Training Project Report

On

COVID-19 analysis and prediction

In partial fulfillment of requirements for the degree

of

BACHELOR OF

TECHNOLOGY IN

INFORMATION TECHNOLOGY/
COMPUTER SCIENCE

Submitted by:

ADITYA CHOUHAN [17100BTCSE01203]
KRATIKA KOTHARI [17100BTIT01437]
MANAN SETHI [17100BTIT01439]
SANYA TARE [17100BTIT01460]
SAPNESH KHANDELWAL[17100BTIT01461]

Under the guidance of

MS. MOUSITA DHAR
(Webtek labs Pvt Ltd)

SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA
INDORE, JULY-AUGUST 2020

CANDIDATE'S DECLARATION

We hereby declare that we have undertaken industrial training at “WEBTEK LABS PVT. LTD.” during a period from 27th July to 16th August 2020 in partial fulfilment of requirements for the award of degree of B.Tech at SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA. The work which is being presented in the training report submitted to Department of INFORMATION TECHNOLOGY/COMPUTER SCIENCE at SHRI VAISHNAV VIDYAPEETH VISHWAVIDYALAYA.

ACKNOWLEDGEMENT

It gives us great pleasure to acknowledge the guidance, assistance and support of Mrs. Mousita Dhar in making the Project and this Project report successful, which has been structured under her valued suggestion. She has helped us to accomplish the challenging task in a very short period of time. Finally, we express the constant support of our friends, family and professors for inspiring us throughout and encouraging us.

Aditya Chouhan
Kratika Kothari
Manan Sethi
Sanya Tare
Sapnesh Khandelwal

CERTIFICATE OF APPROVAL

The project “COVID-19 ANALYSIS ” made by the efforts of Aditya Chouhan, Kratika Kothari, Manan Sethi, Sanya Tare, Sapnesh Khandelwal is hereby approved as a creditable study for the Bachelor of Technology and presented in a manner of satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood that by this approval they undersigned this project only for the purpose for which it is submitted.

_____

Ms. Mousita
(Project In charge)

INTRODUCTION

1.1 PYTHON

About Python:

- Python is a high-level, general-purpose, open source, strictly typed programming language. The language provides constructs intended to enable clear programs on both a small and large scale.
- Python was created By Guido van Rossum.
- The Python Software Foundation (PSF) is the organization behind Python.

Python versions:

- First released in 1991.
- Python 2.0 was released on 16 October 2000
- Python 3.0 was released on 3 December 2008

Current Versions:

- 3.8

Python features:

- Easy to understand
- Dynamic
- Object oriented multipurpose
- Strongly typed
- Open Sourced

Python is mainly used in many domains:

- Web Development
- Data Analysis
- Machine Learning
- Internet Of Things
- GUI Development
- Image processing
- Data visualization
- Game Development

IDLE:

IDLE is an integrated development environment for Python, which has been bundled with the default implementation of the language.

1.2 ANACONDA

Anaconda is an open source Distribution for data science and machine learning using python. It includes hundreds of popular data science packages and the conda package and virtual environment manager for Windows, Linux, and MacOS. Conda makes it quick and easy to install, run, and upgrade complex data science and machine learning environments like scikitlearn, TensorFlow, and SciPy. Anaconda Distribution is the foundation of millions of data science projects as well as Amazon Web Service Machine Learning AMIs and Anaconda for Microsoft on Azure and Windows.

1.3 PACKAGES

1.3.1 NumPy

NumPy is the fundamental package for scientific computing with Python.
It contains among other things:

