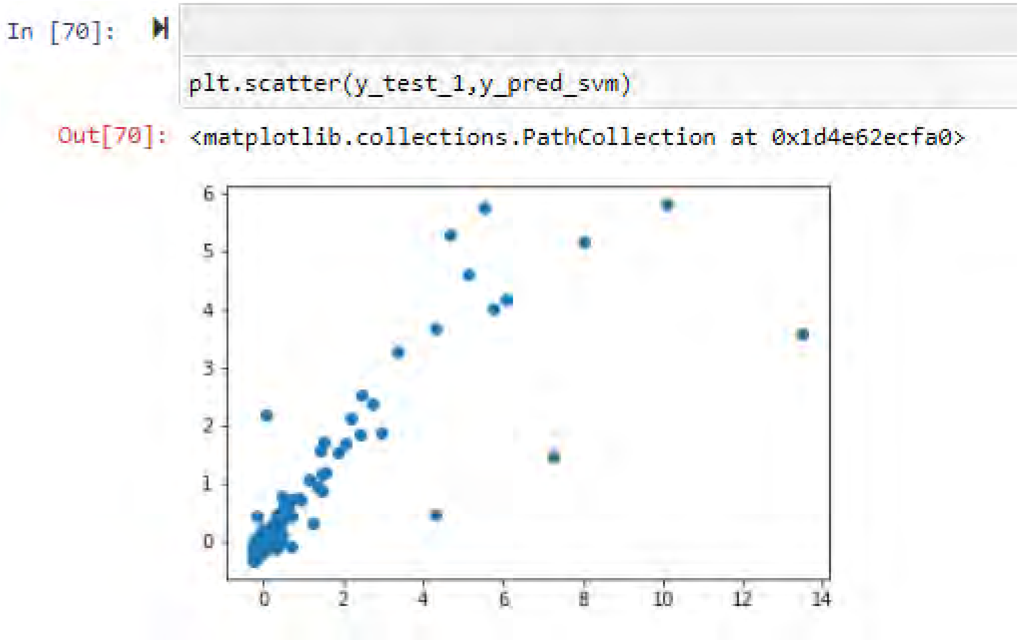


```
In [45]:  resg=SVR(kernel='rbf',degree=5)
           resg.fit(X_train_1,y_train_1)
           y_pred_svm=resg.predict(X_test_1)
```

Prediction of model:



```
In [49]:  y_pred_svm=pd.DataFrame(y_pred_svm)
          y_pred_svm.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_svm)

     Predicted
0    -0.148371
1    -0.253175
2    -0.146729
3    -0.169664
4    -0.185439
..         ...
702  -0.037144
703  -0.156493
704  -0.183320
```



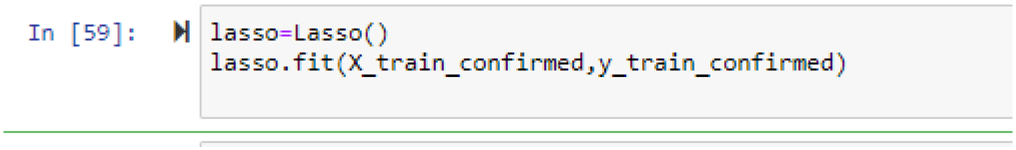
```
In [70]:  

          plt.scatter(y_test_1,y_pred_svm)

Out[70]: <matplotlib.collections.PathCollection at 0x1d4e62ecfa0>
```

6.2.3 Lasso Regression:

Lasso regression is a type of **linear regression** that use shrinkage. Shrinkage is where data values are shrunk towards a central point, like the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of muticollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

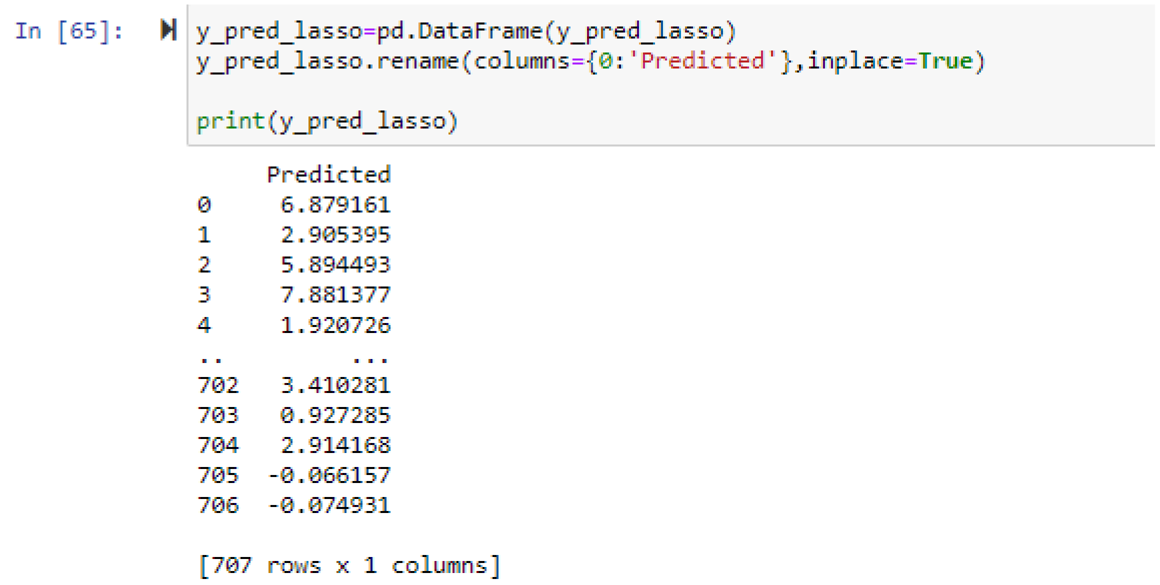
Lasso regression performs L1 regularization, which adds a penalty equal to the absolute value of the magnitude of coefficients. This type of regularization can result in sparse models with few coefficients; Some coefficients can become zero and eliminated from the model. Larger penalties result in coefficient values closer to zero, which is the ideal for producing simpler models. On the other hand, L2 regularization (e.g. Ridge regression) doesn’t result in elimination of coefficients or sparse models. This makes the Lasso far easier to interpret than the Ridge.

Fitting the model

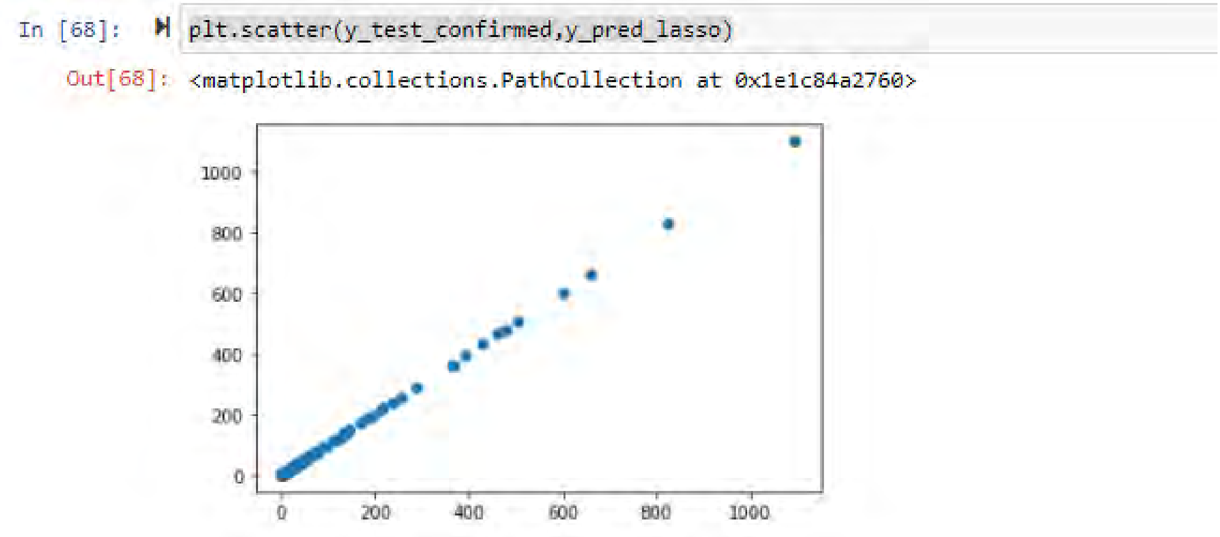


```
In [59]:  lasso=Lasso()
          lasso.fit(X_train_confirmed,y_train_confirmed)
```

Prediction of cases



```
In [65]:  y_pred_lasso=pd.DataFrame(y_pred_lasso)
          y_pred_lasso.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_lasso)

     Predicted
0     6.879161
1     2.905395
2     5.894493
3     7.881377
4     1.920726
..         ...
702   3.410281
703   0.927285
704   2.914168
705  -0.066157
706  -0.074931

[707 rows x 1 columns]
```

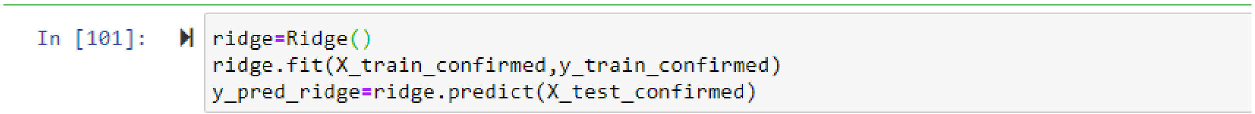


```
In [68]:  plt.scatter(y_test_confirmed,y_pred_lasso)
Out[68]:  <matplotlib.collections.PathCollection at 0x1e1c84a2760>
```

6.2.4 Ridge Regression:

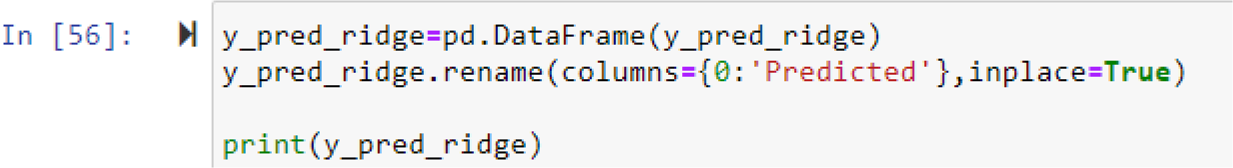
Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

Fitting the model:



```
In [101]:  ridge=Ridge()
            ridge.fit(X_train_confirmed,y_train_confirmed)
            y_pred_ridge=ridge.predict(X_test_confirmed)
```

Prediction:



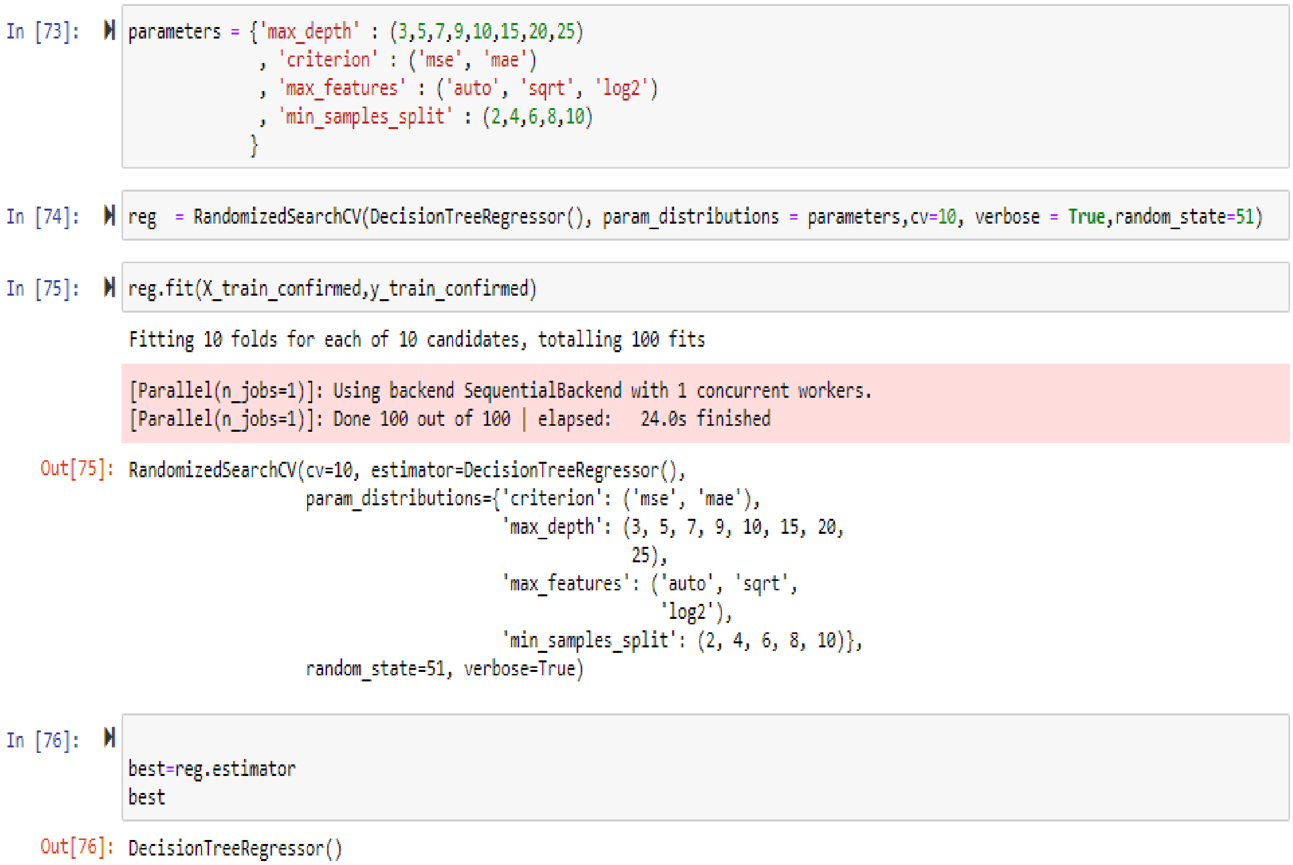
```
In [56]:  y_pred_ridge=pd.DataFrame(y_pred_ridge)
          y_pred_ridge.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_ridge)
```

