Distribution & Requirement of Medical Resources for Covid-19 and Factors Affecting Hospitalization.

Group Members: Gopal Ramesh Dahale(11840520) , Asad Abidi(11840220), Abinash Acharya(11840050), Ashutosh Soni(11840270), Himanshu Sekhar Nayak(11840560)..

About the Data Source:

Uncover COVID 19, Medical Conditions, Covid net hospitalization rates, Covid 19 in the USA, CoVCSD, Covid 19 and its clinical spectrum

Progress:

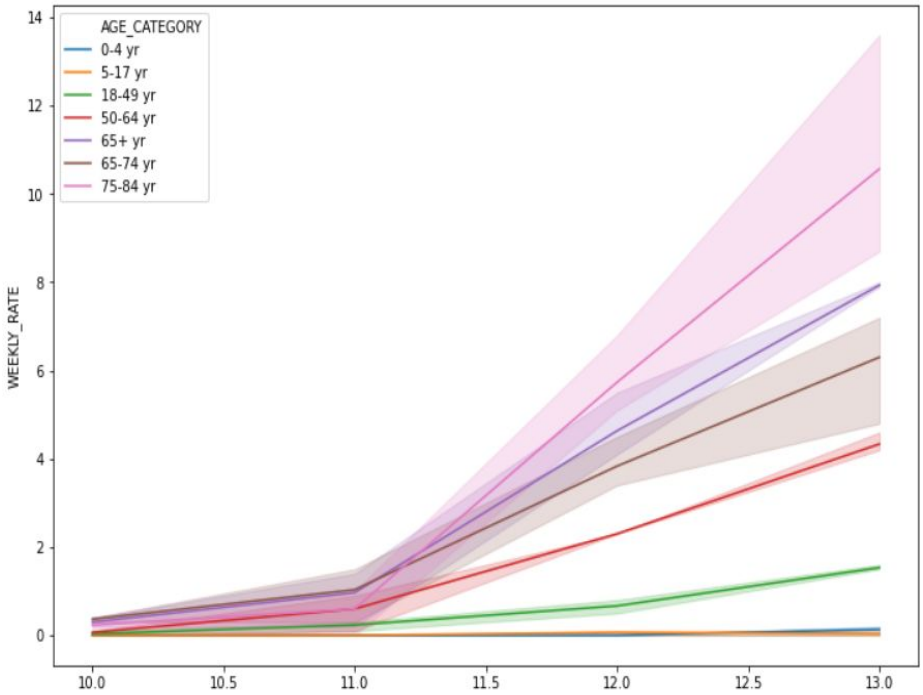
There are 225 CSV files in the UNCOVER dataset as of now. So the first task was to choose the datasets that would be handy in achieving the aim of the project. We selected the relevant datasets(CSV Files) from the Uncover Dataset to analyze as well as visualize hospitalization data and to assess the number of ventilators required as the pandemic progresses.
To choose the relevant datasets, we did the following operations :

- Since a major section of our analysis concerns the availability/prediction of hospital resources, we searched for keywords like ventilators, ICU Beds, etc in the headers/columns of the CSV files and we assigned each of them a score on this basis.
- Then, we sorted them based on this score and chose the files with the highest scores.
- Then, we manually went through the files to select those files that we needed to analyze and also observed how many non-null rows the files had.
- One problem was that a lot of files had non-English headers. So, we had to work with the translations of various other languages like Spanish, Italian, French, etc. (since most files had data of North American/European nations) and included the keywords in these languages.

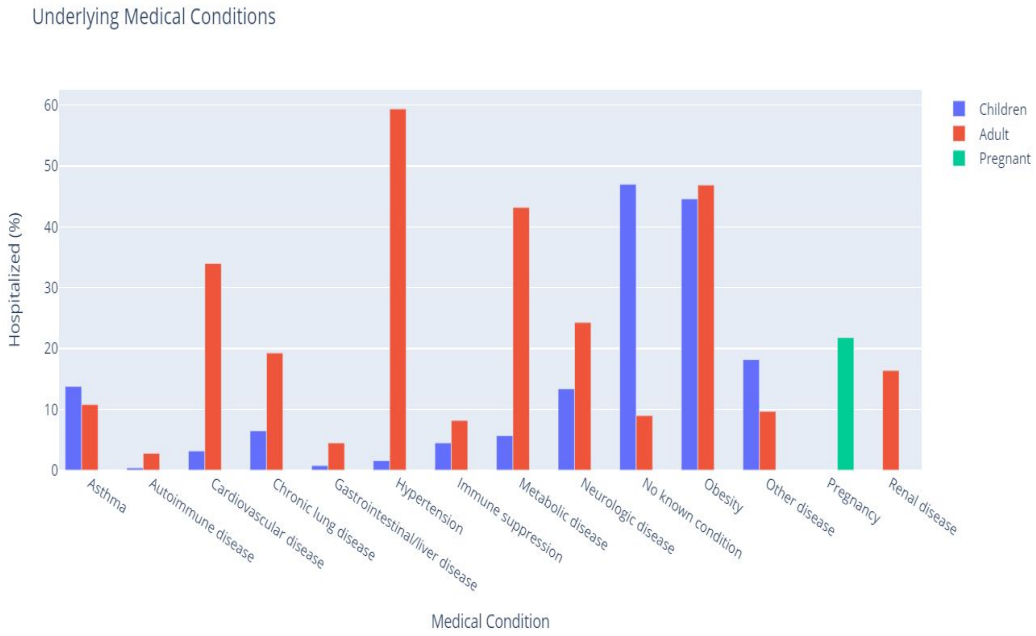
Who actually got hospitalized in covid 19?

Most of the data is related to the USA and Spain

- Percentage of hospitalization in the USA: Which states have a high hospitalization rate?
- Hospitalization and ICU rate analysis in Spain: Analyzed the relation between ICU rate and Hospitalization rate.
- What age-group got hospitalized: Analyzed the distribution of Weekly hospitalization rates of various age-groups.



- Medical conditions that led to hospitalization: What other medical conditions are a major cause of Hospitalization?



Analysis of the important features/medical tests which can indicate the possibility of the patient needing hospitalization, or hospital bed in other words.

The analysis was based on a dataset named:
diagnosis-of-covid-19-and-its-clinical-spectrum.csv.
The dataset contains results of various clinical tests conducted over the patients (which can be both Covid-19 positive or negative), some of which are rods, monocytes

,aspartate_transaminase, po2_venous_blood_gas_analysis,
base_excess_arterial_blood_gas_analysis

The mean and mean variance of the different tests, which are mentioned in the columns, is calculated for both types of patients, i.e. hospitalized and not hospitalized. A score is assigned for each row which is based on the three parameters i.e. difference between means of both types of patients, mean variance of 1st type of patient, mean variance of 2nd type of patient. The score function is to be maximised for getting the features that have high mean difference between patients with low variance for each patient. Top 10 scored rows were selected. The reason for selecting these features as an indicator of the need to hospitalize is: these are the features that help predict the requirement of a bed for a patient since they have scored higher. The analysis was done for both normal hospital beds and intensive care units(ICUs). The figure below illustrates the observation.

