**CH5019 TERM PROJECT**
**GROUP 20**

**QUESTION 1:**

***INTRODUCTION:***

Solving face recognition problem and finding the representative image by applying SINGULAR VALUE DECOMPOSITION. Also finding the accuracy at which the images are being recognized.

***TASK:***

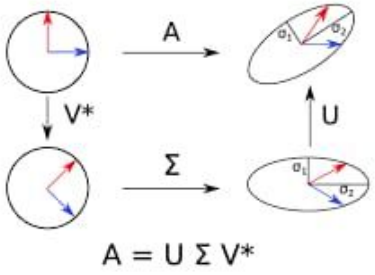
- We have images of 15 people in 10 different conditions each.
- Due to storage limitations, we have to have only one representative image for each person.
- The representative image is used to identify the images of a particular person in different conditions.

***THEORY:***

**SINGULAR VALUE DECOMPOSITION:**

- Singular value decomposition takes a rectangular matrix A (*n* x *p* matrix) in which the *n* rows represents the genes, and the *p* columns represents the experimental conditions.
- The SVD theorem states:

$$\mathbf{A}_{nxp} = \mathbf{U}_{nxn}\,\mathbf{S}_{nxp}\,\mathbf{V}^{\mathbf{T}}_{pxp}$$



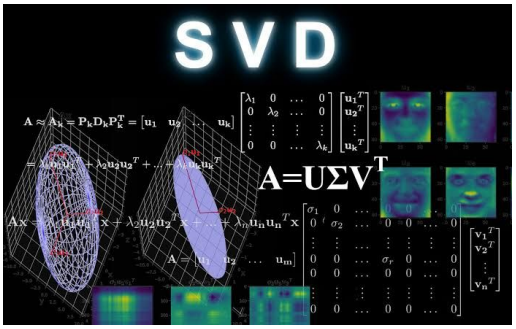

- Calculating the SVD consists of finding the eigenvalues and eigenvectors of $AA^T$ and $A^TA$.
- The eigenvectors of $A^TA$ make up the columns of *V,* the eigenvectors of $AA^T$ *make* up the columns of *U*.
- Also, the singular values in *S* are square roots of eigenvalues from $AA^T$ or $A^TA$.
- The singular values are the diagonal entries of the *S* matrix and are arranged in descending order.

**REPRESENTATIVE IMAGES:**

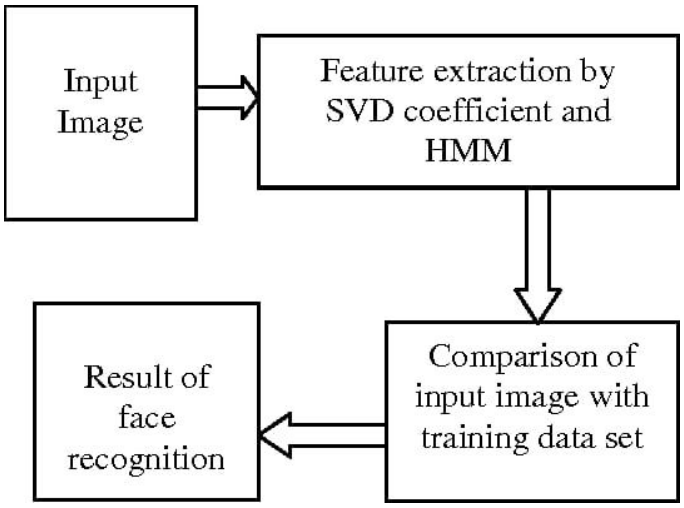
- $A = U_1S_1V_1' + U_2S_2V_2'$
- $A = U_1S_1V_1'$

  $AV_1 = U_1S_1$

  → Representative image




***ALGORITHM:***




- Using the "*imread*" command in MATLAB we convert the images into a pixel matrix of dimensions mxn
- We then use the "*reshape*" command to convert the pixel matrix of dimensions mxn into a column vector of dimensions mnx1.
- We get 10 such column vectors of dimensions mnx1.
- We put these column vectors together as a single matrix of dimensions mnx10.
- To this matrix we apply Singular Value Decomposition.

  i.e., A = U S V'

  mnx10 ← mnxmn ← 10x10 ← mnx10

- We get the representative image by taking product of eigenvectors and eigenvalues (first 2 terms)
- We get 10 such representative images.
- We now compare the different images with the representative image using the "*norm*" command in MATLAB.
- If the norm between a person's image and its corresponding representative image is minimum then it is identifying correctly.

***CODE EXPLANATION:***

***READING THE IMAGES:***

- We define images matrix of dimensions 64*64x10x15 and fill it with zeroes.
- We then read the 150 images using the "*imread*" command and update the images matrix.
- We then change the shape of the matrix using the "*reshape*" command.