6.2.5 Decision Tree:

Decision Tree is a **Supervised learning technique** that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where **internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**

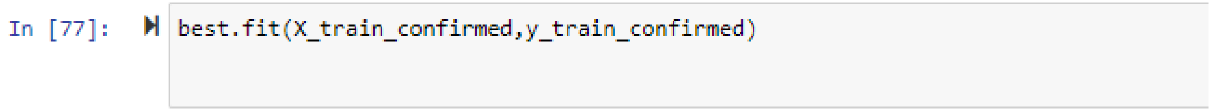
In a Decision tree, there are two nodes, which are the **Decision Node** and **Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

- **Hyperparameter tunning using Randomized SearchCV.**



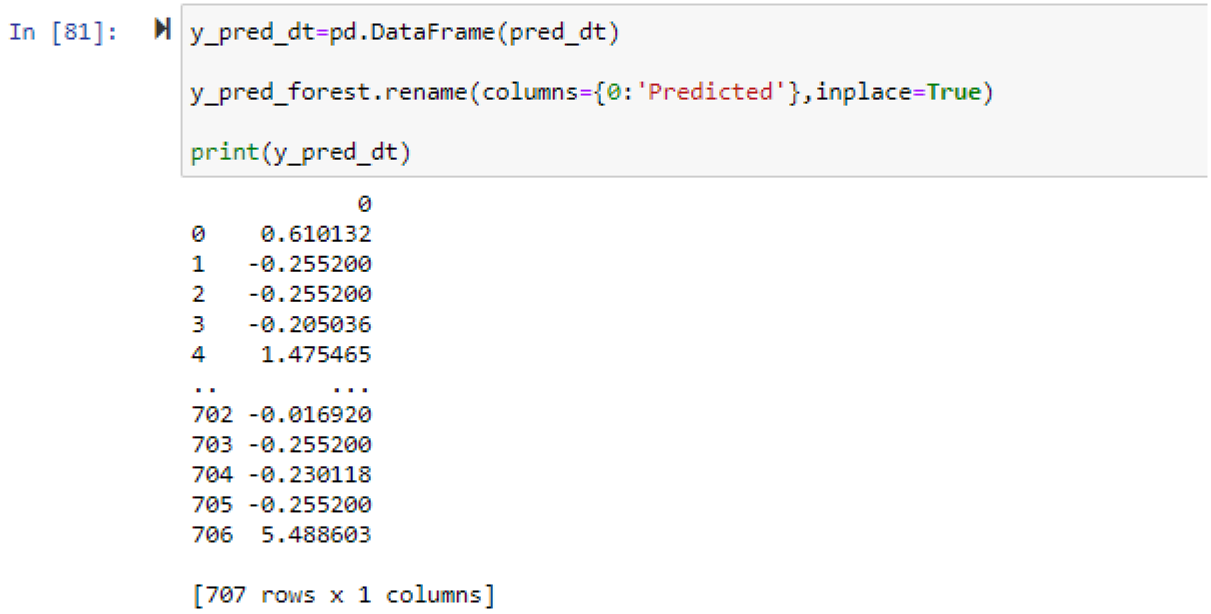
```
In [73]:  parameters = {'max_depth' : (3,5,7,9,10,15,20,25)
                       , 'criterion' : ('mse', 'mae')
                       , 'max_features' : ('auto', 'sqrt', 'log2')
                       , 'min_samples_split' : (2,4,6,8,10)
                      }

In [74]:  reg  = RandomizedSearchCV(DecisionTreeRegressor(), param_distributions = parameters,cv=10, verbose = True,random_state=51)

In [75]:  reg.fit(X_train_confirmed,y_train_confirmed)

          Fitting 10 folds for each of 10 candidates, totalling 100 fits

          [Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
          [Parallel(n_jobs=1)]: Done 100 out of 100 | elapsed:   24.0s finished

Out[75]:  RandomizedSearchCV(cv=10, estimator=DecisionTreeRegressor(),
                             param_distributions={'criterion': ('mse', 'mae'),
                                                  'max_depth': (3, 5, 7, 9, 10, 15, 20,
                                                                25),
                                                  'max_features': ('auto', 'sqrt',
                                                                   'log2'),
                                                  'min_samples_split': (2, 4, 6, 8, 10)},
                             random_state=51, verbose=True)

In [76]:  
          best=reg.estimator
          best

Out[76]:  DecisionTreeRegressor()
```

Fit the model:



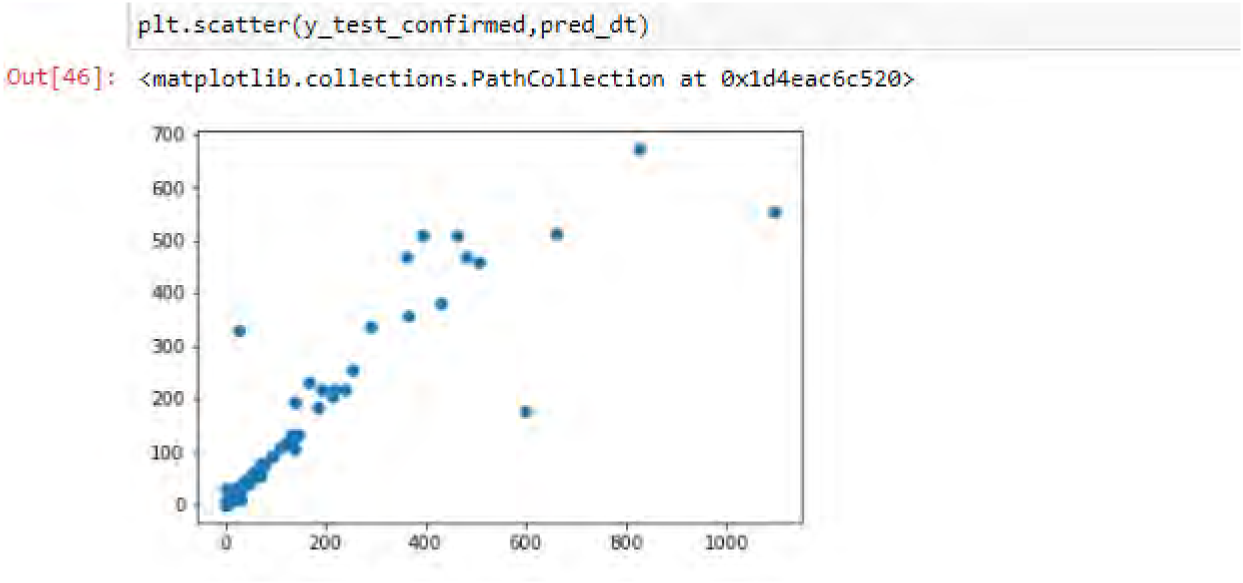
```
In [77]:  best.fit(X_train_confirmed,y_train_confirmed)
```

Predictions:



```
In [81]:  y_pred_dt=pd.DataFrame(pred_dt)

          y_pred_forest.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_dt)
```

```
            0
0    0.610132
1   -0.255200
2   -0.255200
3   -0.205036
4    1.475465
..        ...
702 -0.016920
703 -0.255200
704 -0.230118
705 -0.255200
706  5.488603

[707 rows x 1 columns]
```



```
          plt.scatter(y_test_confirmed,pred_dt)

Out[46]: <matplotlib.collections.PathCollection at 0x1d4eac6c520>
```

6.2.6 Random Forest Regression

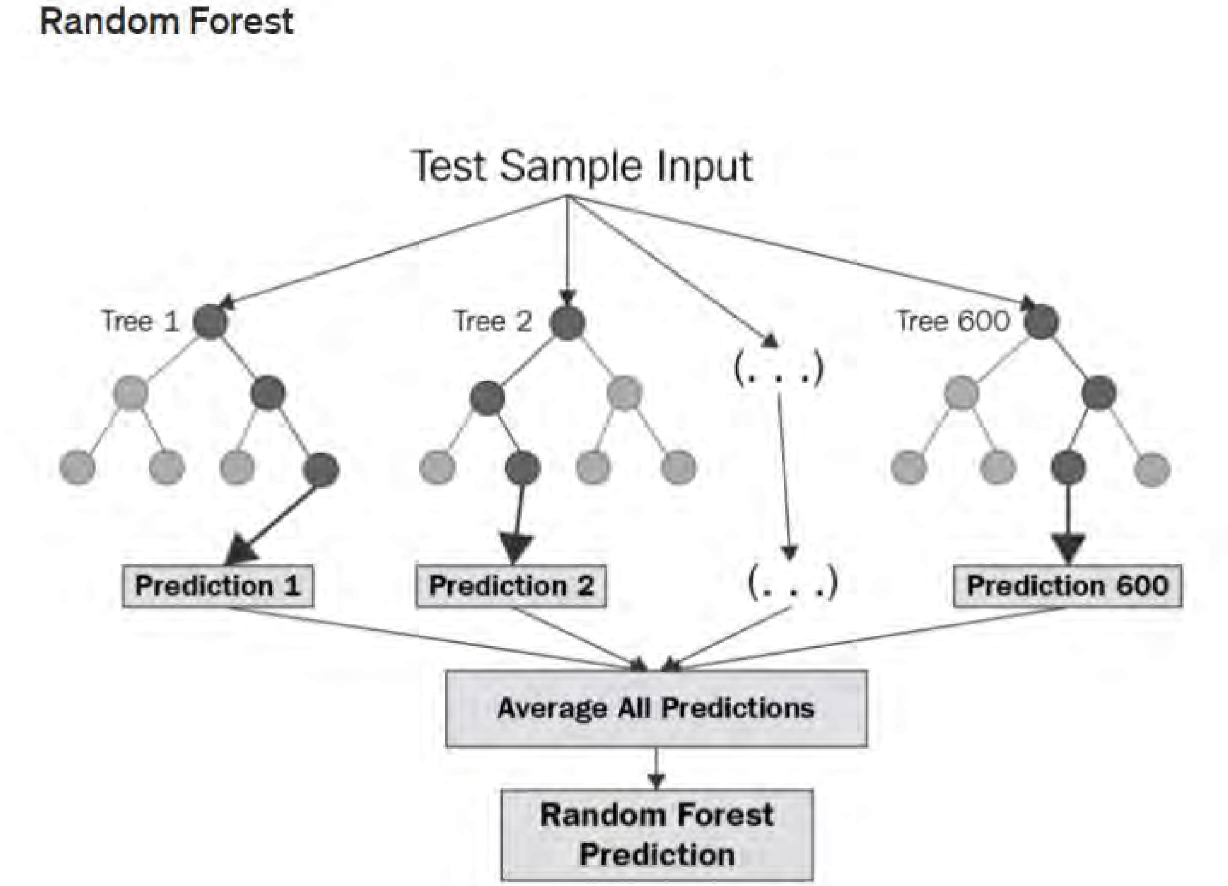
Random Forest Regression is a supervised learning algorithm that uses ensemble learning method for **regression**. A **Random Forest** operates by constructing several **decision** trees during training time and outputting the mean of the classes as the prediction of all the trees.

Random forest is a **bagging** technique and **not a boosting** technique. The trees in **random forests** are run in parallel. There is no interaction between these trees while building the trees.

It operates by constructing a multitude of decision trees at training time and outputting the class that is the **mode** of the **classes (classification)** or **mean prediction (regression)** of the individual trees.

