CH5019 Mathematical Foundations of Data Science

Team Project

Team Members:

N Sowmya Manojna	BE17B007
VVS Lalitha	CE17B063
Lakshman Kanth Boyina	ME16B021
S Rahul	ME16B036
Kamesh K	MM16B107

Indian Institute of Technology, Madras

Contents

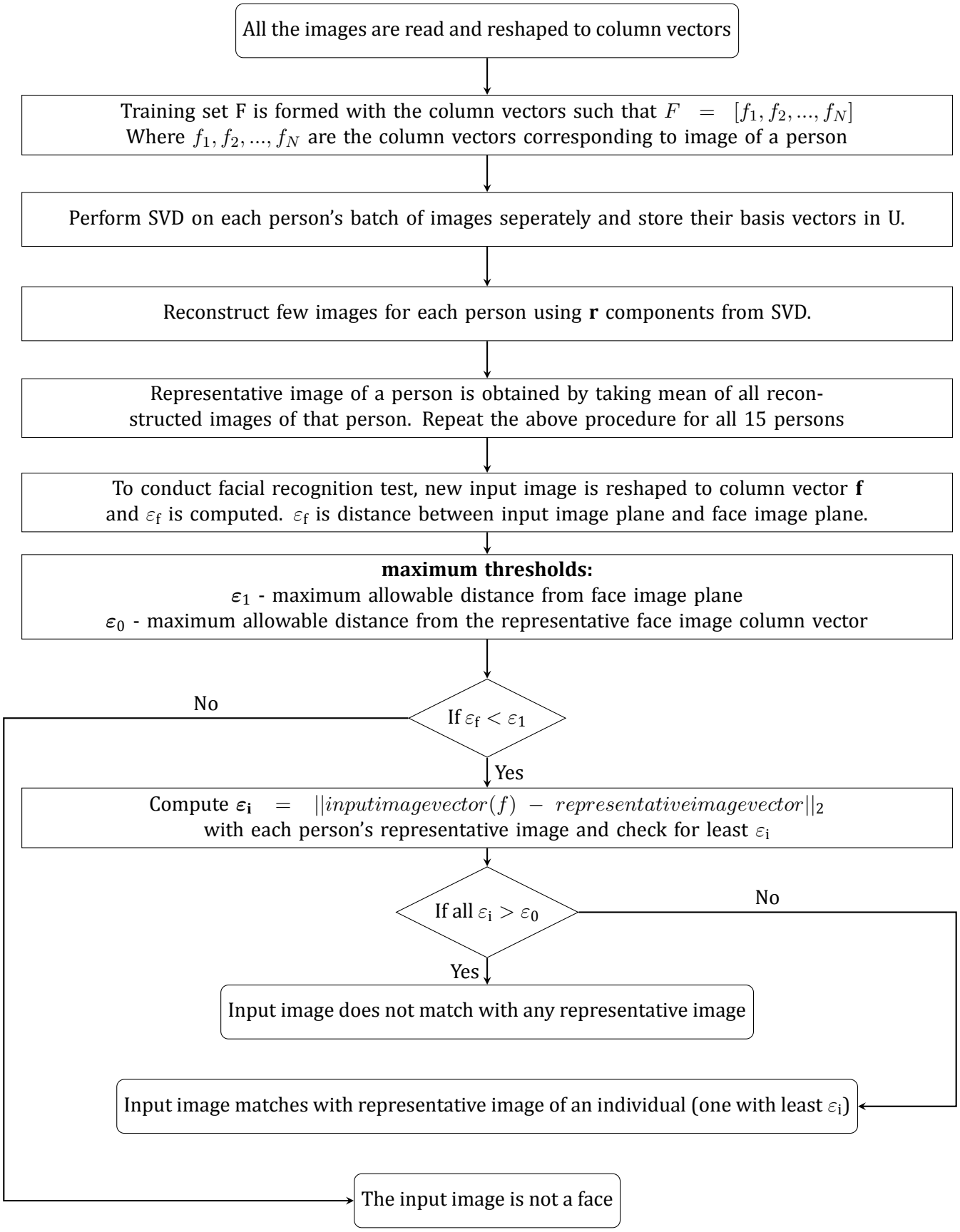
1 Question 1 . . . **2**
- 1.1 Result . . . 3
  - 1.1.1 Representative images of all subjects . . . 3
  - 1.1.2 Face recognition test output . . . 3

2 Question 2 . . . **4**
- 2.1 Result Statistics . . . 5
- 2.2 Data Partitioning . . . 6
- 2.3 Logistic Regression . . . 6
  - 2.3.1 Sigmoid Function . . . 6
  - 2.3.2 Cost function . . . 6
  - 2.3.3 Gradient Descent . . . 7
- 2.4 Performance . . . 7
  - 2.4.1 Confusion matrix . . . 7
  - 2.4.2 F1 Score . . . 7

3 Question 3 . . . **8**
- 3.1 Most affected age group . . . 8
  - 3.1.1 Approach . . . 8
  - 3.1.2 Result . . . 8
- 3.2 Plots of Observed, Recovered and Death cases . . . 8
  - 3.2.1 Approach . . . 8
  - 3.2.2 Results . . . 8
- 3.3 State level intensity measurement . . . 9
  - 3.3.1 Approach . . . 9
  - 3.3.2 Results . . . 9
- 3.4 Active Hotspots identification . . . 10
  - 3.4.1 Approach . . . 10
  - 3.4.2 Results . . . 10
- 3.5 State with maximum change in number of hotspots . . . 10
  - 3.5.1 Approach . . . 10
  - 3.5.2 Results . . . 11
- 3.6 Primary, secondary and tertiary transmissions . . . 11
  - 3.6.1 Approach . . . 11
  - 3.6.2 Result . . . 11
- 3.7 Estimate the number of additional labs required . . . 12
  - 3.7.1 Approach . . . 12
  - 3.7.2 Results . . . 12
- 3.8 The notion of ‘flattening the curve’ . . . 12
  - 3.8.1 Approach . . . 12
  - 3.8.2 Results . . . 13

1 Question 1

The flowchart of the approach used is as follows:



1.1 Result

1.1.1 Representative images of all subjects

- All the images of the person are reconstructed using the components obtained from SVD.
- The representative image is chosen to be the average image of the images reconstructed for the given person using the chosen components.