***APPLICATION OF SVD AND GETTING REPRESENTATIVE IMAGES:***

- We define a zero matrix of dimensions 4096x1x15 and name it as "*rep_images*". This matrix is to store the representative image.
- We create a "*for*" loop which goes from *i=1 to i=15*.
- Inside the loop,
  - we define *matrix D* as
    D = images(:,:,i)
    i.e., it is a matrix containing all images of one particular person.
  - We now use the "*eig*" command on matrix D to get the eigenvectors and their eigenvalues.
    i.e., *[U,u] = eig (D*D')*
  - *"U"* matix stores the eigenvectors while "*u"* matrix stores the eigenvalues.
  - Representative images are obtained by
    *Rep_images(:,1,i) = D*U(:,10) + D*U(:,9)*
    (We consider the first two terms)

***COMPARING AND CHECKING:***

- We define a zero matrix of dimensions 1x15 and name it as *matrix N* (to store the Norm)
- We define three "*for*" loops
  *Loop i=1:15* **=>** *fix person*
  *Loop j=1:10* **=>***fix condition*
  *Loop k=1:15* **=>** *for comparing image with rep image*
- Inside the loops, we use the "*norm*" command which gives the Euclidean distance between the representative images and the images and stores the value in the N *matrix.*
- We define a variable "*count*" to find out how many images are being correctly recognized.
- If the norm between a person's image and its corresponding representative image is the minimum, then it is being correctly recognized.
- To find the minimum norm we used a "*for*" loop where *z* runs from *1 to 15.*
- We find accuracy which is equal to the number of images being identified correctly (count) for each "*i*" value.
- We store these accuracy values in a matrix of dimensions 15x1, "*correct_identification*"
- We define overall performance as the percentage of images being identified correctly out of the given 150 images.

*overall_performance = sum (correct_identification) *100 /150.*

***RESULTS:***

**Overall performance = 78%**

| Persons | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No.of images correctly identified | 8 | 8 | 9 | 9 | 8 | 8 | 5 | 8 | 10 | 6 | 10 | 9 | 7 | 5 | 7 |

The total number of images identified correctly out of the given 150 images is 117.

We are unable to achieve 100% performance because the representative image is not very accurate in terms of representing all the features of a person. This is because we used less number of images to get the representative image. If we were to obtain a representative image of each person using a larger number of images, the overall performance can be improved.

**QUESTION 2**

INTRODUCTION:

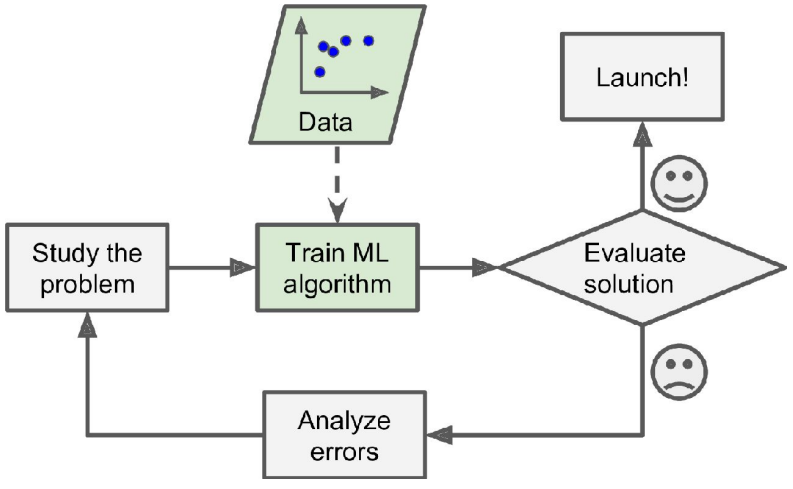
- Fitting logistic Regression Model(Classification Technique) for the given dataset.
- Current data is binary classification problem(Contains Classes : Pass ,Fail ) .
- Goal is to predict whether reactor will operate or fail under the given conditions using above fitted model

TASK DEFINITION:

- Inputs: **Temperature:** 400-700 K
  **Pressure:** 1-50 bar
  **Feed Flow Rate:** 50-200 kmol/hr
  **Coolant Flow Rate:** 1000-3600 L/hr
  **Inlet Reactant Concentration:** 0.1-0.5 mol fraction
- Output:Reactor Operating condition (Pass,Fail)

ALGORITHM DEFINITION:

- Basic ML approach



METHODOLOGY:

- Get the data : Using the pandas library
- Discover and visualize the data to gain insights :
  Observe the top five rows using the DataFrame's head() method

| | Temperature | Pressure | Feed Flow rate | Coolant Flow rate | Inlet reactant concentration | Test | |
|---|---|---|---|---|---|---|---|
| 0 | 406.86 | 17.66 | 121.83 | 2109.20 | 0.1033 | 1 | |
| 1 | 693.39 | 24.66 | 133.18 | 3138.96 | 0.3785 | 1 | |
| 2 | 523.10 | 23.23 | 146.55 | 1058.24 | 0.4799 | 0 | |
| 3 | 612.86 | 40.97 | 94.44 | 1325.12 | 0.3147 | 0 | |
| 4 | 500.28 | 37.44 | 185.48 | 2474.51 | 0.2284 | 1 | |

Each row represents one sample. There are 6 attributes.

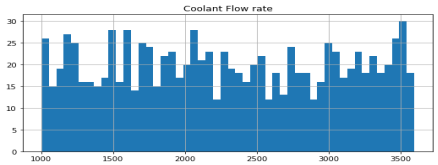
The info() method is useful to get a quick description of the data, regarding the total number of rows, each attribute's type, and the number of non-null values.

