HEALTH CARE DATA ANALYSIS USING MACHINE LEARNING

Project report Submitted in full fulfillment of the requirement for the degree of
Bachelor of Technology

In

Computer Science and Engineering

By

Harshita Chauhan(161251)

Under the supervision of
Dr. Yugal Kumar

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,
Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled " HEALTH CARE DATA ANALYSIS USING MACHINE LEARNING " in fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering submitted in the department of Computer Science & Engineering , Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from December 2019 to May 2020 under the supervision of Dr.Yugal Kumar (Assistant Professor(Senior Grade) and CSE).

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Harshita Chauhan

Harshita Chauhan (161251)

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

YUGAL

15/07/2020

(Supervisor Signature)

Dr.Yugal Kumar

Assistant Professor

Senior Grade

CSE

Dated:

ACKNOWLEDGEMENT

My special gratitude to my project guide Dr.Yugal Kumar for his inspiration, adroit guidance,constant supervision and constructive in successful completion of the project.
I am very grateful to all the professors and lecturers of our department for their cooperation and keen interest throughout this project.

My sincere thanks to all my friends who helped me during this project.

LIST OF FIGURES AND OUTPUT

INDEX

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT:

We are together facing the pandemic of COVID-19 worldwide. This has led to horrible defamation of the economy even the developed countries have tremendously gone down and the human resources of every country have gone in trap of this virus. With daily exponentially increasing cases all over the world our project aims to analyze and predict the number of deaths and affected people from COVID-19 using data.

1.2 CONTRIBUTION OF PROJECT:

In the first half of our project we developed a Drug recommender system which can be a huge contribution for this pandemic. In the time where the whole world is going through this pandemic every country will be requiring drugs to treat people, this might lead to shortage of the medicine the recommender system will help us know the nearest and most appropriate medicine required to treat a person. This system has been trained for a huge dataset and hence can be really reliable and helpful to doctors to contribute for other alternative options.

1.3 AIM OF PROJECT:

1)GLOBAL IMPACT OF CORONAVIRUS

2)CORONAVIRUS OUTBREAK ANALYSIS

We will see its impact so far in terms of total case emerging. Total deaths reported and the total number of recoveries across the globe. We will analyze the outbreak of corona using different tools visualise them using charts and graphs and predict the number of upcoming cases for next ten days i.e between 16-March to 26-March using linear regression model and support vector machine using python.The data used is from 22January to 15March 2020.

This majorly focuses on how you can use machine learning algorithms to summarise such sensitive issues. We will certainly cover the current scenario and impact that has been created world wide. Predictions may vary as the cases are now varying and this is data from march.

1.4 Analysis on worldwide dataset

COVID-19 reported its 562 total confirmed cases with 9 deaths on 25th March Afternoon in India

1)Trend in INDIA VS ITALY,CHINA & SOUTH KOREA

2)Trend across the world

3)Forecast & prediction of the virus

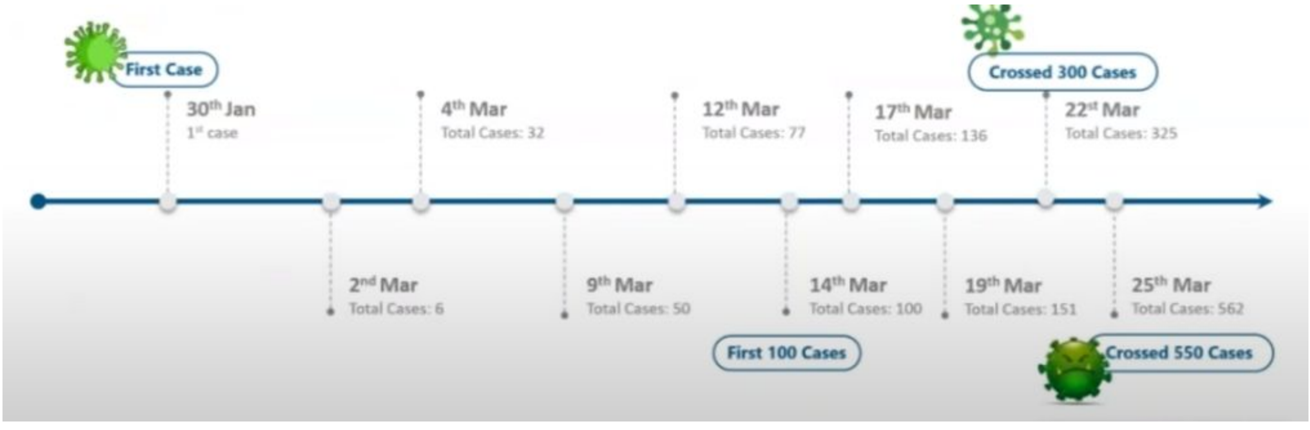


FIG i) TIMELINE OF COVID-19 CASES IN INDIA

CHAPTER 2

LITERATURE SURVEY

2.1 What is Coronavirus?

Coronavirus(CoV) are largely family of virus that cause illness ranging from common cold to more severe diseases such as Middle East Respiratory Syndrome(MERS-Cov) and Severe Acute Respiratory Syndrome (SARS-CoV)

Coronaviruses are zoonotic, i.e it can be transmitted between animals and human beings.

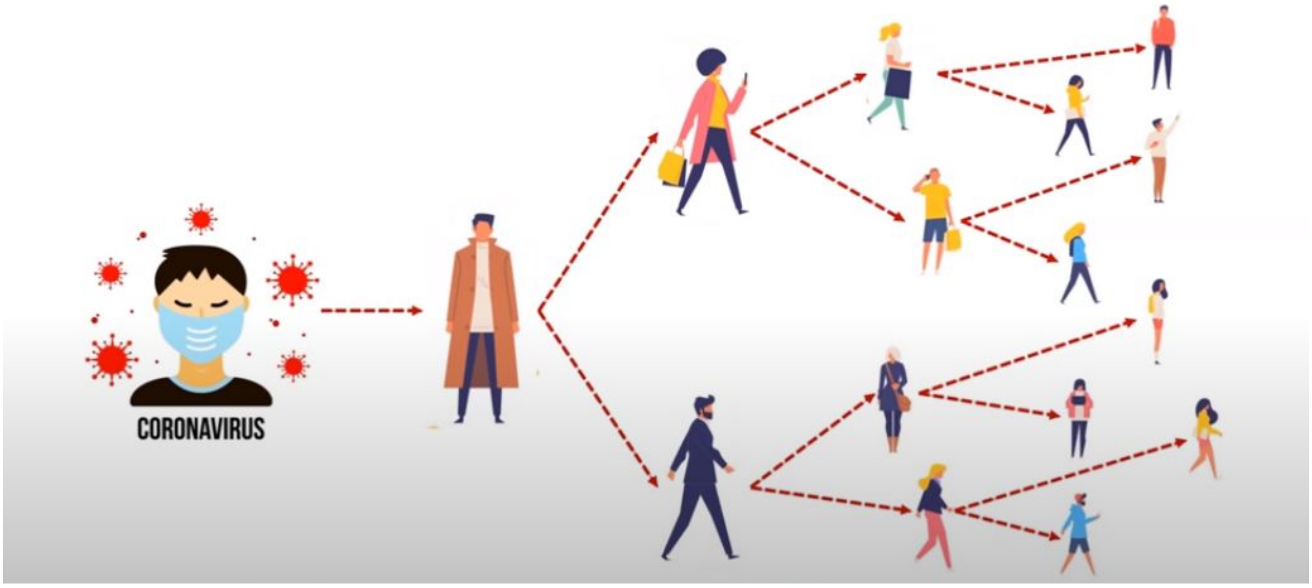


FIG ii) Spread of COVID-19 VIRUS

2.2 How COVID -19 emerged?

COVID-19 is the disease caused by the new coronavirus that emerged in China in December 2019. The source of coronavirus is believed to be a “wet market” in Wuhan which sold both dead and live animals including fish and birds.

2.3 Symptoms of coronavirus

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

2.4 Global impact of coronavirus

The novel coronavirus is now a public health emergency of international concern, killing more than 319000 people and infecting more than 4.18M people worldwide.
A lockdown has been put over world wide to stop community transmission which has led to major economic crises and deflation of GDP growth of even the developed nations.

This has led to thousand of unemployed labour in every country.WHO has declared COVID-19 as pandemic. Researchers and all countries are striving and making collective efforts to find solution or vaccine to get over this but yet no such solution has come up. WHO has also quotes we might have to live with this and this might end up being a endemic.

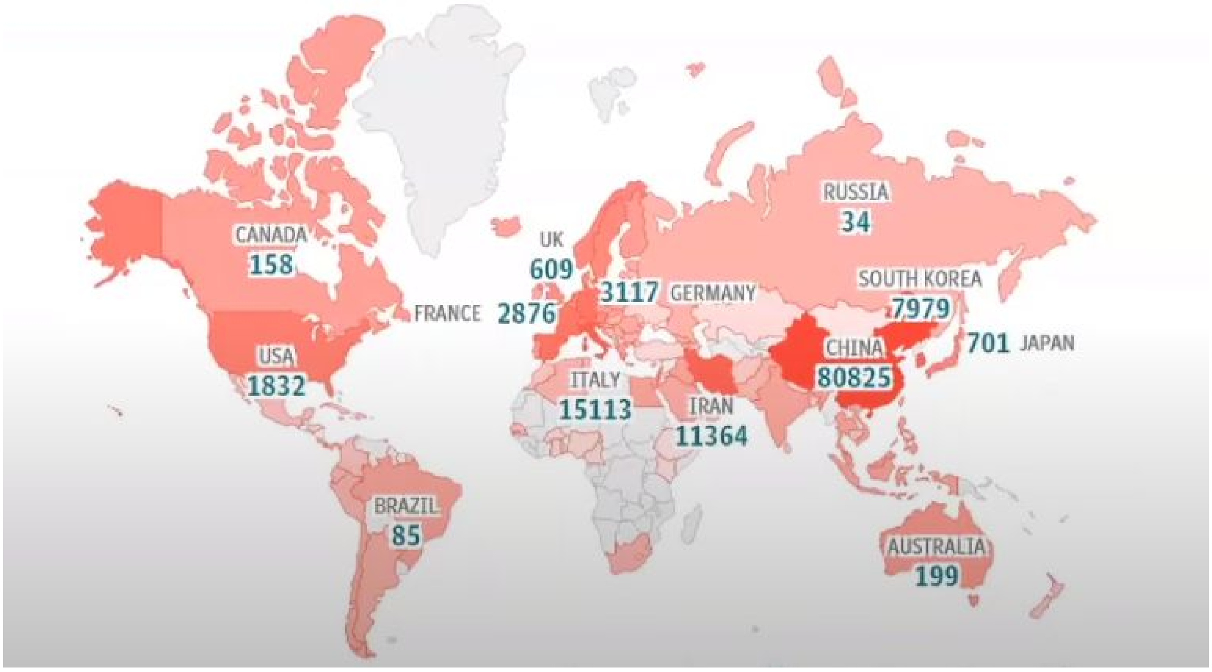


Fig iii) Total Coronavirus cases till 13th March

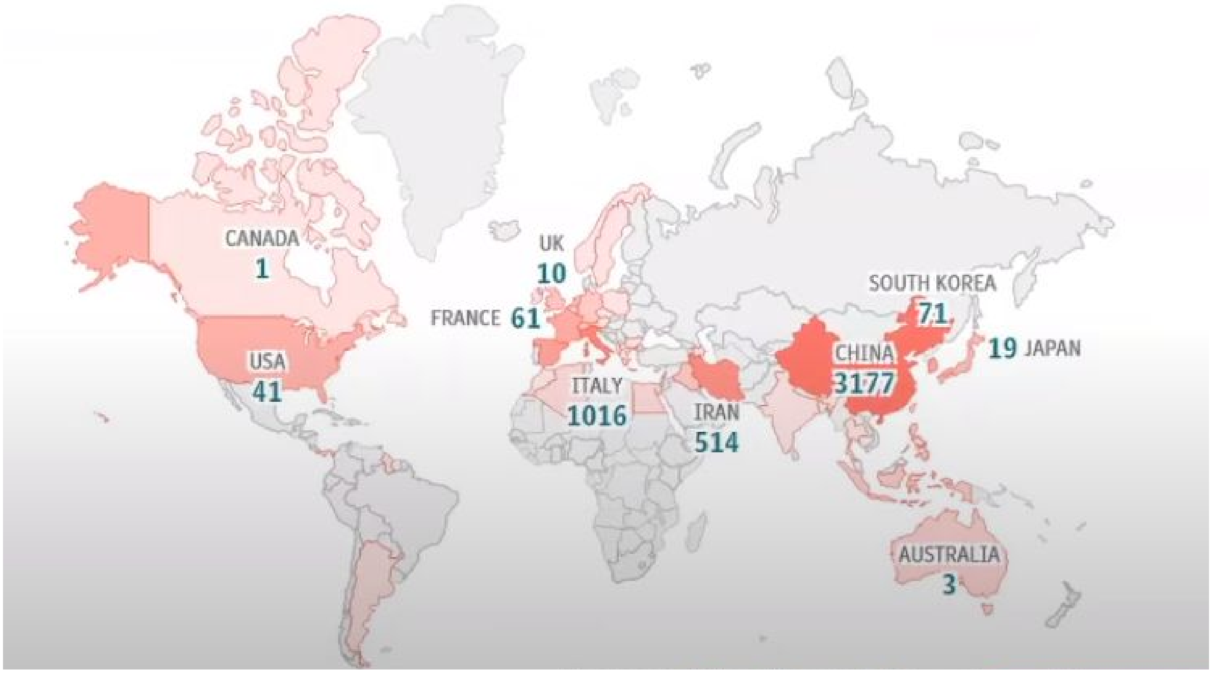


Fig iv)Total Coronavirus cases deaths till 13th march

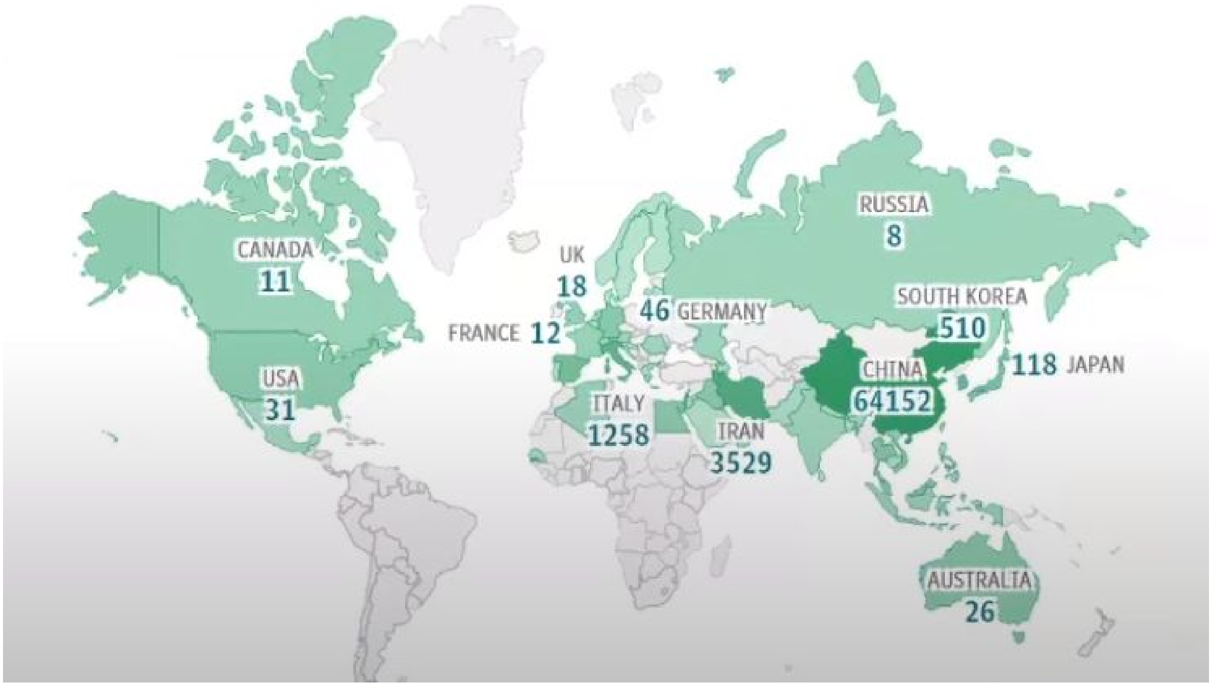


Fig v)Total Coronavirus cases recoveries till 13th march

In the global effort to slow the spread of COVID-19, many countries have adopted social distancing and quarantine measures to mitigate the impact of the pandemic. Now, we are applying machine learning tools to analyze the data and predict the oute quantify the effects of these measures in specific parts of the world.