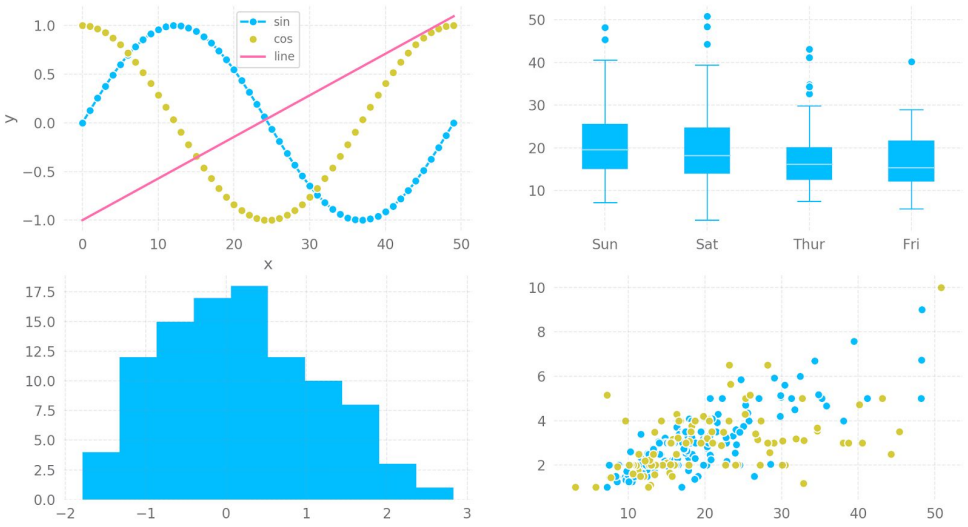
- a powerful N-dimensional array object
- sophisticated (broadcasting) functions • tools for integrating C/C++ and Fortran code
- useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

1.3.2 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits.

Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatterplots, etc., with just a few lines of code. For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.



1.3.3 Scikit-learn

Scikit-learn provides machine learning libraries for python.Some of the features of Scikit-learn includes:

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

1.3.4 Pandas

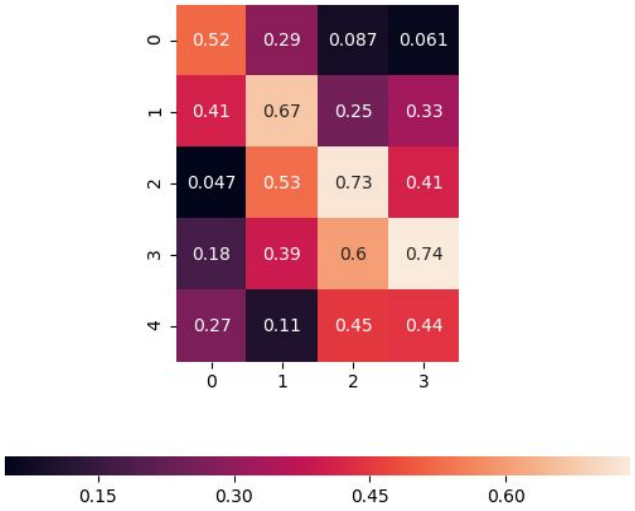
Pandas is an open source, BSD-licensed library providing higher performance, easy-to-use data structures and data analysis tools for the Python programming language.

Pandas library is well suited for data manipulation and analysis using python. In particular, it offers data structures and operations for manipulating numerical tables and time series.

1.3.5 Seaborn

Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

Eg:- Heatmap



2. TRAINING WORK UNDERTAKEN

2.1 Collecting data from kaggle

Kaggle is a platform for predictive modelling and analytics competitions in which statisticians and data miners compete to produce the best models for predicting and describing the datasets uploaded by companies and users. This crowdsourcing approach relies on the fact that there are countless strategies that can be applied to any predictive modelling task and it is impossible to know beforehand which technique or analyst will be most effective. On 8 March 2017.Google announced that they were acquiring Kaggle. They will join the Google Cloud team and continue to be a distinct brand. In January 2018, Booz Allen and Kaggle launched Data Science Bowl, a machine learning competition to analyze cell images and identify nuclei.