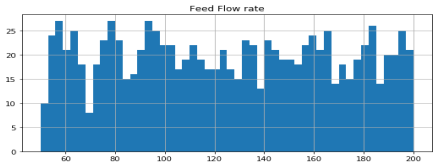
```
<class 'pandas.core.frame.DataFrame'>
        RangeIndex: 1000 entries, 0 to 999
        Data columns (total 6 columns):
        Temperature                 1000 non-null float64
        Pressure                    1000 non-null float64
        Feed Flow rate              1000 non-null float64
        Coolant Flow rate           1000 non-null float64
        Inlet reactant concentration 1000 non-null float64
        Test                        1000 non-null object
        dtypes: float64(5), object(1)
        memory usage: 47.0+ KB
```

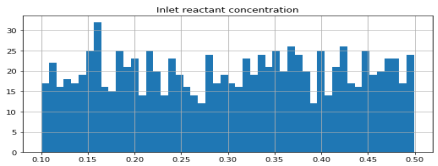
All attributes are numerical except Test field. It's type is an object. It may be a categorical attribute. Machine Learning Techniques gives a better result if data is numeric. Finding out any categorical features exist or not and Encoding them if they exist.

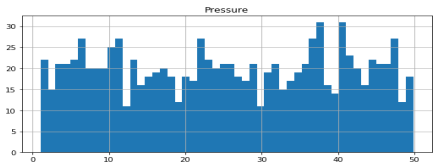
```
Pass    585
Fail    415
Name: Test, dtype: int64
```

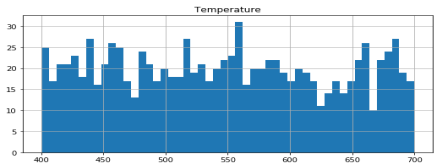
Visualizing the Data




Computing the standard correlation coefficient (also called Pearson's r) Test and every attributes using the corr() method:

| | |
|---|---|
| Test | 1.000000 |
| Coolant Flow rate | 0.762192 |
| Inlet reactant concentration | 0.008161 |
| Temperature | -0.008426 |
| Pressure | -0.048925 |
| Feed Flow rate | -0.092982 |
| Name: Test, dtype: float64 | |

- Splitting data into train_set,test_set:

- Training the model:

$$Y\_bar = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \theta_3 X_3 + \theta_4 X_4 + \theta_5 X_5 + \theta_6 X_6$$

$p(\theta) = h\ \theta\ (Y\_bar)$

$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Cost function for logistic Regression(log loss Function) :

$$J(\theta) = -1/m\{ \sum_{i=1-m} y(i) \log p(i) + (1 - y(i)) \log (1 - p(i)) \}$$

Logistic cost function partial derivatives:

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * (\sum_{i=1}^{m} \left[ y^{(i)} * \left(1 - h_\theta(x^{(i)})\right) * x_j^i - (1 - y^{(i)}) * h_\theta(x^{(i)}) * * x_j^i \right]$$

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * (\sum_{i=1}^{m} [\ y^{(i)} - y^{(i)} * h_\theta(x^{(i)}) - h_\theta(x^{(i)}) + y^{(i)} * h_\theta(x^{(i)})] * x_j^i\ )$$

$$\frac{\partial(J(\theta))}{\partial(\theta j)} = -\frac{1}{m} * (\sum_{i=1}^{m} [\ y^{(i)} - h_\theta(x^{(i)})] * x_j^i\ )$$

Gradient_descent:

$\theta$ (next step) = $\theta$ – alpha*grad(J( $\theta$ ))  {alpha = 0.2}

- If Y_bar is non-negative then we get p>0.5(class => 1) otherwise we get p<0.5(class => 0).
- Decision boundary is Y_bar. Fitting the model for X_train,Y_train.
- Predicting test data with the above model. Evaluating the performance of the above model over test data.

Initial parameter values: [ [ 812.25329167]
[ 392.31241846]
[ 496.96732017]
[1009.05307749]
[1001.37857587]
[ 788.45853768] ]

Final Parameters values : [ [7.34796758e+03]
[3.60888596e+06]
[1.74694324e+05]
[8.48815910e+05]
[1.05390770e+07]
[2.74970425e+03] ]

Confusion_matrix : [[ 0 110]
[ 1 189]]

Accuracy_score : 0.63

| Report : | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.00 | 0.00 | 0.00 | 110 |
| 1 | 0.63 | 0.99 | 0.77 | 190 |
| accuracy | | | 0.63 | 300 |
| macro avg | 0.32 | 0.50 | 0.39 | 300 |
| weighted avg | 0.40 | 0.63 | 0.49 | 300 |

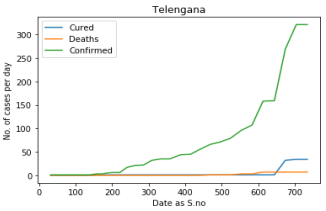
***QUESTION 3***

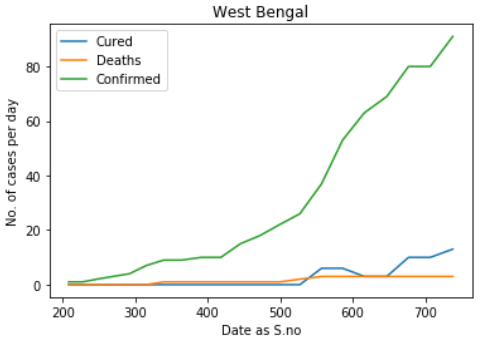
**TASK:** Visualize given CoVid Dataset and draw useful insights.

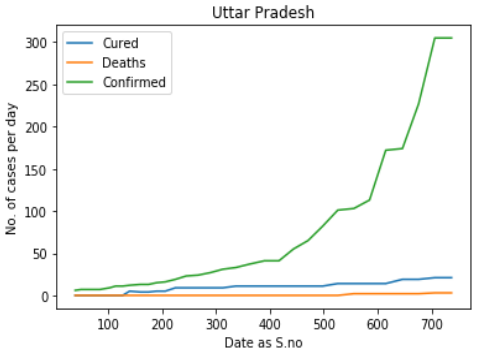
**LIBRARIES USED:** Pandas,Matplotlib.