Figure 1: Representative images

1.1.2 Face recognition test output

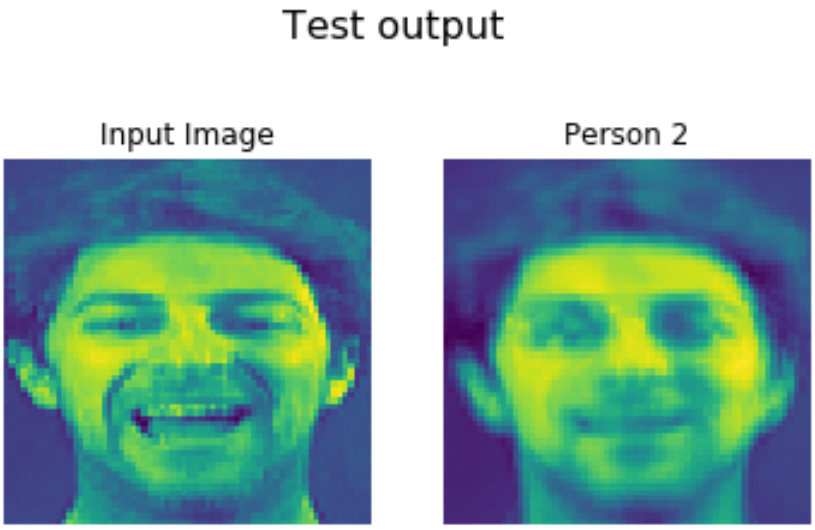
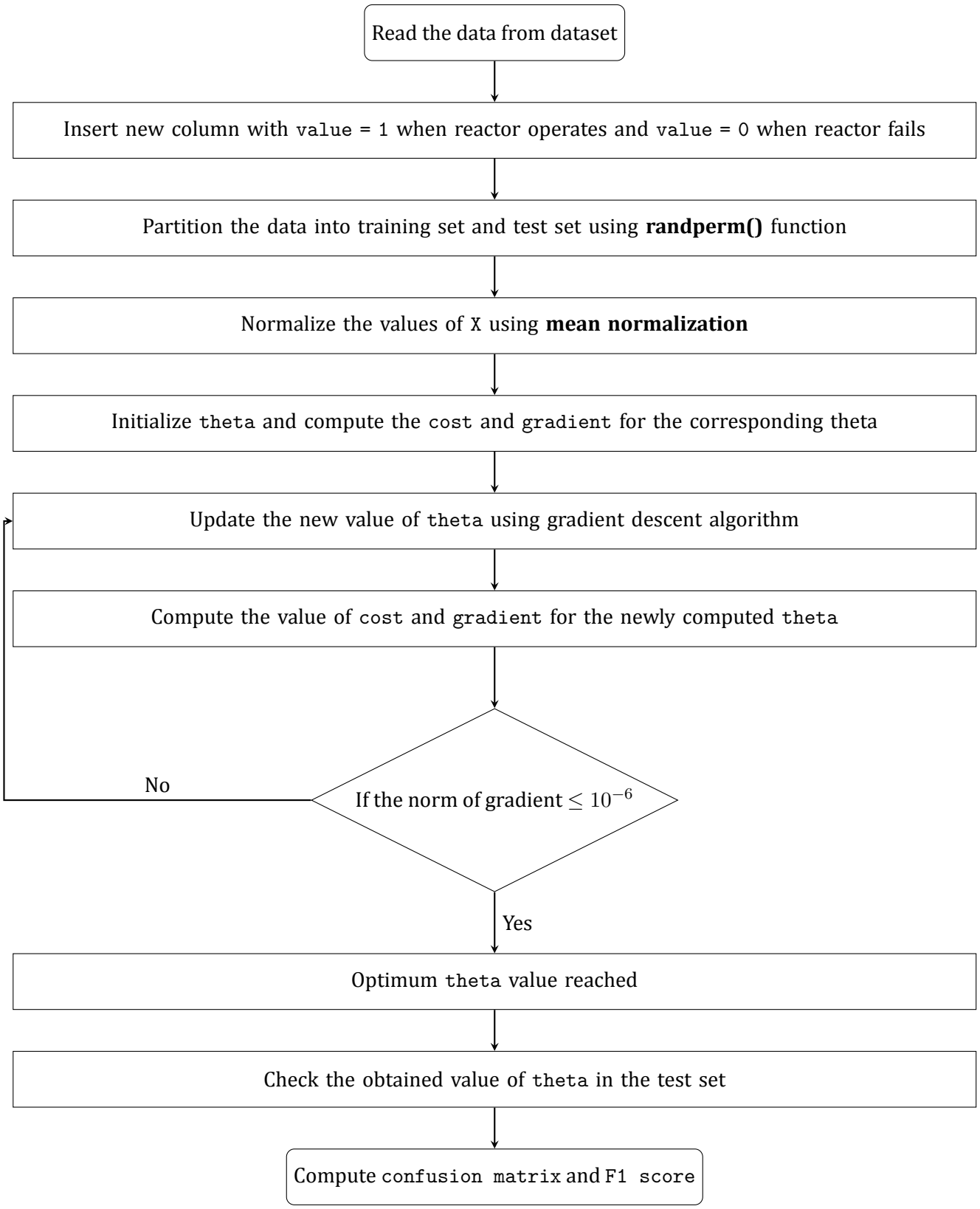


Figure 2: Test Output

- It can be observed given input image matches with person2 of representative images.
- Similarly after testing 150 images, 149 images were recognised and matched with representative images.