2.2 Data Science

Data science is an interdisciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from data in various forms, both structured and unstructured, similar to data mining. Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data. It employs techniques and theories drawn from many fields within the context of mathematics, statistics, information science, and computer science. Turing award winner JiGray imagined data science as a "fourth paradigm" of science (empirical, theoretical, computational and now data-driven) and asserted that "everything about science is changing because of the impact of information technology" and the data deluge. When Harvard Business Review called it "The Sexiest Job of the 21st Century" the term became a buzzword, and is now often applied to business analytics, business intelligence, predictive modeling, or any arbitrary use of data, or used as a glamorized term for statistics. In many cases, earlier approaches and solutions are now simply rebranded as "data science" to be more attractive, which can cause the term to become "dilute[d] beyond usefulness." While many university programs now offer a data science degree, there exists no consensus on a definition or suitable curriculum contents. Because of the current

popularity of this term, there are many "advocacy efforts" surrounding the field. To its discredit, however, many data science and big data projects fail to deliver useful results, often as a result of poor management and utilization of resources.

2.3 Project Description

Coronavirus(COVID-19) has become the most buzzed topic these days. COVID-19 is the disease caused by the new coronavirus that emerged in China in December 2019. The source of this virus is believed to be a 'wet market' in Wuhan which sold both dead and live animals including fishes and birds.

COVID-19 symptoms include cough, fever, shortness of breath,dry cough, headache, pneumonia. COVID-19 can be severe and some cases have caused death.

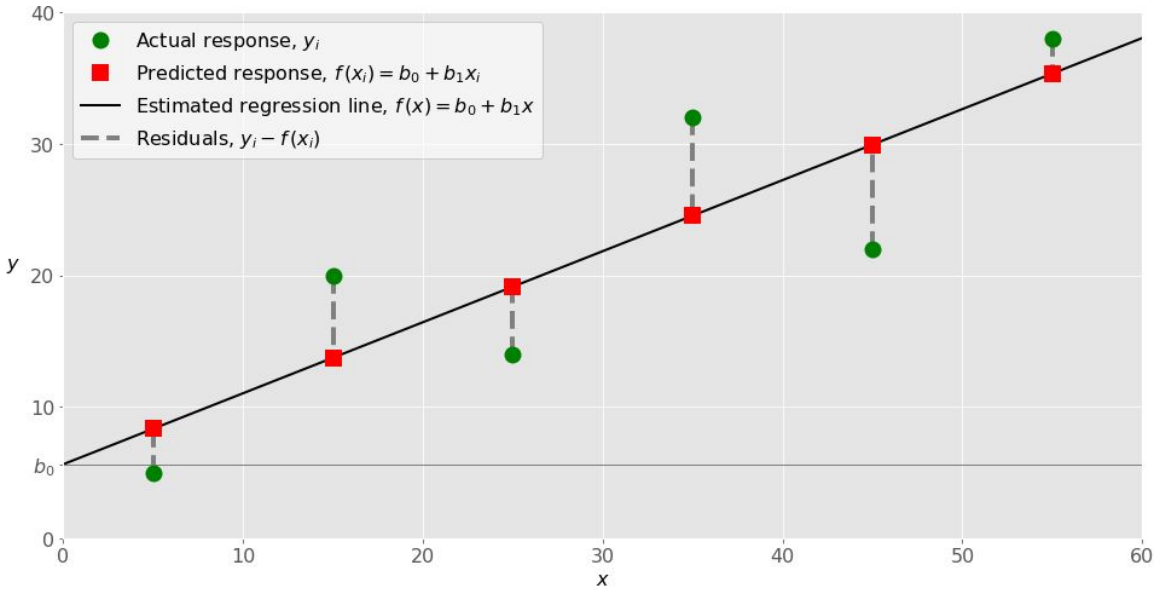
We will analyze the outbreak of coronavirus across various regions,visualize them using charts and graphs and predict the number of upcoming cases for the next 10 days using linear regression and polynomial regression models in python.

The data has information from 31th December 2019 till 10th March 2020.

Simple Linear Regression

Simple or single-variate linear regression is the simplest case of linear regression with a single independent variable, $\mathbf{x} = x$.

The following figure illustrates simple linear regression:



Example of simple linear regression

When implementing simple linear regression, you typically start with a given set of input-output (x-y) pairs (green circles). These pairs are your observations.