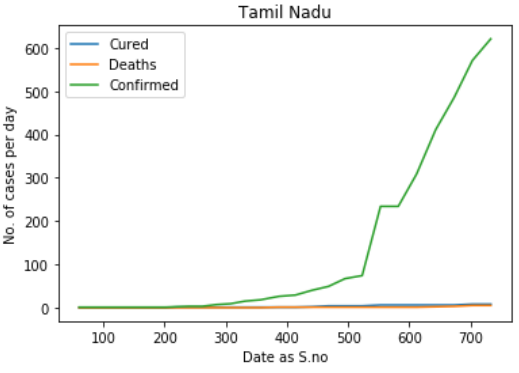
**1)**The age group of 20-29 is the most infected followed by the age group of 30-39.

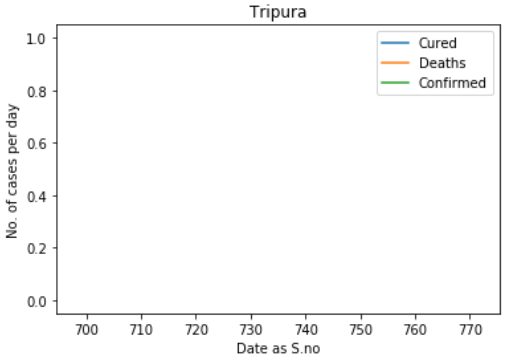
**2)**We isolated the individual states' data and plotted using the plot() function in Pandas.