a_rward_mean: admitted to hospital

na_rward_mean: not admitted to hospital

a_icu_mean: admitted to ICU

na_icu_mean: not admitted to ICU

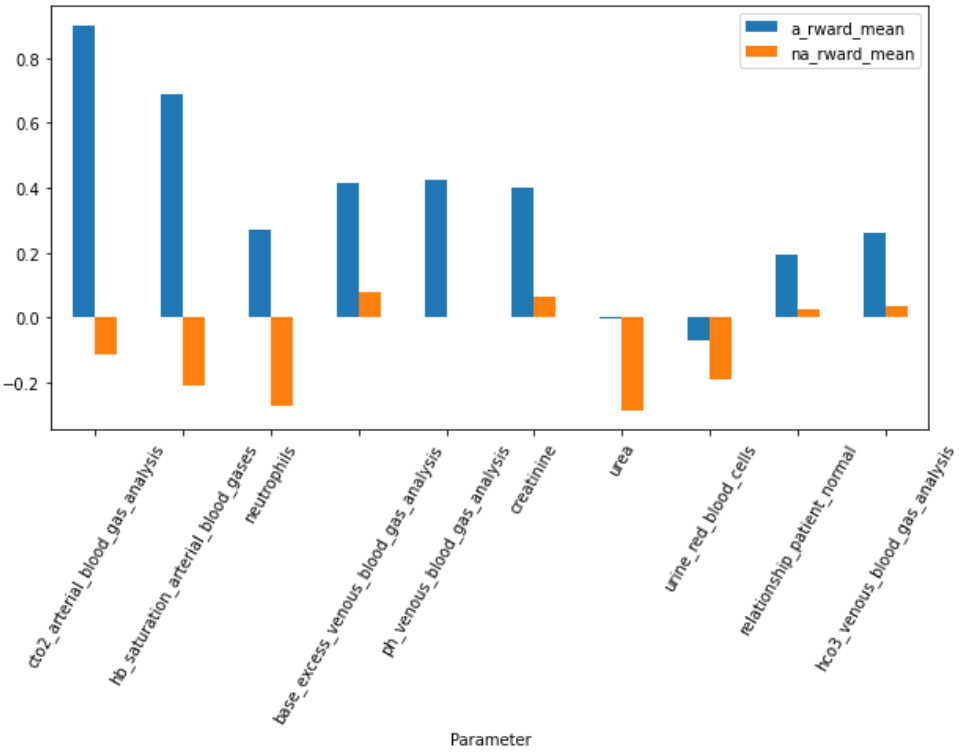


Fig: Features important for predicting normal hospitalization

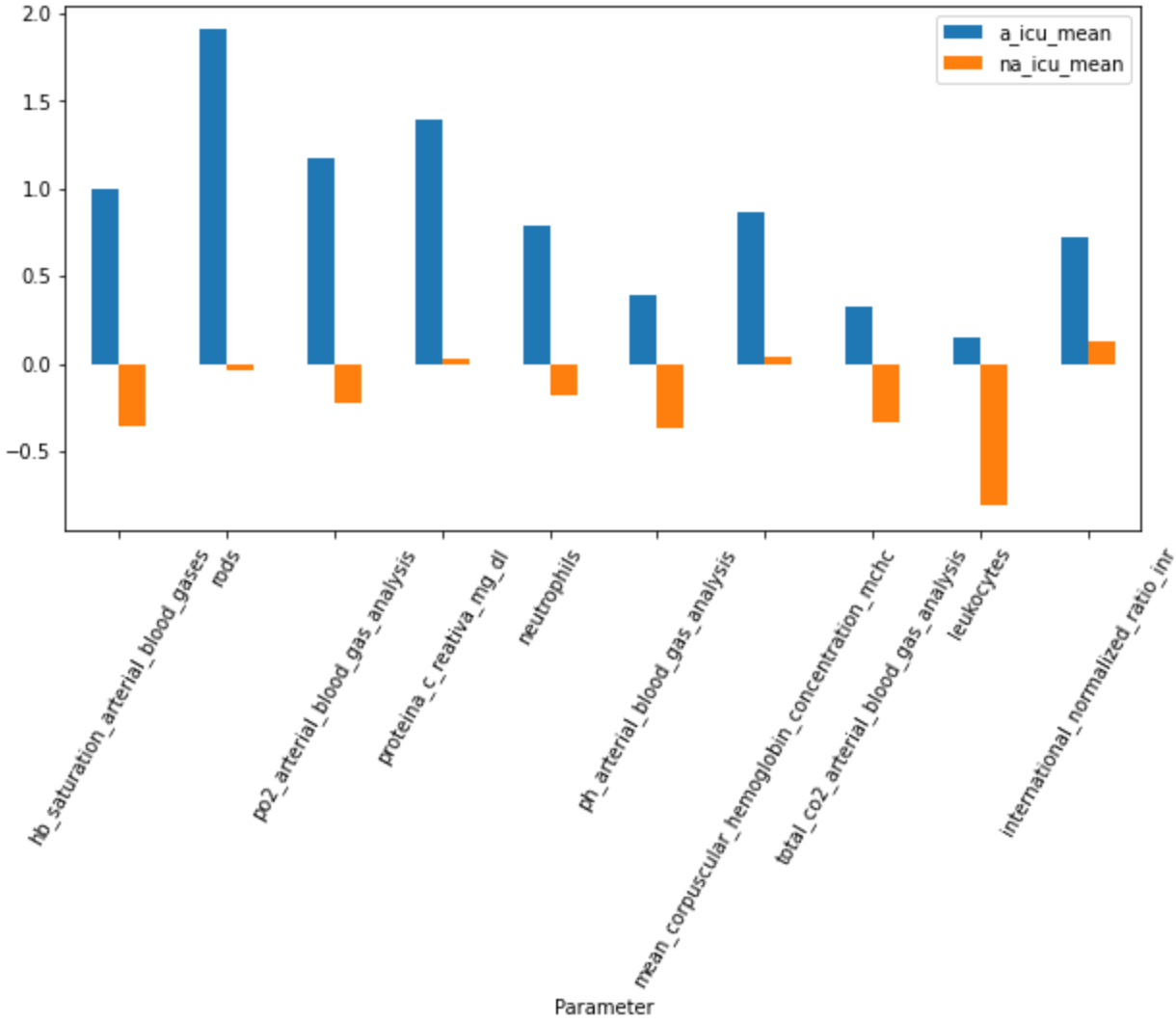


Fig: Features important for predicting ICU bed requirement

Predicting whether a person will require ICU or not using the laboratory tests data of confirmed COVID positive patients.

The dataset was taken from here. Cleaning of the data involved making age percentiles and window's to number. A brief EDA is performed over the dataset to gather the percentage of people who require ICU, whether a person with age above 65 needs ICU or not, and whether the data follows a trend between age percentiles and ICU, etc. We now briefly describe the window concept.

Window	Description
0-2	From 0 to 2 hours of the admission
2-4	From 2 to 4 hours of the admission
4-6	From 4 to 6 hours of the admission
6-12	From 6 to 12 hours of the admission
Above-12	Above 12 hours from admission

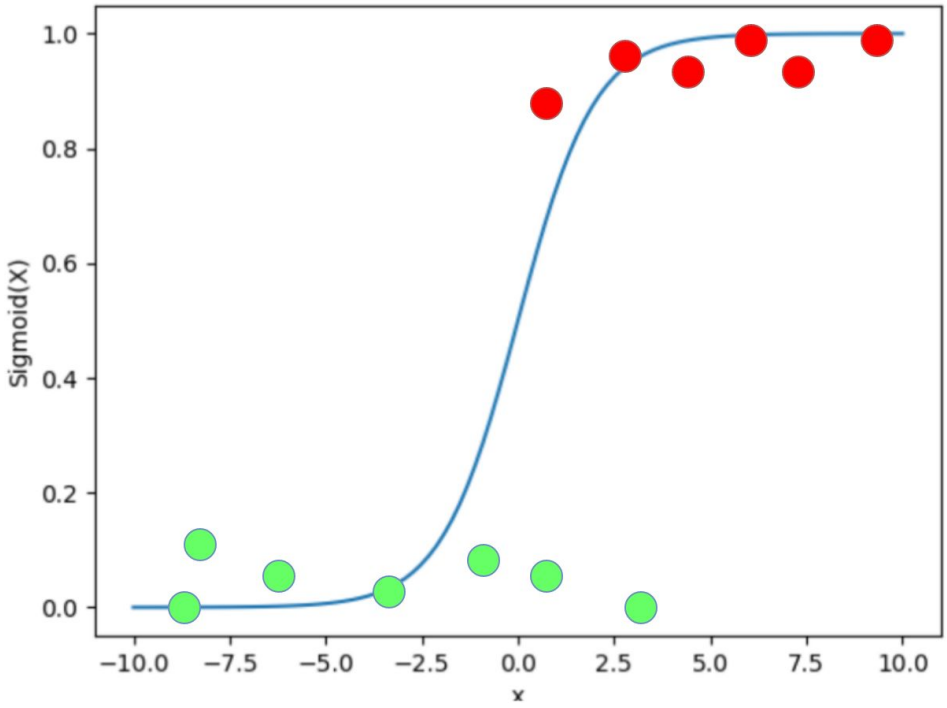
A window describes the time after which the patient needed ICU. For example, a patient might need ICU after 2 hours of admission and so there is a column entry corresponding to that patient which highlights this fact in the dataset. Whereas a predictive model using all-time windows will probably yield greater accuracy, a nice model using only the first (0-2) is likely to be more clinically relevant.

For filling the missing values the following strategy has been used:

- Fill the missing values with the mean value of those rows which have the WINDOW and ICU value same as that of the missing row.
- Fill the missing values with the median value of those rows which have the WINDOW and ICU value same as that of the missing row.
- Remove the rows which have more than 70 percent Nan values and then impute using mean and median values of rows having the same WINDOW and ICU values. The mean and median of rows having the same WINDOW and ICU is used because these values for a patient with 0-2 window and ICU required is not related to a patient with 4-6 window and ICU not required. We assume that different categories of WINDOW and ICU do not have an impact on each other.

The filling with median strategy works better than filling with mean and hence the same has been carried forward in the analysis.

We break the dataset into five different data frames, one for each window. In our notebook, we have used the windows of 0-2 hr and 2-4 hr as the training data, since only window 0-2 did not give us good results, while the rest are used as test sets. Since the prediction involves binary classification, we use the ROC curve (receiver operating characteristic curve) to judge the accuracy of our model. We now briefly describe the ROC using a logistic regression model.



The red dots represent that a patient requires ICU and the green dots represent that a patient does not require ICU. We try to fit a logistic regression model here. The y-axis is now converted to the probability that a patient needs ICU. For simplicity, we have shown only one feature as the x-axis (but there are many in the dataset).