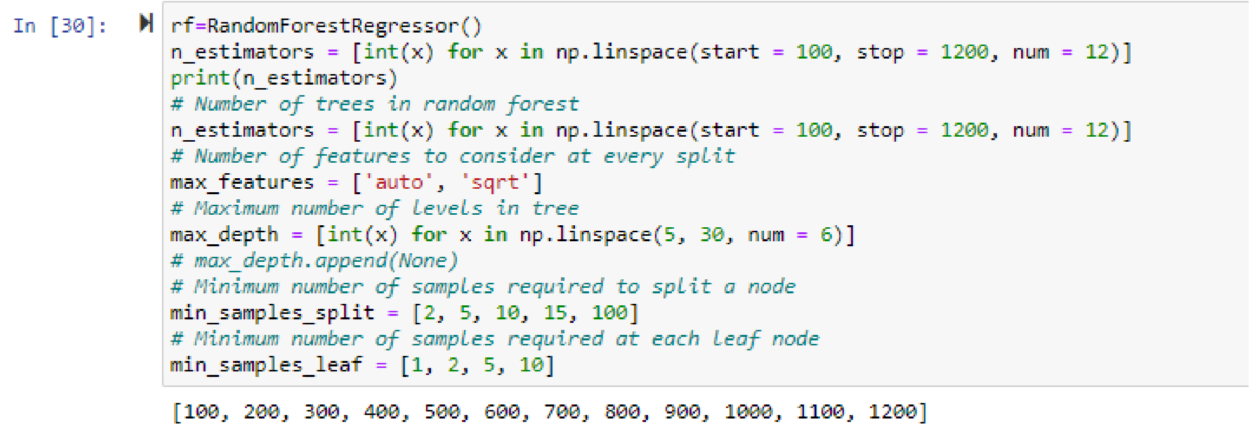
A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which **aggregates many decision trees**, with some helpful modifications:

1. The number of features that can be split on at each node is limited to some percentage of the total (which is known as the **hyperparameter**). This ensures that the ensemble model **does not rely too heavily on any individual feature**, and makes **fair use of all potentially predictive features**.
2. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents **overfitting**.

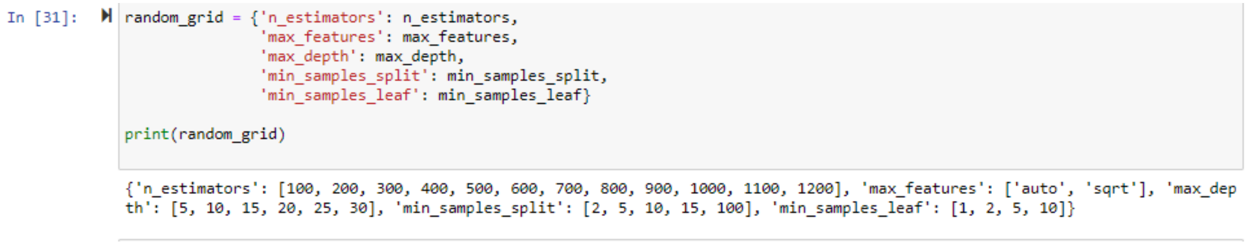


Hyperparameter tunning

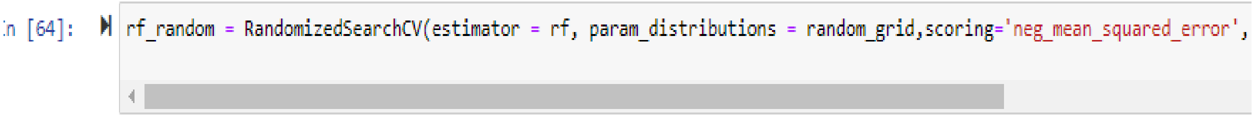
In random forest model we did hyperparamter tunning, to tune the model so that it give us best accuracy.



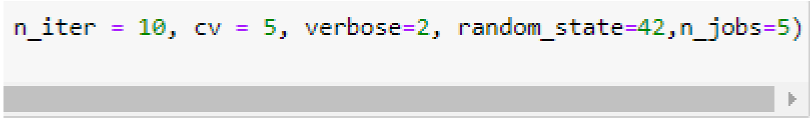
```
In [30]:  rf=RandomForestRegressor()
          n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
          print(n_estimators)
          # Number of trees in random forest
          n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
          # Number of features to consider at every split
          max_features = ['auto', 'sqrt']
          # Maximum number of levels in tree
          max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
          # max_depth.append(None)
          # Minimum number of samples required to split a node
          min_samples_split = [2, 5, 10, 15, 100]
          # Minimum number of samples required at each leaf node
          min_samples_leaf = [1, 2, 5, 10]

          [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200]
```



```
In [31]:  random_grid = {'n_estimators': n_estimators,
                         'max_features': max_features,
                         'max_depth': max_depth,
                         'min_samples_split': min_samples_split,
                         'min_samples_leaf': min_samples_leaf}

          print(random_grid)

          {'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200], 'max_features': ['auto', 'sqrt'], 'max_dep
          th': [5, 10, 15, 20, 25, 30], 'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_leaf': [1, 2, 5, 10]}
```

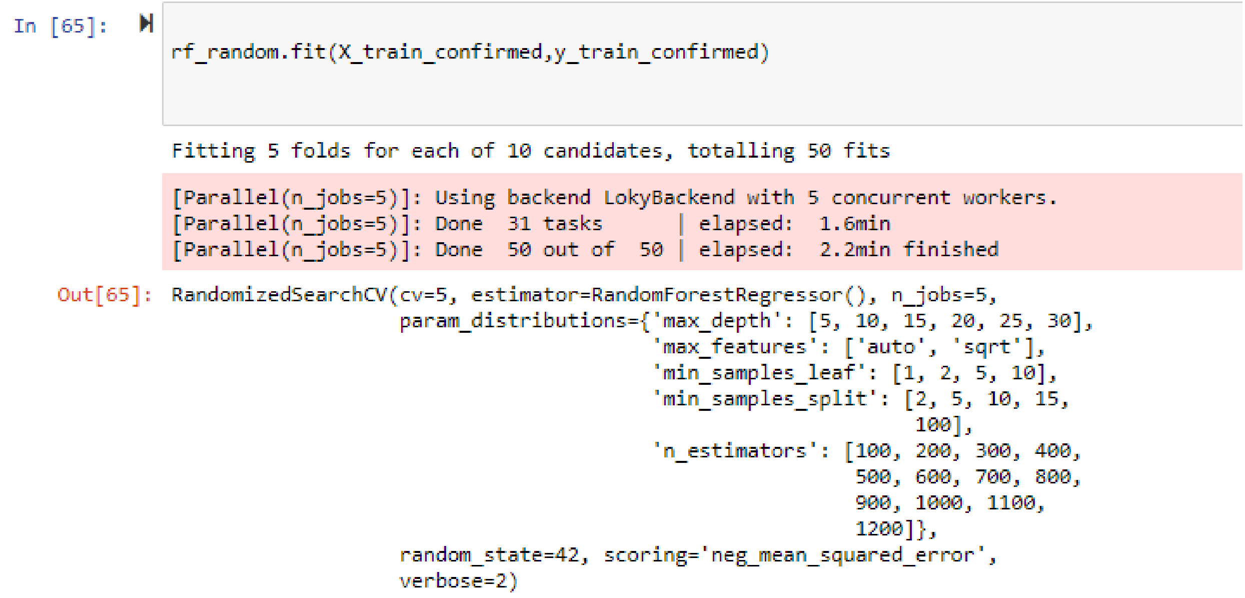


```
In [64]:  rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid,scoring='neg_mean_squared_error',
```



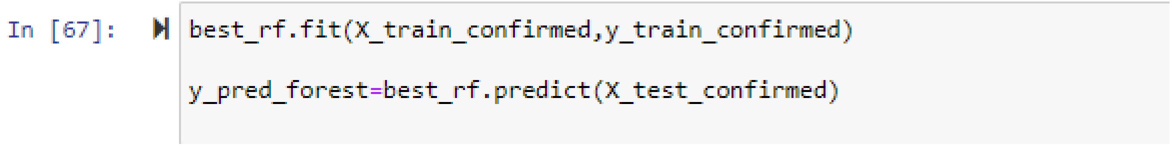
```
n_iter = 10, cv = 5, verbose=2, random_state=42,n_jobs=5)
```

Fit to find best parameters



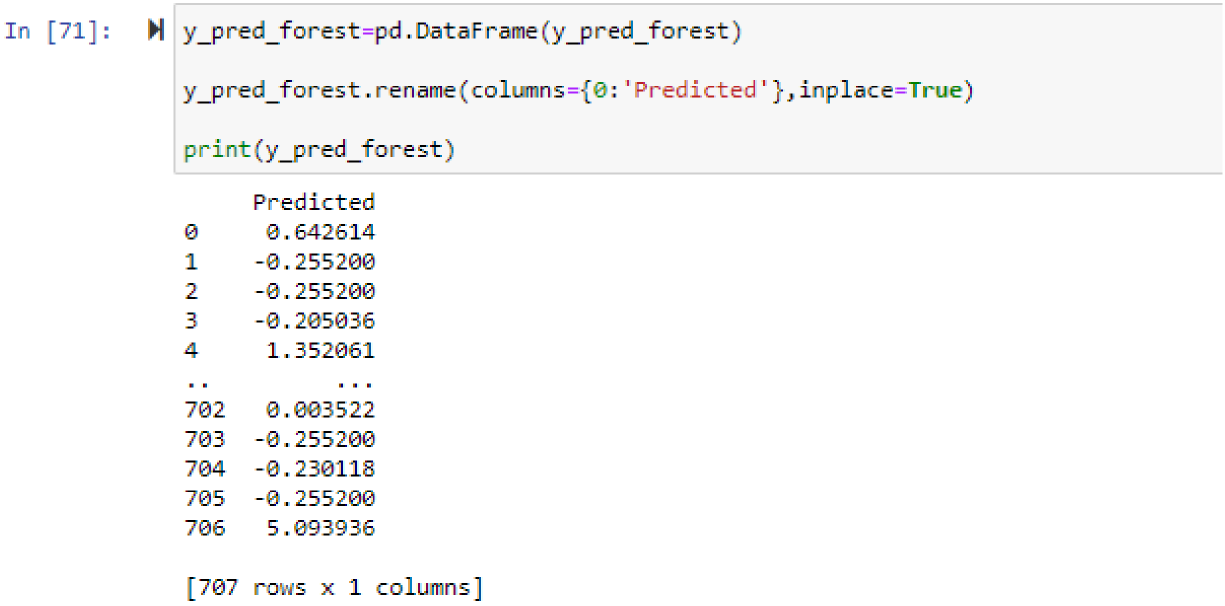
```
In [65]:  rf_random.fit(X_train_confirmed,y_train_confirmed)

          Fitting 5 folds for each of 10 candidates, totalling 50 fits

          [Parallel(n_jobs=5)]: Using backend LokyBackend with 5 concurrent workers.
          [Parallel(n_jobs=5)]: Done  31 tasks      | elapsed:  1.6min
          [Parallel(n_jobs=5)]: Done  50 out of  50 | elapsed:  2.2min finished

Out[65]:  RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=5,
                             param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                                  'max_features': ['auto', 'sqrt'],
                                                  'min_samples_leaf': [1, 2, 5, 10],
                                                  'min_samples_split': [2, 5, 10, 15,
                                                                        100],
                                                  'n_estimators': [100, 200, 300, 400,
                                                                   500, 600, 700, 800,
                                                                   900, 1000, 1100,
                                                                   1200]},
                             random_state=42, scoring='neg_mean_squared_error',
                             verbose=2)
```

Fit the model and find prediction:

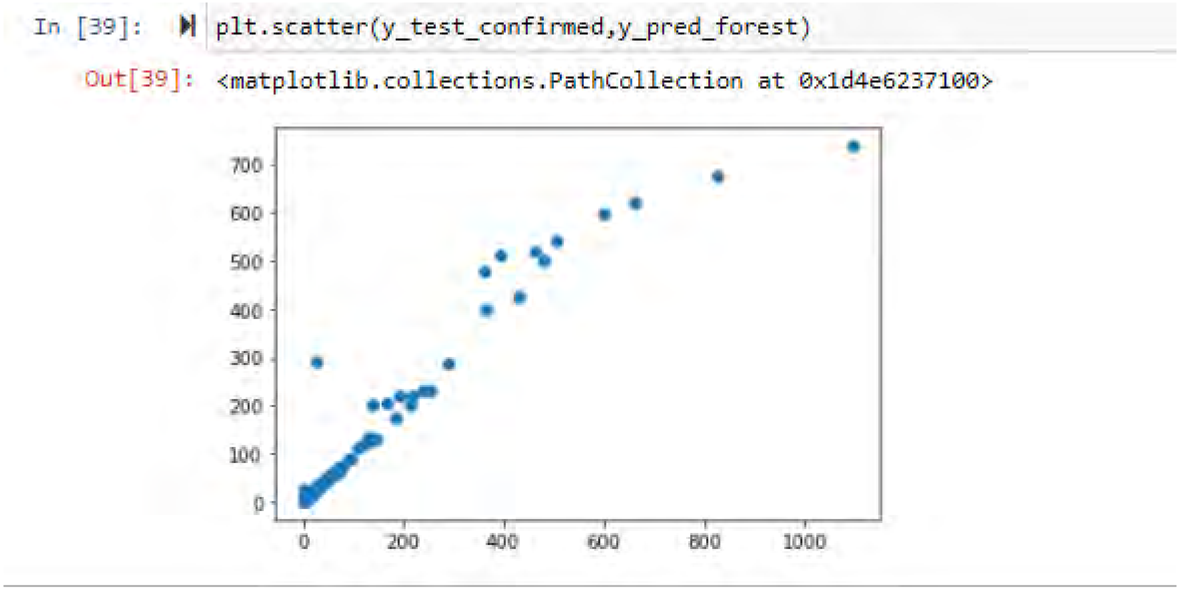


```
In [67]:  best_rf.fit(X_train_confirmed,y_train_confirmed)

          y_pred_forest=best_rf.predict(X_test_confirmed)
```