2.5 LIBRARIES USED IN PROJECT

- **pandas**

The pandas library is worked for cleaning, controlling, changing and picturing information in Python.In expansion to offering a great deal of comfort, pandas is additionally frequently quicker than unadulterated Python for working with information. Like R, pandas exploits vectorization, which accelerates code execution.

- **NumPy**

NumPy is a major Python library that gives usefulness to logical registering. NumPy gives a portion of the center rationale that pandas is based upon. Normally, most information researchers will work with pandas, yet knowing NumPy is significant as it permits you to get to a portion of the center usefulness when you have to.

- **Matplotlib**

The Matplotlib library is a powerful plotting library for Python. Data scientists often use the Pyplot module from the library, which provides a standard interface for plotting data.The plotting usefulness that is remembered for pandas calls Matplotlib in the engine, so understanding matplotlib assists with altering plots you make in pandas

- **Scikit-learn**

It incorporates submodules for such models as:

• Classification: SVM, closest neighbors, arbitrary backwoods, strategic relapse

• Regression: Lasso, edge relapse

• Clustering: k-implies, ghastly bunching

• Dimensionality decrease: PCA, include determination, network factorization,

• Model determination: Grid search, cross-approval, measurements

• Preprocessing: Feature extraction, standardization Along with pandas, statsmodels, and IPython, scikit-learn

● Folium

folium makes it simple to picture information that has been controlled in Python on an intelligent flyer map. It empowers both the official of information to a guide for choropleth representations just as passing rich vector/raster/HTML perceptions as markers on the guide.

● Seaborn

Seaborn is a Python information perception library dependent on matplotlib. It gives a significant level interface to drawing alluring and instructive factual designs.

● Plotly

The plotly Python library (plotly.py) is an intelligent, open-source plotting library that underpins more than 40 one of a kind diagram types covering a wide scope of measurable, budgetary, geographic, logical, and 3-dimensional use-cases.

CHAPTER 3

ALGORITHMS USED IN PROJECT

3.1. Support Vector Machines

Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

- Powerful in high dimensional spaces.
- Still powerful in situations where number of measurements is more noteworthy than the quantity of tests.
- Utilizations a subset of preparing focuses in the choice capacity (called bolster vectors), so it is likewise memory productive.
- Flexible: distinctive Kernel capacities can be determined for the choice capacity. Basic parts are given, yet it is additionally conceivable to indicate custom pieces.

The disadvantages of support vector machines include:

- In the event that the quantity of highlights is a lot more noteworthy than the quantity of tests, maintain a strategic distance from over-fitting in picking Kernel capacities and regularization term is vital.
- SVMs don't legitimately give likelihood appraises, these are determined utilizing a costly five-overlap cross-approval (see Scores and probabilities, underneath).

3.2 Linear Regression

Straight relapse endeavors to show the connection between two factors by fitting a direct condition to watched information. One variable is viewed as an informative variable, and the other is viewed as a needy variable. For instance, a modeler should relate the loads of people to their statures utilizing a straight relapse model.

advantage:

1. The displaying speed is quick, doesn't require extremely convoluted counts, and runs quick when the measure of information is huge.
2. The comprehension and translation of every factor can be offered by the coefficient

Disadvantages:

Non-straight information can't be well fitted. So you have to initially decide if the factors are straight.

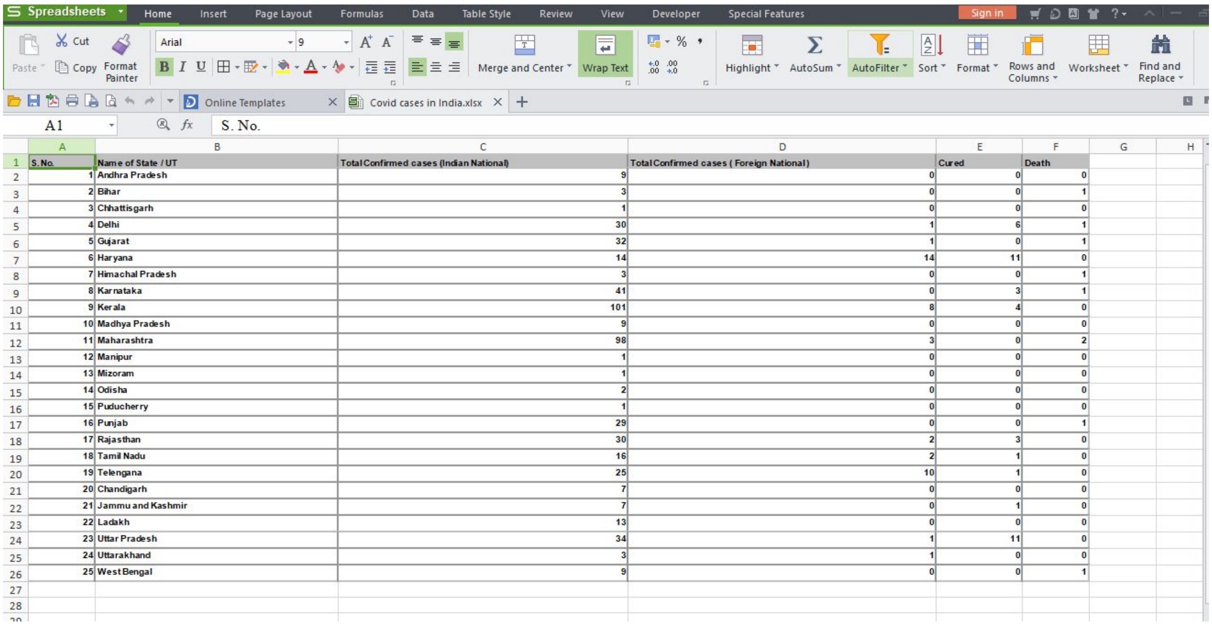
CHAPTER 4

SYSYTEM DEVELOPNMENT

4.1 COLLECTING DATASETS AND MAKING IT READY FOR ANALYSES

Datasets and constraints use in analyze:

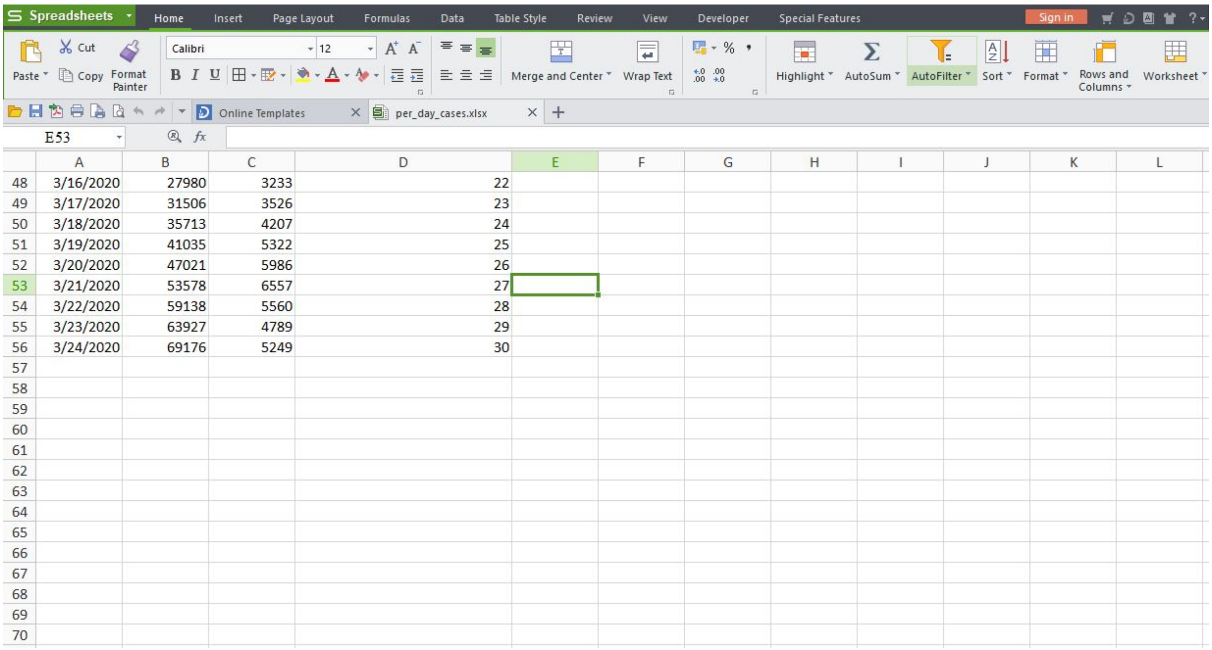
1) Covid Cases in India:



S.No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
1	Andhra Pradesh	9	0	0	0
2	Bihar	3	0	0	1
3	Chhattisgarh	1	0	0	0
4	Delhi	30	1	6	1
5	Gujarat	32	1	0	1
6	Haryana	14	14	11	0
7	Himachal Pradesh	3	0	0	1
8	Karnataka	41	0	3	1
9	Kerala	101	8	4	0
10	Madhya Pradesh	9	0	0	0
11	Maharashtra	98	3	0	2
12	Manipur	1	0	0	0
13	Mizoram	1	0	0	0
14	Odisha	2	0	0	0
15	Puducherry	1	0	0	0
16	Punjab	29	0	0	1
17	Rajasthan	30	2	3	0
18	Tamil Nadu	16	2	1	0
19	Telengana	25	10	1	0
20	Chandigarh	7	0	0	0
21	Jammu and Kashmir	7	0	1	0
22	Ladakh	13	0	0	0
23	Uttar Pradesh	34	1	11	0
24	Uttarakhand	3	1	0	0
25	West Bengal	9	0	0	1

Fig vi) Database of Covid_Cases_in_India.csv

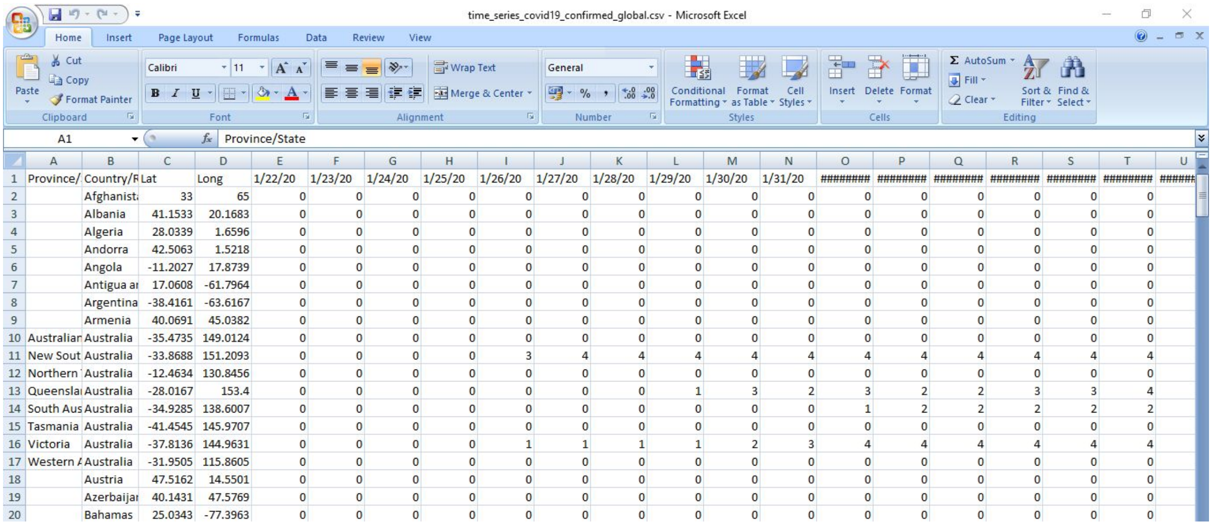
2)Per day cases



	A	B	C	D
48	3/16/2020	27980	3233	22
49	3/17/2020	31506	3526	23
50	3/18/2020	35713	4207	24
51	3/19/2020	41035	5322	25
52	3/20/2020	47021	5986	26
53	3/21/2020	53578	6557	27
54	3/22/2020	59138	5560	28
55	3/23/2020	63927	4789	29
56	3/24/2020	69176	5249	30

Fig vii) Database of per_day_cases.csv

3)time series covid 19 confirmed global



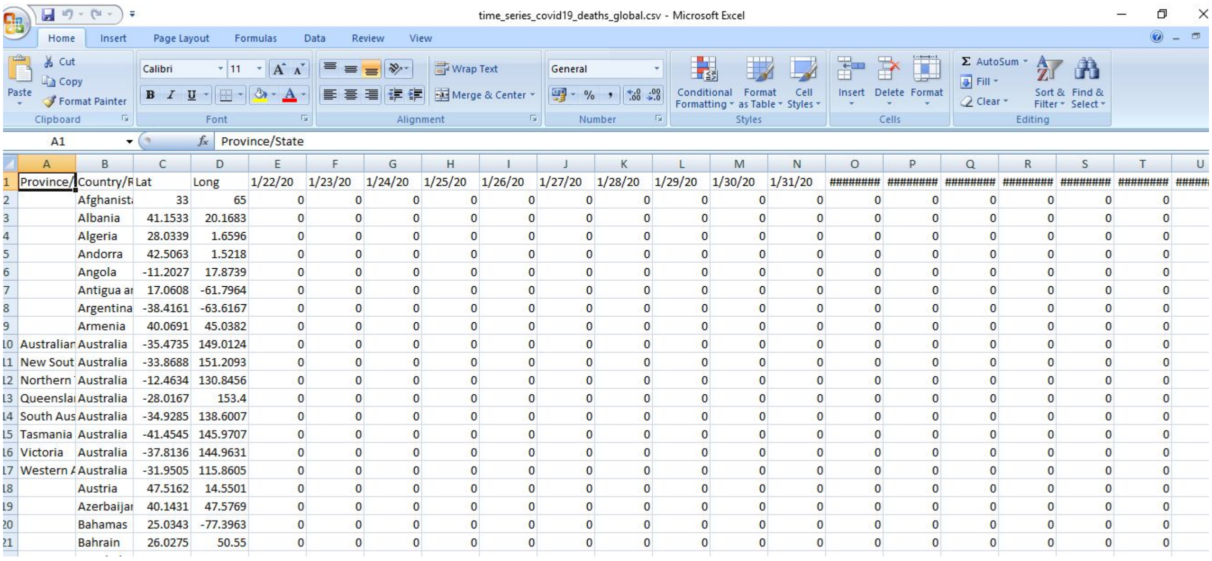
time_series_covid19_confirmed_global.csv - Microsoft Excel

A1 Province/State

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Province/	Country/R	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	########	########	########	########	########	########	######
2		Afghanist	33	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3		Albania	41.1533	20.1683	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4		Algeria	28.0339	1.6596	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5		Andorra	42.5063	1.5218	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6		Angola	-11.2027	17.8739	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7		Antigua ar	17.0608	-61.7964	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8		Argentina	-38.4161	-63.6167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9		Armenia	40.0691	45.0382	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	Australian	Australia	-35.4735	149.0124	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	New Sout	Australia	-33.8688	151.2093	0	0	0	0	3	4	4	4	4	4	4	4	4	4	4	4	
12	Northern	Australia	-12.4634	130.8456	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	Queensla	Australia	-28.0167	153.4	0	0	0	0	0	0	0	1	3	2	3	2	2	3	3	4	
14	South Aus	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	0	0	0	1	2	2	2	2	2	
15	Tasmania	Australia	-41.4545	145.9707	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	Victoria	Australia	-37.8136	144.9631	0	0	0	0	1	1	1	1	2	3	4	4	4	4	4	4	
17	Western A	Australia	-31.9505	115.8605	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18		Austria	47.5162	14.5501	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19		Azerbaijar	40.1431	47.5769	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20		Bahamas	25.0343	-77.3963	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig viii) Database of time_series_covid_19_confirmed_globals.csv