We start by setting threshold values (0 to 1) i.e a line parallel to the x-axis and then try to figure out how our model worked for that threshold. We make the following table

		Acutal	
		Need ICU	Does not need ICU
Predicted	Need ICU	281	5
	Does not need ICU	86	13

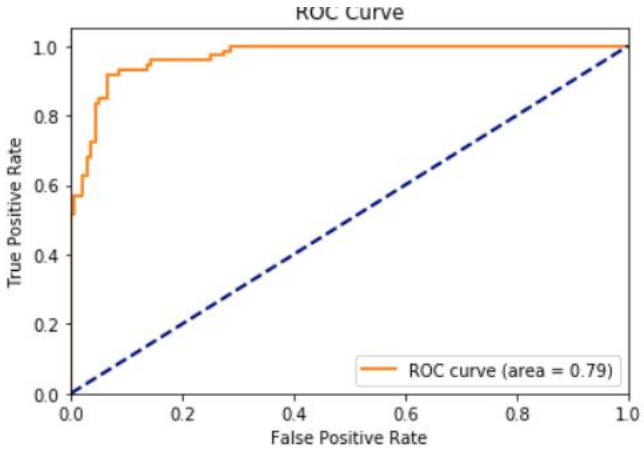
This is a confusion matrix that tells the count of what we predicted and what it actually was. We make more such matrices for different thresholds and use ROC which provides a simple way to summarize the information. The y-axis is the true positive rate which is defined as

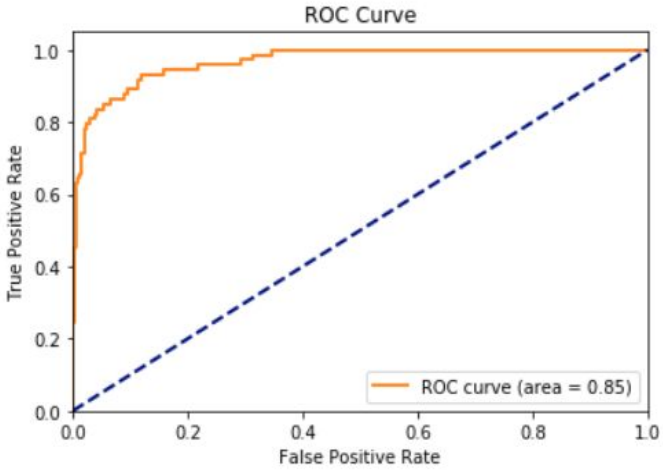
$$True\ positive\ rate\ =\ sensitivity\ =\ \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

The x-axis is the false positive rate which is defined as

$$False\ positive\ rate\ =\ 1 -\ specificity\ =\ \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

Using this we get the curve similar to this:





Our goal is to maximize the area under the curve (AOC) and R2 scores:

The best R2 score values when rows with max Nan values are not removed are depicted below and it is achieved with MLPClassifier.

clf= MLPClassifier(alpha=1, max_iter=10000).fit(X_train,y_train)

```
training data prediction score
R2 score: 0.958
------------------------------
window 4-6 prediction score
R2 score: 0.896
------------------------------
window 6-12 prediction score
R2 score: 0.912
------------------------------
window above 12 prediction score
R2 score: 0.823
------------------------------
```

The best R2 scores and ROC curve areas after removing the rows with 70% or more Nan values and imputing with the grouped median was also achieved with

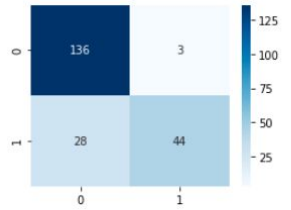
MLPClassifier.

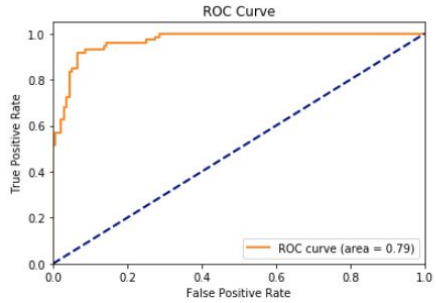
```
------------------------------
window 4-6 prediction score
              precision    recall  f1-score   support

           0       0.83      0.98      0.90       139
           1       0.94      0.61      0.74        72

   micro avg       0.85      0.85      0.85       211
   macro avg       0.88      0.79      0.82       211
weighted avg       0.87      0.85      0.84       211

R2 score: 0.853
```



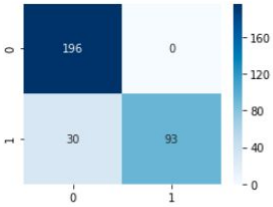


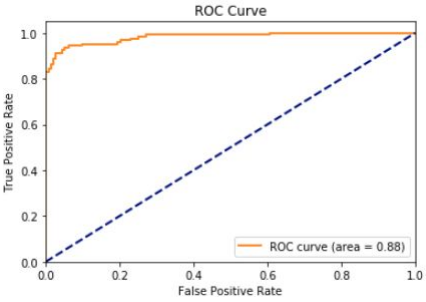

```
window 6-12 prediction score
              precision    recall  f1-score   support

           0       0.87      1.00      0.93       196
           1       1.00      0.76      0.86       123

   micro avg       0.91      0.91      0.91       319
   macro avg       0.93      0.88      0.90       319
weighted avg       0.92      0.91      0.90       319

R2 score: 0.906
```



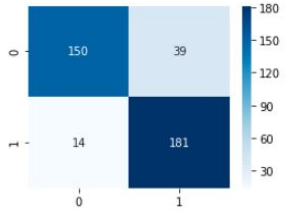


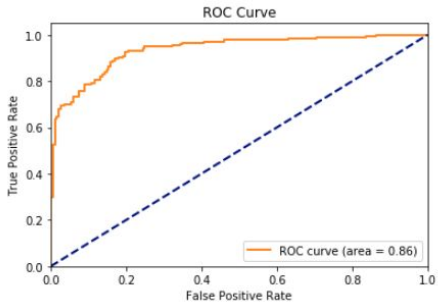

```
window above 12 prediction score
              precision    recall  f1-score   support

           0       0.91      0.79      0.85       189
           1       0.82      0.93      0.87       195

   micro avg       0.86      0.86      0.86       384
   macro avg       0.87      0.86      0.86       384
weighted avg       0.87      0.86      0.86       384

R2 score: 0.862
```