Polynomial Regression

You can regard polynomial regression as a generalized case of linear regression. You assume the polynomial dependence between the output and inputs and, consequently, the polynomial estimated regression function.

In other words, in addition to linear terms like b_1x_1, your regression function f can include non-linear terms such as $b_2x_1^2$, $b_3x_1^3$, or even $b_4x_1x_2$, $b_5x_1^2x_2$, and so on.

The simplest example of polynomial regression has a single independent variable, and the estimated regression function is a polynomial of degree 2: $f(x) = b_0 + b_1x + b_2x^2$.

Now, remember that you want to calculate b_0, b_1, and b_2, which minimize SSR. These are your unknowns!

Implementing Linear Regression in Python

Python Packages for Linear Regression: The package NumPy is a fundamental Python scientific package that allows many high-performance operations on single- and multi-dimensional arrays. It also offers many mathematical routines. Of course, it's open source.

The package scikit-learn is a widely used Python library for machine learning, built on top of NumPy and some other packages. It provides the means for preprocessing data, reducing dimensionality, implementing regression, classification, clustering, and more. Like NumPy, scikit-learn is also open source.

If you want to implement linear regression and need the functionality beyond the scope of scikit-learn, you should consider `statsmodels`. It's a powerful Python package for the estimation of statistical models, performing tests, and more. It's open source as well.

We have first implemented our project through a linear regression model but the accuracy hasn't come very well. The accuracy came out through this model is very less .

So, we have to switch over Polynomial Regression.

Implementing Polynomial Regression With scikit-learn

Implementing polynomial regression with scikit-learn is very similar to linear regression. There is only one extra step: you need to transform the array of inputs to include non-linear terms such as x^2.

Degree is an integer (2 by default) that represents the degree of the polynomial regression function.

In our project, we have taken the degree 3 of polynomial regression for better accuracy.

The result came out through this model is very good and has very good accuracy of 98.7

2.4 Snapshots of our Project

```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sb
        import datetime as dt
        from sklearn.preprocessing import StandardScaler
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from sklearn.preprocessing import PolynomialFeatures
        from sklearn.metrics import r2_score,mean_squared_error

In [2]: #reading data from excel file
        url=r'dataset2\covid-data.csv'
        covid=pd.read_csv(url)

In [3]: #checking for rows and columns in table
        covid.shape

Out[3]: (36347, 36)
```

Fig.1 Import all packages

```
In [4]: #viewing first five entries
        covid.head()
```

Out[4]:

	iso_code	continent	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million	...	aged_70_olde
0	AFG	Asia	Afghanistan	2019-12-31	0.0	0.0	0.0	0.0	0.0	0.0	...	1.33
1	AFG	Asia	Afghanistan	2020-01-01	0.0	0.0	0.0	0.0	0.0	0.0	...	1.33
2	AFG	Asia	Afghanistan	2020-01-02	0.0	0.0	0.0	0.0	0.0	0.0	...	1.33
3	AFG	Asia	Afghanistan	2020-01-03	0.0	0.0	0.0	0.0	0.0	0.0	...	1.33
4	AFG	Asia	Afghanistan	2020-01-04	0.0	0.0	0.0	0.0	0.0	0.0	...	1.33

5 rows × 36 columns

```
In [5]: covid.tail()
```

Fig.2 Print top 5 rows

```
In [5]: covid.tail()
```

Out[5]:

	iso_code	continent	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million	...	aged_70
36342	NaN	NaN	International	2020-02-28	705.0	0.0	4.0	0.0	NaN	NaN	...	
36343	NaN	NaN	International	2020-02-29	705.0	0.0	6.0	2.0	NaN	NaN	...	
36344	NaN	NaN	International	2020-03-01	705.0	0.0	6.0	0.0	NaN	NaN	...	
36345	NaN	NaN	International	2020-03-02	705.0	0.0	6.0	0.0	NaN	NaN	...	
36346	NaN	NaN	International	2020-03-10	696.0	-9.0	7.0	1.0	NaN	NaN	...	