4)time series covid 19 deaths global:

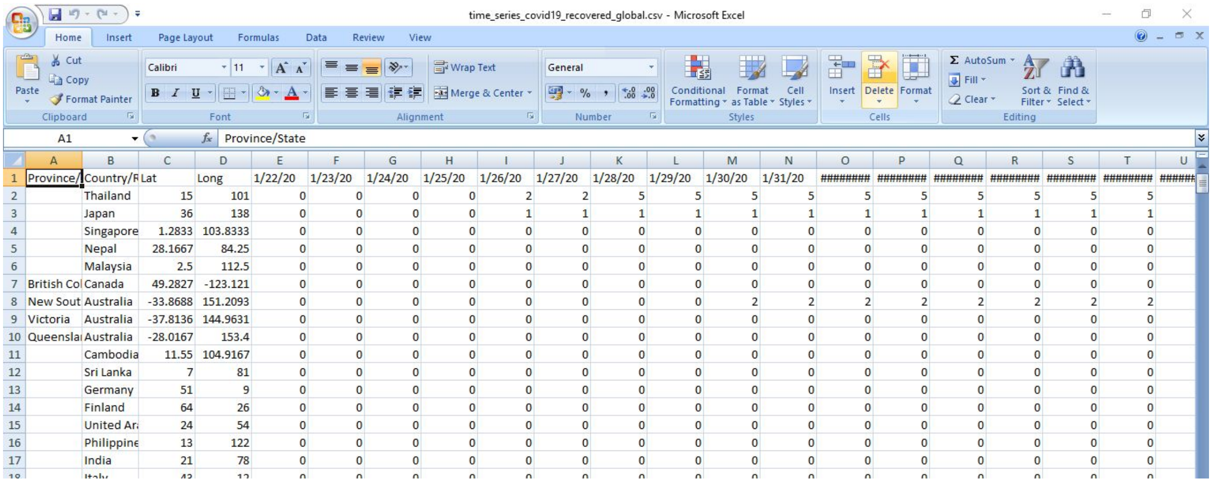

time_series_covid19_deaths_global.csv - Microsoft Excel

A1 Province/State

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Province/	Country/R	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	########	########	########	########	########	########	######
2		Afghanist	33	65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3		Albania	41.1533	20.1683	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4		Algeria	28.0339	1.6596	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5		Andorra	42.5063	1.5218	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6		Angola	-11.2027	17.8739	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7		Antigua ar	17.0608	-61.7964	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8		Argentina	-38.4161	-63.6167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
9		Armenia	40.0691	45.0382	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	Australian	Australia	-35.4735	149.0124	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	New Sout	Australia	-33.8688	151.2093	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	Northern	Australia	-12.4634	130.8456	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	Queenslar	Australia	-28.0167	153.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	South Aus	Australia	-34.9285	138.6007	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	Tasmania	Australia	-41.4545	145.9707	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	Victoria	Australia	-37.8136	144.9631	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	Western A	Australia	-31.9505	115.8605	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
18		Austria	47.5162	14.5501	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19		Azerbaijar	40.1431	47.5769	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
20		Bahamas	25.0343	-77.3963	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21		Bahrain	26.0275	50.55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig ix)Database of time_series_covid_19_deaths_global.csv

5)time series covid 19 recovered global:


time_series_covid19_recovered_global.csv - Microsoft Excel

A1 Province/State

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	Province/	Country/R	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	1/28/20	1/29/20	1/30/20	1/31/20	########	########	########	########	########	########	######
2		Thailand	15	101	0	0	0	0	2	2	5	5	5	5	5	5	5	5	5	5	
3		Japan	36	138	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	
4		Singapore	1.2833	103.8333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5		Nepal	28.1667	84.25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6		Malaysia	2.5	112.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	British Col	Canada	49.2827	-123.121	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	New Sout	Australia	-33.8688	151.2093	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	
9	Victoria	Australia	-37.8136	144.9631	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	Queenslar	Australia	-28.0167	153.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11		Cambodia	11.55	104.9167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12		Sri Lanka	7	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13		Germany	51	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14		Finland	64	26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15		United Ar	24	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16		Philippine	13	122	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17		India	21	78	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Fig x)Database of time_series_covid_19_recovered_global.csv

4.2IMPLEMENTING SVM AND LINEAR REGRESSION TO IMPLEMENT USING COVID-19

Step 1: Importing necessary libraries like:

1)numpy

2)pandas

3)matplotlib

4)math

5)sklearn.svm

6)sklearn.model_selection

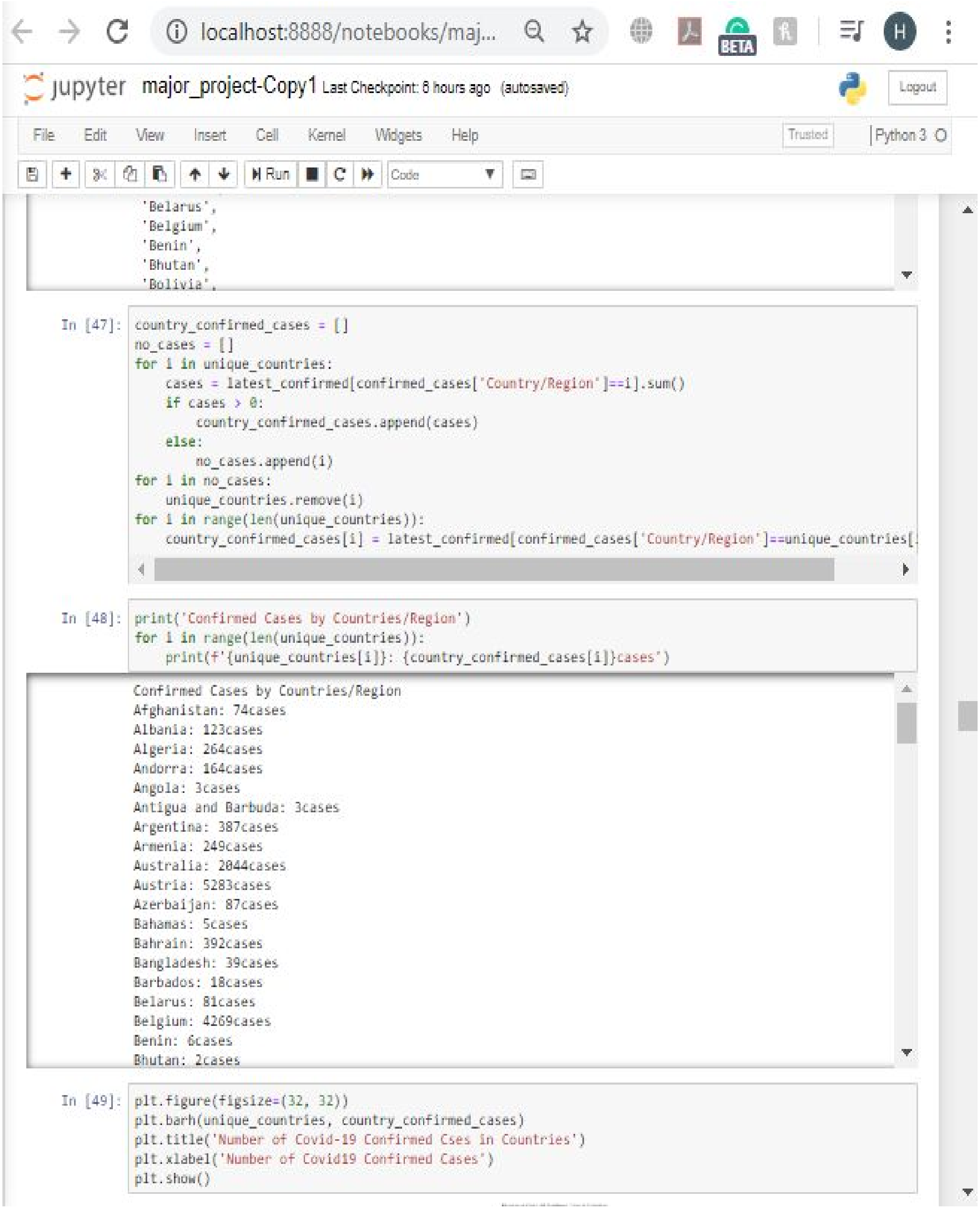
7)time

Step 2: Gathering confirmed_cases, recovered_cases, total_deaths, mortality_rate and world_cases using data

Step 3: Converting all the data in numpy arrays and reshaping them

Step 4: View all the above by printing these variables to have look at the stored data.

Step 5: Visualizing data using plt.


localhost:8888/notebooks/maj...
BETA
H
jupyter major_project-Copy1 Last Checkpoint: 8 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
 'Belarus',
 'Belgium',
 'Benin',
 'Bhutan',
 'Bolivia',
```

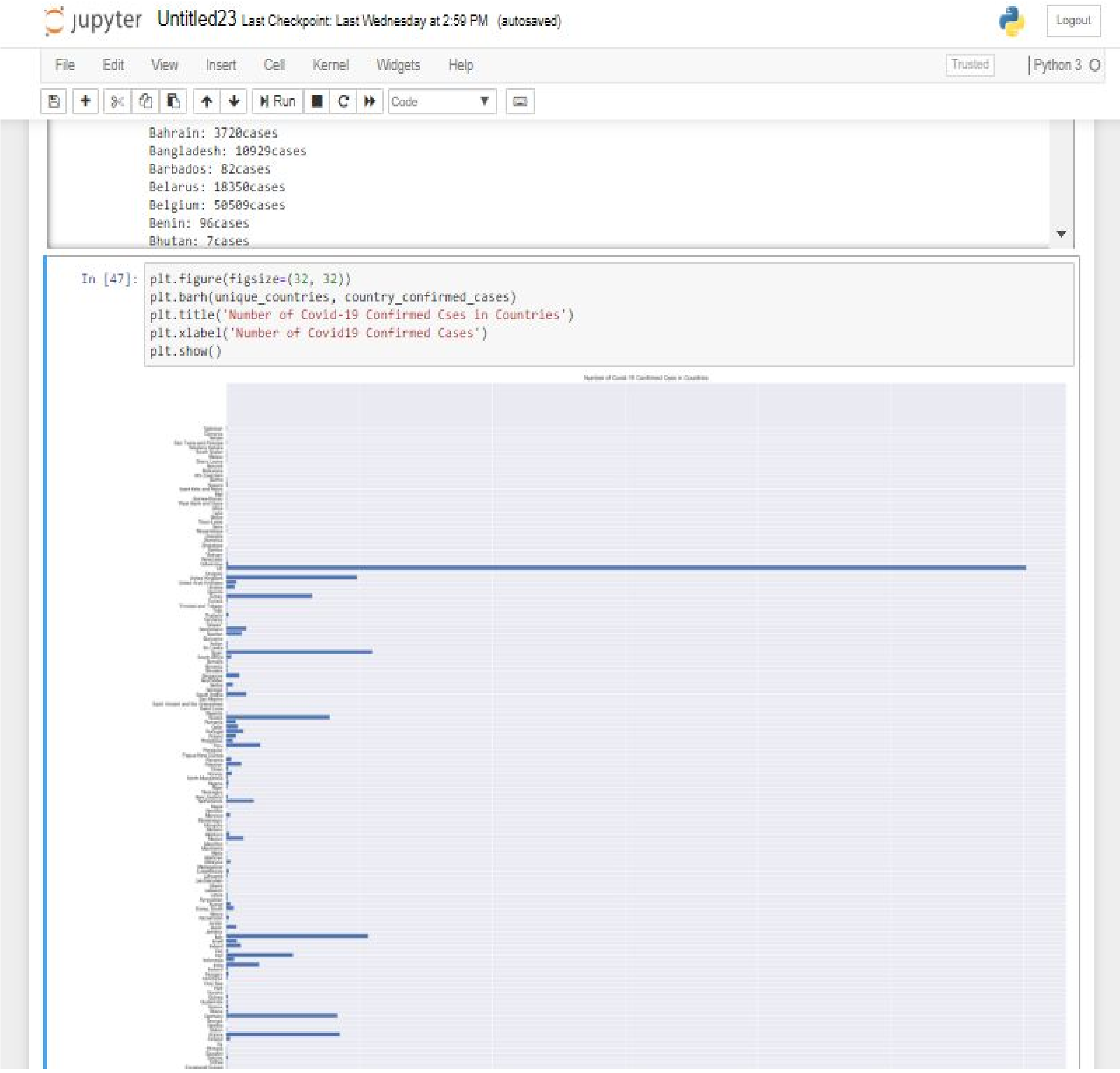
```
In [47]: country_confirmed_cases = []
         no_cases = []
         for i in unique_countries:
             cases = latest_confirmed[confirmed_cases['Country/Region']==i].sum()
             if cases > 0:
                 country_confirmed_cases.append(cases)
             else:
                 no_cases.append(i)
         for i in no_cases:
             unique_countries.remove(i)
         for i in range(len(unique_countries)):
             country_confirmed_cases[i] = latest_confirmed[confirmed_cases['Country/Region']==unique_countries[
```

```
In [48]: print('Confirmed Cases by Countries/Region')
         for i in range(len(unique_countries)):
             print(f'{unique_countries[i]}: {country_confirmed_cases[i]}cases')
```

```
Confirmed Cases by Countries/Region
Afghanistan: 74cases
Albania: 123cases
Algeria: 264cases
Andorra: 164cases
Angola: 3cases
Antigua and Barbuda: 3cases
Argentina: 387cases
Armenia: 249cases
Australia: 2044cases
Austria: 5283cases
Azerbaijan: 87cases
Bahamas: 5cases
Bahrain: 392cases
Bangladesh: 39cases
Barbados: 18cases
Belarus: 81cases
Belgium: 4269cases
Benin: 6cases
Bhutan: 2cases
```

```
In [49]: plt.figure(figsize=(32, 32))
         plt.barh(unique_countries, country_confirmed_cases)
         plt.title('Number of Covid-19 Confirmed Cses in Countries')
         plt.xlabel('Number of Covid19 Confirmed Cases')
         plt.show()
```

Fig xi) Extracting number of cases in the world according to country


jupyter
Untitled23 Last Checkpoint: Last Wednesday at 2:59 PM (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code

```
Bahrain: 3720cases
Bangladesh: 10929cases
Barbados: 82cases
Belarus: 18350cases
Belgium: 50509cases
Benin: 96cases
Bhutan: 7cases
```