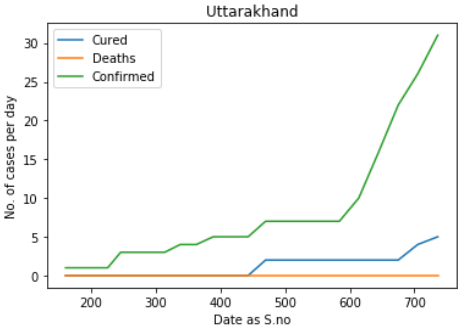


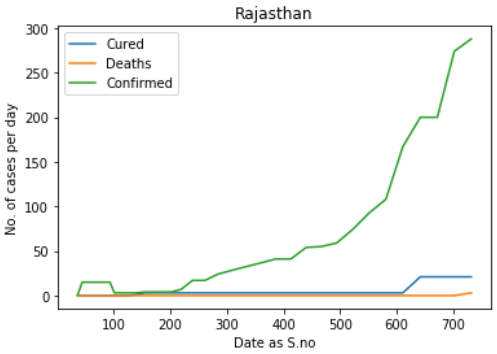



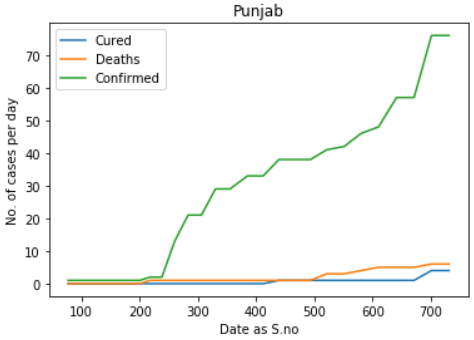



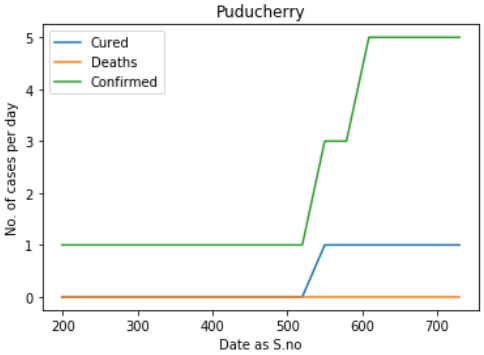



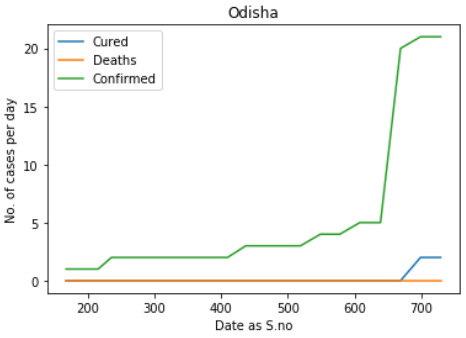



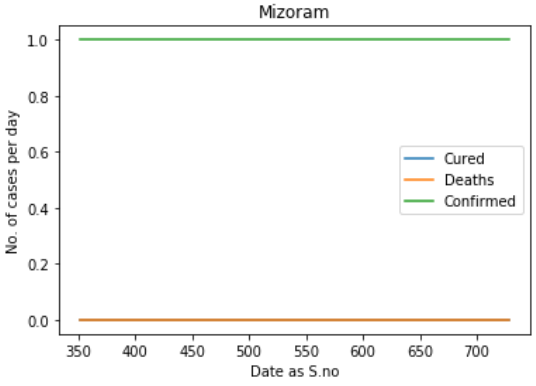



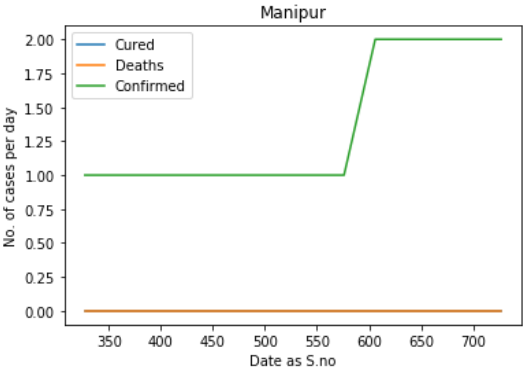

Rajasthan
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

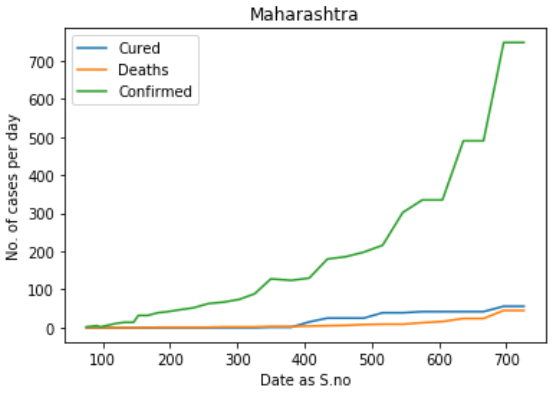

Punjab
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

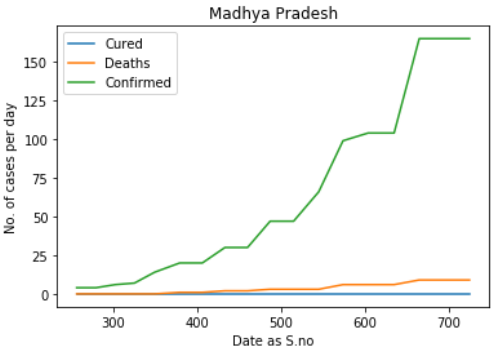

Puducherry
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

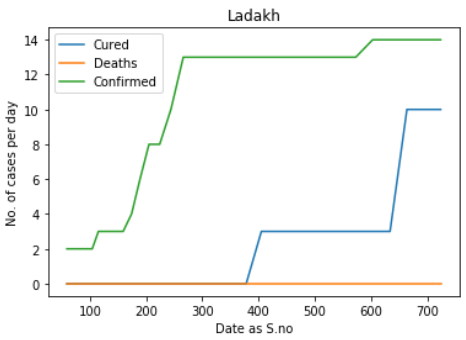

Odisha
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

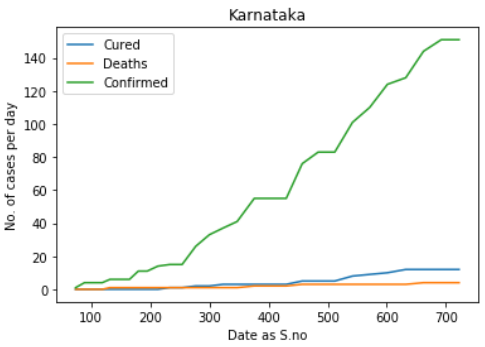

Mizoram
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

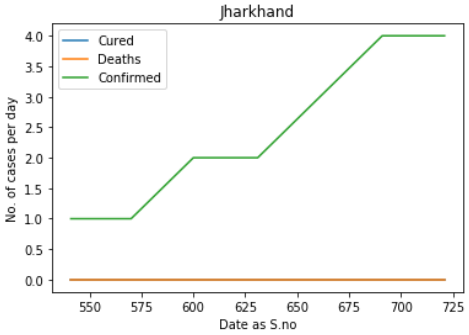

Manipur
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

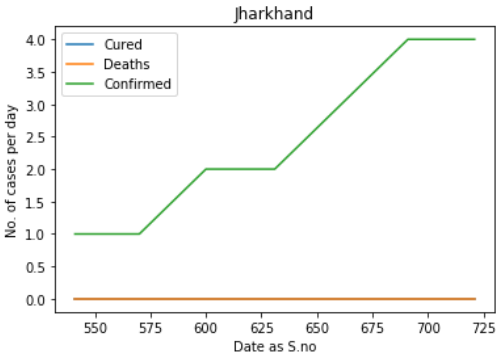

Maharashtra
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

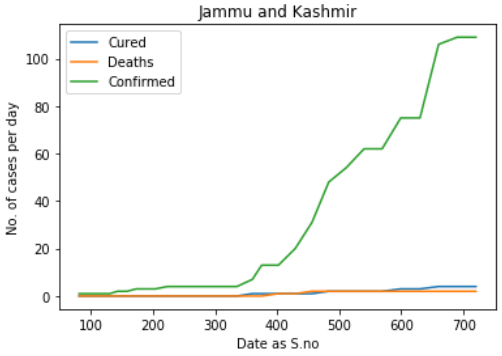

Madhya Pradesh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