5 rows × 36 columns

```
In [6]: #Looking for no. of empty rows
        covid.isnull().sum()
```

Fig.3 Print last 5 rows

```
In [9]: india_case.tail()
```

Out[9]:

	iso_code	continent	location	date	total_cases	new_cases	total_deaths	new_deaths	total_cases_per_million	new_cases_per_million	...	aged_70_ol
15443	IND	Asia	India	2020-08-08	2088611.0	61537.0	42518.0	933.0	1513.481	44.592	...	3.
15444	IND	Asia	India	2020-08-09	2153010.0	64399.0	43379.0	861.0	1560.147	46.666	...	3.
15445	IND	Asia	India	2020-08-10	2215074.0	62064.0	44386.0	1007.0	1605.121	44.974	...	3.
15446	IND	Asia	India	2020-08-11	2268675.0	53601.0	45257.0	871.0	1643.962	38.841	...	3.
15447	IND	Asia	India	2020-08-12	2329638.0	60963.0	46091.0	834.0	1688.138	44.176	...	3.

5 rows × 36 columns

Fig.4 Print top 5 rows from India's cases

```
In [10]: #
sb.set(rc={'figure.figsize':(15,10)})
sb.lineplot(x='date',y='total_cases',data=india_case)
plt.show()
```

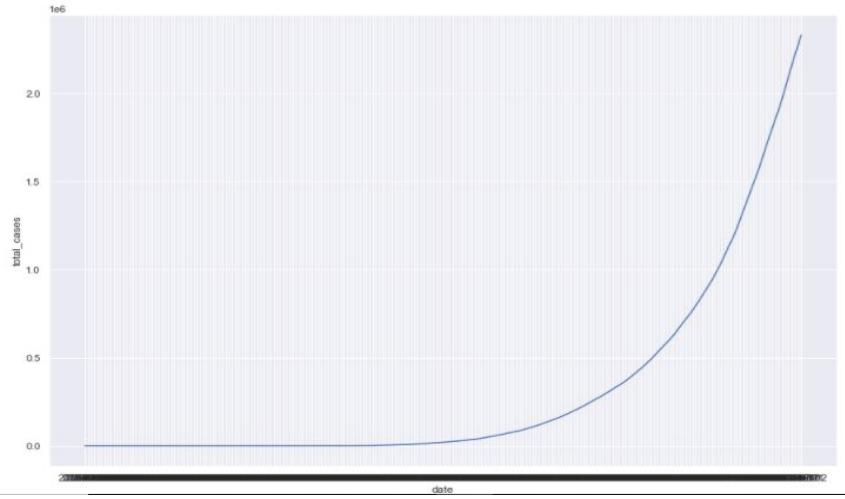


Fig.5 India cases graph

```
In [11]: #getting last 10 days record of india
lastmonth_india=india_case.tail(10)
sb.lineplot(x='date',y='total_cases',data=lastmonth_india)
plt.show()
```