```
In [47]: plt.figure(figsize=(32, 32))
         plt.barh(unique_countries, country_confirmed_cases)
         plt.title('Number of Covid-19 Confirmed Cses in Countries')
         plt.xlabel('Number of Covid19 Confirmed Cases')
         plt.show()
```

Number of Covid-19 Confirmed Cses in Countries

Fig xii) Unique countries plot with number of infected people

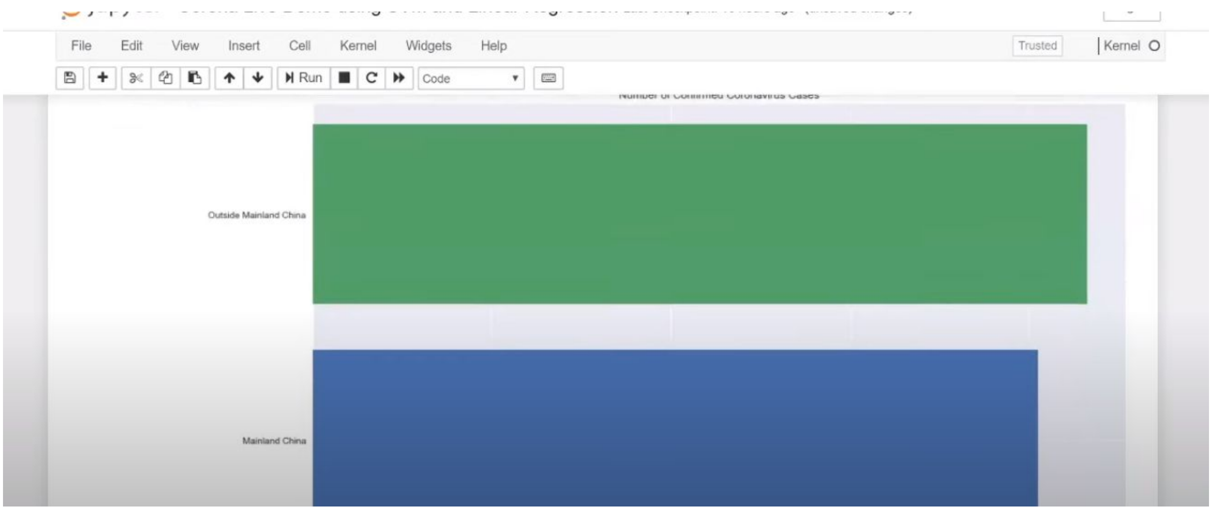
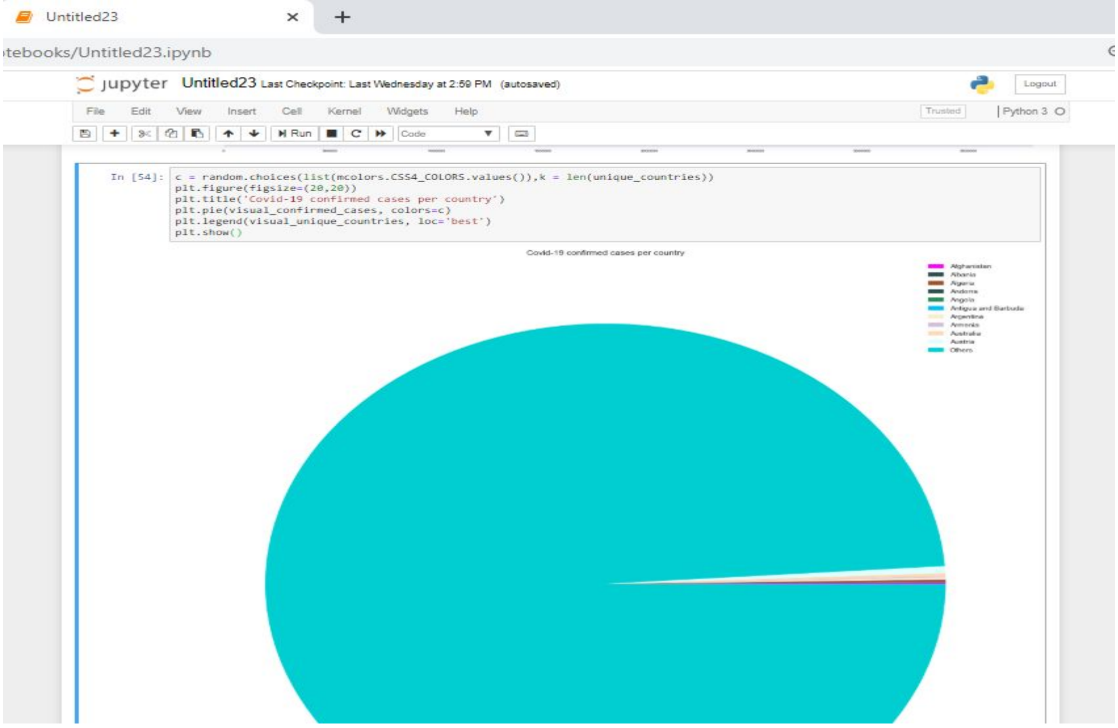


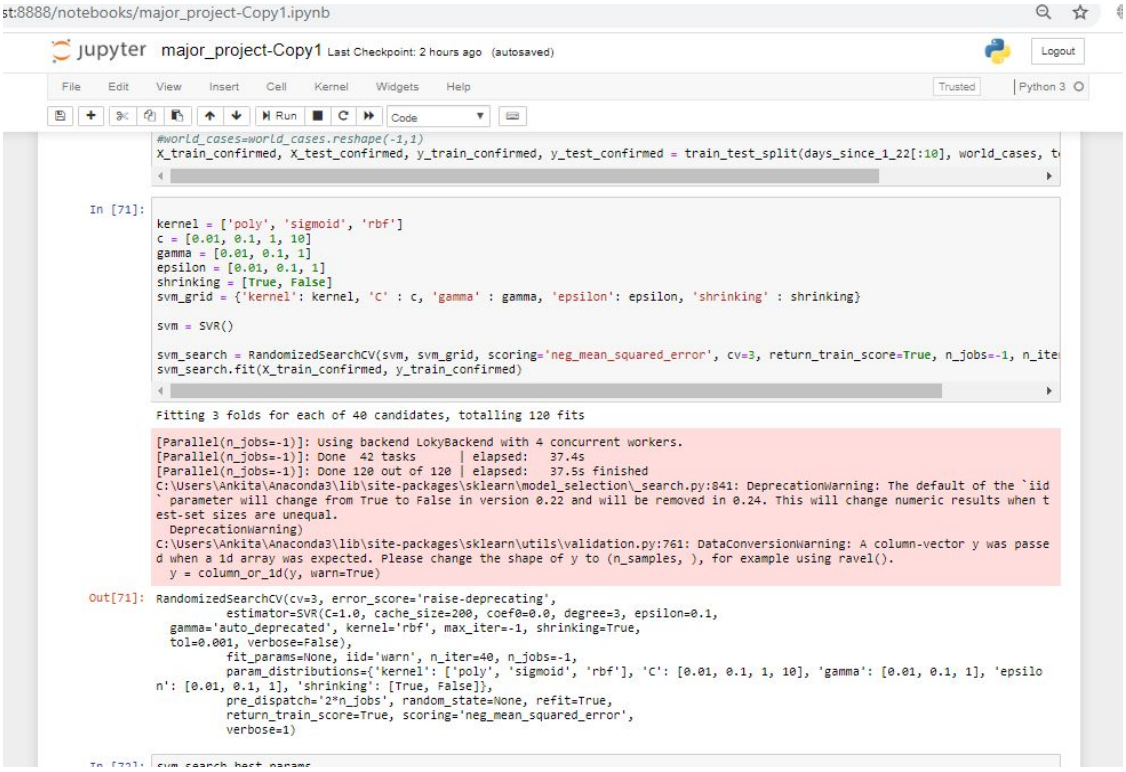
Fig xiii) .Infected Population count of Mainland China v/s Outside Mainland China



```
In [54]: c = random.choices(list(mcolors.CSS4_COLORS.values()),k = len(unique_countries))
         plt.figure(figsize=(20,20))
         plt.title('Covid-19 confirmed cases per country')
         plt.pie(visual_confirmed_cases, colors=c)
         plt.legend(visual_unique_countries, loc='best')
         plt.show()
```

Fig xiv) This data represents the infected people in china and other countries (in start china was the most infected place to be , this chart does not include italy

Step 6: Building of SVM MODEL using kernal,c,gamma,epsilon,shrinking and svm_grid . Building X and Y dataset using two constraints days_since_1_22 and world_cases



st:8888/notebooks/major_project-Copy1.ipynb
jupyter major_project-Copy1 Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
#world_cases=world_cases.reshape(-1,1)
X_train_confirmed, X_test_confirmed, y_train_confirmed, y_test_confirmed = train_test_split(days_since_1_22[:10], world_cases, t
```

In [71]:

```
kernel = ['poly', 'sigmoid', 'rbf']
c = [0.01, 0.1, 1, 10]
gamma = [0.01, 0.1, 1]
epsilon = [0.01, 0.1, 1]
shrinking = [True, False]
svm_grid = {'kernel': kernel, 'C' : c, 'gamma' : gamma, 'epsilon': epsilon, 'shrinking' : shrinking}

svm = SVR()

svm_search = RandomizedSearchCV(svm, svm_grid, scoring='neg_mean_squared_error', cv=3, return_train_score=True, n_jobs=-1, n_ite
svm_search.fit(X_train_confirmed, y_train_confirmed)
```

```
Fitting 3 folds for each of 40 candidates, totalling 120 fits
```

```
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 4 concurrent workers.
[Parallel(n_jobs=-1)]: Done  42 tasks      | elapsed:   37.4s
[Parallel(n_jobs=-1)]: Done 120 out of 120 | elapsed:   37.5s finished
C:\Users\Ankita\Anaconda3\lib\site-packages\sklearn\model_selection\_search.py:841: DeprecationWarning: The default of the `iid
` parameter will change from True to False in version 0.22 and will be removed in 0.24. This will change numeric results when t
est-set sizes are unequal.
  DeprecationWarning)
C:\Users\Ankita\Anaconda3\lib\site-packages\sklearn\utils\validation.py:761: DataConversionWarning: A column-vector y was passe
d when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

Out[71]:

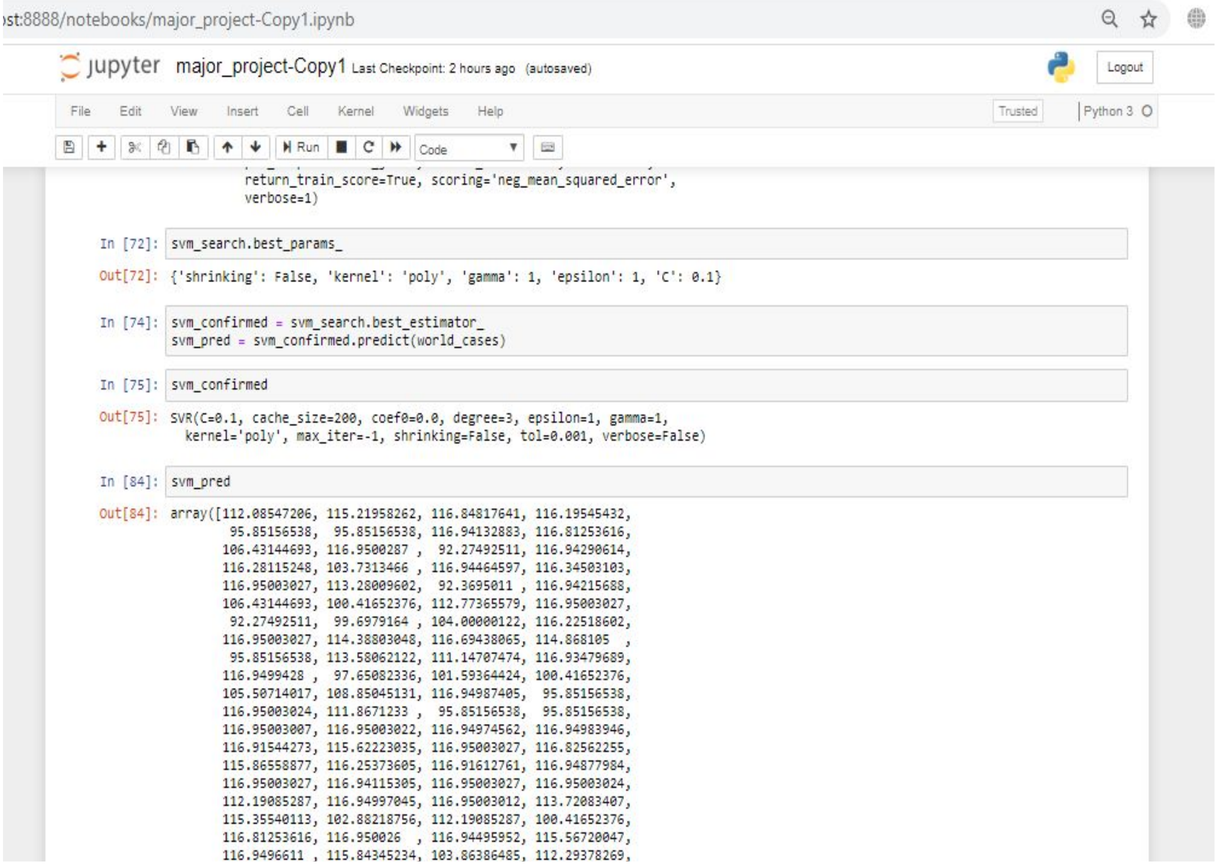
```
RandomizedSearchCV(cv=3, error_score='raise-deprecating',
          estimator=SVR(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1,
  gamma='auto_deprecated', kernel='rbf', max_iter=-1, shrinking=True,
  tol=0.001, verbose=False),
          fit_params=None, iid='warn', n_iter=40, n_jobs=-1,
          param_distributions={'kernel': ['poly', 'sigmoid', 'rbf'], 'C': [0.01, 0.1, 1, 10], 'gamma': [0.01, 0.1, 1], 'epsilo
n': [0.01, 0.1, 1], 'shrinking': [True, False]},
          pre_dispatch='2*n_jobs', random_state=None, refit=True,
          return_train_score=True, scoring='neg_mean_squared_error',
          verbose=1)
```

In [72]:

```
svm_search.best_params_
```

Fig xv) Applying SVM model

Step 7: Calculating the best parameters and its values. Best estimators for SVM functiona and its predicted values.

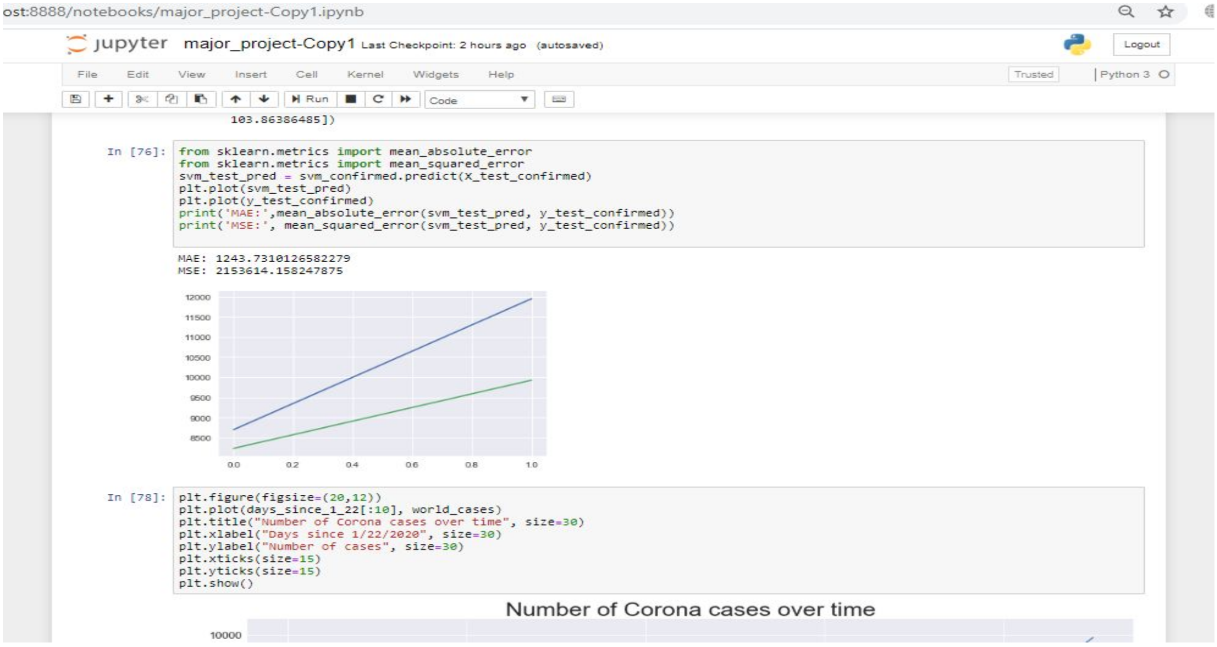


ost:8888/notebooks/major_project-Copy1.ipynb
jupyter major_project-Copy1 Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

```
          return_train_score=True, scoring='neg_mean_squared_error',
          verbose=1)

In [72]:  svm_search.best_params_

Out[72]:  {'shrinking': False, 'kernel': 'poly', 'gamma': 1, 'epsilon': 1, 'C': 0.1}

In [74]:  svm_confirmed = svm_search.best_estimator_
          svm_pred = svm_confirmed.predict(world_cases)

In [75]:  svm_confirmed

Out[75]:  SVR(C=0.1, cache_size=200, coef0=0.0, degree=3, epsilon=1, gamma=1,
            kernel='poly', max_iter=-1, shrinking=False, tol=0.001, verbose=False)