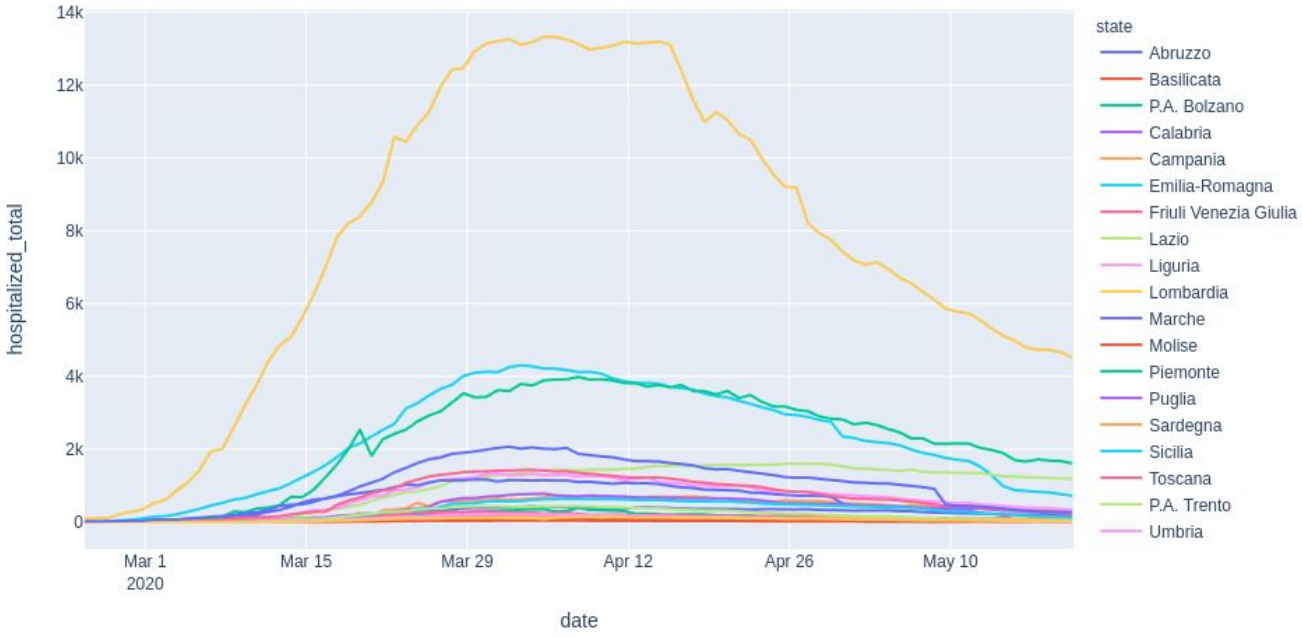




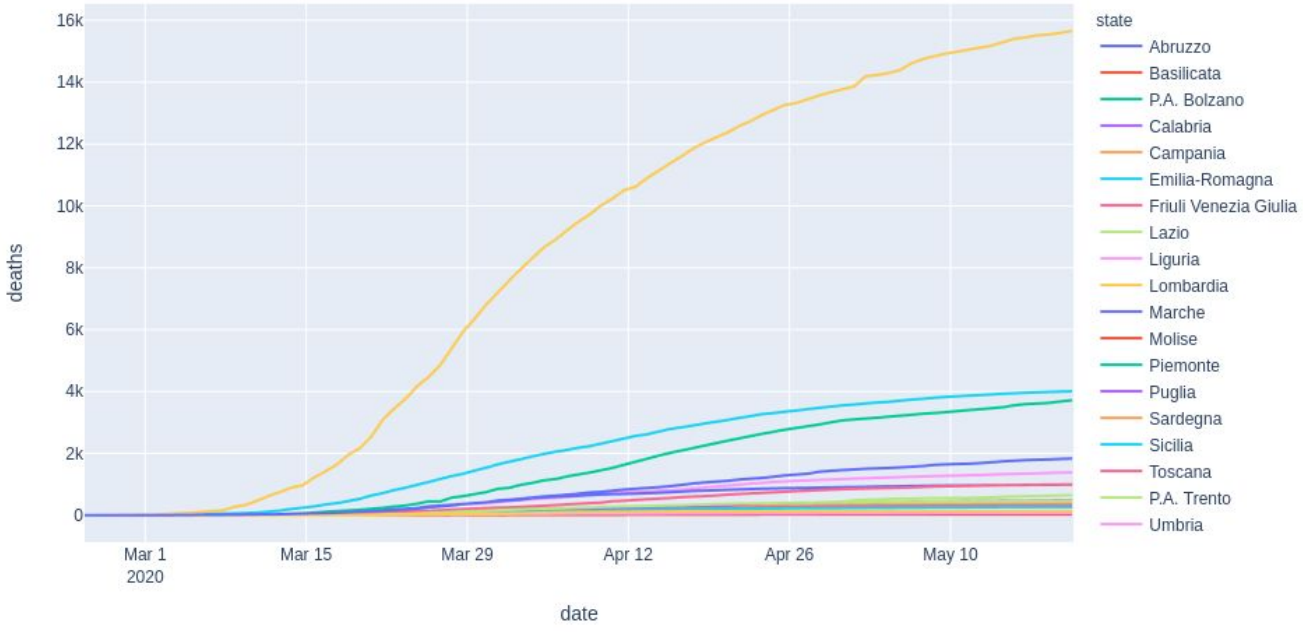
Predicting The Number of Ventilators that will be Required By Different countries (and their most affected regions in particular)

Data Visualization (Italy - by region)

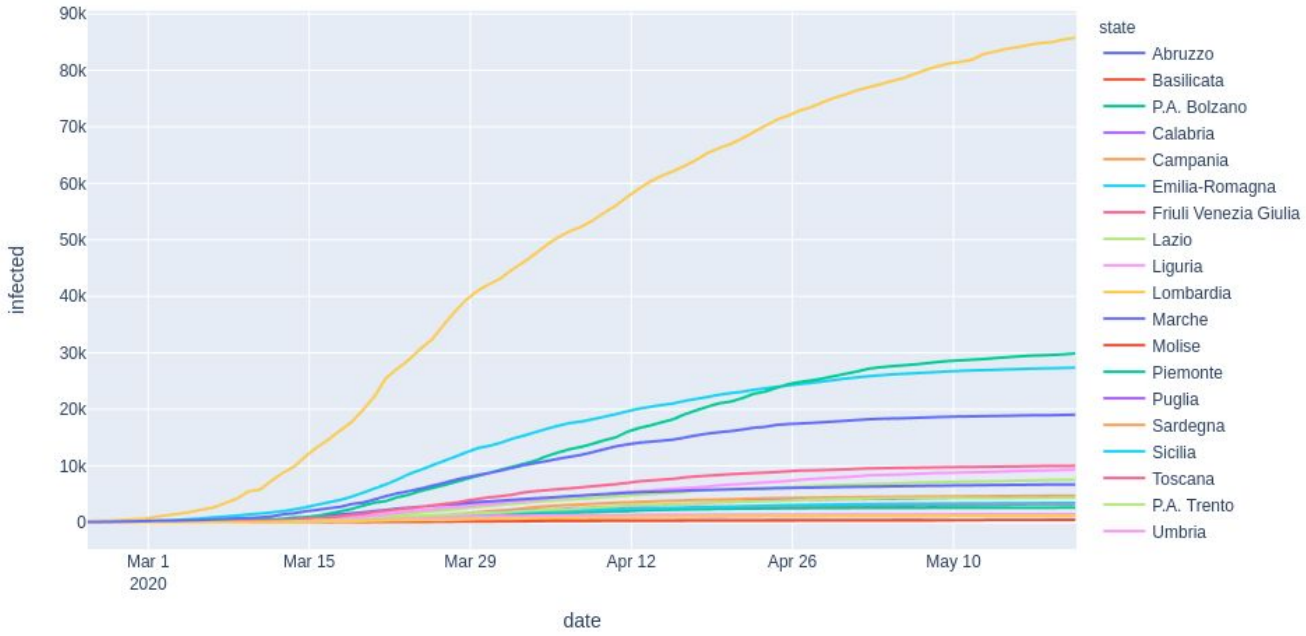
Hospitalization vs Time



Deaths vs Time



Infected people vs Time



We notice that the region of Lombardia is the most affected region by Covid-19. So we will be focusing on predicting the number of ventilators that Lombardia will require in the future based on our predictive model.

Proposed Approach:

● We analyzed the data from Italy and used some assumptions (based on previous studies) to build a model by using a modified susceptible-exposed-infectious-recovered (SEIR) compartmental mathematical model for our predictions.

The following are the Ordinary Differential Equations for the SEIR model:-

$$\frac{ds}{dt} = -a.\ s(t).\ i(t) \qquad \text{i.}$$

$$\frac{de}{dt} = a.\ s(t).\ i(t) - b.\ e(t) \qquad \text{ii.}$$

$$\frac{di}{dt} = b.\ e(t) - c.\ i(t) \qquad \text{iii.}$$

$$\frac{dr}{dt} = c.\ i(t) \qquad \text{iv.}$$

s(t): Susceptible cases - Population fraction that can get a disease

e(t): Exposed cases - Population fraction that got the virus, but doesn't present any symptom (asymptomatic)

i(t): Infected cases - Those exposed, but present some symptoms

r(t): Recovered cases - Population fraction that got infected and recovered after some time.

In order to solve the above Ordinary Differential Equations we'll use python's **scipy.integrate.odeint** function.

SEIR for Lombardia

We define some initial population (N) of Lombardia: N = 10103969

We need to assign some initial values to our parameters to solve the differential equations.

```
In [10]: R_start = region_data.loc[0, 'recovered']/N
         I_start = region_data.loc[0, 'infected']/N
         E_start = (region_data.loc[6, 'infected'] - region_data.loc[5, 'infected'])/N
         S_start = 1 - E_start - I_start
```

Taking the initial R parameter to be the fraction of recovered people on day 1.
Taking the initial I parameter to be the fraction of infected people on day 1.
Taking the initial E parameter to be the fraction of infected people on day 7 - infected people on day 6.
Taking the initial S parameter to be 1 - E(start) - I(start)

'region_data' stores the the data for Lombardia

Define the differential equations system.

```
# Differential Equations System
dSdt = -exposed_rate*s*i
dEdt = (exposed_rate*s*i) - (infection_rate*e)
dIdt = (infection_rate*e) - (recovery_rate*i)
dRdt = recovery_rate*i
```

Calculating the SEIR Model:

```
def odeint_model(params, t, initial_condition):
    # Create an alias to our ode model to pass guessed params
    ODE_SEIR = lambda y,t:seir_diff_eqn(y, t, params)


    # Calculate ode solution, return values to each
    ode_result = integrate.odeint(func=ODE_SEIR, y0=initial_condition, t=t)

    # Return results
    return ode_result
```

‘seir_diff_eqn()’ returns the differential equation model. Then we integrate it after passing the initial/ starting values and store it in ‘ode_result’ and then return it.

Optimizing the parameters

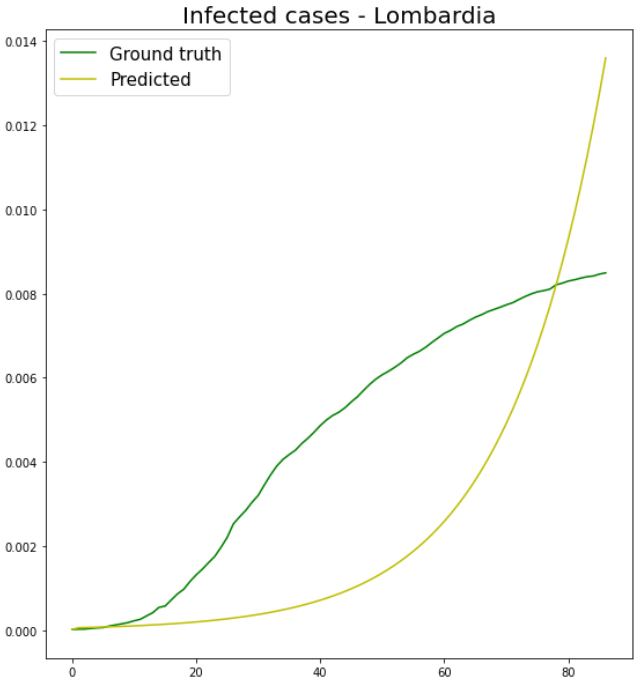
We are measuring the residual value (error to minimize) as a ‘Least Square’ over a residual for each point in the ‘infected cases’ and ‘recovered cases’ curves. I.e., we are minimizing two cost functions at the same time using scipy’s ‘optimize.leastsq’.