2 Question 2

This question was coded using `MATLAB`.
The flowchart of the approach used is as follows:



2.1 Result Statistics

	Temperature	Pressure	Feed Flow rate	Coolant Flow rate	Inlet reactant concentration
count	1000.00000	1000.000000	1000.000000	1000.000000	1000.000000
mean	546.76643	25.493270	125.029060	2295.797770	0.302692
std	86.85878	14.252407	43.508159	763.680625	0.116062
min	400.31000	1.060000	50.030000	1002.530000	0.100300
25%	469.73500	12.725000	88.587500	1635.682500	0.199075
50%	545.80000	25.375000	124.590000	2268.710000	0.308850
75%	618.87750	37.820000	162.562500	2983.692500	0.401625
max	699.87000	49.890000	199.960000	3595.620000	0.499600

Figure 3: Input statistics

	Temperature	Pressure	Feed Flow rate	Coolant Flow rate	Inlet reactant concentration
count	585.000000	585.000000	585.000000	585.000000	585.000000
mean	546.150291	24.906256	121.623419	2785.807744	0.303489
std	86.431031	14.241507	44.018933	528.726067	0.115592
min	400.630000	1.060000	50.030000	1006.650000	0.100300
25%	469.060000	11.510000	84.500000	2365.590000	0.198900
50%	543.220000	24.710000	116.170000	2828.230000	0.313000
75%	619.560000	37.380000	160.230000	3238.550000	0.403000
max	699.870000	49.890000	199.960000	3595.620000	0.499600

Figure 4: pass test statistics

	Temperature	Pressure	Feed Flow rate	Coolant Flow rate	Inlet reactant concentration
count	415.000000	415.000000	415.000000	415.000000	415.000000
mean	547.634964	26.320747	129.829783	1605.060819	0.301568
std	87.555359	14.243827	42.367161	442.049908	0.116852
min	400.310000	1.190000	50.110000	1002.530000	0.104700
25%	470.295000	14.370000	92.925000	1246.415000	0.200750
50%	547.380000	25.800000	134.650000	1557.140000	0.304400
75%	618.400000	39.185000	164.720000	1846.395000	0.400900
max	698.880000	49.790000	199.320000	3587.690000	0.498800

Figure 5: fail test statistics

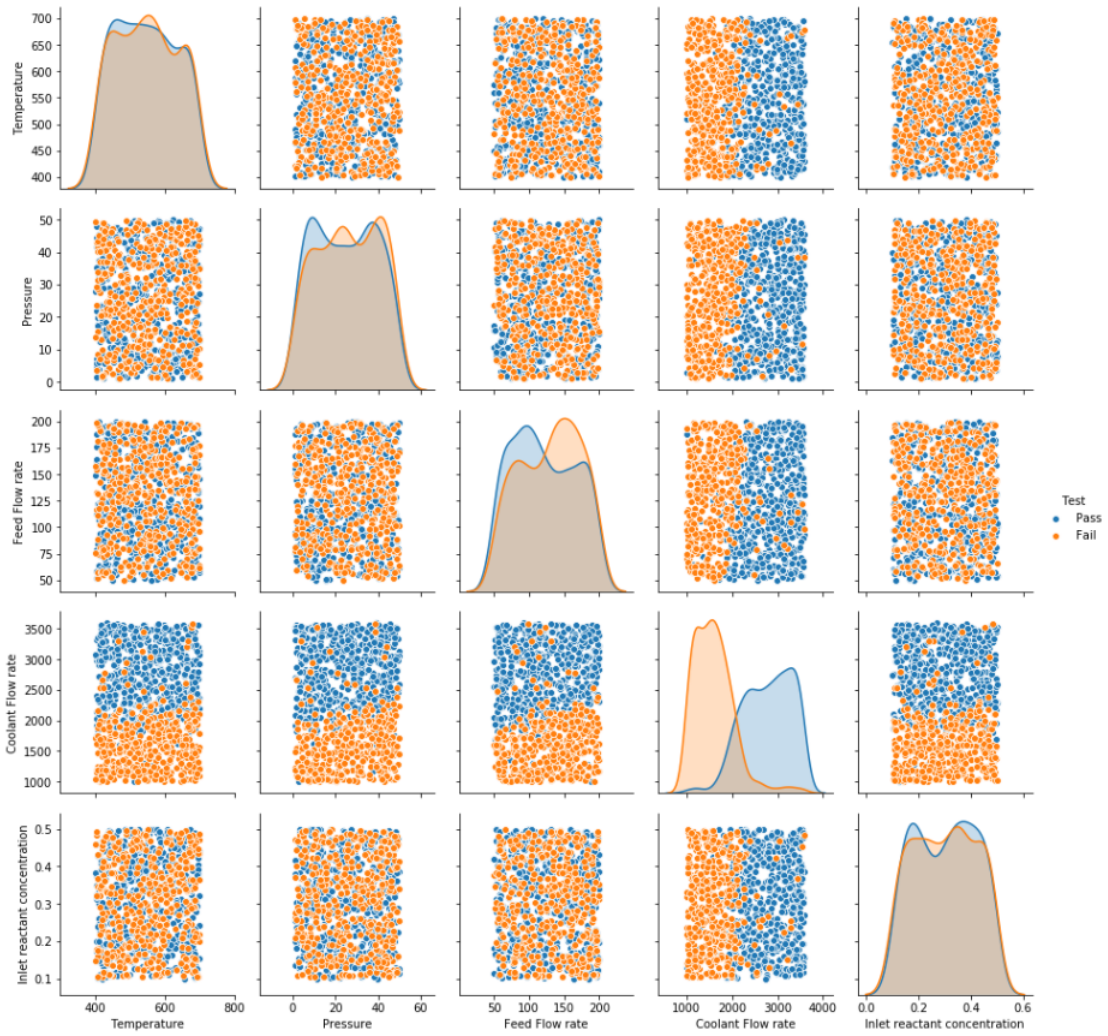


Figure 6: Pair plot

2.2 Data Partitioning

Data was partitioned using the MATLAB inbuilt function `randperm(n)`. This function returns an array containing the random permutation of integers from 1 to n (in our case 1000) without repeating elements. The first 700 elements of this array and the corresponding entries in the data comprise the training set. The remaining 300 entries were made as the testing set.