Predictions:



```
In [71]:  y_pred_forest=pd.DataFrame(y_pred_forest)

          y_pred_forest.rename(columns={0:'Predicted'},inplace=True)

          print(y_pred_forest)

     Predicted
0     0.642614
1    -0.255200
2    -0.255200
3    -0.205036
4     1.352061
..         ...
702   0.003522
703  -0.255200
704  -0.230118
705  -0.255200
706   5.093936

[707 rows x 1 columns]
```



```
In [39]:  plt.scatter(y_test_confirmed,y_pred_forest)

Out[39]: <matplotlib.collections.PathCollection at 0x1d4e6237100>
```

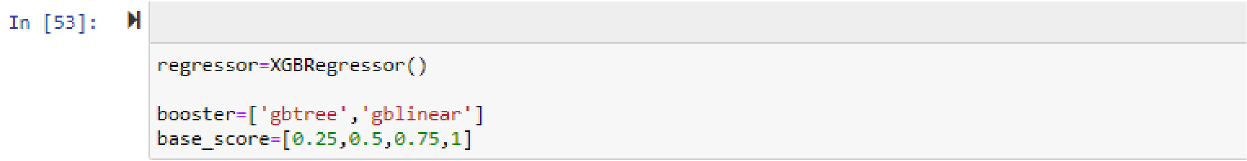
6.2.7 Xgboost Regression

XGBoost is a popular and efficient open-source implementation of the gradient boosted trees algorithm. Gradient boosting is a supervised learning algorithm, which attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models.

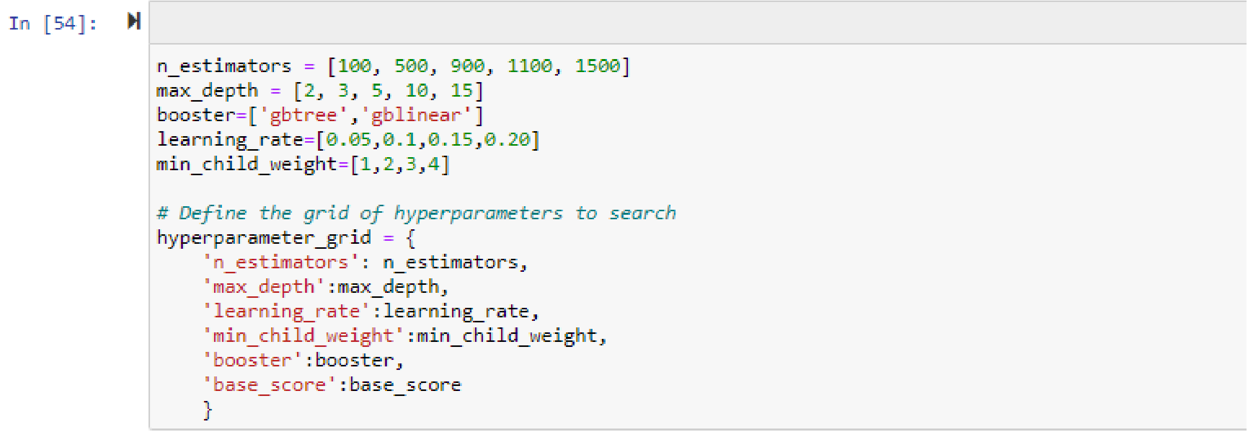
When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leafs that contains a continuous score. XGBoost minimizes a regularized (L1 and L2) objective function that combines a convex loss function (based on the difference between the predicted and target outputs) and a penalty term for model complexity (in other words, the regression tree functions).

The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees that are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

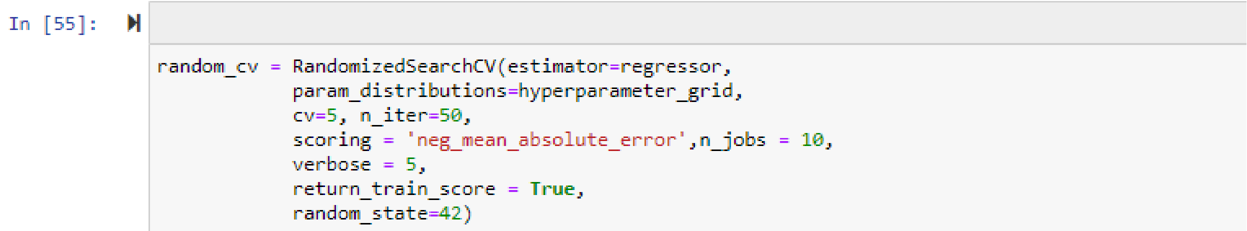
Hyperparameter tuning using Randomized Searchcv



```
In [53]:

regressor=XGBRegressor()

booster=['gbtree','gblinear']
base_score=[0.25,0.5,0.75,1]
```

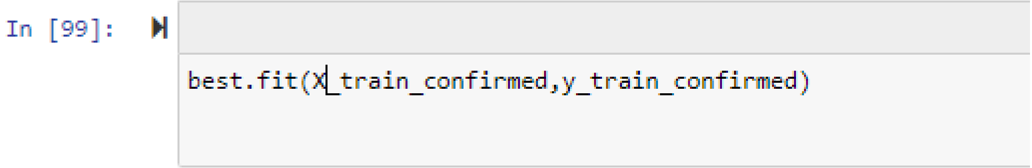


```
In [54]:

n_estimators = [100, 500, 900, 1100, 1500]
max_depth = [2, 3, 5, 10, 15]
booster=['gbtree','gblinear']
learning_rate=[0.05,0.1,0.15,0.20]
min_child_weight=[1,2,3,4]

# Define the grid of hyperparameters to search
hyperparameter_grid = {
    'n_estimators': n_estimators,
    'max_depth':max_depth,
    'learning_rate':learning_rate,
    'min_child_weight':min_child_weight,
    'booster':booster,
    'base_score':base_score
    }
```



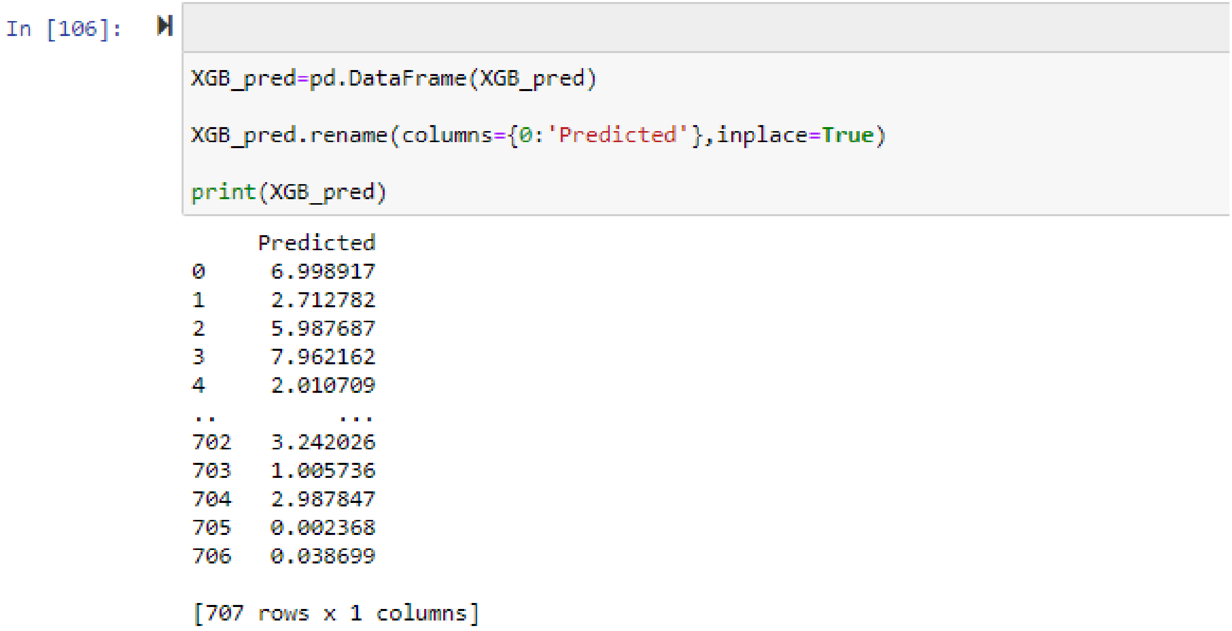
```
In [55]:

random_cv = RandomizedSearchCV(estimator=regressor,
            param_distributions=hyperparameter_grid,
            cv=5, n_iter=50,
            scoring = 'neg_mean_absolute_error',n_jobs = 10,
            verbose = 5,
            return_train_score = True,
            random_state=42)
```

Fitting Model

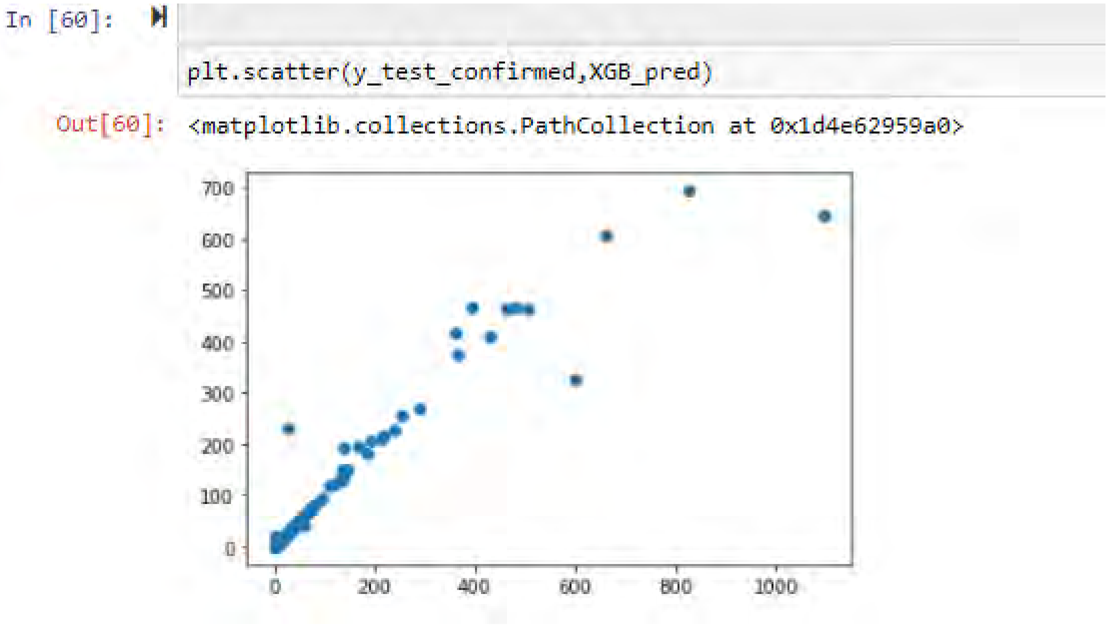


```
In [99]:

best.fit(X_train_confirmed,y_train_confirmed)
```

Predictions



```
In [106]:  XGB_pred=pd.DataFrame(XGB_pred)

           XGB_pred.rename(columns={0:'Predicted'},inplace=True)

           print(XGB_pred)

                Predicted
           0     6.998917
           1     2.712782
           2     5.987687
           3     7.962162
           4     2.010709
           ..         ...
           702   3.242026
           703   1.005736
           704   2.987847
           705   0.002368
           706   0.038699

           [707 rows x 1 columns]
```



```
In [60]:  plt.scatter(y_test_confirmed,XGB_pred)

Out[60]:  <matplotlib.collections.PathCollection at 0x1d4e62959a0>
```

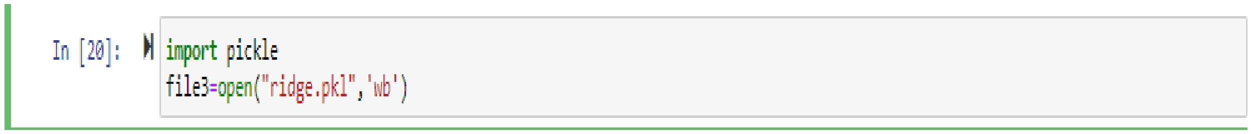
Chapter No 7

Model Deployment

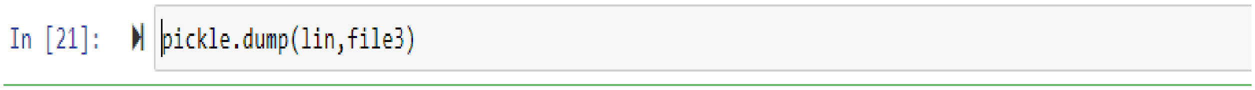
In this chapter I will discuss all the process which I take to deploy my model.I use Flask for deployment of my model which will pick the result from pickle file and show to the end user.