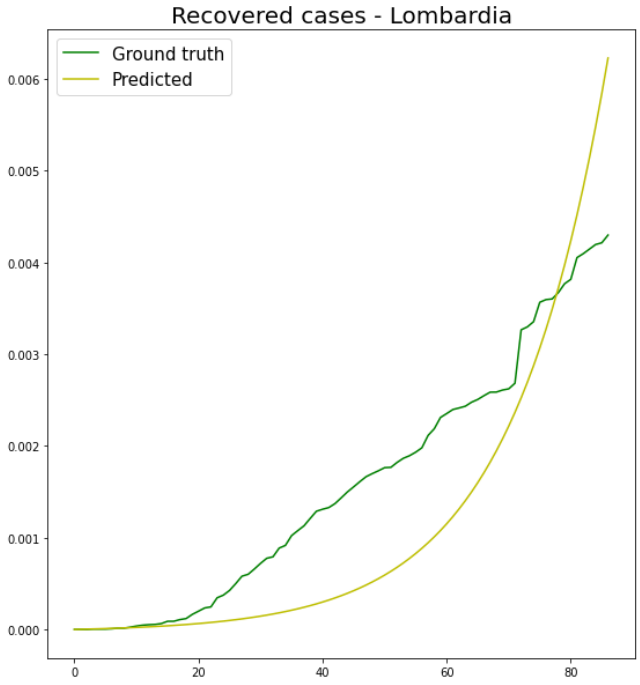
```
optimal_parameters, acceptance = optimize.leastsq(Fit_SEIR, x0=initial_parameters_guess, args=(time_period,
                                                                initial_conditions, true_vals), ftol=1.49012e-20)
```

```
print('\tLombardia')
print('Optimized Infection rate: ', optimal_parameters[0])
print('Optimized Recovered rate: ', optimal_parameters[1])
print('Optimized Exposed rate: ', optimal_parameters[2])
```

```
        Lombardia
Optimize infection rate:  517.3699984973765
Optimize recovered rate:  0.029339849305285735
Optimize exposed rate:  0.09405890570512981
```

Plotting the results for Infected/Recovered



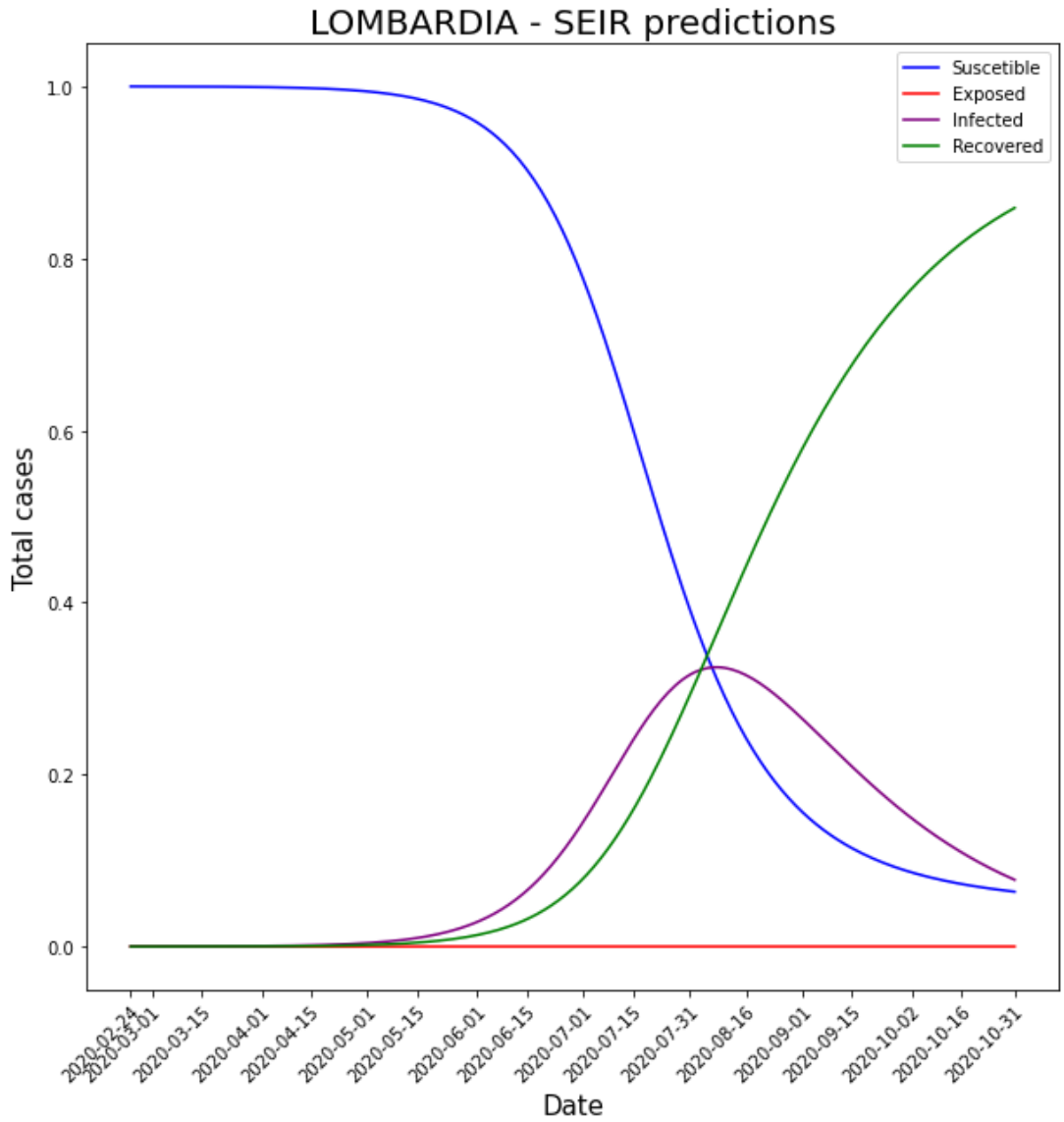


We observe that:

- The Infected cases graph is not very accurate but the Recovery rate graph is relatively more accurate.
- This is because SEIR is just a short term mathematical model.

- This may also be attributed to the fallacies in the assumptions of the SEIR model, some of which were that the population remains constant during the pandemic (clearly not true in our case) and that people 'must' transmit the virus if they interact with each other.
- Many possibilities like people interacting while wearing proper protective gear and asymptomatic people are not accounted for in the model. Nonetheless, it provides some estimation for short term scenarios.

SEIR predictions until October for Lombardia



Calculate ventilators curve based in SEIR infected curve

According to nsmedicaldevices.com we have the following:-

Analytics company GlobalData estimates the US needs about 75,000 ventilators, while France, Germany, Italy, Spain and the UK collectively require 74,000 devices to make up the gap.

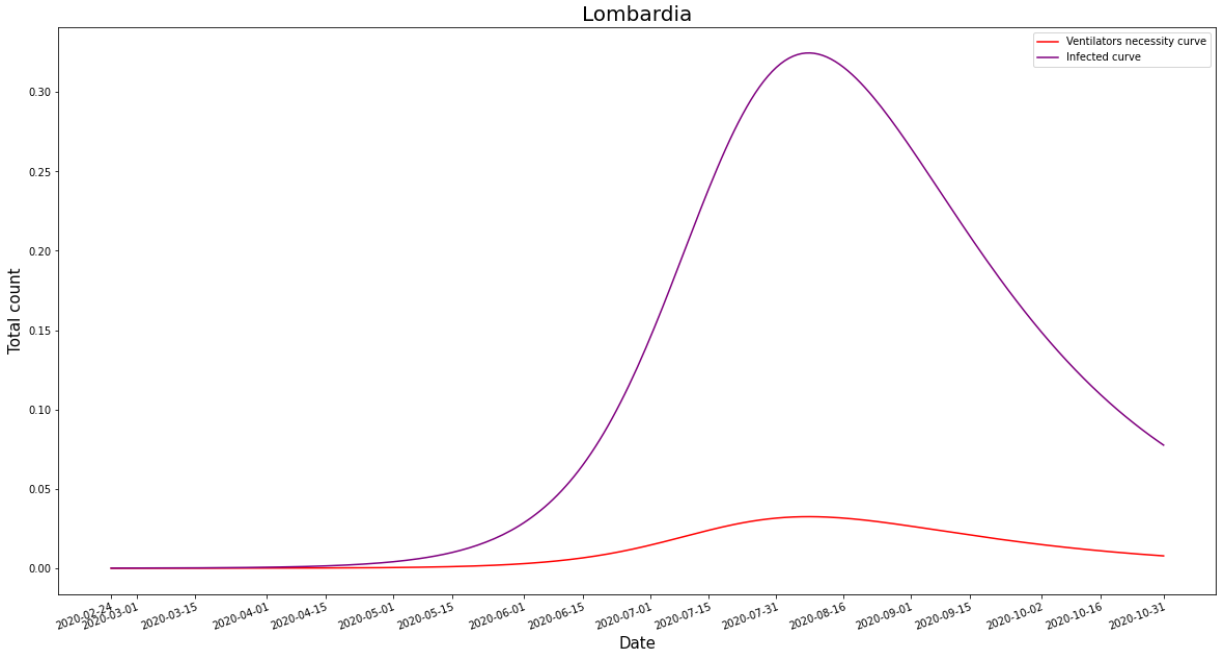
It's estimated that, of the Covid-19 cases occurring worldwide, about 10% of patients need ventilators.