2.3 Logistic Regression

$$\boxed{h(X) = \text{sigmoid}(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5)}$$

Where, $h(X)$ is the probability that y=1 for the value of X
- x_1 is Temperature
- x_2 is Pressure
- x_3 is Feed flow rate
- x_4 is Coolant flow rate
- x_5 is Inlet reactant concentration

2.3.1 Sigmoid Function

$$\boxed{\text{Sigmoid}(z) = \frac{1}{1 + e^{-z}}}$$

From the fig. we can see that if $z > 0$ output will be greater than 0.5 otherwise, output is less than or equal to 0.5.

So we predict as follows: $y = \begin{cases} 1 & h(X) \leq 0.5 \\ 0 & otherwise \end{cases}$

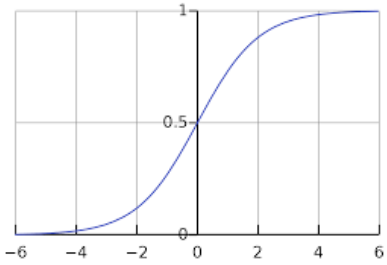


Figure 7: The sigmoid function

2.3.2 Cost function

Since it is a classification problem the cost depends both on the values of $h(X)$ and y. So, we take cost as:

$$\boxed{cost(h(X), y) = \begin{cases} -log(h(X)) & y = 1 \\ -log(1 - h(X)) & y = 0 \end{cases}}$$

By assigning costs like this, we penalize the algorithm if it predicts wrongly (by increasing the cost by a large amount).
In the generalized form, we take the cost function to be:

$$\boxed{J(\theta) = \frac{-1}{m}\left(\sum_{i=1}^{m}(y_i * log(h(X_i)) + (1 - y_i * log(1 - h(X_i)))\right)}$$

Where, J is the cost and m is total number of training examples taken.

As y can only take values 0 and 1, the cost in the generalized formulation reduces as below:
When $y = 1$ only the first term is active which implies that $cost = -log(h(X))$
When $y = 0$ only the second term is active which implies that $cost = -log(1 - h(X))$

So we can infer that both formulations are same.
Since each parameter had different ranges, **mean normalization** for each parameter was done before

starting Gradient descent. The formula used for the same is as follows:

$$\boxed{X_{norm}(i,j) = \frac{X(i,j) - mean(j)}{Range(j)}}$$

2.3.3 Gradient Descent

Our objective is to minimize the cost function value (J).
So we iterate for the values of theta in the following manner:

$$\theta_{j,new} = \theta_{j,old} - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

Since

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\Big(\sum_{i=1}^{m}(h(X_i) - y_i)X_i(j)\Big)$$

we get,

$$\boxed{\theta_{j,new} = \theta_{j,old} - \alpha \frac{1}{m}\Big(\sum_{i=1}^{m}(h(X_i) - y_i)X_i(j)\Big)}$$

The learning rate α was taken to be 0.01.
Convergence condition: Norm of the gradients of $\theta \leq 10^{-5}$

We iterate for the values of theta by using the above condition until the convergence condition is reached.

2.4 Performance

Obtained values of theta are: 1.2369, -0.6470, -1.6551, -2.1753, 11.9063, -0.4493
Obtained Cost = 0.2720

Number of cases predicted correctly = 286
Number of cases predicted wrongly = 14
Accuracy = 286/300 = 0.9533 = 95.33%

Number of True positive obtained (TP) = 163 ($y_{obtained} = 1$ and $y_{test} = 1$)
Number of False positive obtained (FP) = 9 ($y_{obtained} = 1$ and $y_{test} = 0$)
Number of False negative obtained (FN) = 5 ($y_{obtained} = 0$ and $y_{test} = 1$)
Number of True negative obtained (TN) = 123 ($y_{obtained} = 0$ and $y_{test} = 0$)

2.4.1 Confusion matrix

		Actual Values	
		Positive ($y_{test} = 1$)	Negative ($y_{test} = 0$)
Predicted Values	Positive ($y_{pred} = 1$)	163 *(TP)*	9 *(FP)*
	Negative ($y_{pred} = 0$)	5 *(FN)*	123 *(TN)*

2.4.2 F1 Score

$$\text{Precision(P)} = \frac{TP}{TP + FP} = \frac{163}{172} = 0.9477$$

$$\text{Recall(R)} = \frac{TP}{TP + FN} = \frac{163}{168} = 0.9702$$

$$\text{F1 score} = \frac{2RP}{R + P} = 0.9588$$

3 Question 3

This question was coded using `Python`. All datasets except `StatewiseTestingDetails.csv` and `IndividualDetails` are as provided on Moodle. The aforementioned datasets have been sourced from Kaggle.

3.1 Most affected age group

3.1.1 Approach

- The index corresponding to the maximum 'Total Cases' in `AgeGroupDetails.csv` was found.
- The age group corresponding to this index was reported.

3.1.2 Result

Age Group with maximum cases is 20-29 with 172 cases, constituting 24.86% of total cases.

3.2 Plots of Observed, Recovered and Death cases

3.2.1 Approach

- Data from `covid_19_india.csv` was segregated based on 'Cured', 'Deaths' and 'Confirmed'.
- The number of cases were summed for each day
- The 'Cured', 'Deaths' and 'Confirmed' data was then plotted across dates.

3.2.2 Results

Detailed state-wise distribution and contribution plots are available with the code.