In [84]:  svm_pred

Out[84]:  array([112.08547206, 115.21958262, 116.84817641, 116.19545432,
                  95.85156538,  95.85156538, 116.94132883, 116.81253616,
                 106.43144693, 116.9500287 ,  92.27492511, 116.94290614,
                 116.28115248, 103.7313466 , 116.94464597, 116.34503103,
                 116.95003027, 113.28009602,  92.3695011 , 116.94215688,
                 106.43144693, 100.41652376, 112.77365579, 116.95003027,
                  92.27492511,  99.6979164 , 104.00000122, 116.22518602,
                 116.95003027, 114.38803048, 116.69438065, 114.868105  ,
                  95.85156538, 113.58062122, 111.14707474, 116.93479689,
                 116.9499428 ,  97.65082336, 101.59364424, 100.41652376,
                 105.50714017, 108.85045131, 116.94987405,  95.85156538,
                 116.95003024, 111.8671233 ,  95.85156538,  95.85156538,
                 116.95003007, 116.95003022, 116.94974562, 116.94983946,
                 116.91544273, 115.62223035, 116.95003027, 116.82562255,
                 115.86558877, 116.25373605, 116.91612761, 116.94877984,
                 116.95003027, 116.94115305, 116.95003027, 116.95003024,
                 112.19085287, 116.94997045, 116.95003012, 113.72083407,
                 115.35540113, 102.88218756, 112.19085287, 100.41652376,
                 116.81253616, 116.950026  , 116.94495952, 115.56720047,
                 116.9496611 , 115.84345234, 103.86386485, 112.29378269,
```

Fig xvi) Applying SVM model amd extracting other factors

Step 8:Creating plot between svm_test_predicted data and svm_test_confirm data. Also print values of MAE and MSE i.e Mean_absolute_error and mean_squared_error

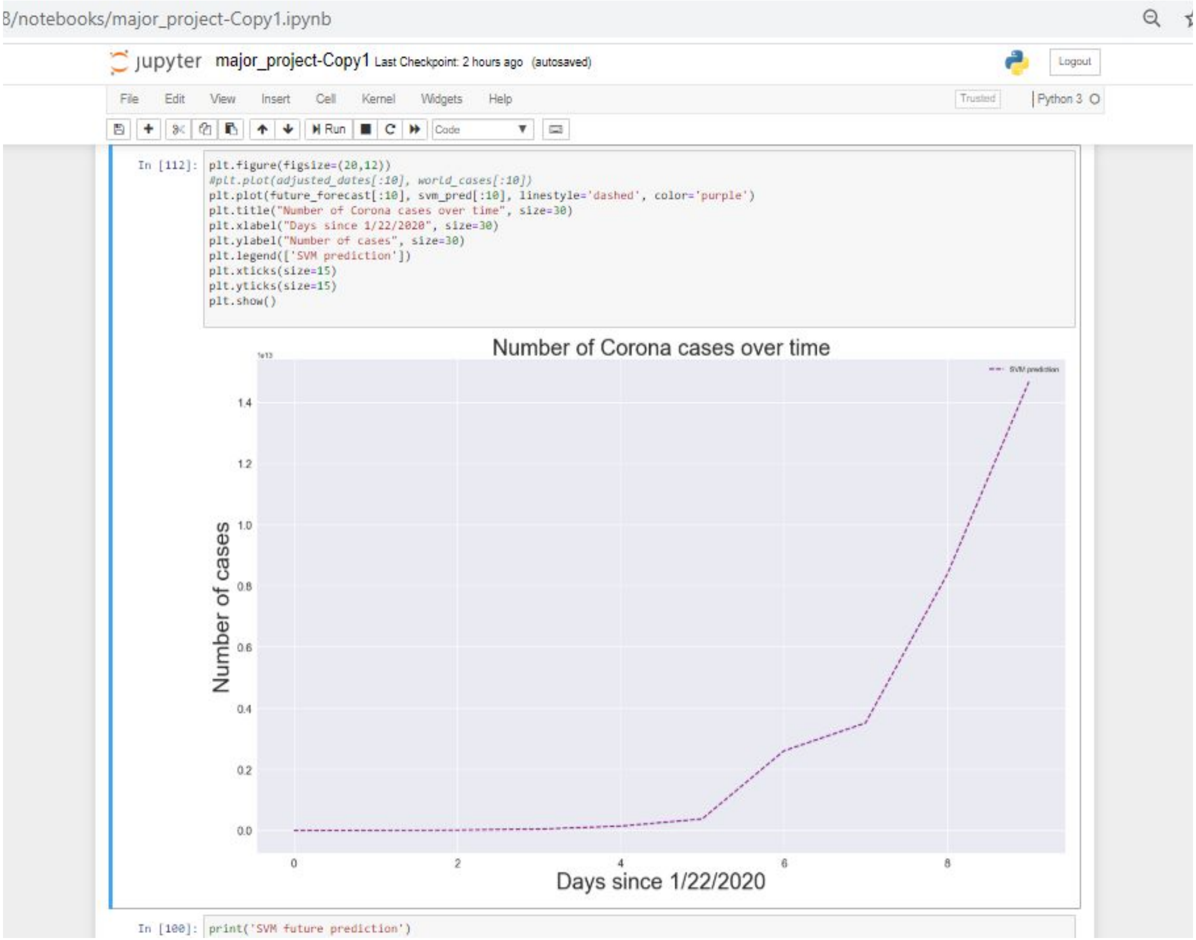


ost:8888/notebooks/major_project-Copy1.ipynb
jupyter major_project-Copy1 Last Checkpoint: 2 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
12000
11500
11000
10500
10000
9500
9000
8500
0.0
0.2
0.4
0.6
0.8
1.0
Number of Corona cases over time
10000

```
          103.86386485])

In [76]:  from sklearn.metrics import mean_absolute_error
          from sklearn.metrics import mean_squared_error
          svm_test_pred = svm_confirmed.predict(X_test_confirmed)
          plt.plot(svm_test_pred)
          plt.plot(y_test_confirmed)
          print('MAE:',mean_absolute_error(svm_test_pred, y_test_confirmed))
          print('MSE:', mean_squared_error(svm_test_pred, y_test_confirmed))

          MAE: 1243.7310126582279
          MSE: 2153614.158247875

In [78]:  plt.figure(figsize=(20,12))
          plt.plot(days_since_1_22[:10], world_cases)
          plt.title("Number of Corona cases over time", size=30)
          plt.xlabel("Days since 1/22/2020", size=30)
          plt.ylabel("Number of cases", size=30)
          plt.xticks(size=15)
          plt.yticks(size=15)
          plt.show()
```

Fig xvii) Printing and plotting the value of MAE and MSE

Step 9: Representing and plotting data of 40 days where x axis has the dats and y axis has the number of cases. The plot in the below figure represents the SVM prediction of trained model.

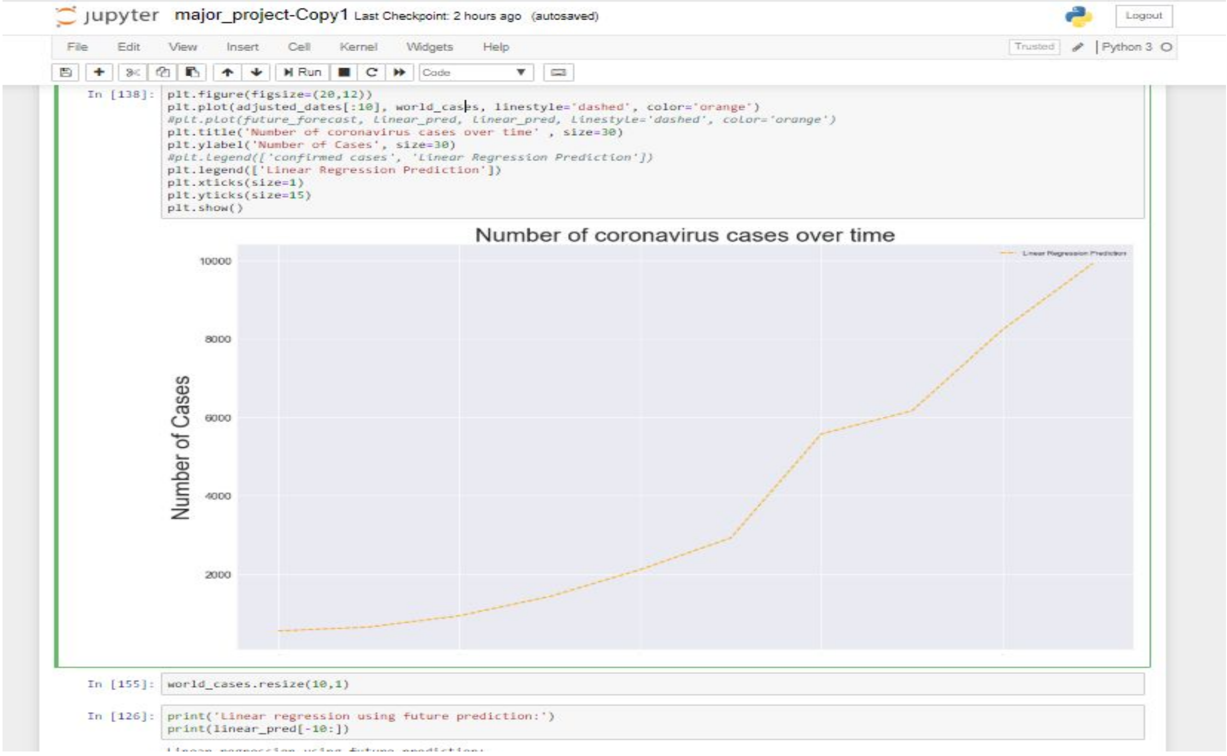


```
In [112]: plt.figure(figsize=(20,12))
          #plt.plot(adjusted_dates[:10], world_cases[:10])
          plt.plot(future_forecast[:10], svm_pred[:10], linestyle='dashed', color='purple')
          plt.title("Number of Corona cases over time", size=30)
          plt.xlabel("Days since 1/22/2020", size=30)
          plt.ylabel("Number of cases", size=30)
          plt.legend(['SVM prediction'])
          plt.xticks(size=15)
          plt.yticks(size=15)
          plt.show()
```

```
In [100]: print('SVM future prediction')
```

Fig xviii) SVM prediction plot with number of cases and days Step 10)

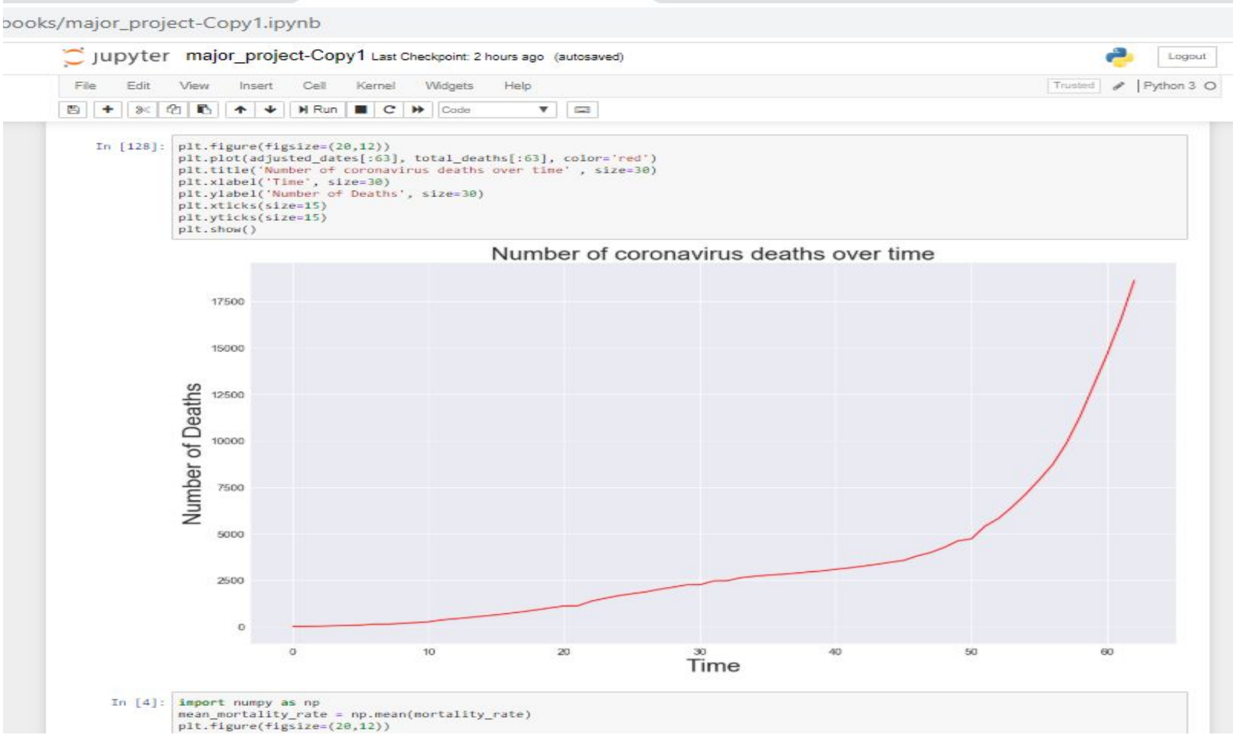
Step 10: Predicting data using Linear Regression plotting and analyzing the trend of disease using this technique from scikit library.


jupyter
major_project-Copy1 Last Checkpoint: 2 hours ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Number of coronavirus cases over time
Linear Regression Prediction
10000
8000
6000
4000
2000
Number of Cases

```
In [138]: plt.figure(figsize=(20,12))
          plt.plot(adjusted_dates[:10], world_cases, linestyle='dashed', color='orange')
          #plt.plot(future_forecast, linear_pred, linear_pred, linestyle='dashed', color='orange')
          plt.title('Number of coronavirus cases over time' , size=30)
          plt.ylabel('Number of Cases', size=30)
          #plt.legend(['confirmed cases', 'Linear Regression Prediction'])
          plt.legend(['Linear Regression Prediction'])
          plt.xticks(size=1)
          plt.yticks(size=15)
          plt.show()

In [155]: world_cases.resize(10,1)

In [126]: print('Linear regression using future prediction:')
          print(linear_pred[-10:])

          Linear regression using future prediction:
```

Fig xix) Applying Linear Regression on Data

STEP 11: Linear Regression has been implemented on data, where the red line plotted in the graph is representing the output obtained by plotting two constraints on a graph.