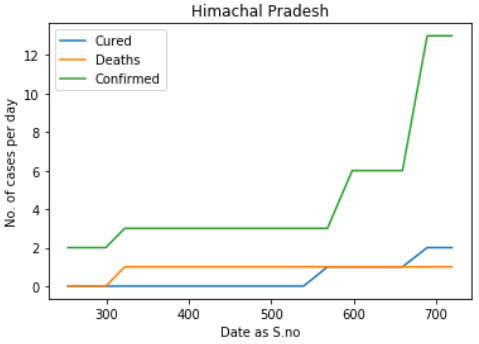

Ladakh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

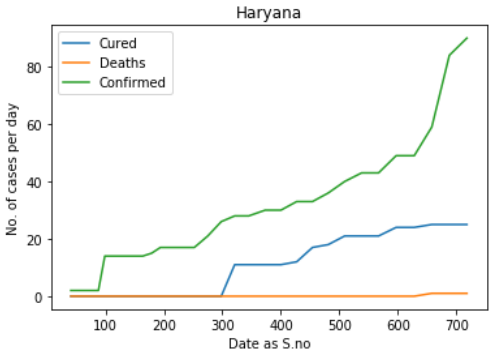

Karnataka
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

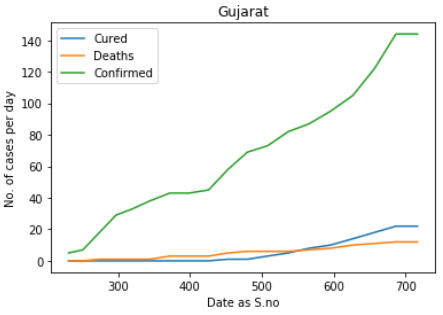

Jharkhand
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

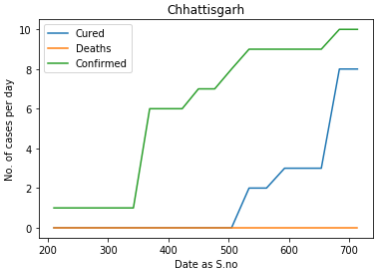

Jharkhand
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

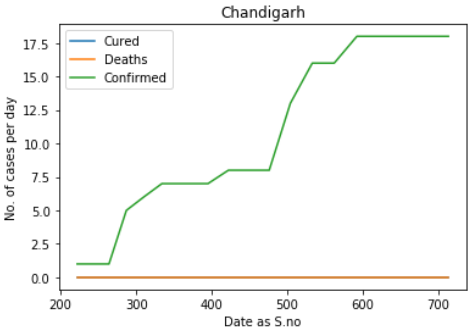

Jammu and Kashmir
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

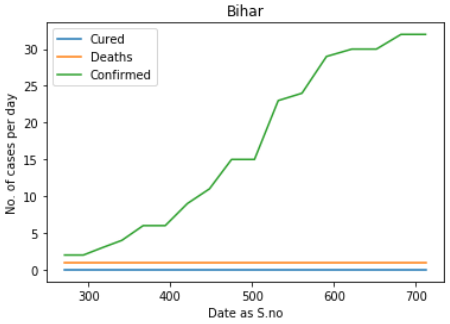

Himachal Pradesh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

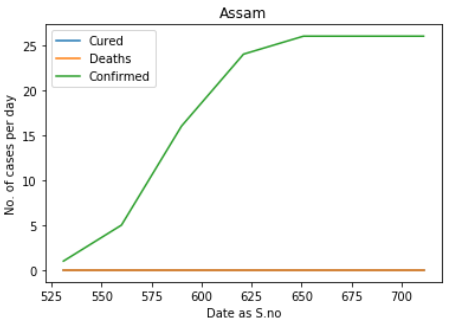

Haryana
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

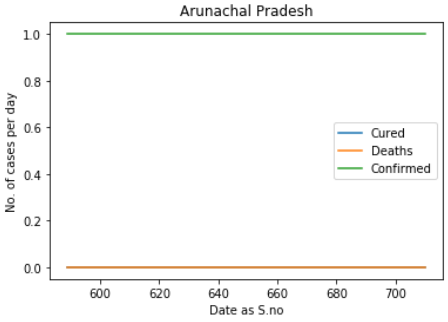

Gujarat
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

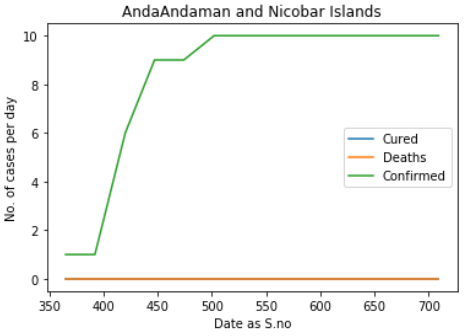

Chhattisgarh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

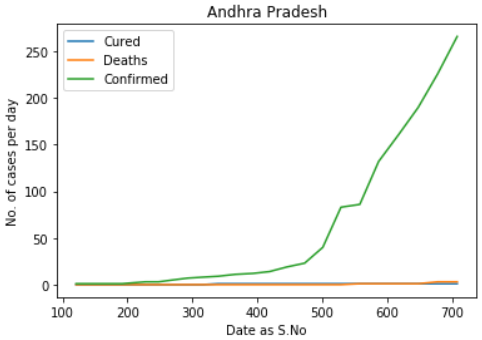

Chandigarh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