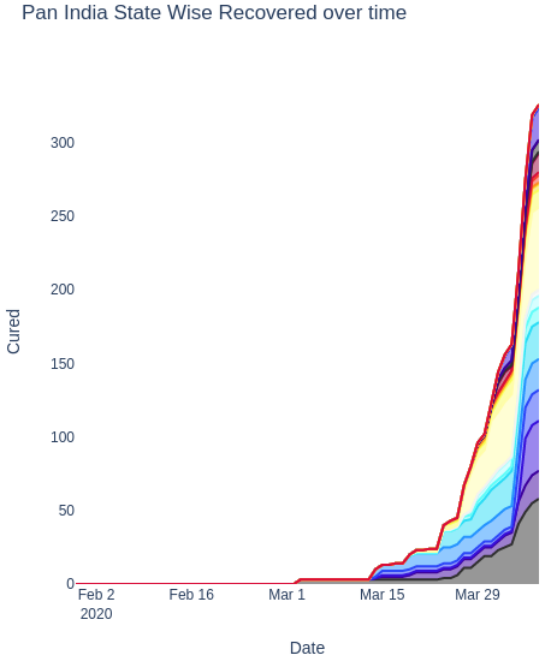


Figure 8: India-wide recovered cases vs Time

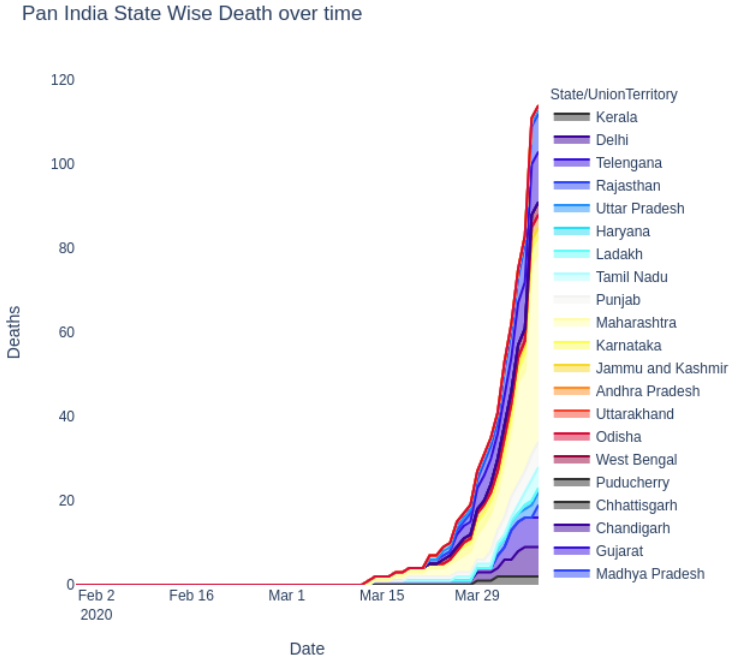


Figure 9: India-wide death cases vs Time

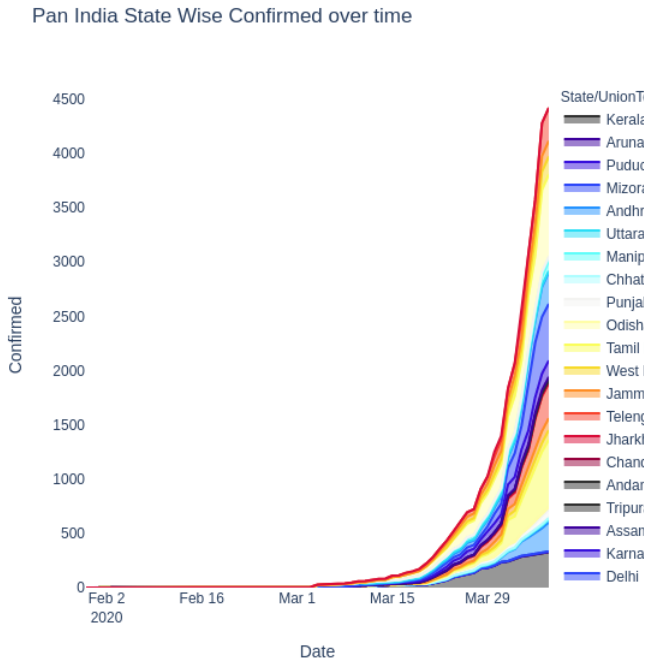


Figure 10: India-wide observed cases vs Time

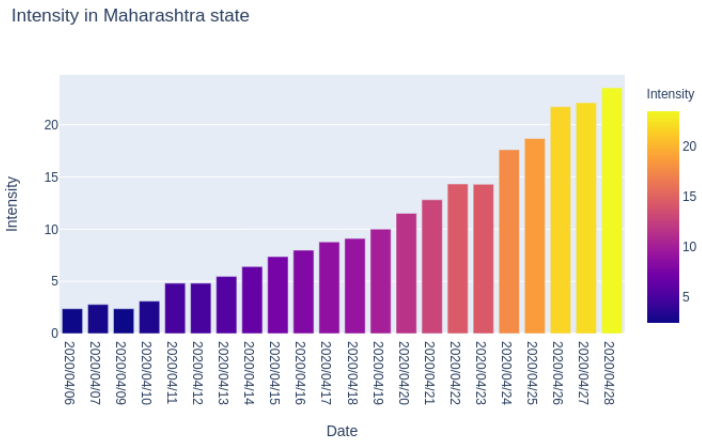


Figure 11: Intensity distribution of Maharastra - maximum intensity (greater that 20)