GlobalData's medical devices analyst Tina Deng said: "Ventilator shortages are a crucial reality as the Covid-19 outbreak continues to worsen globally.

Based on this, we're assuming that **10%** of all infected people will require ventilators.

```
ventilators_future_pred = 0.10*predictions_future[:,2]
# 0.10 for the 10% assumption
```

The below graph just depicts the infected people and the ventilator requirements curve using our 10% assumption



Predicting the ventilator requirements in Lombardia

```
# Get the maximum curve value and transform to absolute value multiplying by region population
max_vent_necessity = N*max(ventilators_future_pred)
# Show results
print('Lombardia would require: ', int(max_vent_necessity),'ventilators at the peak of the Pandemic')
```

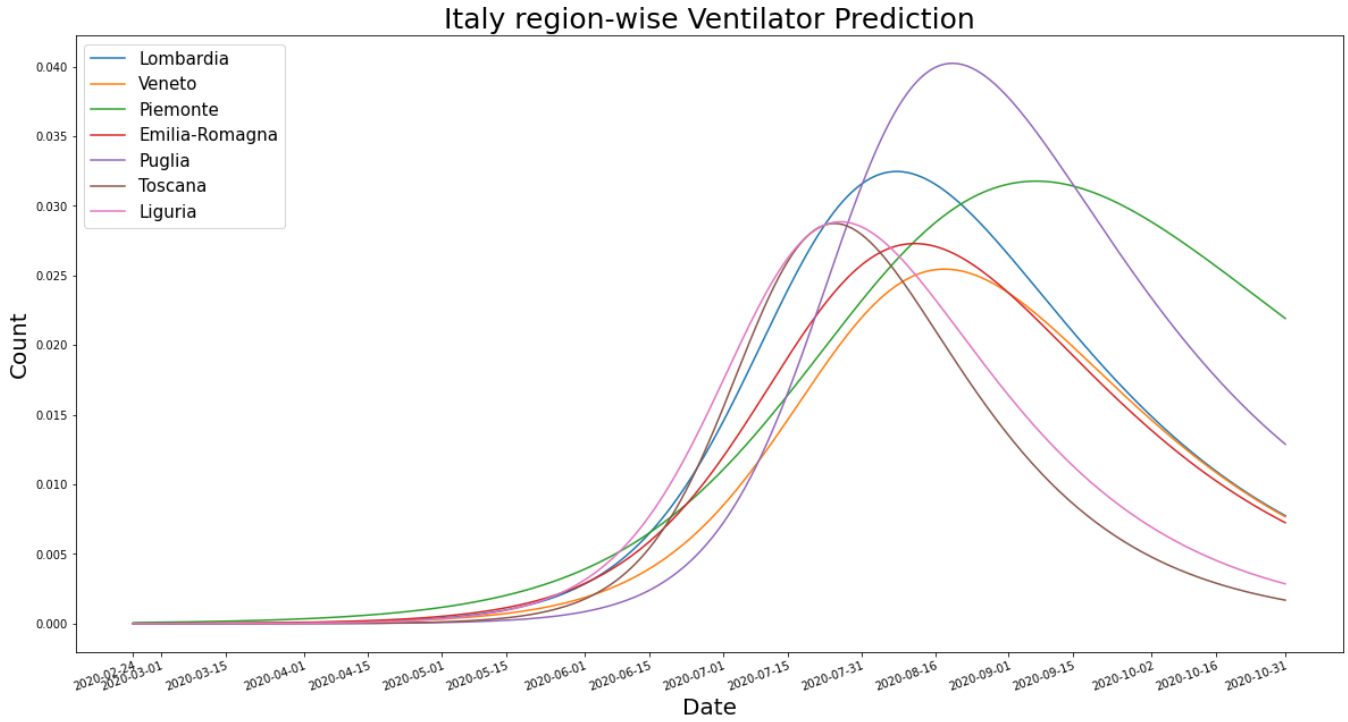
```
Lombardia would require:  340357 ventilators at the peak of the Pandemic
```

Now, we do the same for the other risk regions of Italy:

We added 6 more regions of italy which were at high risk and performed the above analysis on them to compare the results.

```
italy_risk_regions = ['Veneto', 'Piemonte', 'Emilia-Romagna','Puglia','Toscana','Liguria']
# Population for each risk region
pop_states = [4907704.0, 4341375.0, 4467118.0,4031885.0, 3722729.0, 1543127]
```

Final Graph Showing the predicted ventilator requirements of different regions over time.



From the above graph we observe that even though the region of **Puglia** had the lowest cases and ventilator requirements initially as compared to the other regions shown in the graph, it requires more ventilators than any other region around August 2020 (even more than Lombardia!).

Printing the Final Ventilator Requirements of the different regions/states:

```
          HIGH RISK REGIONS

1   Lombardia
Number of Ventilators required:  340357

2   Veneto
Number of Ventilators required:  128419

3   Piemonte
Number of Ventilators required:  137935

4   Emilia-Romagna
Number of Ventilators required:  119141

5   Puglia
Number of Ventilators required:  168679

6   Toscana
Number of Ventilators required:  106789

7   Liguria
Number of Ventilators required:  44797

Total:  1046117
```

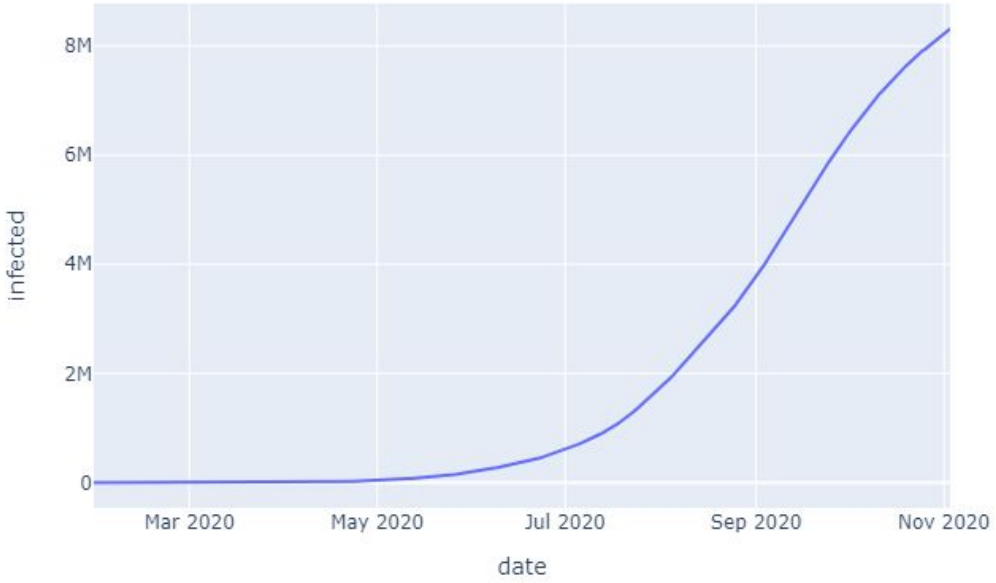
In total Italy would need **1,046,117** Ventilators for just these 7 regions.

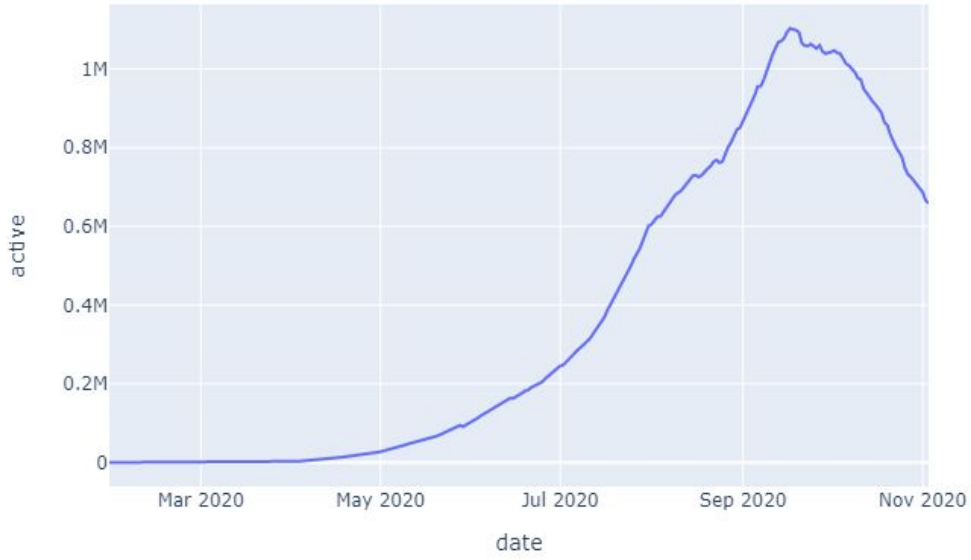
Note: We weren't able to cross check our predictions or try other optimization techniques due to lack of time. And we had to use the percentage of hospitalized people requiring ventilators from external sources because we were not to compute that in the given timeframe.