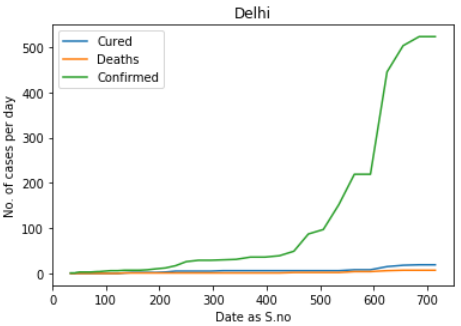

Bihar
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

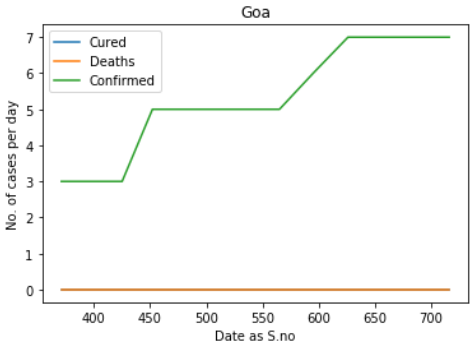

Assam
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no


Arunachal Pradesh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

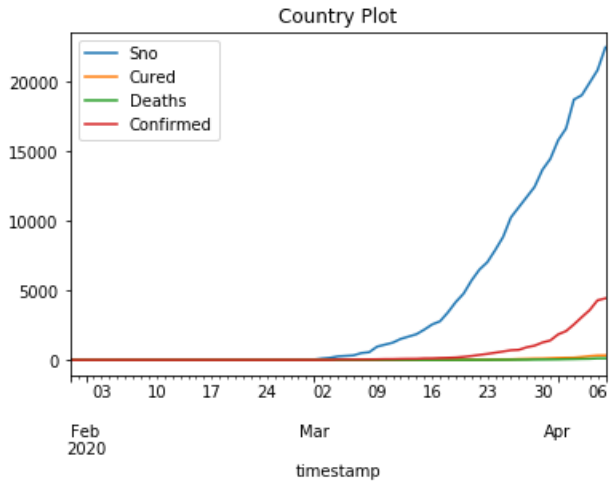

AndaAndaman and Nicobar Islands
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no


Andhra Pradesh
Cured
Deaths
Confirmed
No. of cases per day
Date as S.No

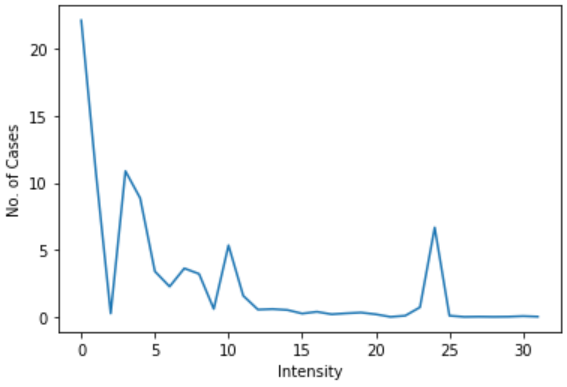

Delhi
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no


Goa
Cured
Deaths
Confirmed
No. of cases per day
Date as S.no

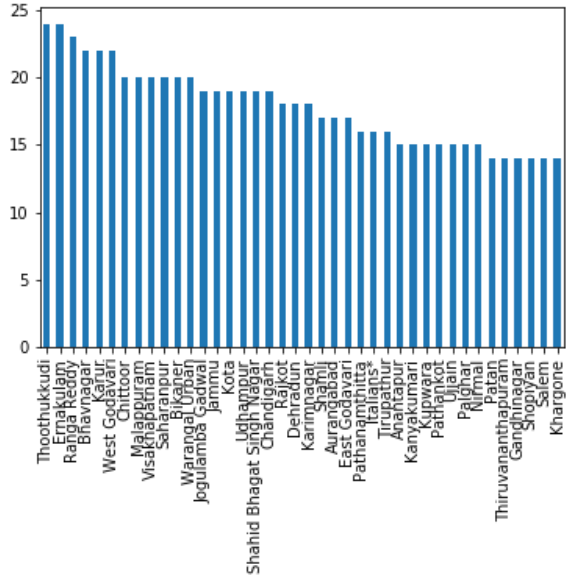
It can be observed that north-eastern states(Tripura,Assam,etc) have extremely less number of cases compared to rest of India.

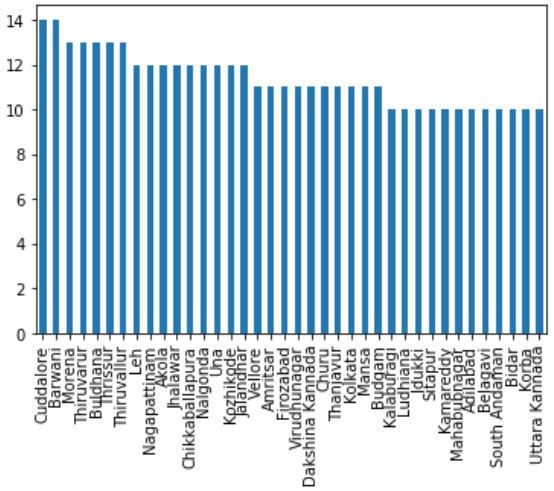


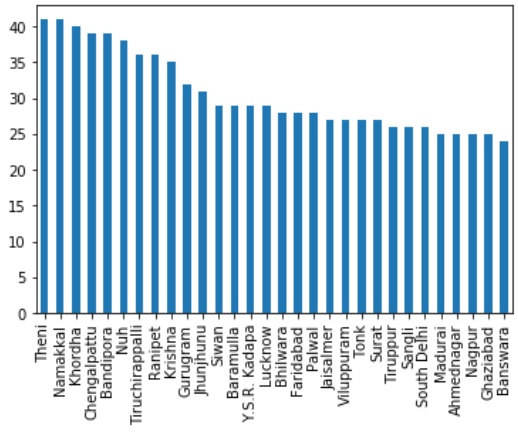

**3)** The following graph quantifies the Intensity of Virus Spread across the nation. It can be observed from the following graph that No. of cases is less where the Intensity is high(No. of cases/Population Density)