7.1 Store Result In Pickle File

Firstly I have tried many models on dataset and after getting best accuracy model we store thosemodel result in pickle file as shown below. All of these steps have done in jupyter notebook.



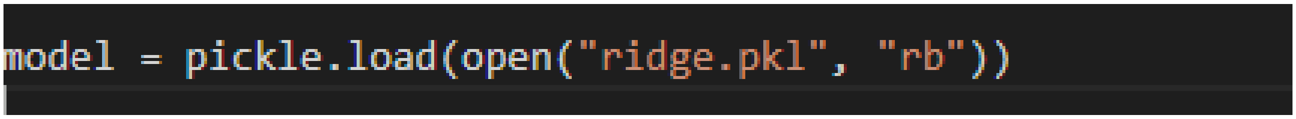
```
In [20]: import pickle
         file3=open("ridge.pkl",'wb')
```



```
In [21]: pickle.dump(lin,file3)
```

7.2 Read data from Pickle File

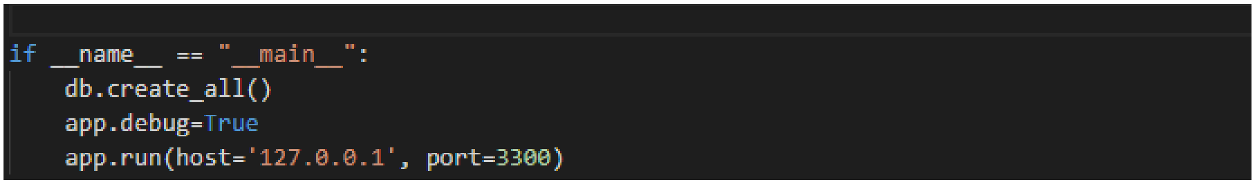
Next I read data from the pickle file and use it in prediction of the data which will be entered by the user.



```
model = pickle.load(open("ridge.pkl", "rb"))
```

7.3 Running Port

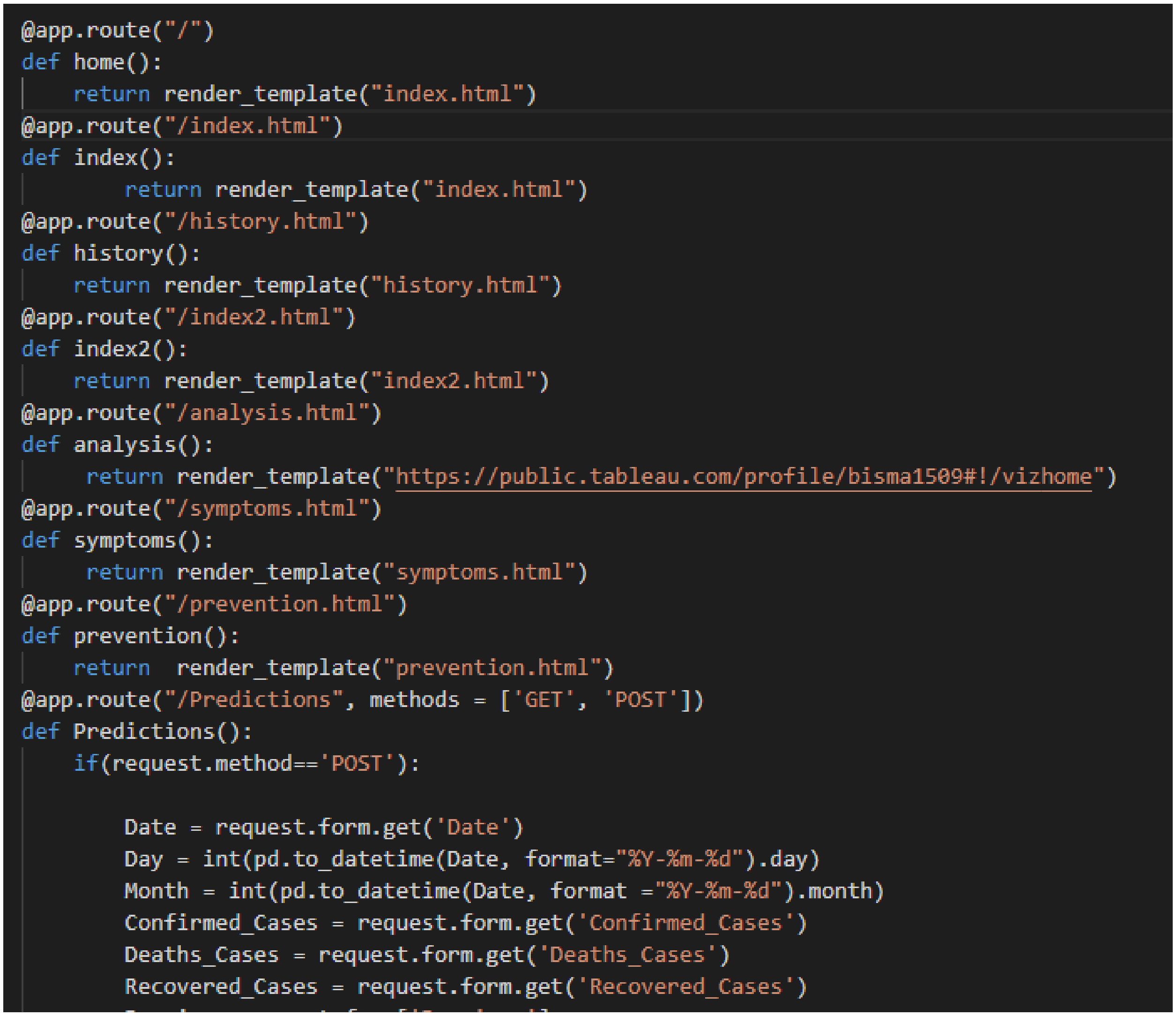
When I run the code the browser is open on port 3300 which code are given below.



```
if __name__ == "__main__":
    db.create_all()
    app.debug=True
    app.run(host='127.0.0.1', port=3300)
```

7.4 Routing for moving pages

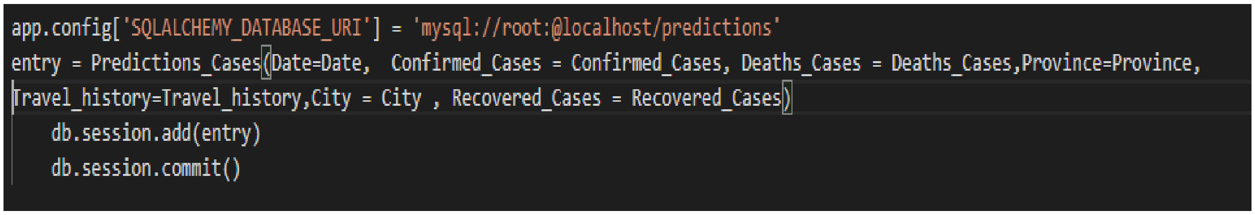
Here are the code through which pages can be moved. User can click on their desired page and with the help of these code selected pages will be displayed to end user.



```
@app.route("/")
def home():
    return render_template("index.html")
@app.route("/index.html")
def index():
        return render_template("index.html")
@app.route("/history.html")
def history():
    return render_template("history.html")
@app.route("/index2.html")
def index2():
    return render_template("index2.html")
@app.route("/analysis.html")
def analysis():
     return render_template("https://public.tableau.com/profile/bisma1509#!/vizhome")
@app.route("/symptoms.html")
def symptoms():
     return render_template("symptoms.html")
@app.route("/prevention.html")
def prevention():
    return  render_template("prevention.html")
@app.route("/Predictions", methods = ['GET', 'POST'])
def Predictions():
    if(request.method=='POST'):

        Date = request.form.get('Date')
        Day = int(pd.to_datetime(Date, format="%Y-%m-%d").day)
        Month = int(pd.to_datetime(Date, format ="%Y-%m-%d").month)
        Confirmed_Cases = request.form.get('Confirmed_Cases')
        Deaths_Cases = request.form.get('Deaths_Cases')
        Recovered_Cases = request.form.get('Recovered_Cases')
```

7.5 Store Result In Database

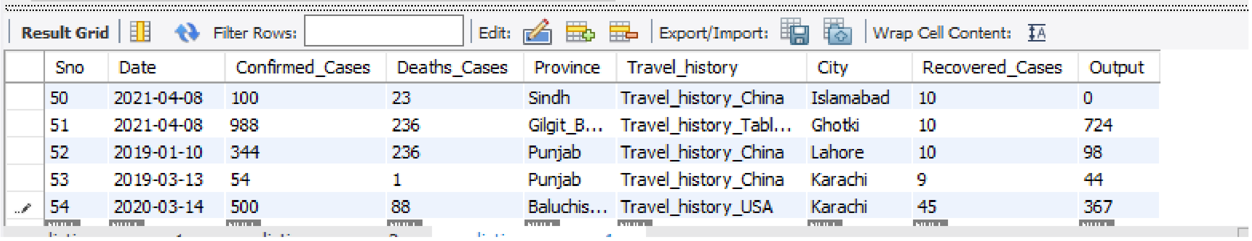
As database plays an important role in every project because data can be saved for long time in database. Anytime we can access data from database and use it for other purposes.



```
app.config['SQLALCHEMY_DATABASE_URI'] = 'mysql://root:@localhost/predictions'
entry = Predictions_Cases(Date=Date,  Confirmed_Cases = Confirmed_Cases, Deaths_Cases = Deaths_Cases,Province=Province,
Travel_history=Travel_history,City = City , Recovered_Cases = Recovered_Cases)
    db.session.add(entry)
    db.session.commit()
```

Result of Database

Below are the figure which shows the data entered by user into database.



Sno	Date	Confirmed_Cases	Deaths_Cases	Province	Travel_history	City	Recovered_Cases	Output
50	2021-04-08	100	23	Sindh	Travel_history_China	Islamabad	10	0
51	2021-04-08	988	236	Gilgit_B...	Travel_history_Tabl...	Ghotki	10	724
52	2019-01-10	344	236	Punjab	Travel_history_China	Lahore	10	98
53	2019-03-13	54	1	Punjab	Travel_history_China	Karachi	9	44
54	2020-03-14	500	88	Baluchis...	Travel_history_USA	Karachi	45	367

Chapter No 8

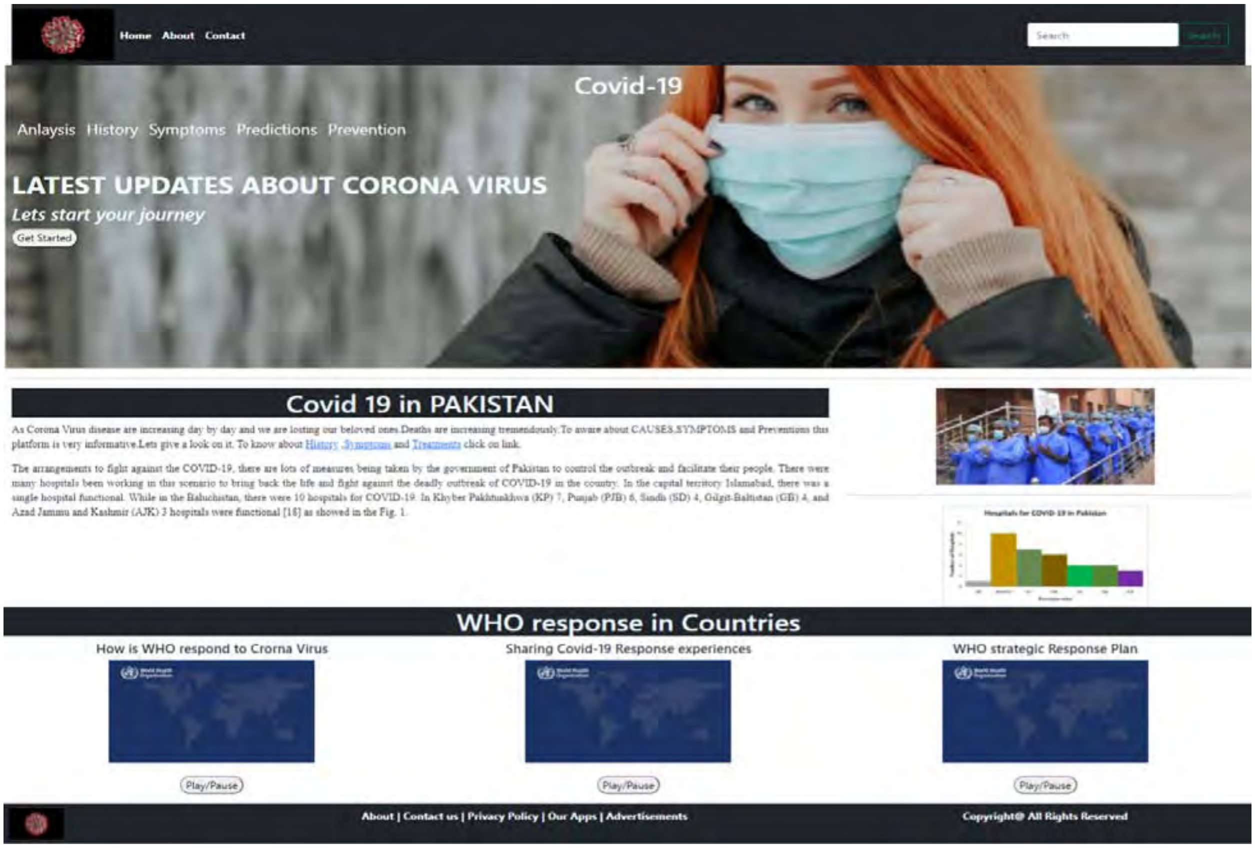
Interfaces

8.1 Introductions

Screen images are interfaces through which user interacts with the system .A user interface, also called a “UI” or simply an “interface” is the mean through which a person controls a software application or hardware device. A good user interface provides a “user friendly” experience, allowing the user to interact with the software or hardware in a natural or intuitive way. The User interface is one of most important parts of any program because it determines how easily you can make the program do what you want. A powerful program with poorly designed user interface has little value. Graphical User interface (GUI) that has windows ,icons , pop-up menus have become standard on personal computers.