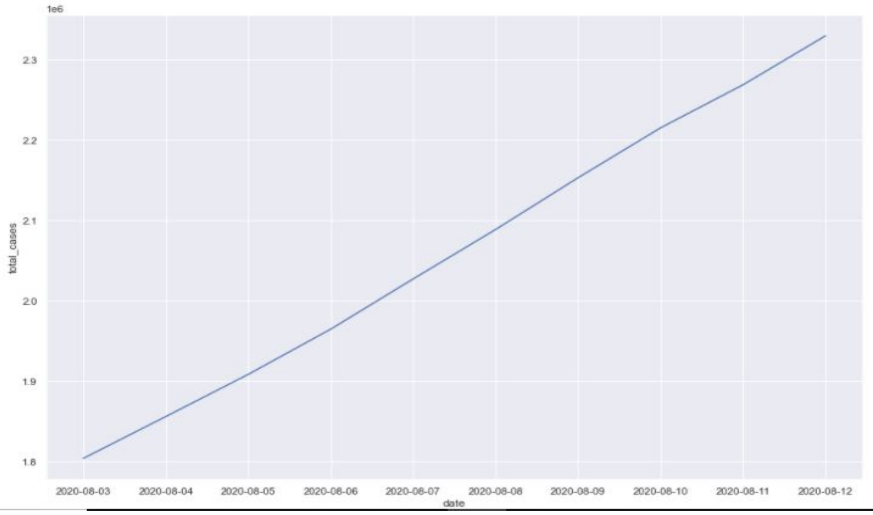


Fig.6 Last 10 days record of India Cases

```
In [14]: #India Vs China Vs USA , india=red, china=blue, green= USA
sb.lineplot(x='date',y='total_cases',data=india_case,color='red')
sb.lineplot(x='date',y='total_cases',data=china_case,color='blue')
sb.lineplot(x='date',y='total_cases',data=USA_case,color='green')
plt.show()
```

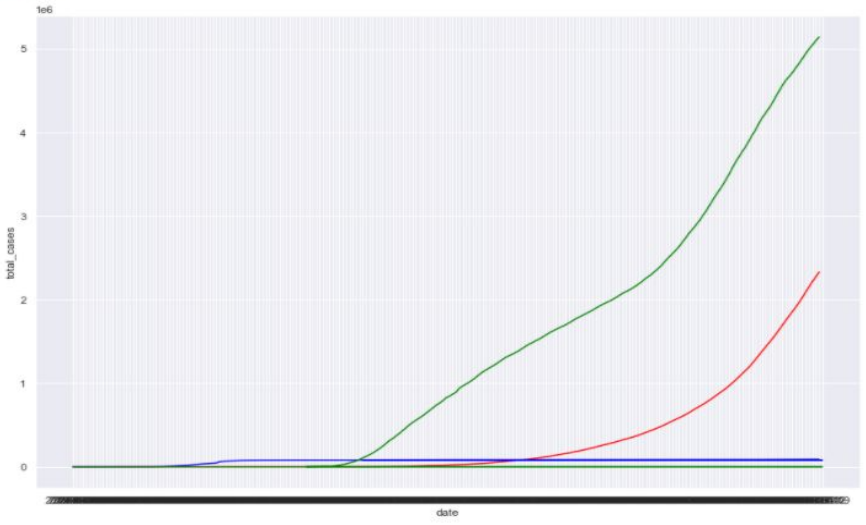


Fig.7 Comparison of cases between India,China and US

```
In [17]: #Top 10 countries having maximum total cases
max_case_countries=last_day_cases.sort_values(by='total_cases',ascending=False)
max_case_countries.loc[:,['location','total_cases']].head(10)
#sb.barplot(x='location',y='total_cases',data=max_case_countries[:10],hue='location')
#plt.show()
```

Out[17]:

	location	total_cases
36282	World	20330351.0
34561	United States	5141207.0
4913	Brazil	3109630.0
15447	India	2329638.0
27398	Russia	897599.0
30132	South Africa	566109.0
21694	Mexico	492522.0
26041	Peru	489680.0
7381	Colombia	410453.0
6994	Chile	376616.0

```
In [18]: #Visualization of top 10 countries having maximum no of cases
sb.barplot(x='location',y='total_cases',data=max_case_countries[1:10],hue='location')
plt.show()
```