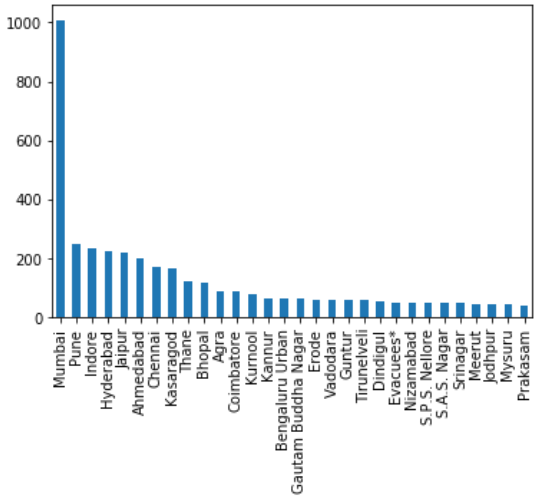


**4)** Active Hotspots across the nation plotted against No. of cases as of 10.04.2020



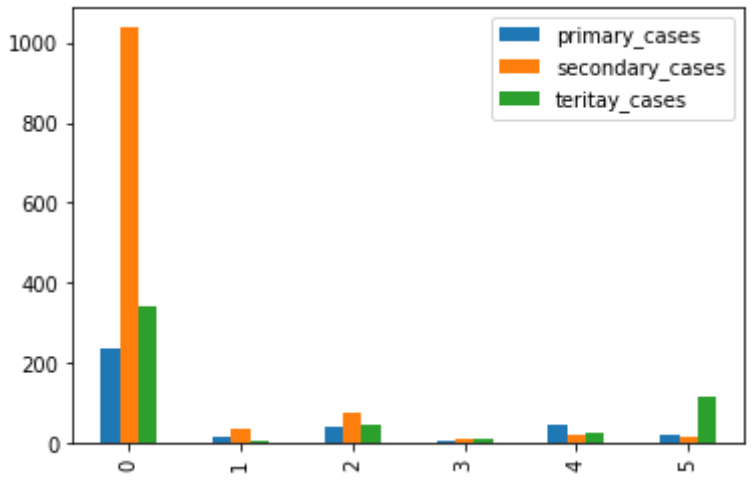

**5)** Maharashtra being the most affected state displayed the maximum increase in the number of hotspots(11) followed by Andhra Pradesh(3) and Tamil Nadu(3) during the 3 weeks period(20.03.2020 to 10.04.2020). It is found by iterating over the active cases dataframe obtained from the original dataframe(IndividualDetails.csv)

**6)** The indices on the x-axis :

- 0-India
- 1-Delhi
- 2-Rajasthan
- 3-Madhya Pradesh
- 4-Maharashtra
- 5-Gujarat

We found the top 5 states with the maximum number of cases,checked and sorted manually the cases into Primary,Secondary,Tertiary categories.

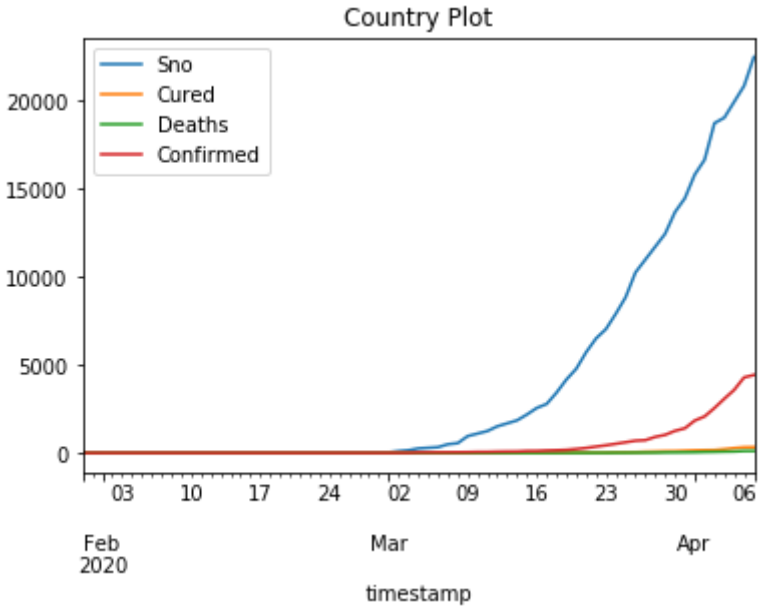
**Assumption :** Considering only the top 100 reasons of CoVid Transmission as they constitute the majority of the cases.




**7)** No. of labs across the nation = 268(100 tests per lab => 26800 tests per day.)

- No. of cases as on 10.04.2020 = 6872. The 10% increase rate per day over the next 7 days gives us 6872(1.1)^10 = 17824 cases which is less than the capacity(26800). So no additional labs are required.

**8)** Though it is too early(10.04.2020)to comment since the curve is still at the foot of the curve,it makes sense to assume that the nation is going to be successful in flattening the curve




**9)** Based on the timeseries data(covid_19_india.csv),The 21 day lockdown is successful to little extent as the rate of increase of cases per day remained almost constant. But it is to be noted that had there been no lockdown,the rate of increase in the number of cases per day would have increased at an alarming rate. So the nationwide lockdown had successfully suppressed the increase rate from growing