8.2 Home Interface

Here is the first page the user first seen on the website.



Covid 19 in PAKISTAN

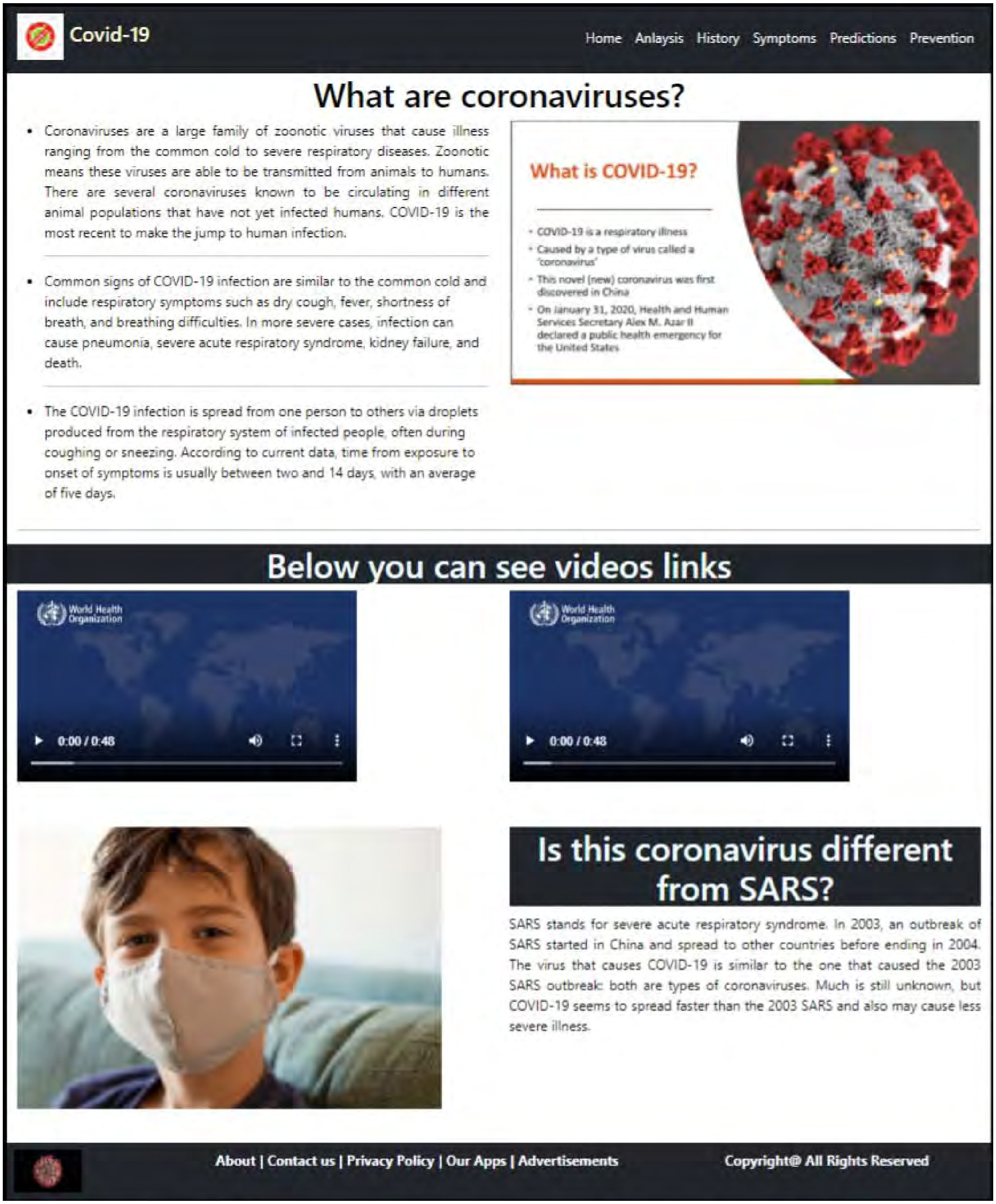
As Corona Virus disease are increasing day by day and we are losting our beloved ones.Deaths are increasing tremendously.To aware about CAUSES,SYMPTOMS and Preventions this platform is very informative.Lets give a look on it. To know about History ,Symptom and Treatments click on link.

The arrangements to fight against the COVID-19, there are lots of measures being taken by the government of Pakistan to control the outbreak and facilitate their people. There were many hospitals been working in this scenario to bring back the life and fight against the deadly outbreak of COVID-19 in the country. In the capital territory Islamabad, there was a single hospital functional. While in the Baluchistan, there were 10 hospitals for COVID-19. In Khyber Pakhtunkhwa (KP) 7, Punjab (PJB) 6, Sindh (SD) 4, Gilgit-Baltistan (GB) 4, and Azad Jammu and Kashmir (AJK) 3 hospitals were functional [16] as showed in the Fig. 1.

Hospitals for COVID-19 in Pakistan
WHO response in Countries
How is WHO respond to Crorna Virus
Sharing Covid-19 Response experiences
WHO strategic Response Plan
Play/Pause
Play/Pause
Play/Pause
About | Contact us | Privacy Policy | Our Apps | Advertisements
Copyright@ All Rights Reserved

8.3 History Page Interface

Below are the page where user can see history of the Corona Virus.It also includes video links so that user can easily understand the history of current panademic



What are coronaviruses?

- Coronaviruses are a large family of zoonotic viruses that cause illness ranging from the common cold to severe respiratory diseases. Zoonotic means these viruses are able to be transmitted from animals to humans. There are several coronaviruses known to be circulating in different animal populations that have not yet infected humans. COVID-19 is the most recent to make the jump to human infection.
- Common signs of COVID-19 infection are similar to the common cold and include respiratory symptoms such as dry cough, fever, shortness of breath, and breathing difficulties. In more severe cases, infection can cause pneumonia, severe acute respiratory syndrome, kidney failure, and death.
- The COVID-19 infection is spread from one person to others via droplets produced from the respiratory system of infected people, often during coughing or sneezing. According to current data, time from exposure to onset of symptoms is usually between two and 14 days, with an average of five days.

What is COVID-19?

- COVID-19 is a respiratory illness
- Caused by a type of virus called a 'coronavirus'
- This novel (new) coronavirus was first discovered in China
- On January 31, 2020, Health and Human Services Secretary Alex M. Azar II declared a public health emergency for the United States

Below you can see videos links

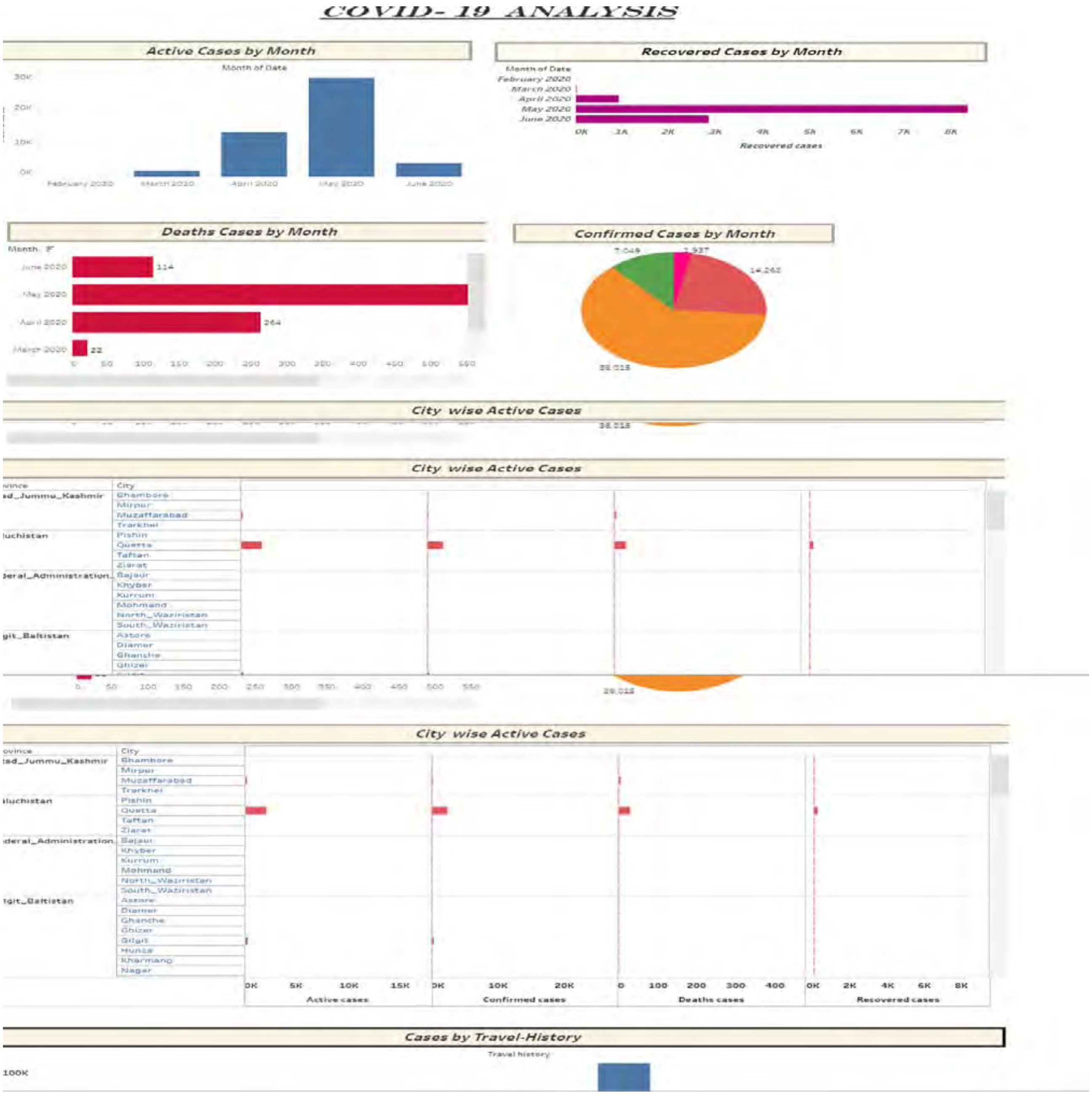
Is this coronavirus different from SARS?

SARS stands for severe acute respiratory syndrome. In 2003, an outbreak of SARS started in China and spread to other countries before ending in 2004. The virus that causes COVID-19 is similar to the one that caused the 2003 SARS outbreak: both are types of coronaviruses. Much is still unknown, but COVID-19 seems to spread faster than the 2003 SARS and also may cause less severe illness.