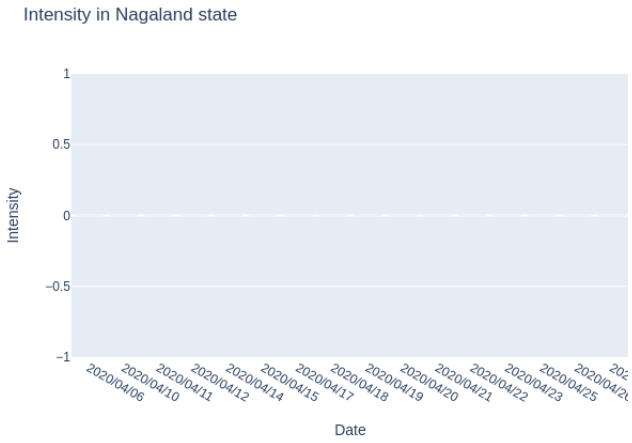


Figure 12: Intensity distribution of Nagaland - zero intensity

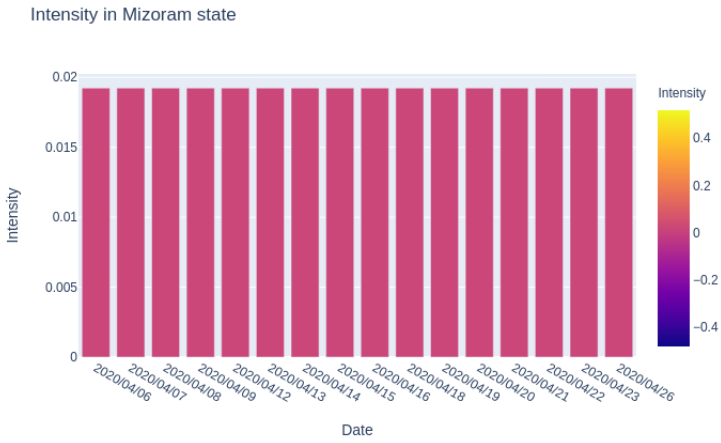


Figure 13: Intensity distribution of Mizoram - constant intensity

3.3 State level intensity measurement

3.3.1 Approach

- The `population density` of each state was obtained from `population_india_census2011.csv` and the positive cases from `StatewiseTestingDetails.csv` were grouped and summed based on states.
- The intensity was calculate and plotted across dates for all states.

3.3.2 Results

Detailed intensity distribution plots for all 32 states and union territories is available with the code.

As on 28^{th} April, Maharashtra is found to have the highest intensity (greater that 20) and as of 27^{th} April, Nagaland was found to have the least intensity (equal to zero).

3.4 Active Hotspots identification

3.4.1 Approach

- Week based approach of summing cases across a week and a day based cases count approach was used for hotspot identification.
- District wise count of patients was taken from `IndividualDetails.csv` between $3^{rd} - 10^{th}$ April (for the week based approach) and on 10^{th} April (for the day based approach).

3.4.2 Results

The hotspots as of April 10 (based on week 3/4/20 to 10/4/20) are:
Adilabad, Agra, Ahmedabad, Akola, Anantapur, Aurangabad, Banswara, Baramulla, Barwani, Bengaluru Urban, Bhavnagar, Bhopal, Bikaner, Chengalpattu, Chennai, Chittoor, Coimbatore, Cuddalore, Dindigul, Erode, Evacuees*, Faridabad, Gautam Buddha Nagar, Ghaziabad, Guntur, Gurugram, Hyderabad, Indore, Italians*, Jaipur, Jaisalmer, Jammu, Jhalawar, Jhunjhunu, Jodhpur, Jogulamba Gadwal, Kamareddy, Kannur, Kanyakumari, Karimnagar, Kasaragod, Khargone, Khordha, Kota, Krishna, Kupwara, Kurnool, Lucknow, Madurai, Mahabubnagar, Mansa, Meerut, Mumbai, Mysuru, Nalgonda, Namakkal, Nirmal, Nizamabad, Nuh, Palghar, Palwal, Patan, Pathanamthitta, Pathankot, Prakasam, Pune, Ranipet, S.A.S. Nagar, S.P.S. Nellore, Shamli, Shopiyan, Sitapur, Siwan, Srinagar, Surat, Thane, Thanjavur, Theni, Thiruvallur, Thoothukkudi, Tiruchirappalli, Tirunelveli, Tiruppur, Tonk, Udhampur, Vadodara, Vellore, Viluppuram, Y.S.R. Kadapa