SEIR Prediction - India

- We collected Indian Covid Time Series Data Using COVID19-India API | api.
- The data was Time Series data consisting of the number of confirmed cases, recovered cases ,deaths and daily changes in each of these.
- We plotted all the above columns in order to understand and gain insights into the data
- Below are some of these visualizations :

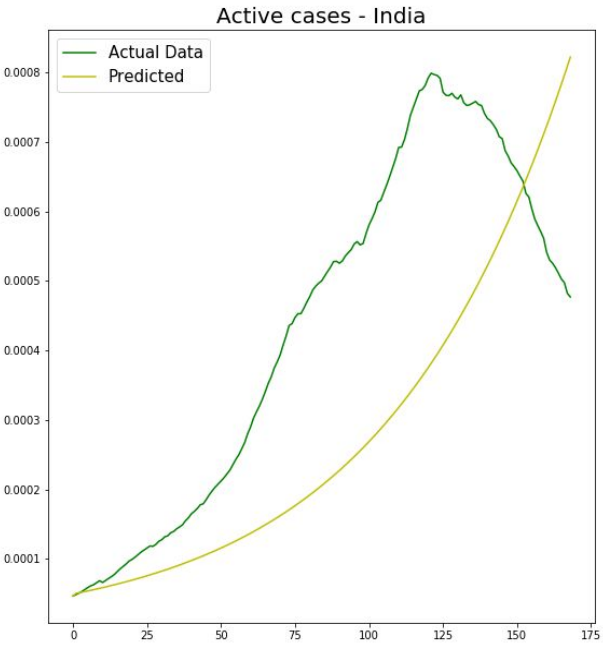
Infected Cases vs Time

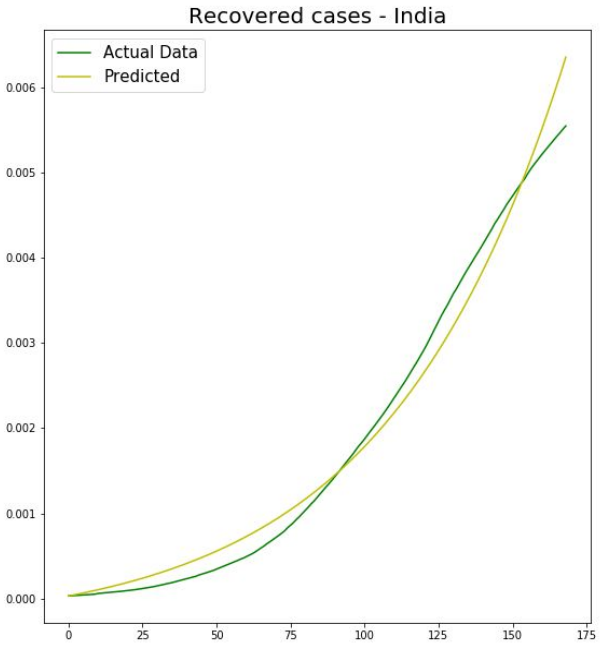




Active Cases vs Time

- Then, we took the initial parameters for S,E,I,R from 19th May (since after this point of time, we had a substantial number of cases.)
- We created a loss function from the actual data and the predicted model we got from the Odeint library.
- We minimized this loss function using the least method in scipy.optimize and obtained the optimal parameters for the exposed rate, infection rate, and recovery rate
- We used these rates and the initial values of S, E,I, R to calculate the number of Susceptible, Exposed, Infected, and Recovered People over time
- We predicted the number of active cases and recovered cases till November using the model





The X-axis denotes the number of days after 19th May and the Y-axis denotes the number of predicted/actual Active and recovered cases as a fraction of India's total population.

We got a low training accuracy on the infected cases while we got a very high training accuracy on the recovered cases.

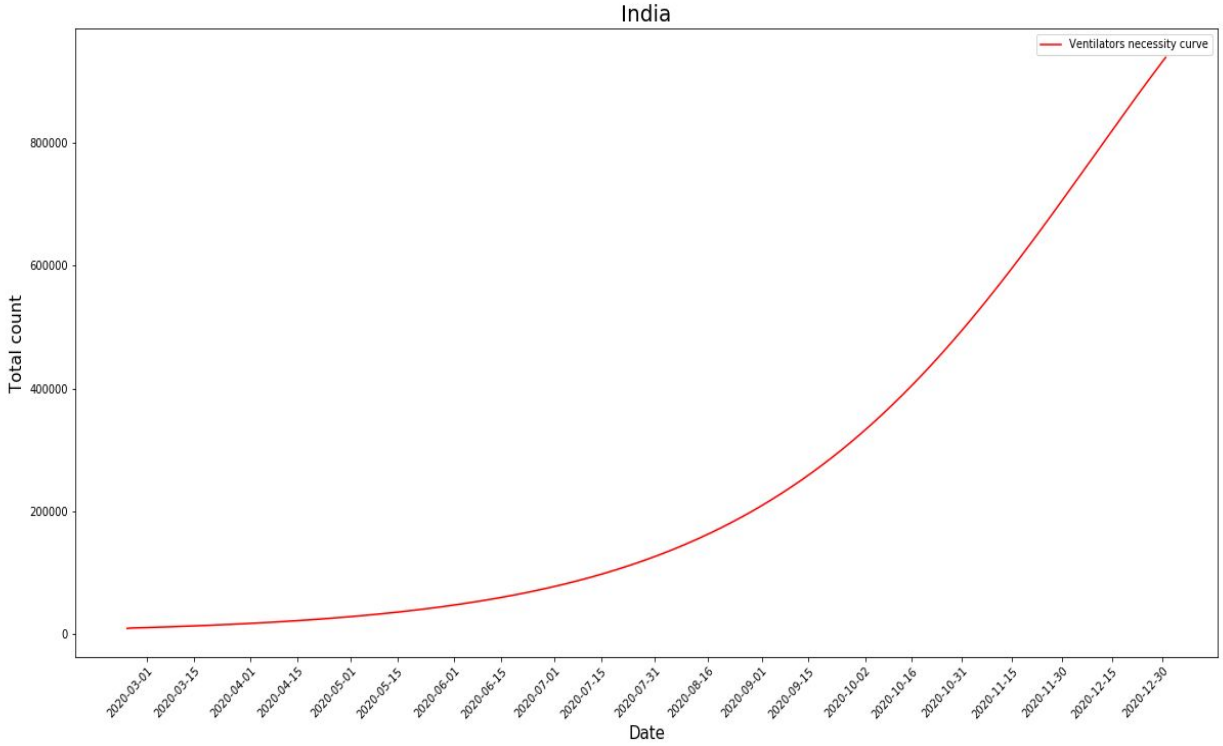
```
print('Accuracy for Active Cases :',r2_score(inf_rec_true[0],y_pred[:,2]))
```

```
Accuracy for Active Cases : 0.14926685564320719
```

```
print('Accuracy for Recovered Cases :',r2_score(inf_rec_true[1],y_pred[:,3]))
```

```
Accuracy for Recovered Cases : 0.9833269360105391
```

We predicted the number of ventilators India would require from March till the end of December using this model using the assumption that 13% of all infected people required ventilators.



Predicted Number of Ventilators vs Time

<u>Alternative Model for Predicting India's Ventilator Requirement</u>

We decided to apply a more accurate and elaborate model which is called SEIR HCD.
Here, the additional 3 parameters stand for hospitalized, Critical, and Deceased.
We used the following set of differential equations for more accuracy :-