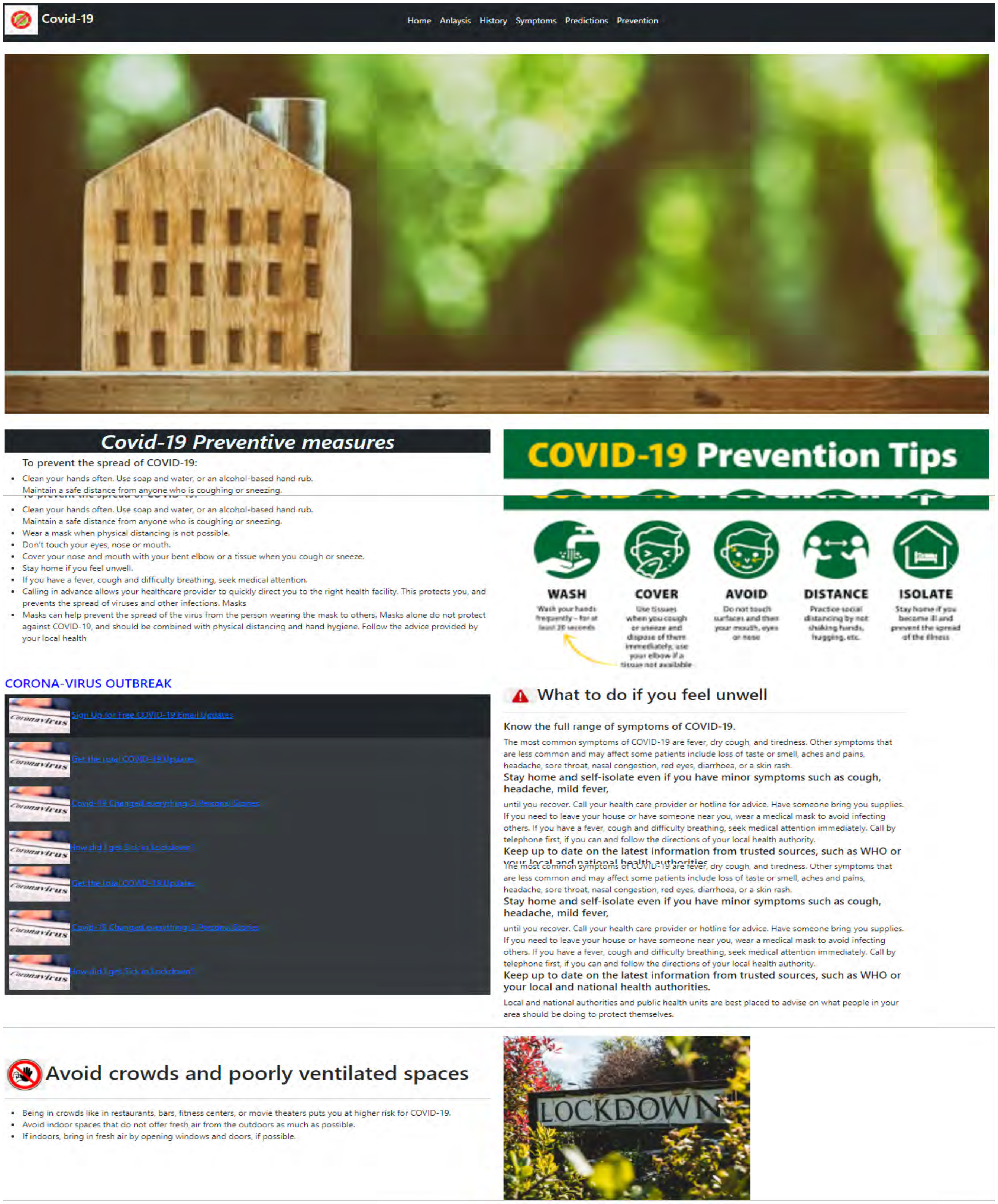
About | Contact us | Privacy Policy | Our Apps | Advertisements
Copyright@ All Rights Reserved

8.4 Analysis Page Interface

Here are the most important page in my website. As we can see that people(end user) can get quick information with the help of graphs and charts. I have used Tableau dashboard for better visualization of the data.



8.5 Prevention Page interface



Covid-19 Preventive measures

To prevent the spread of COVID-19:

- Clean your hands often. Use soap and water, or an alcohol-based hand rub. Maintain a safe distance from anyone who is coughing or sneezing.

To prevent the spread of COVID-19:

- Clean your hands often. Use soap and water, or an alcohol-based hand rub. Maintain a safe distance from anyone who is coughing or sneezing.
- Wear a mask when physical distancing is not possible.
- Don't touch your eyes, nose or mouth.
- Cover your nose and mouth with your bent elbow or a tissue when you cough or sneeze.
- Stay home if you feel unwell.
- If you have a fever, cough and difficulty breathing, seek medical attention.
- Calling in advance allows your healthcare provider to quickly direct you to the right health facility. This protects you, and prevents the spread of viruses and other infections. Masks
- Masks can help prevent the spread of the virus from the person wearing the mask to others. Masks alone do not protect against COVID-19, and should be combined with physical distancing and hand hygiene. Follow the advice provided by your local health

CORONA-VIRUS OUTBREAK

- Sign Up for Free COVID-19 Email Updates
- Get the total COVID-19 Updates
- Covid-19 Changed everything: 3 Personal Stories
- How did I get Sick in Lockdown?
- Get the total COVID-19 Updates
- Covid-19 Changed everything: 3 Personal Stories
- How did I get Sick in Lockdown?

COVID-19 Prevention Tips

WASH
Wash your hands frequently – for at least 20 seconds
COVER
Use tissues when you cough or sneeze and dispose of there immediately; use your elbow if a tissue not available
AVOID
Do not touch surfaces and then your mouth, eyes or nose
DISTANCE
Practice social distancing by not shaking hands, hugging, etc.
ISOLATE
Stay home if you become ill and prevent the spread of the illness

What to do if you feel unwell

Know the full range of symptoms of COVID-19.

The most common symptoms of COVID-19 are fever, dry cough, and tiredness. Other symptoms that are less common and may affect some patients include loss of taste or smell, aches and pains, headache, sore throat, nasal congestion, red eyes, diarrhoea, or a skin rash.

Stay home and self-isolate even if you have minor symptoms such as cough, headache, mild fever,

until you recover. Call your health care provider or hotline for advice. Have someone bring you supplies. If you need to leave your house or have someone near you, wear a medical mask to avoid infecting others. If you have a fever, cough and difficulty breathing, seek medical attention immediately. Call by telephone first, if you can and follow the directions of your local health authority.

Keep up to date on the latest information from trusted sources, such as WHO or your local and national health authorities

The most common symptoms of COVID-19 are fever, dry cough, and tiredness. Other symptoms that are less common and may affect some patients include loss of taste or smell, aches and pains, headache, sore throat, nasal congestion, red eyes, diarrhoea, or a skin rash.

Stay home and self-isolate even if you have minor symptoms such as cough, headache, mild fever,

until you recover. Call your health care provider or hotline for advice. Have someone bring you supplies. If you need to leave your house or have someone near you, wear a medical mask to avoid infecting others. If you have a fever, cough and difficulty breathing, seek medical attention immediately. Call by telephone first, if you can and follow the directions of your local health authority.

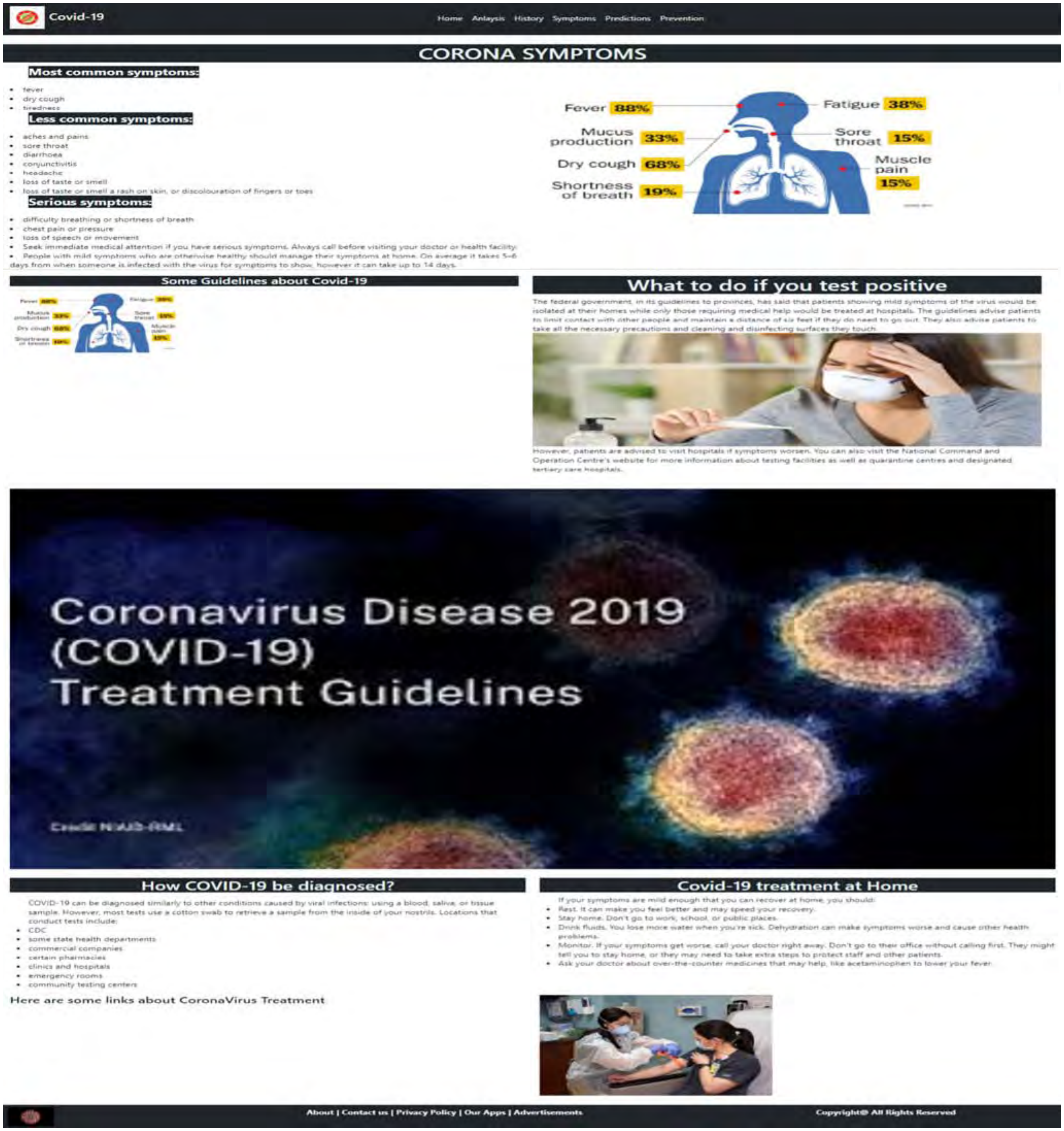
Keep up to date on the latest information from trusted sources, such as WHO or your local and national health authorities.

Local and national authorities and public health units are best placed to advise on what people in your area should be doing to protect themselves.

Avoid crowds and poorly ventilated spaces

- Being in crowds like in restaurants, bars, fitness centers, or movie theaters puts you at higher risk for COVID-19.
- Avoid indoor spaces that do not offer fresh air from the outdoors as much as possible.
- If indoors, bring in fresh air by opening windows and doors, if possible.

8.6 Symptoms page Interface



CORONA SYMPTOMS

Most common symptoms:

- fever
- dry cough
- tiredness

Less common symptoms:

- aches and pains
- sore throat
- diarrhoea
- conjunctivitis
- headache
- loss of taste or smell
- loss of taste or smell a rash on skin, or discolouration of fingers or toes

Serious symptoms:

- difficulty breathing or shortness of breath
- chest pain or pressure
- loss of speech or movement
- Seek immediate medical attention if you have serious symptoms. Always call before visiting your doctor or health facility.
- People with mild symptoms who are otherwise healthy should manage their symptoms at home. On average it takes 5–6 days from when someone is infected with the virus for symptoms to show, however it can take up to 14 days.

Some Guidelines about Covid-19

What to do if you test positive

The federal government, in its guidelines to provinces, has said that patients showing mild symptoms of the virus would be isolated at their homes while only those requiring medical help would be treated at hospitals. The guidelines advise patients to limit contact with other people and maintain a distance of six feet if they do need to go out. They also advise patients to take all the necessary precautions and cleaning and disinfecting surfaces they touch.

However, patients are advised to visit hospitals if symptoms worsen. You can also visit the National Command and Operation Centre's website for more information about testing facilities as well as quarantine centres and designated tertiary care hospitals.

How COVID-19 be diagnosed?