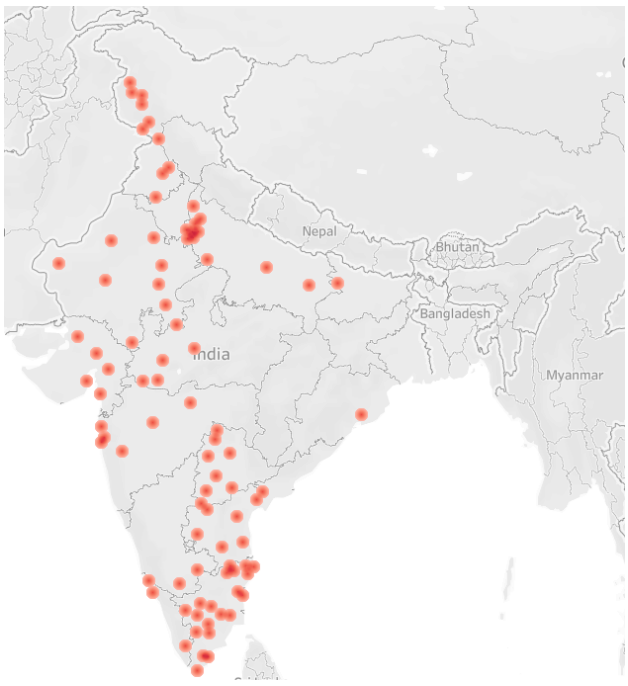


Figure 14: Hotspot visualisation based on week count

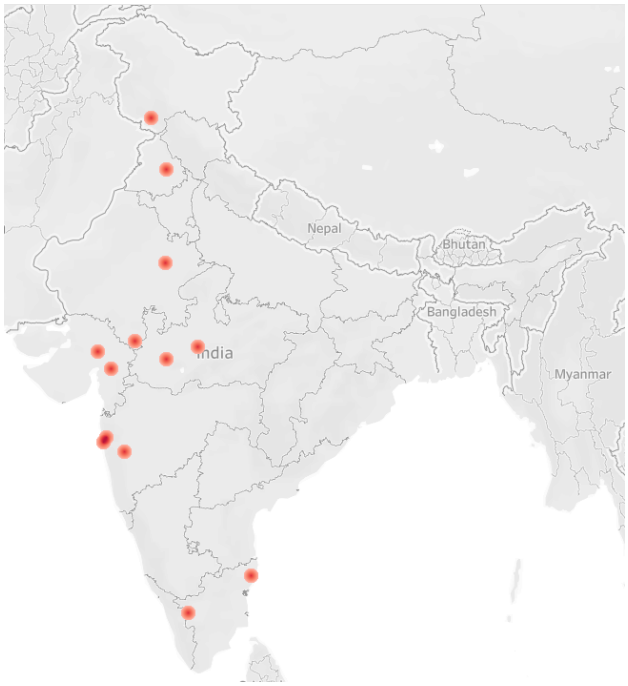


Figure 15: Hotspot visualisation based on day count

The hotspots as of April 10 (based on the day 10/4/20) are:
Ahmedabad, Banswara, Bhopal, Chengalpattu, Coimbatore, Indore, Jaipur, Mumbai, Pune, S.A.S. Nagar, Thane, Udhampur, Vadodara

3.5 State with maximum change in number of hotspots

3.5.1 Approach

- District wise count of number of positive patients was taken from `IndividualDetails.csv`. All districts with greater than 10 cases on each week were designated as hotspots.
- Week IDs were assigned for all positive cases that were reported between 20^{th} March and 10^{th} April. This data was then grouped based on States and hotspots were counted for each State.

3.5.2 Results

The state with maximum increase in hotspots is Tamil Nadu and the state with maximum decrease in hotspots is Kerala.

3.6 Primary, secondary and tertiary transmissions

3.6.1 Approach

- Data from `IndividualDetails.csv` with entries before 10/04/20 was used for this question.
- Primary cases were identified with the word 'Travelled' (case sensitive) which weren't followed by to/from 'Delhi', 'Bangalore', 'Rajasthan', 'Kolkata', 'Mumbai', 'WB', 'Phuket' (there cases were due to travel within India).
- Secondary cases were identified with words such as 'Contact', Patient IDs (queried using `Regex`), 'Doctors', 'Hospitals', 'Family', 'NRI', 'Patient', 'Positive'. Already identified primary cases were excluded from this list.
- Tertiary cases were identified by removing all primary and secondary cases.
- The primary, secondary and tertiary data thus obtained were grouped by State name and the cases were summed.

3.6.2 Result

Detailed primary, secondary and tertiary plots are available with the code.
The results of top 5 states are as follows:

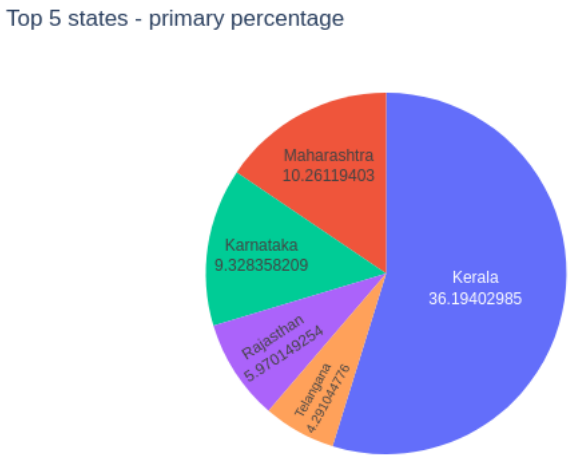


Figure 16: Primary case distribution across states

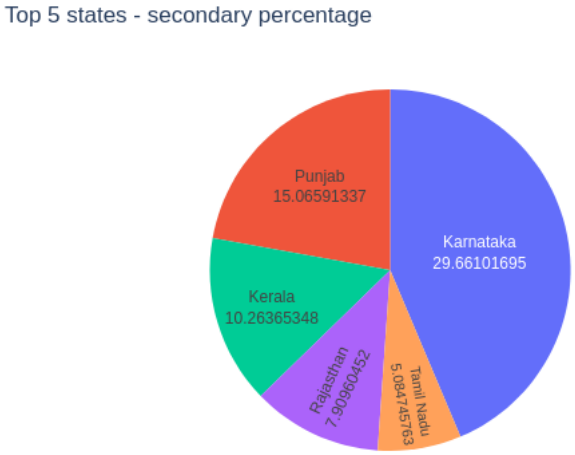


Figure 17: Secondary case distribution across states

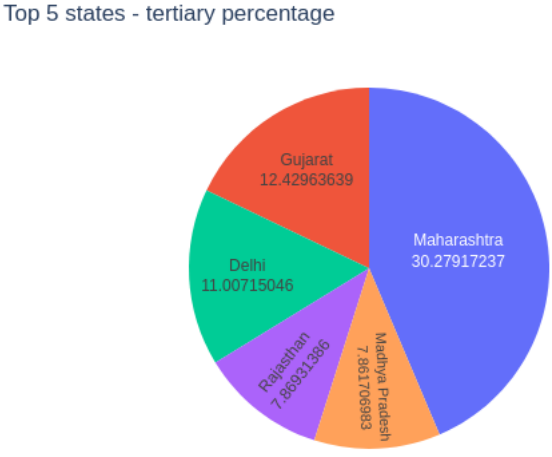


Figure 18: Tertiary case distribution across states

3.7 Estimate the number of additional labs required

3.7.1 Approach

- Two approaches were used to answer this question. They are - 10% increase in the positive cases per day and 10% increase in the total number of cases per day.
- Data from `ICMRTestingDetails.csv` has been used for both the approaches.
- **Approach 1:** The total positive cases was plotted against the total tested cases and the possible relation between the two were analysed.
- Barring the first four days, the relation between the two were found to be linear. The parameters for the linear fit were found using the `numpy polyfit` function.
- The future positive cases were built on the last positive case count data available and the count was compounded by 10% as in the question.
- Using the linear fit parameters returned, the future total cases were also estimated. The required number of testing centers were found using the future total cases.
- **Approach 2:** The future total cases were built on the last total case count data available and the count was compounded by 10% as in the question
- The required number of testing centers were found using the future total cases