Fig.8 Top 10 countries having maximum total cases

Fig.9 Visualization of top 10 countries having maximum no of cases

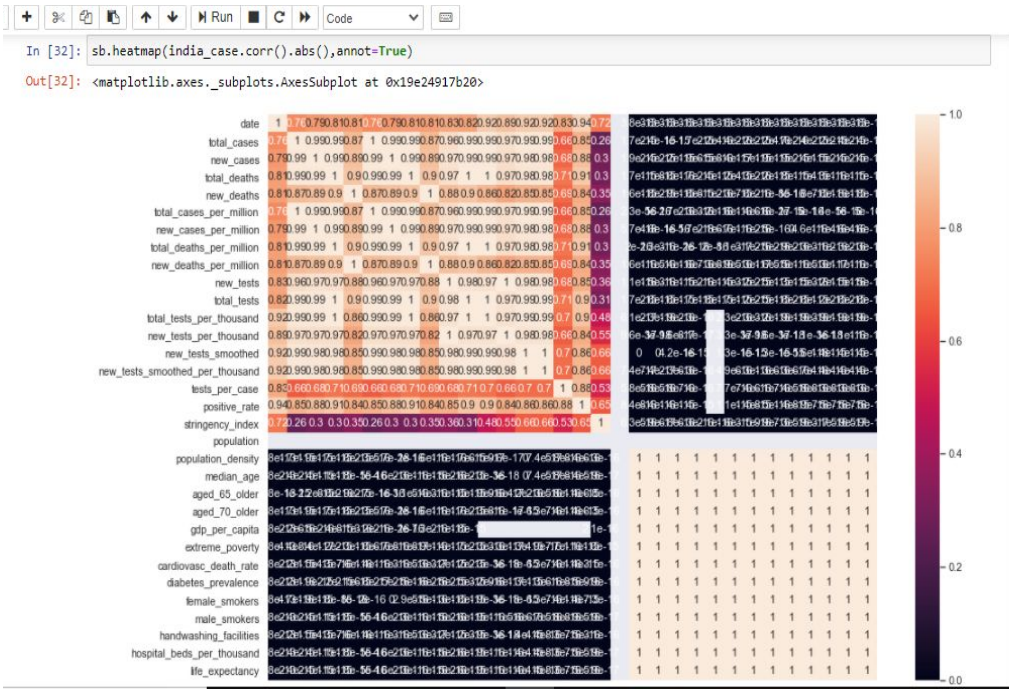


Fig.10 Heatmap representation(checking correlation)

```
sc=StandardScaler() x=sc.fit_transform(india_case.iloc[:,:16]) y=sc.fit_transform(india_case.iloc[:,16:])

In [37]: sc=StandardScaler()
         india_case.to_csv('India_case.csv')

In [38]: x=sc.fit_transform(india_case.iloc[:,0:1].values)
         y=sc.fit_transform(india_case.iloc[:,1:2].values)

In [39]: x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

In [41]: lr=LinearRegression()
         lr.fit(x_train,y_train)
Out[41]: LinearRegression()

In [42]: lr.coef_
Out[42]: array([[0.81119309]])

In [43]: lr.intercept_
Out[43]: array([0.05633201])

In [44]: y_pred=lr.predict(x_test)

In [45]: y1_pred=sc.inverse_transform(y_pred)
         y1_test=sc.inverse_transform(y_test)

In [46]: r2_score(y_test,y_pred)*100
Out[46]: 2.0486502378601723
```

Fig.11 Output with Linear regression

```
Run | Code

In [45]: y1_pred=sc.inverse_transform(y_pred)
         y1_test=sc.inverse_transform(y_test)

In [46]: r2_score(y_test,y_pred)*100
Out[46]: 2.0486502378601723

In [47]: poly=PolynomialFeatures(degree=3)

In [48]: x_poly_train=poly.fit_transform(x_train)
         x_poly_test=poly.fit_transform(x_test)
         ro_2=LinearRegression()
         ro_2.fit(x_poly_train,y_train)
Out[48]: LinearRegression()

In [49]: y_pred_poly=ro_2.predict(x_poly_test)

In [50]: r2_score(y_pred_poly,y_test)
Out[50]: 0.975909875999799

In [51]: ro_2.score(x_poly_train,y_train)
Out[51]: 0.9878431493211576

In [52]: mean_squared_error(y_test,y_pred_poly)
Out[52]: 0.010389034898536733

In [ ]:
```

Fig.12 Output with Polynomial Regression

3. RESULT

Accuracy achieved with using polynomial regression =98.7%

Error comes out = 0.010

4.CONCLUSION

COVID-19 is increasing rapidly throughout the world. So we have made this project that is predicting the number of corona cases in upcoming days so that one can take safety measures and precautions in advance.

In this project we have applied a linear regression model first but didn’t get good accuracy. So we applied polynomial regression model having 3rd degree to achieve good accuracy.