```
In [128]: plt.figure(figsize=(20,12))
          plt.plot(adjusted_dates[:63], total_deaths[:63], color='red')
          plt.title('Number of coronavirus deaths over time' , size=30)
          plt.xlabel('Time', size=30)
          plt.ylabel('Number of Deaths', size=30)
          plt.xticks(size=15)
          plt.yticks(size=15)
          plt.show()

In [4]:   import numpy as np
          mean_mortality_rate = np.mean(mortality_rate)
          plt.figure(figsize=(20,12))
```

Fig xx) Linear regression plot

CHAPTER 5

ANALYSIS OF WORLDWIDE COVID-19 CASES

COVID-19 reported its 562 total confirmed cases with 9 deaths on 25th March Afternoon in India

1)Trend in INDIA VS ITALY,CHINA & SOUTH KOREA

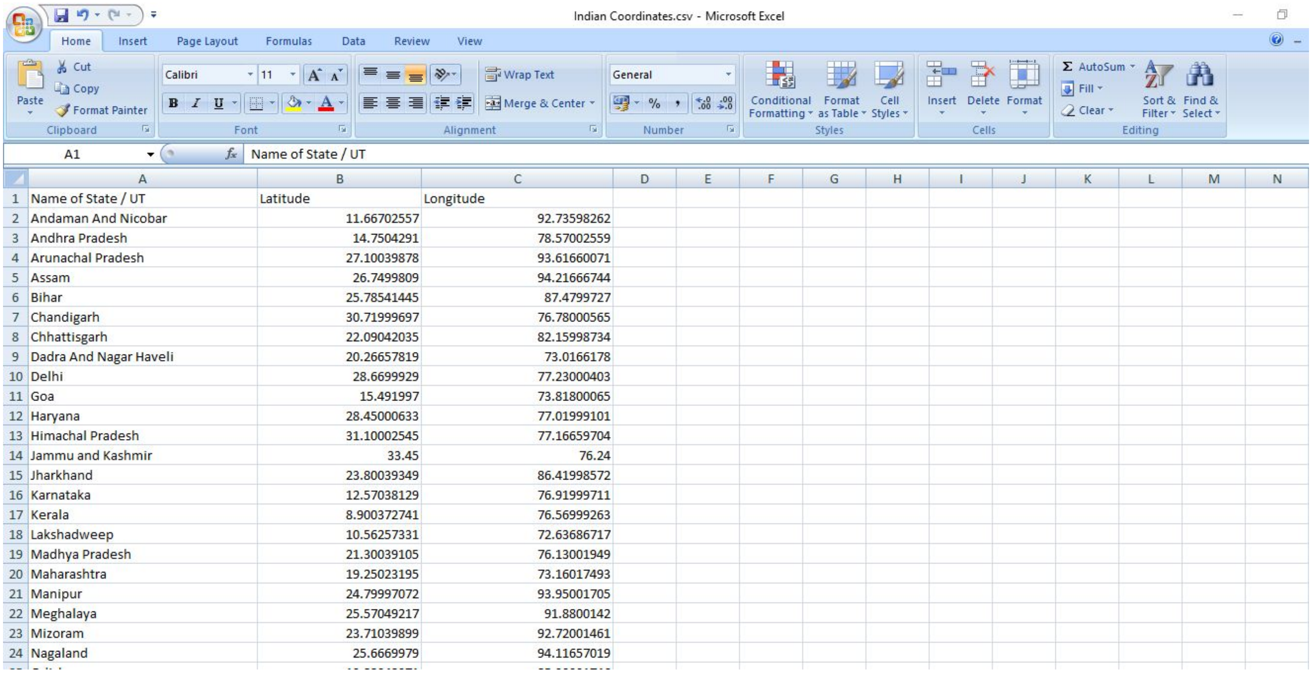
2)Trend across the world

3)Forecast & prediction of the virus

5.1Datasets Used:

Screenshot of some of new datasets that have to be used in the data analysis , others are used which were described in the start of the report.

1)Indian coordinates used to plot the map for representation

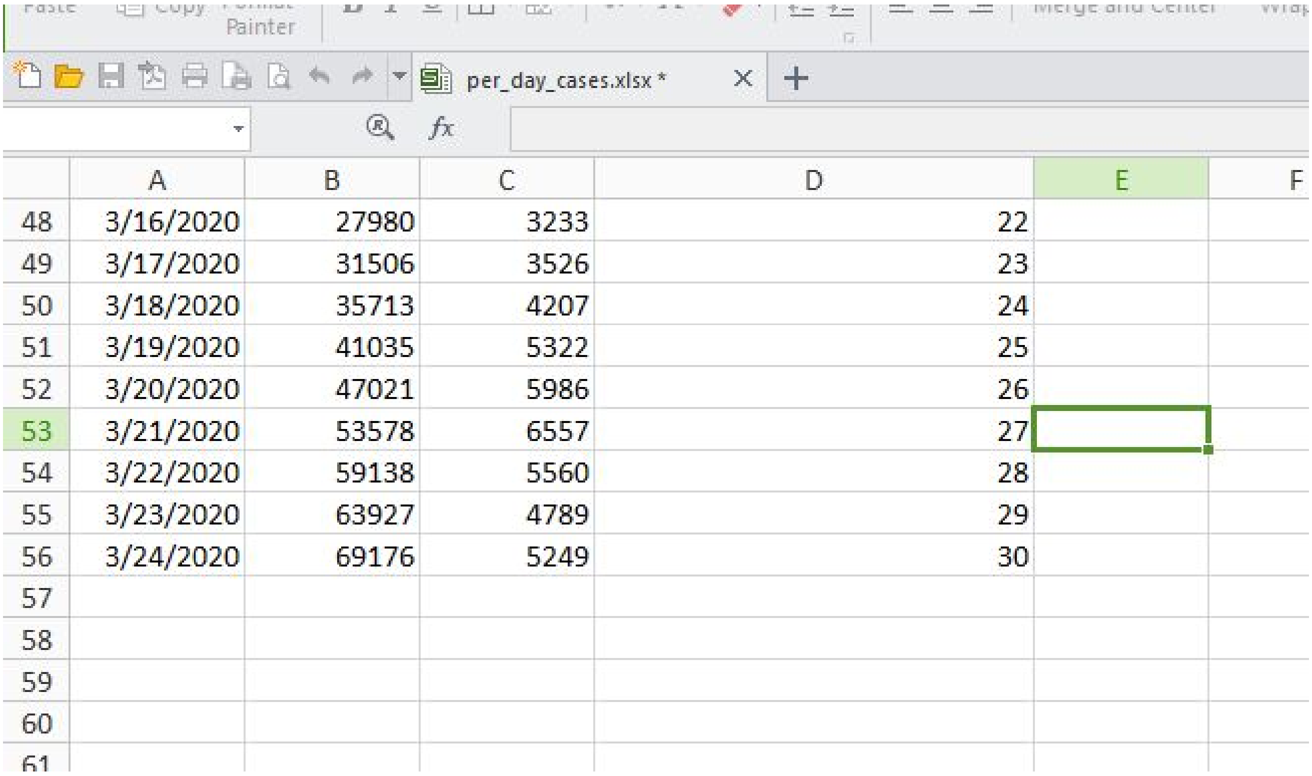


A1 | Name of State / UT

	A	B	C
1	Name of State / UT	Latitude	Longitude
2	Andaman And Nicobar	11.66702557	92.73598262
3	Andhra Pradesh	14.7504291	78.57002559
4	Arunachal Pradesh	27.10039878	93.6166071
5	Assam	26.7499809	94.21666744
6	Bihar	25.78541445	87.4799727
7	Chandigarh	30.71999697	76.78000565
8	Chhattisgarh	22.09042035	82.15998734
9	Dadra And Nagar Haveli	20.26657819	73.0166178
10	Delhi	28.6699929	77.23000403
11	Goa	15.491997	73.81800065
12	Haryana	28.45000633	77.01999101
13	Himachal Pradesh	31.10002545	77.16659704
14	Jammu and Kashmir	33.45	76.24
15	Jharkhand	23.80039349	86.41998572
16	Karnataka	12.57038129	76.91999711
17	Kerala	8.900372741	76.56999263
18	Lakshadweep	10.56257331	72.63686717
19	Madhya Pradesh	21.30039105	76.13001949
20	Maharashtra	19.25023195	73.16017493
21	Manipur	24.79997072	93.95001705
22	Meghalaya	25.57049217	91.8800142
23	Mizoram	23.71039899	92.72001461
24	Nagaland	25.6669979	94.11657019

Fig xxi) Indian_coordinates.csv

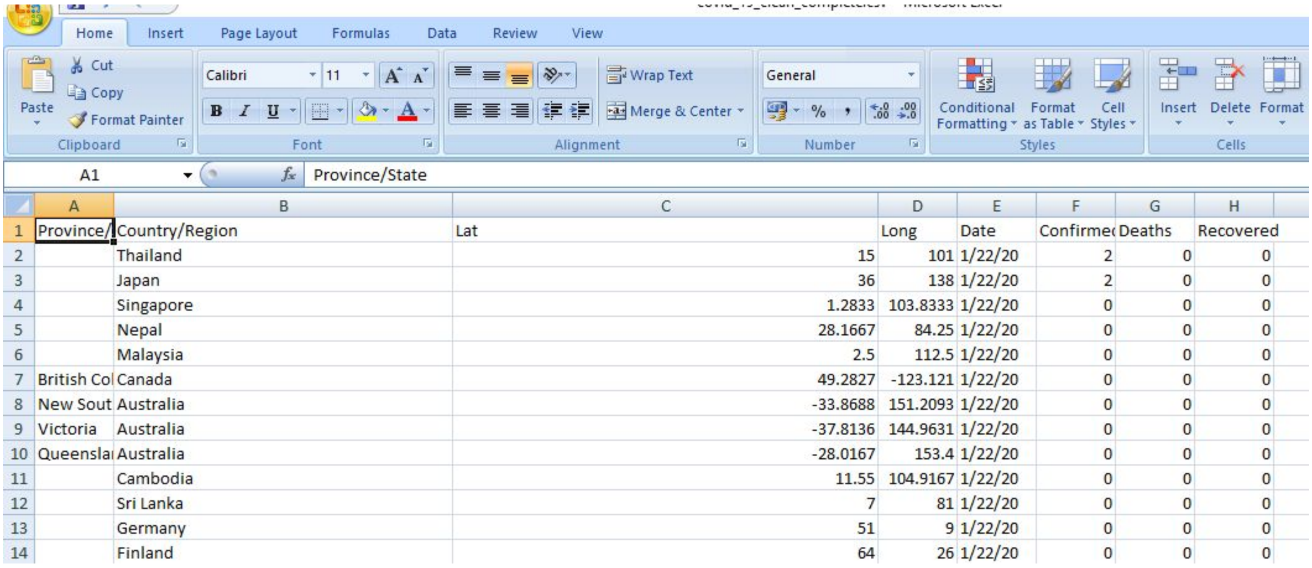
2) per_day_cases describes the updates new cases



	A	B	C	D	E	F
48	3/16/2020	27980	3233	22		
49	3/17/2020	31506	3526	23		
50	3/18/2020	35713	4207	24		
51	3/19/2020	41035	5322	25		
52	3/20/2020	47021	5986	26		
53	3/21/2020	53578	6557	27		
54	3/22/2020	59138	5560	28		
55	3/23/2020	63927	4789	29		
56	3/24/2020	69176	5249	30		
57						
58						
59						
60						

Fig xxii) Per_day_cases.csv

3)Cleaned_dataset_for_indian_corona_cases

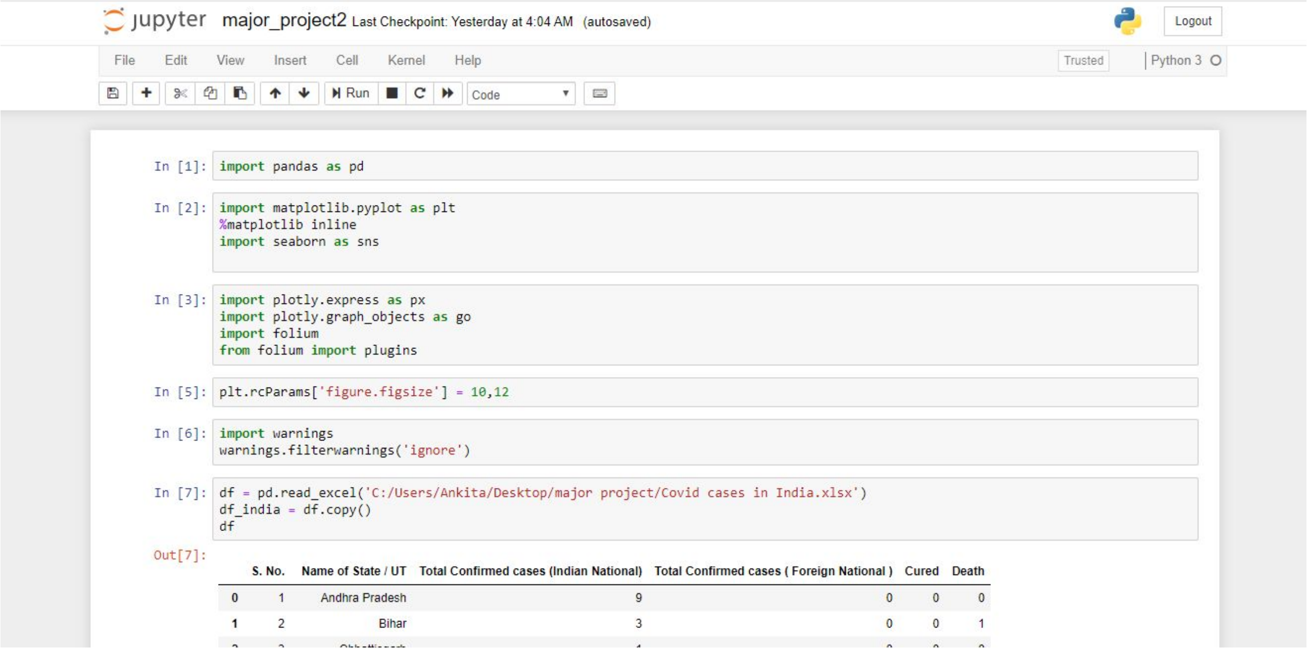


A1 Province/State

	A	B	C	D	E	F	G	H
1	Province/	Country/Region	Lat	Long	Date	Confirmec	Deaths	Recovered
2		Thailand	15	101	1/22/20	2	0	0
3		Japan	36	138	1/22/20	2	0	0
4		Singapore	1.2833	103.8333	1/22/20	0	0	0
5		Nepal	28.1667	84.25	1/22/20	0	0	0
6		Malaysia	2.5	112.5	1/22/20	0	0	0
7	British Col	Canada	49.2827	-123.121	1/22/20	0	0	0
8	New Sout	Australia	-33.8688	151.2093	1/22/20	0	0	0
9	Victoria	Australia	-37.8136	144.9631	1/22/20	0	0	0
10	Queenslai	Australia	-28.0167	153.4	1/22/20	0	0	0
11		Cambodia	11.55	104.9167	1/22/20	0	0	0
12		Sri Lanka	7	81	1/22/20	0	0	0
13		Germany	51	9	1/22/20	0	0	0
14		Finland	64	26	1/22/20	0	0	0

5.2Procedure:

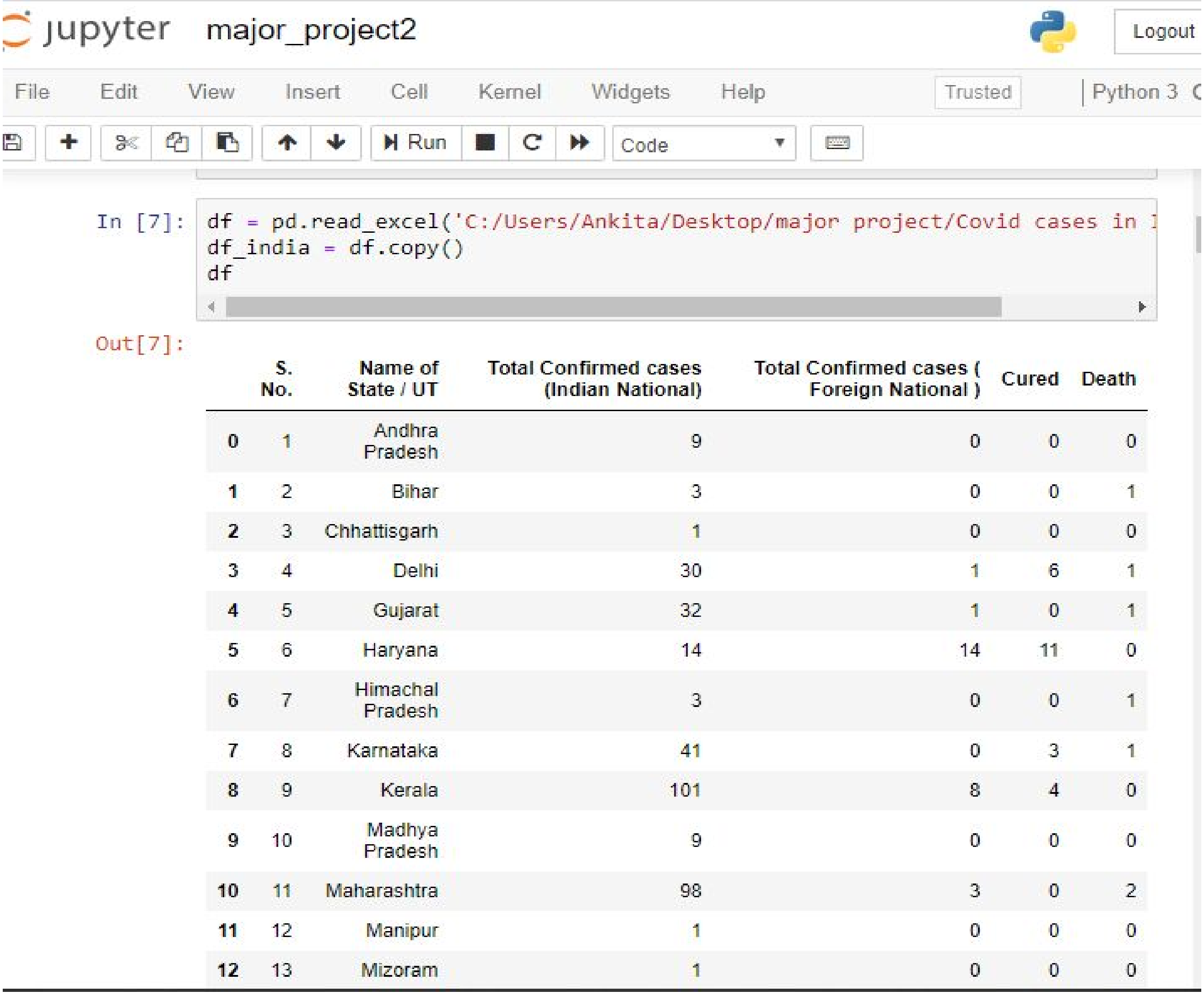
1)Importing libraries and datasets and obtaining heads


jupyter major_project2 Last Checkpoint: Yesterday at 4:04 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Help
Trusted
Python 3
Run
Code