COVID-19 can be diagnosed similarly to other conditions caused by viral infections: using a blood, saliva, or tissue sample. However, most tests use a cotton swab to retrieve a sample from the inside of your nostrils. Locations that conduct tests include:

- CDC
- some state health departments
- commercial companies
- certain pharmacies
- clinics and hospitals
- emergency rooms
- community testing centers

Here are some links about CoronaVirus Treatment

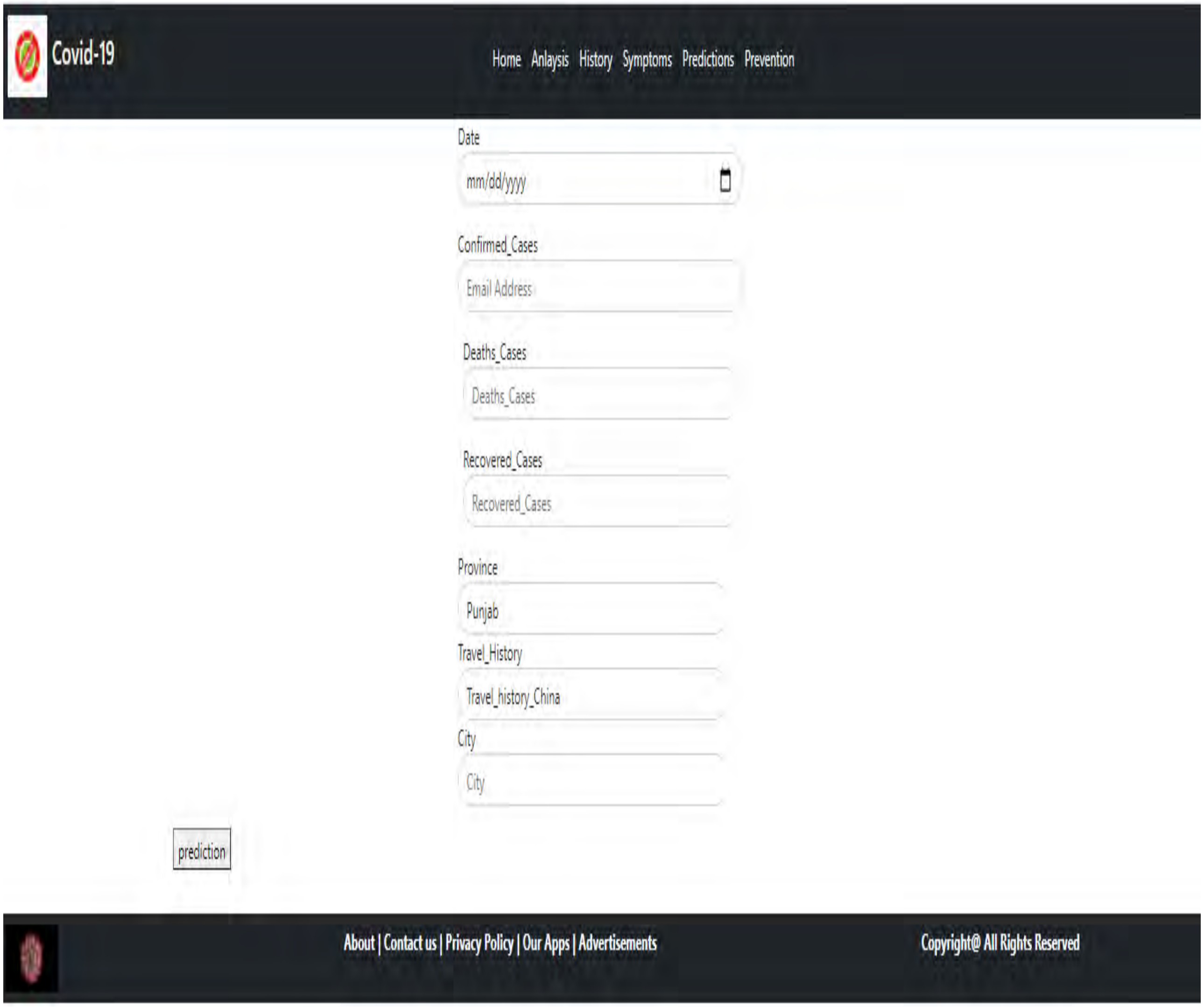
Covid-19 treatment at Home

If your symptoms are mild enough that you can recover at home, you should:

- Rest. It can make you feel better and may speed your recovery.
- Stay home. Don't go to work, school, or public places.
- Drink fluids. You lose more water when you're sick. Dehydration can make symptoms worse and cause other health problems.
- Monitor. If your symptoms get worse, call your doctor right away. Don't go to their office without calling first. They might tell you to stay home, or they may need to take extra steps to protect staff and other patients.
- Ask your doctor about over-the-counter medicines that may help, like acetaminophen to lower your fever.

8.7 Predictions

Below are the page through which user can get quick information about the ActiveCases of a City or a Province .User give different inputs and through machine learning model this page give predicted(ActiveCases) result to the user.


Covid-19
Home Anlaysis History Symptoms Predictions Prevention
prediction
About | Contact us | Privacy Policy | Our Apps | Advertisements
Copyright@ All Rights Reserved

Date
mm/dd/yyyy

Confirmed_Cases
Email Address

Deaths_Cases
Deaths_Cases

Recovered_Cases
Recovered_Cases

Province
Punjab

Travel_History
Travel_history_China

City
City

Chapter No 9

Testing and Evaluation

9.1 Introduction:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a weak product.

Software testing is the process of verifying a system with the purpose of identifying any errors, gaps or missing requirement versus the actual requirement.

Software testing is categorized into two types :

- Functional testing
- Non functional testing

When to start test activities:

Testing should be started as early as possible to reduce the cost and time to rework and produce software that is bug-free so that it can be delivered to the client. However, in Software Development Life Cycle (SDLC), testing can be started from the Requirements Gathering phase and continued till the software is out there in productions. It also depends on the development model that is being used. For example, in the Waterfall model, testing starts from the testing phase which is quite below in the tree,; but in the V-model, testing is performed parallel to the development phase.

9.2 Software Testing Methodologies:

Are the various strategies or approaches used to test an application to ensure it behaves and looks as expected. Theses encompasses everything from front to back end testing including unit testing and system testing.

.9.2.1 Functional Testing

Functional testing involves testing the application against the business requirements. It incorporates all test types designed to guarantee each part of a piece of software behaves as expected by using uses cases provided by the design team or business analyst. These testing methods are usually conducted in order and include:

9.2.1.1 Unit testing

Unit testing is the first level of testing and is often performed by the developers themselves. It is the process of ensuring individual components of a piece of software at the code level are functional and work as they were designed to. Developers in a test-driven environment will typically write and run the tests prior to the software or feature being passed over to the test team. Unit testing can be conducted manually, but automating the process will speed up delivery cycles and expand test coverage. Unit testing will also make debugging easier because finding issues earlier means they take less time to fix than if they were discovered later in the testing process.

9.2.2.2 Integration testing

After each unit is thoroughly tested, it is integrated with other units to create modules or components that are designed to perform specific tasks or activities. These are then tested as group through integration testing to ensure whole segments of an application behave as expected (i.e, the interactions between units are seamless). These tests are often framed by user scenarios, such as logging into an application or opening files. Integrated tests can be conducted by either developers or independent testers and are usually comprised of a combination of automated functional and manual tests.

9.2.1.3System testing

System testing is a black box testing method used to evaluate the completed and integrated system, as a whole, to ensure it meets specified requirements. The functionality of the software is tested from end-to-end and is typically conducted by a separate testing team than the development team before the product is pushed into production.

9.2.1.4 Acceptance testing

Acceptance testing is the last phase of functional testing and is used to assess whether or not the final piece of software is ready for delivery. It involves ensuring that the product is in compliance with all of the original business criteria and that it meets the end user’s needs. This requires the product be tested both internally and externally, meaning you’ll need to get it into the hands of your end users for beta testing along with those of your QA team. Beta testing is key to getting real feedback from potential customers and can address any final usability concerns.

9.2.2 Non-Functional Testing

Non-functional testing methods incorporate all test types focused on the operational aspects of a piece of software. These include:

- Performance testing
- Security testing
- Usability testing
- Compatibility testing

9.2.2.1 Performance Testing

is a non-functional testing technique used to determine how an application will behave under various conditions. The goal is to test its responsiveness and stability in real user situations. Performance testing can be broken down into four types:

- **Load testing** is the process of putting increasing amounts of simulated demand on your software, application, or website to verify whether or not it can handle what it's designed to handle.
- **Stress testing** takes this a step further and is used to gauge how your software will respond at or beyond its peak load. The goal of stress testing is to overload the application on purpose until it breaks by applying both realistic and unrealistic load scenarios. With stress testing, you'll be able to find the failure point of your piece of software.
- **Endurance testing,** also known as soak testing, is used to analyze the behaviorof an application under a specific amount of simulated load over longer amounts of time. The goal is to understand how your system will behave under sustained use, making it a longer process than load or stress testing (which are designed to end after a few hours). A critical piece of endurance testing is that it helps uncover memory leaks.
- **Spike testing** is a type of load test used to determine how your software will respond to substantially larger bursts of concurrent user or system activity over varying amounts of time. Ideally, this will help you understand what will happen when the load is suddenly and drastically increased.

9.2.2.2.Security Testing

With the rise of cloud-based testing platforms and cyber attacks, there is a growing concern and need for the security of data being used and stored in software. Security testing is a non-functional software testing technique used to determine if the information and data in a system is protected. The goal is to purposefully find loopholes and security risks in the system that could result in unauthorized access to or the loss of information by probing the application for weaknesses. There are multiple types of this testing method, each of which aimed at verifying six basic principles of security:

- Integrity
- Confidentiality
- Authentication
- Authorization
- Availability
- Non-repudiation

9.2.2.3 Usability Testing

Usability testing is a testing method that measures an application's ease-of-use from the end-user perspective and is often performed during the system or acceptance testing stages. The goal is to determine whether or not the visible design and aesthetics of an application meet the intended workflow for various processes, such as logging into an application. Usability testing is a great way for teams to review separate functions, or the system as a whole, is intuitive to use.

9.2.2.4 Compatibility Testing

Compatibility testing is used to gauge how an application or piece of software will work in different environments. It is used to check that your product is compatible with multiple operating systems, platforms, browsers, or resolution configurations. The goal is to ensure that your software's functionality is consistently supported across any environment you expect your end users to be using.

9.2.3 White Box Testing

White Box Testing is software testing technique in which internal structure, design and coding of software are tested to verify flow of input-output and to improve design, usability and security. In white box testing, code is visible to testers so it is also called Clear box testing, Open box testing, Transparent box testing, Code-based testing and Glass box testing.

9.2.4 Black Box Testing

Black Box Testing is a software testing method in which the functionalities of software applications are tested without having knowledge of internal code structure, implementation details and internal paths. Black Box Testing mainly focuses on input and output of software applications and it is entirely based on software requirements and specifications. It is also known as Behavioral Testing.

Input
Black Box
Output

Test Case:

Table 9.1: Generic Test Case

Tester	Tester Name?
Test Type	What Testing Technique is used?
Test Case Name	Name of the test case.
Description	Description of functional requirement.
Procedure	Describes the steps of that function.
Expected Result	What should it do?
Actual Result	What it did?
Status	Success or Fail?

9.3 Test Cases of Project:

The Test cases planned for testing this system as follows:

9.3.1 Test Case 1:

Table 9.1: Test Case Model Fitting

Tester	Bisma Ishfaq
Test Type	Black Box Testing
Test Case Name	Model Development test
Description	Purpose of this test is whether the applied model is properly fitting.
Procedure	I fit different models on my data and then check whether it is fitting succcessfully and show predictions
Expected result	Successfully fit model and predict result.
Actual Result	Successful
Status	Success

9.3.2 Test Case 2:

Table 9.2: Test Case Database Insertion

Tester	Bisma Ishfaq
Test Type	Black Box Testing
Test Case Name	Database entries
Description	Purpose of this test is whether the user input is stored in database or not.
Procedure	User give inputs (Date, Confirmed Cases, Deaths Cases, Recovered Cases,Province , City, travel history)and click Prediction button.
Expected result	Successfully insert datainto database.
Actual Result	Successful
Status	Success

9.3.3 Test Case 3 :

Table 9.3: Test Case Shows Predictions(Output)

Tester	Bisma Ishfaq
Test Type	Black Box Testing
Test Case Name	Model Predictions
Description	Purpose of this test is to see when the user give inputs the website give prediction result to end user.
Procedure	User give different inputs and click on submit button it shows prediction results.
Expected result	Successfully show predictions result..
Actual Result	Successful
Status	Success

9.3.4 Test Case 4 :

Table 9.4: Test Case to Check Errors

Tester	Bisma Ishfaq
Test Type	White Box Testing
Test Case Name	Checking Bug and Errors
Description	Purpose of this test is to see whether code is running properly and it is bug free.
Procedure	I run code line by line to see whether any error is present in my code or it is working properly.
Expected result	Code Running Successfully.
Actual Result	Successful
Status	Success

Chapter No 10

Conclusions & Future Work

10.1 Conclusions

In this project, a detailed analysis of Covid-19 Cases in Different Cities is performed. I also, apply different prediction models (Regression) were trained using few machine learning algorithms but linear regression gives a good as compared to others.

I have also used latest technologies such as (FLASK) for the deployment of my model. I have also develop a website using **Javascript** and **Bootstrap**. I have also made connection to database (**MYSQL**) which store the user inputs into the database.

10.2 Future Work

As a part of Future work, It is suggested to add more data into dataset and also insert different factors such as Quarantine days, Vaccination etc. Also model used on this dataset can be applied on different Covid-19 datasets which cover worldwide Covid-19 data.

10.3 References:

Retrieved from https://code.visualstudio.com/

Retrieved from https://www.kaggle.com/zusmani/pakistan-corona-virus-citywise-data

Retrieved from https://www.mysql.com/products/workbench/

Retrieved from https://www.coursera.org/professional-certificates/ibm-data-science

Retrieved from https://www.youtube.com/watch?v=cvvwkgp4HBg&list=PLu0W_9lII9ajyk081To1Cbt2eI5913SsL

The End