$dS/dt = -(R_t / t_{inf})SI$

$dE/dt = (R_t / t_{inf})SI - E/t_{inc}$

$dI/dt = E/t_{inc} - I/t_{inf}$

$dR/dt = (m_a*I/t_{inf}) + (1-c_a)*H/t_{hosp}$

$dH/dt = (1-m_a)*I/t_{inf} + (1-f_a)*C/t_{crit} - H/t_{hosp}$

$dC/dt = c_a*H/t_{hosp} - C/t_{crit}$

$dD/dt = f_a* C/t_{crit}$

The above variables denote the following:-

R_t = Reproduction Number

t_{inc}= Incubation Period

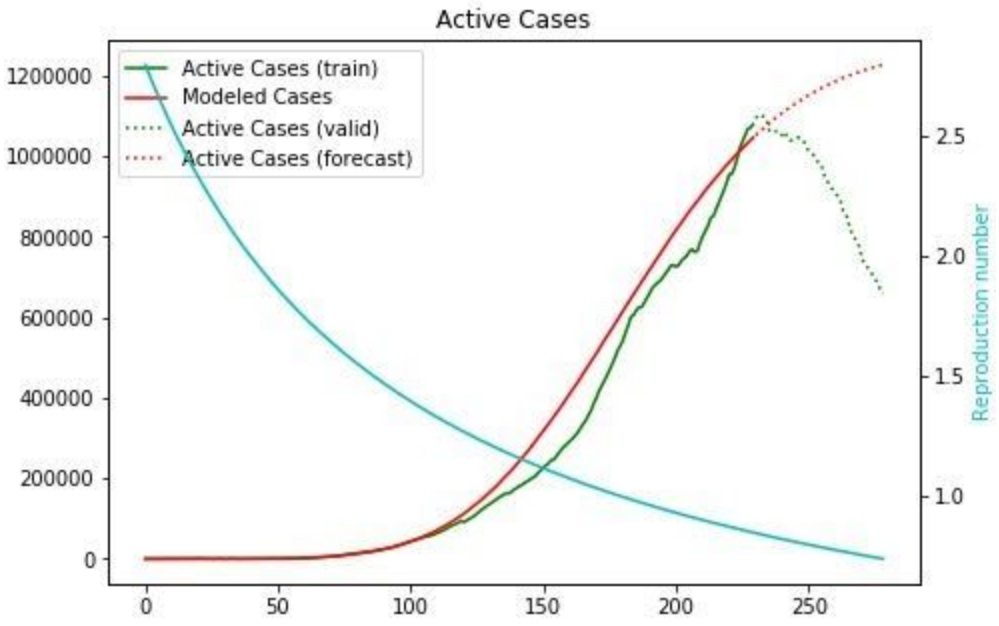
t_{inf}= Infectious Period

t_{hosp}= Average time of a patient while hospitalization (After this he/she either recovers or becomes critical)
t_{crit}= Average time while a patient is in a critical state (Either he recovers or dies)
m_a=Fraction of asymptomatic infections
c_a=Fraction of critical cases
F_a=Fraction of fatal cases
All the above variables(except R_t) were set to default values with regards to global statistics of COVID-19 and Reproduction Number was modeled using 2 methods:-

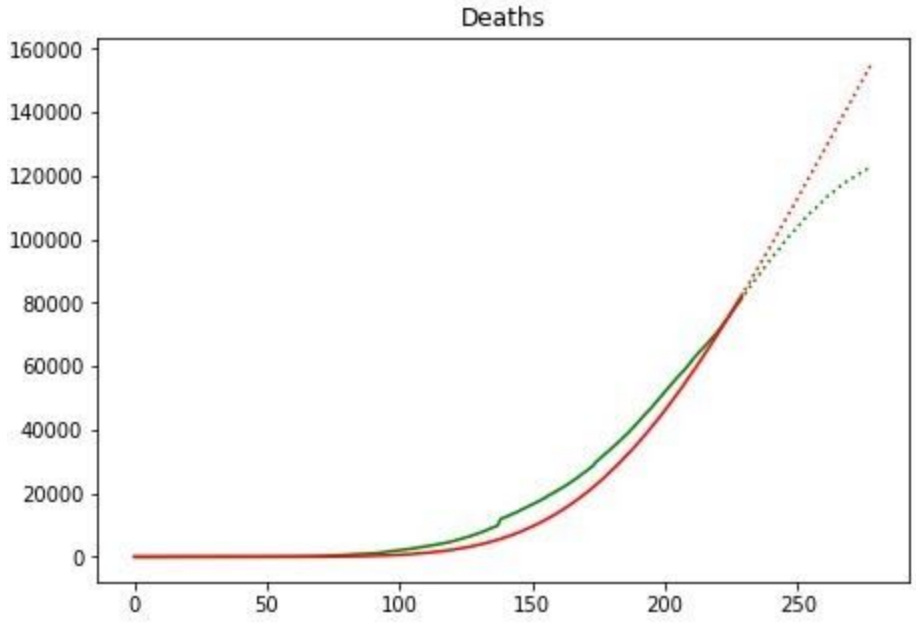
1. Constant Reproduction Number:- Reproduction Number stays constant throughout.
2. Hill Decayed Reproduction Number:- $R_t = R_0/(1+(t/L)^k)$
   Where t=time elapsed and L and K are parametric constants.
   We used Scipy.integrate.solve_ivp to integrate the differential equations and obtain the number of infected people, deaths, and hospitalized people(in need of ventilators).

Here, are the predicted active cases, deaths and ventilator requirements for India.

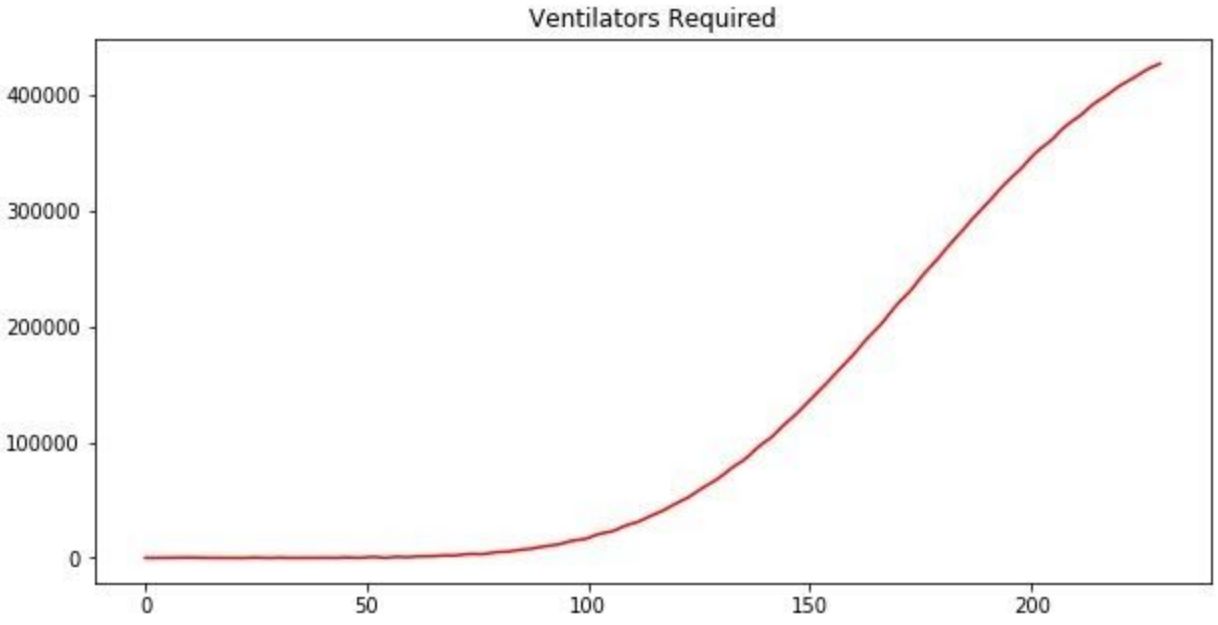
Active Cases vs Time



Here, the blue line denotes the value of Reproduction Number over time . The Y-axis for this is present on the right side.
The red line is the predicted number of active cases.
The green line is the actual number of active cases over time
The X-axis is the number of days elapsed from 30th January 2020



The red line is the predicted number of deaths over time
The green line is the actual number of deaths over time
The X-axis is the number of days elapsed from 30th January 2020



Here, The red line shows the number of (hospitalized + critical) people over time . This denotes the number of ventilators required over time.
The X-axis is the number of days elapsed from 30th January 2020.

Accuracy :- We used the metric Mean Squared Logarithmic Error for testing the accuracy of our model on the validation set (This is from September 15 onwards). We obtained an MSLE of 0.05393 for this model.

```
Validation MSLE: 0.05393
R: 2.800, t_hosp: 9.101, t_crit: 14.250, m: 0.501, c: 0.998, f: 0.078
```

Final Conclusion from SEIR-HCD Model of India :- India would need around 4,00,000 ventilators from till November 2020 to provide to every critical and hospitalized patient.

Member Name	**Data Collect/Clean**	**Data Model/Analysis**	**Data Viz./Report**
Abinash Acharya	Used a python script to select the set of datasets which were relevant to hospitalization resources from the UNCOVER folder	Collected India's Time Series Data Applied the SEIR Model to India and found the model to be insufficient Applied an alternative SEIR-HCD model to India	Visualizations of Active Cases,Recoveries, deaths,etc for India's Time Series Data Visualization of prediction of ventilator requirement for India Report on SEIR Predictions - India
Asad Abidi	Wrote a script to find the set of datasets which were relevant to Italy from the UNCOVER folder.	Used the SEIR Model to predict the number of Ventilators required in Italy. Found it to be a little loose (Not good for accurate predictions).	Visualizations of Hospitalized, Infected and Recovered people in Italy. Visualization of prediction of ventilator requirement for the 7 most affected states of Italy.
Ashutosh Soni	Preprocessed the data and imputed missing values for ICU dataset. Tried different imputation methods.	Used different classifiers and Predicted the ICU requirement of a patient using ROC and R2.	Report on Predictions of ICU requirement. ROC visualisation for best performing classifier.

Gopal Ramesh Dahale	Data is collected by me and basic data cleaning for ICU dataset. Collection and cleaning of Hospitalization data of the USA.	Gave the idea of using ROC-AUC for ICU prediction.	Exploratory data analysis of ICU dataset. Report on EDA. EDA on hospitalization data and conclusion. Report on the same.
Himanshu Sekhar Nayak	Selected the relevant dataset for predicting no of beds and ICU required, from UNCOVER folder.	Used scoring function to get top 10 medical parameters helpful for determining if a COVID-19 patient might hospitalised, require an ICU.	Visualization of different medical parameters affecting a patient's possibility to get to hospital, ICU.