```
In [1]: import pandas as pd

In [2]: import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns

In [3]: import plotly.express as px
        import plotly.graph_objects as go
        import folium
        from folium import plugins

In [5]: plt.rcParams['figure.figsize'] = 10,12

In [6]: import warnings
        warnings.filterwarnings('ignore')

In [7]: df = pd.read_excel('C:/Users/Ankita/Desktop/major project/Covid cases in India.xlsx')
        df_india = df.copy()
        df
```

Out[7]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	9	0	0	0
1	2	Bihar	3	0	0	1



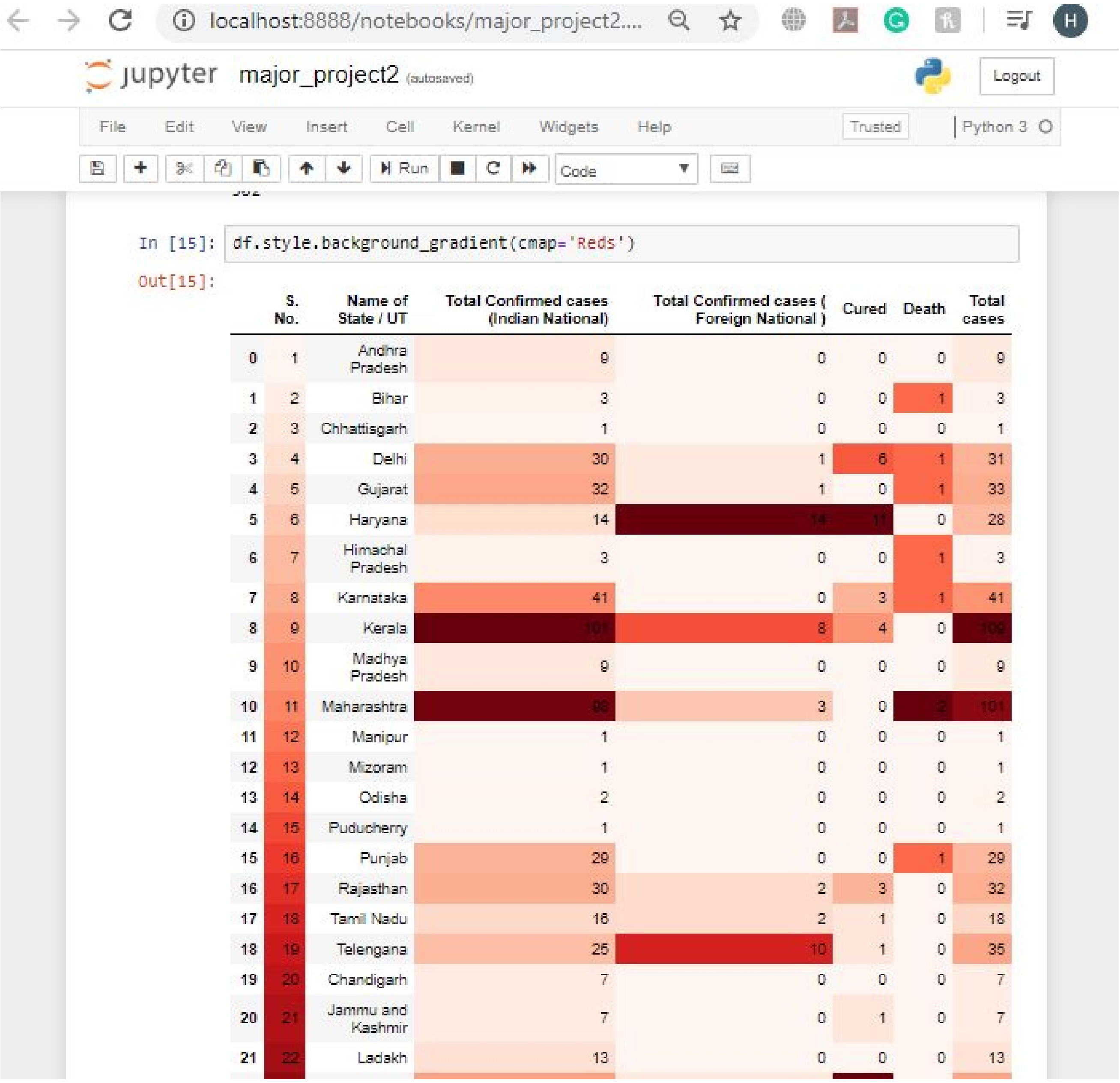
```
In [7]: df = pd.read_excel('C:/Users/Ankita/Desktop/major project/Covid cases in I
        df_india = df.copy()
        df
```

Out[7]:

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death
0	1	Andhra Pradesh	9	0	0	0
1	2	Bihar	3	0	0	1
2	3	Chhattisgarh	1	0	0	0
3	4	Delhi	30	1	6	1
4	5	Gujarat	32	1	0	1
5	6	Haryana	14	14	11	0
6	7	Himachal Pradesh	3	0	0	1
7	8	Karnataka	41	0	3	1
8	9	Kerala	101	8	4	0
9	10	Madhya Pradesh	9	0	0	0
10	11	Maharashtra	98	3	0	2
11	12	Manipur	1	0	0	0
12	13	Mizoram	1	0	0	0

Fig xxiii) Exploring dataset using head function

Step 2: Analyzing data by setting shades of red the greater the number, the darker it is else it is light and has varying shades of set colour red.

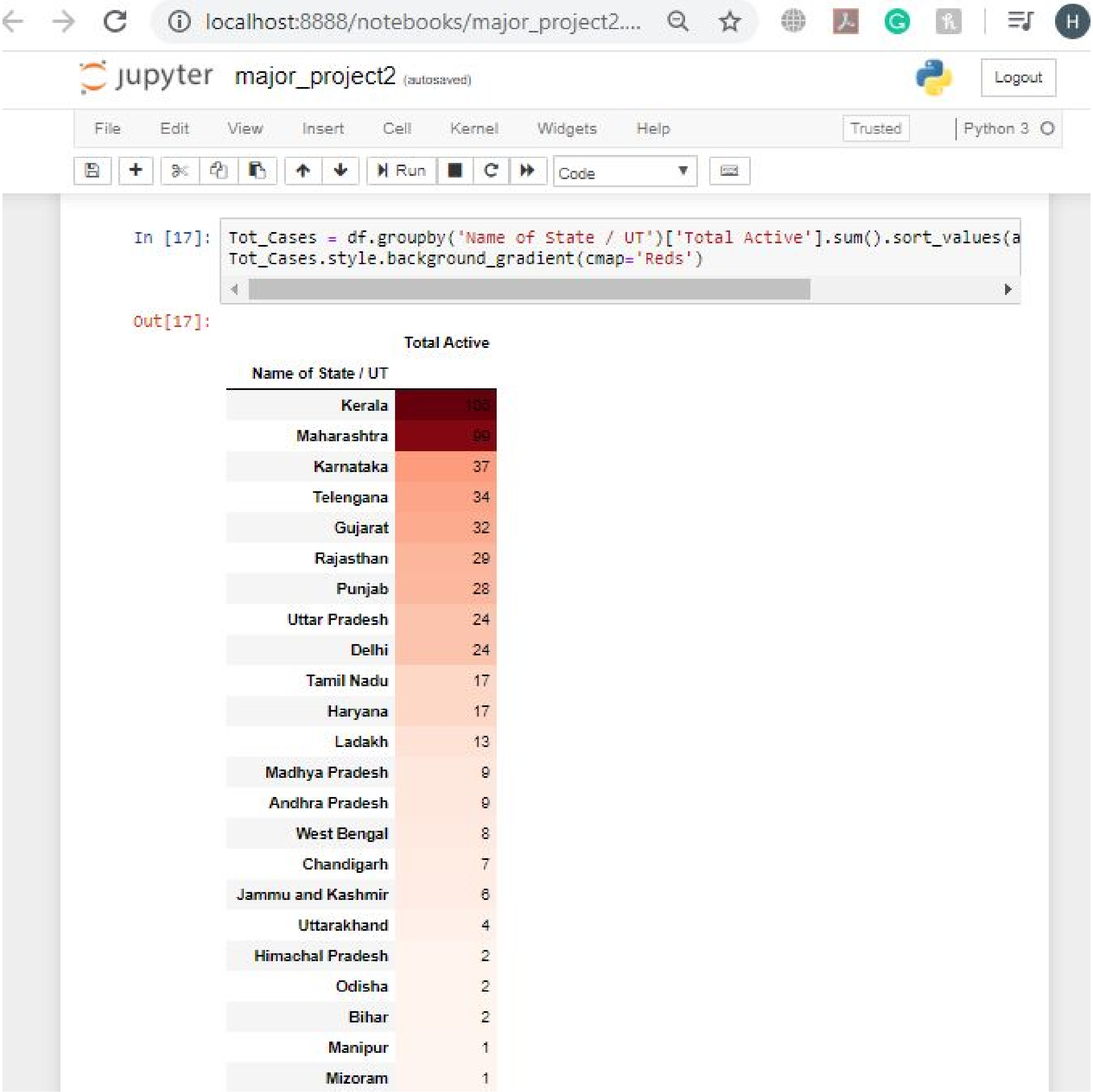


```
In [15]: df.style.background_gradient(cmap='Reds')
Out[15]:
```

	S. No.	Name of State / UT	Total Confirmed cases (Indian National)	Total Confirmed cases (Foreign National)	Cured	Death	Total cases
0	1	Andhra Pradesh	9	0	0	0	9
1	2	Bihar	3	0	0	1	3
2	3	Chhattisgarh	1	0	0	0	1
3	4	Delhi	30	1	6	1	31
4	5	Gujarat	32	1	0	1	33
5	6	Haryana	14	14	11	0	28
6	7	Himachal Pradesh	3	0	0	1	3
7	8	Karnataka	41	0	3	1	41
8	9	Kerala	101	8	4	0	109
9	10	Madhya Pradesh	9	0	0	0	9
10	11	Maharashtra	98	3	0	2	101
11	12	Manipur	1	0	0	0	1
12	13	Mizoram	1	0	0	0	1
13	14	Odisha	2	0	0	0	2
14	15	Puducherry	1	0	0	0	1
15	16	Punjab	29	0	0	1	29
16	17	Rajasthan	30	2	3	0	32
17	18	Tamil Nadu	16	2	1	0	18
18	19	Telengana	25	10	1	0	35
19	20	Chandigarh	7	0	0	0	7
20	21	Jammu and Kashmir	7	0	1	0	7
21	22	Ladakh	13	0	0	0	13

Fig xxiv) Highlighting more numbers by darker shades and light with less number

Step3: Number of Cases of effected people i.e hospital subtract cured and deaths to calculate total number of active cases.


localhost:8888/notebooks/major_project2....
jupyter major_project2 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

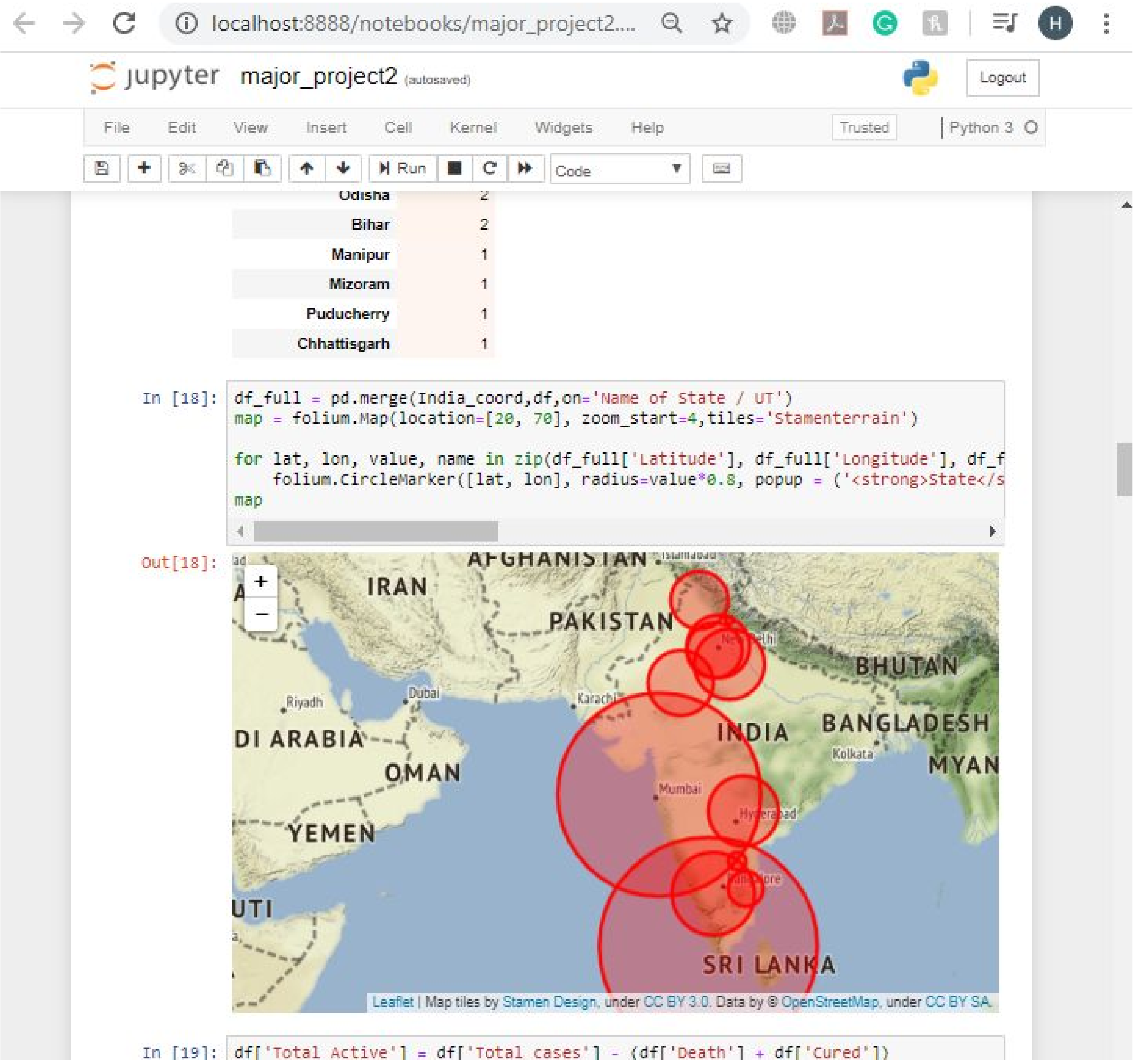
In [17]:

```
Tot_Cases = df.groupby('Name of State / UT')['Total Active'].sum().sort_values(a
Tot_Cases.style.background_gradient(cmap='Reds')
```

Out[17]:

	Total Active
Name of State / UT	
Kerala	105
Maharashtra	99
Karnataka	37
Telengana	34
Gujarat	32
Rajasthan	29
Punjab	28
Uttar Pradesh	24
Delhi	24
Tamil Nadu	17
Haryana	17
Ladakh	13
Madhya Pradesh	9
Andhra Pradesh	9
West Bengal	8
Chandigarh	7
Jammu and Kashmir	6
Uttarakhand	4
Himachal Pradesh	2
Odisha	2
Bihar	2
Manipur	1
Mizoram	1

Step 4: Plotting on coordinates using red circles with the use of indian coordinates data using library folium.



localhost:8888/notebooks/major_project2....
jupyter major_project2 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code

Odisha	2
Bihar	2
Manipur	1
Mizoram	1
Puducherry	1
Chhattisgarh	1

```
In [18]: df_full = pd.merge(India_coord,df,on='Name of State / UT')
         map = folium.Map(location=[20, 70], zoom_start=4,tiles='Stamenterrain')

         for lat, lon, value, name in zip(df_full['Latitude'], df_full['Longitude'], df_f
             folium.CircleMarker([lat, lon], radius=value*0.8, popup = ('<strong>State</s
         map
```

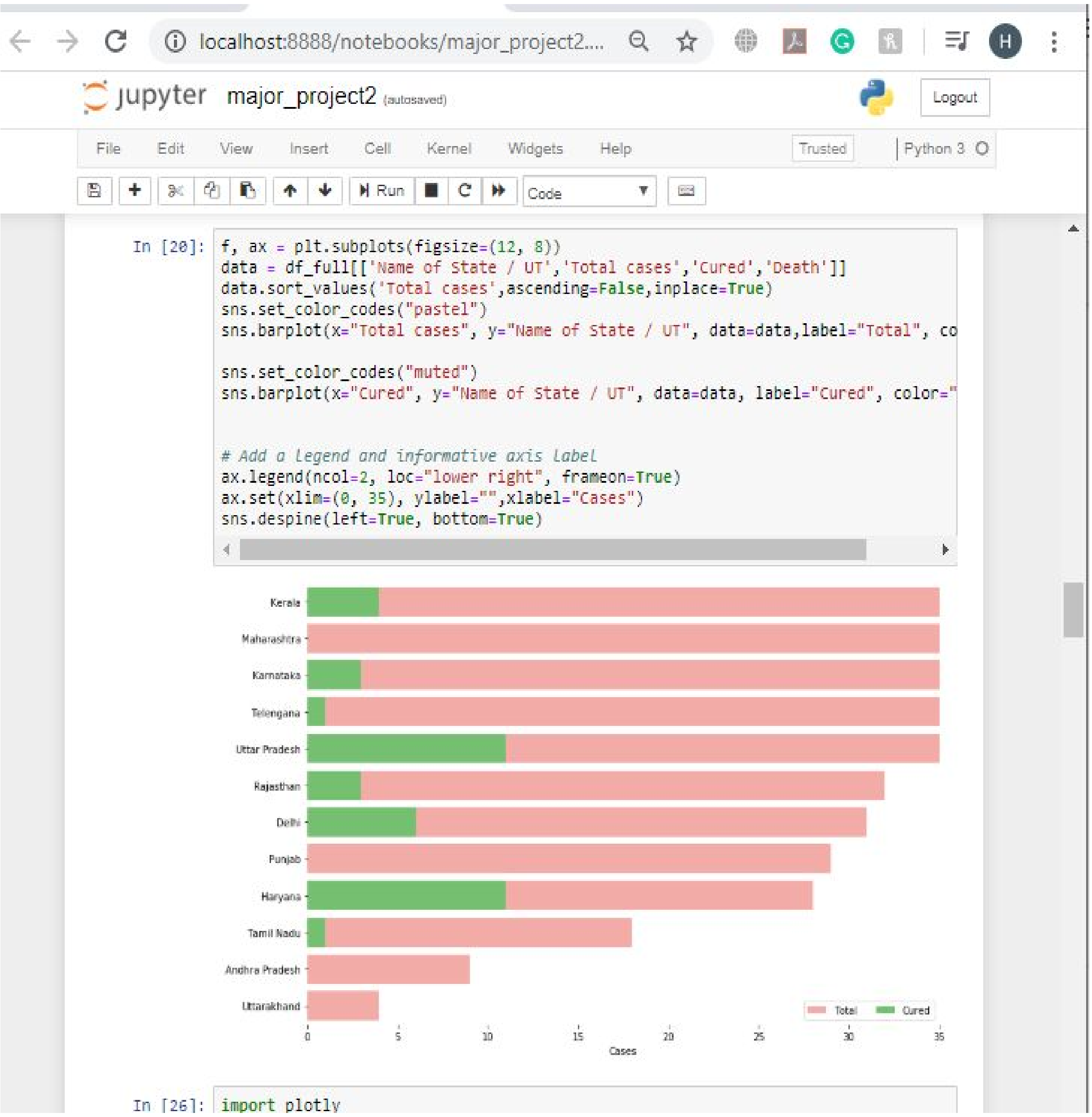
Out[18]:

+
−
AFGHANISTAN
Islamabad
IRAN
PAKISTAN
New Delhi
BHUTAN
Riyadh
Dubai
Karachi
DI ARABIA
INDIA
BANGLADESH
Kolkata
MYAN
OMAN
Mumbai
Hyderabad
YEMEN
Bangalore
UTI
SRI LANKA
Leaflet | Map tiles by Stamen Design, under CC BY 3.0. Data by © OpenStreetMap, under CC BY SA.

```
In [19]: df['Total Active'] = df['Total cases'] - (df['Death'] + df['Cured'])
```

Fig xxv) Plotting a countries map with active case using circles

Step 6:Using seaborn to represent bar plots in pink total cases and in green the cured cases.



```
In [20]: f, ax = plt.subplots(figsize=(12, 8))
         data = df_full[['Name of State / UT','Total cases','Cured','Death']]
         data.sort_values('Total cases',ascending=False,inplace=True)
         sns.set_color_codes("pastel")
         sns.barplot(x="Total cases", y="Name of State / UT", data=data,label="Total", co

         sns.set_color_codes("muted")
         sns.barplot(x="Cured", y="Name of State / UT", data=data, label="Cured", color="


         # Add a legend and informative axis label
         ax.legend(ncol=2, loc="lower right", frameon=True)
         ax.set(xlim=(0, 35), ylabel="",xlabel="Cases")
         sns.despine(left=True, bottom=True)
```

```
In [26]: import plotly
```

Fig xxvi) Plotting countries recovered and active number of cases

Chapter -6

6.1 CONCLUSION

Considering this project this will help us analyze the trend of **data and predict the amount of growth this virus** will take.

The project done in previous semester i.e **DRUG RECOMMENDER SYSTEM** can provide great contribution to the doctors and researchers as once the medicine is entered and in future it is needed for the cure of covid-19 this drug recommender system can help us get the most appropriate drug that can be used to cure this , this will help us in reducing researching time and easily obtaining the name of the alternative drug that can be used.

6.2 How to protect yourself from covid -19?

Corona Virus spreads in a similar way to flow basically in large drops of flu and germs.Germs can stay on any surface for several hours public health england quotes the spread of virus by contact with person in two metres for 15 minutes.There is no vaccine uptil now. Staying away ,social
Distancing ,using sanitizer or washing hands is the possible way to avoid in coming in close contact with germs.

6.3 FUTURE SCOPE

Analyses can be done again once this pandemic comes to end or we are able to overcome it , this will help us not only summarize the data and statistics but also the loss of human resources we have faced. Also the drug recommender system can be updated as the new drugs are introduced in market this will be a really essential and vital tool for doctors and health staff.

6.4 REFERENCES

https://www.worldometers.info/coronavirus/

https://www.kaggle.com/data

https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3546113

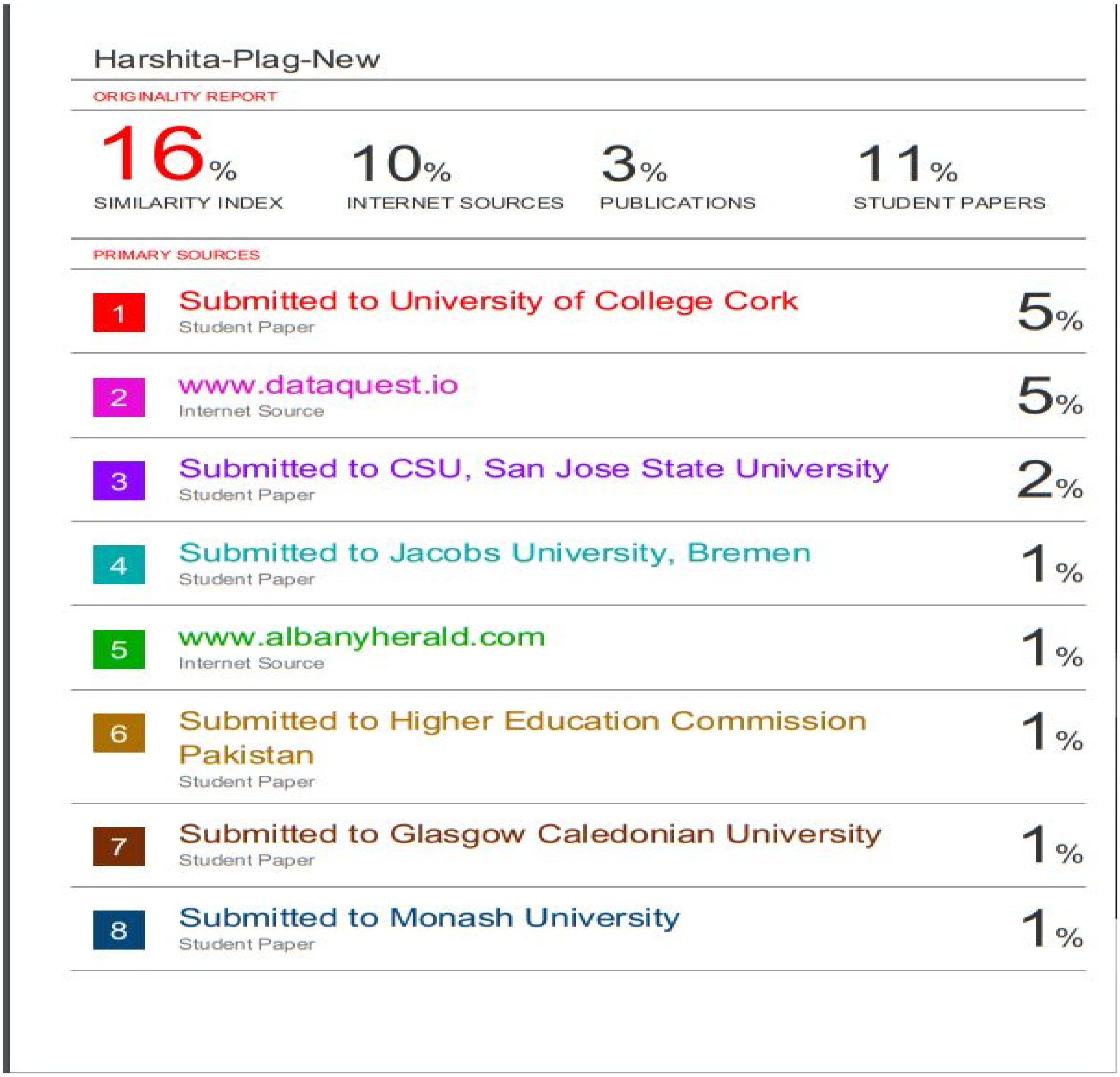
https://www.youtube.com/

https://www.opendataphilly.org/dataset/covid-cases/resource/d4d1e48a-d401-405c-972b-c45292c3d4f5

https://www.edureka.co/

https://www.python.org/

https://scikit-learn.org/



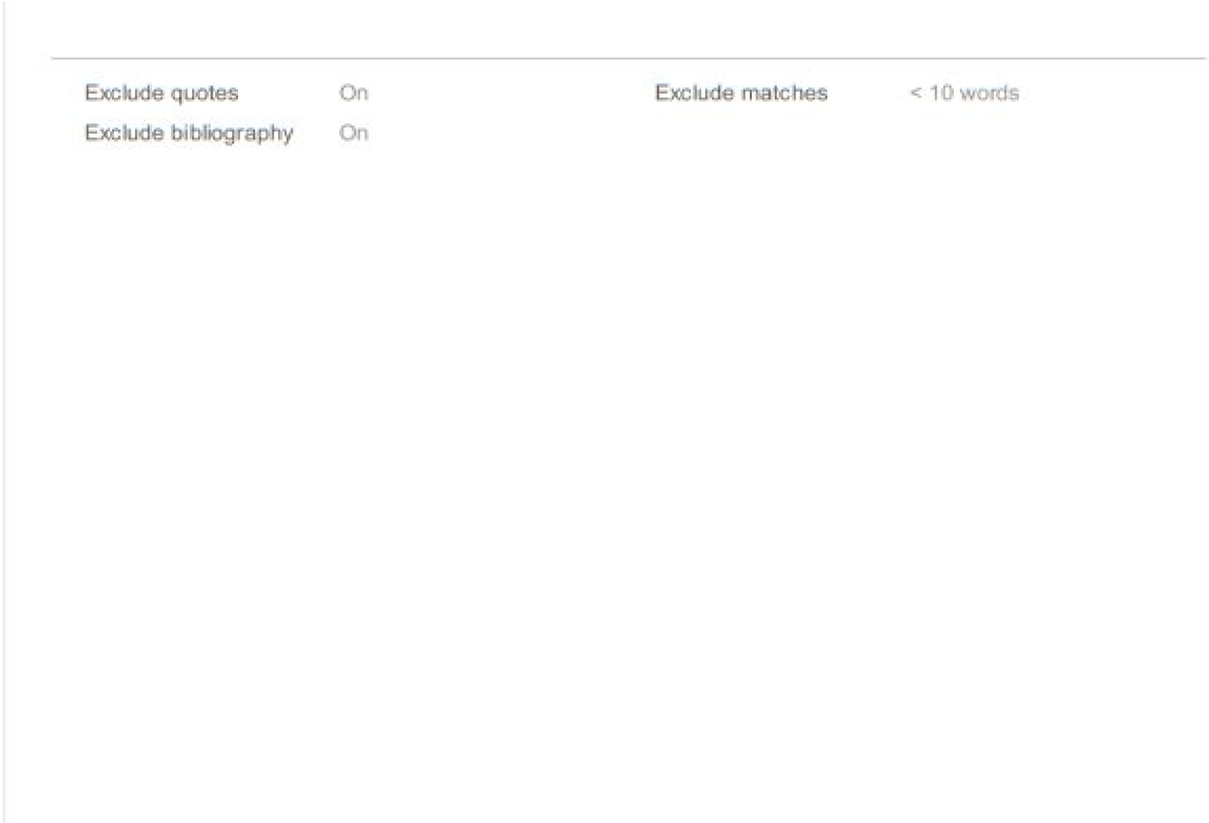
Harshita-Plag-New

ORIGINALITY REPORT

16%	10%	3%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of College Cork Student Paper	5%
2	www.dataquest.io Internet Source	5%
3	Submitted to CSU, San Jose State University Student Paper	2%
4	Submitted to Jacobs University, Bremen Student Paper	1%
5	www.albanyherald.com Internet Source	1%
6	Submitted to Higher Education Commission Pakistan Student Paper	1%
7	Submitted to Glasgow Caledonian University Student Paper	1%
8	Submitted to Monash University Student Paper	1%



Exclude quotes	On	Exclude matches	< 10 words
Exclude bibliography	On		

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: 15/7/20

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report [✓ B.Tech Project Report] Paper

Name: Harshita chauhan **Department:** cse **Enrolment No** 161251

Contact No. 9805108267 **E-mail.** harshitachauhan18.1998@gmail.com

Name of the Supervisor: Dr.Yugal kumar khola

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): HEALTH CARE DATA ANALYSIS USING MACHINE LEARNING

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 40
- Total No. of Preliminary pages = 6
- Total No. of pages accommodate bibliography/references = 2

Harshita chauhan

tudent)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at 16 (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

YUGAL 15/07/2020

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String	16	Word Counts	2753
Report Generated on			Character Counts	1400
Harshita chauhan		**Submission ID**	Total Pages Scanned	37
			File Size	40

Checked by
Name & Signature **Librarian**

………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com