3.7.2 Results

Assuming that total positive cases compound by 10%
Number of additional testing centers required (as of 20th April): 286

Assuming that total tested cases compound by 10%
Number of additional testing centers required (as of 20th April): 466

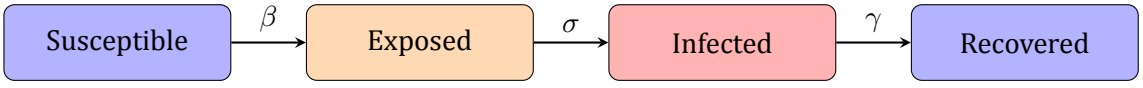
3.8 The notion of 'flattening the curve'

The mathematical interpretation of flattening the curve would be an increase in the mean and standard deviation of the total cases vs time distribution. Physiological interpretation of the same would be a decline in the number of new hosts of the virus.

3.8.1 Approach

An SEIR (Susceptible - Exposed - Infected - Recovered) model has been used to expain the notion of flattening the curve.

The flowchart of the model is as follows:



The medical capacity of India was found by calculating the total number of hospital beds (from `HospitalBedsIndia.csv`) and this value was set as India's medical limit/upper bound.

The susceptible population was calculated by taking in consideration the fraction of the total working population in India who showed movement during this lockdown period [1], [2].

[1] https://data.gov.in/resources/working-population-according-2011-census-states
[2] https://zeenews.india.com/economy/delhi-reports-slowest-return-to-workplace-during-lockdown-3-0-says-google-mobility-report-2283439.html

Here,

- β represents the rate constant for transition from susceptible to exposed per infected person (or) the number of people each infected person can affect.
- σ represents the rate at which the exposed person becomes infected. It has been known from literature [3] [4] that the incubation period can vary from 1 to 12.5 days, with a median of 5 or 6 days. The value 5 has been taken as the incubation period for this model. Hence, $\sigma = 1/5$
- γ represents the rate with which an infected individual recovers. Using the data available from `IndividualDetails.csv`, and taking the mean of the duration for recovery, the duration was found to be equal to 16.1 days. Hence, $\gamma = 1/16.1$

Simulations were carried out by varying the β value (as a representation of possible interactions among people). The value of β was varied from 10 (indicating free movement/absence of restrictions) to 0.1 (indicating stringent restrictions).

The model however assumes, a homogeneous population, uniform probability of infection, all interaction in the population is equally likely and no individual dies, while in practise, these could be skewed by other spatial movements, past exposure to other viruses and the patient's current health conditions.

3.8.2 Results

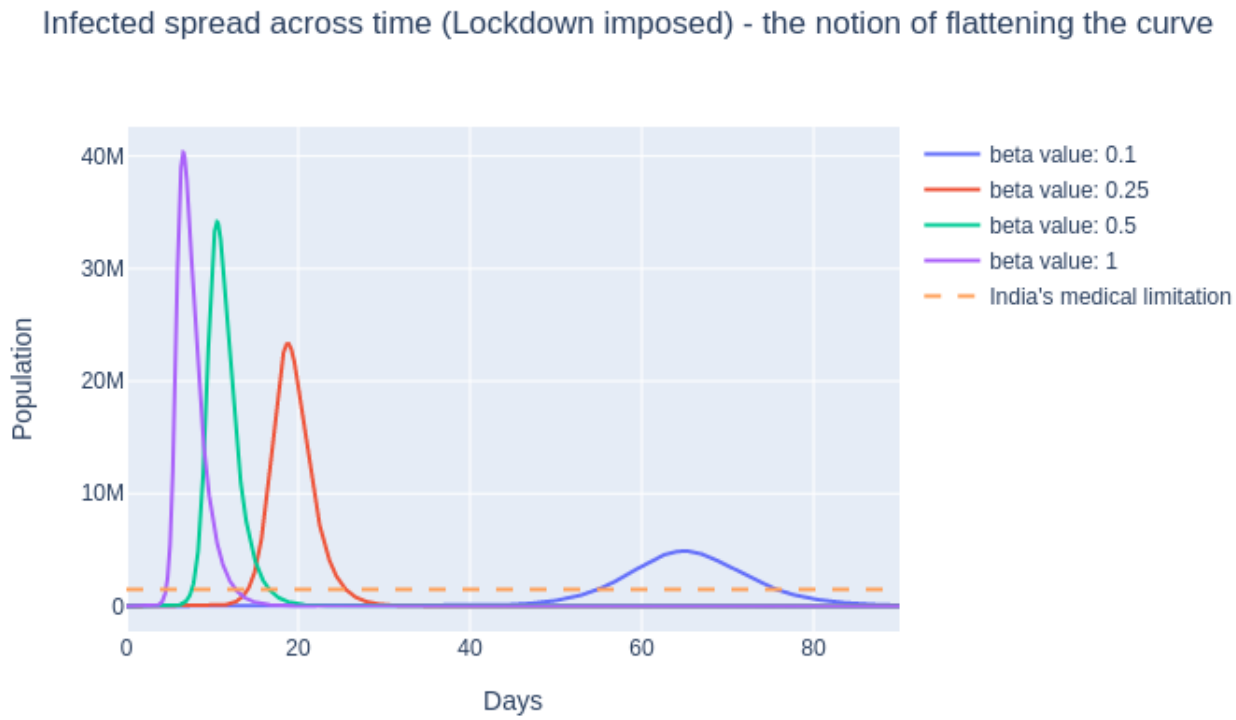


Figure 19: The notion of flattening the curve

As seen below, flattening the curve can be achieved by decreasing the possible interaction among people (by imposing stringent lockdown measures). This would also help reduce the number of affected patients at a given time and hence, distribute or decrease the load on the country's health-care system temporally, as fewer people would require medical attention at any given time.

[3] `https://www.mohfw.gov.in/pdf/DGSOrder04of2020.pdf`

[4] `https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200402-sitrep-73-covid-19.pdf`