Predicting the Daily Deaths due to COVID-19

Jasleen Singh

Introduction

Coronavirus disease (COVID-19) is an infectious disease caused by a newly discovered coronavirus in 2019. The virus that causes COVID-19 is mainly transmitted through droplets generated when an infected person coughs, sneezes, or exhales. These droplets are too heavy to hang in the air, and quickly fall on floors or surfaces.

Most people who fall sick with COVID-19 will experience mild to moderate symptoms and recover without special treatment. As of now there is no vaccine or medicine for this disease. Some of the COVID-19 patients dies as their immune system is not strong enough to fight with the disease.

Number of Death cases highly depends upon various factors like number of new cases, hospitals available, immune system of people, lockdown strictness, human development index, cleanliness facilities, Life expectancy and various other factors.

To predict the deaths, this project will be utilising the historical data of COVID-19. To work on this, we will pass the various parameters that will help us to predict the approximate deaths due to COVID-19 and come out with the best possible COVID-19 death predicting model.

Target Audience

1. WHO (World Health Organization)
   By targeting the factors impacting the COVID-19 patient deaths, organization can minimise the deaths cases by taking preventive measures.
2. Country administration
   As death in country directly leads to economy backlash, so Country administration can take preventive measures, educate people and prepare for required hospital facility
3. Citizens
   Citizens can take precautions based on the most impacting parameter which leads to increase in death cases.

Data

Data Source

To address the problem, we need data with details of all the possible factors of deaths of COVID-19 patients. In this project, we will use the data from Github repository by **owid** in CSV format:

Data: https://covid.ourworldindata.org/data/owid-covid-data.csv

Meta-data: https://covid.ourworldindata.org/data/owid-covid-codebook.csv

We have a lot of fields like new cases, stringency index, median age, cardiovascular death rate, smokers, human development index, life expectancy, etc which gives a lot of options to select the best suited fields for our model.

Methodology

Exploratory Analysis

To find out the key features from the raw data, data processing needs to be normalised. Once data is ready for visualisation, various visualisation techniques can be used to find the hidden secrets. Based on the result model will be developed.

Data Processing

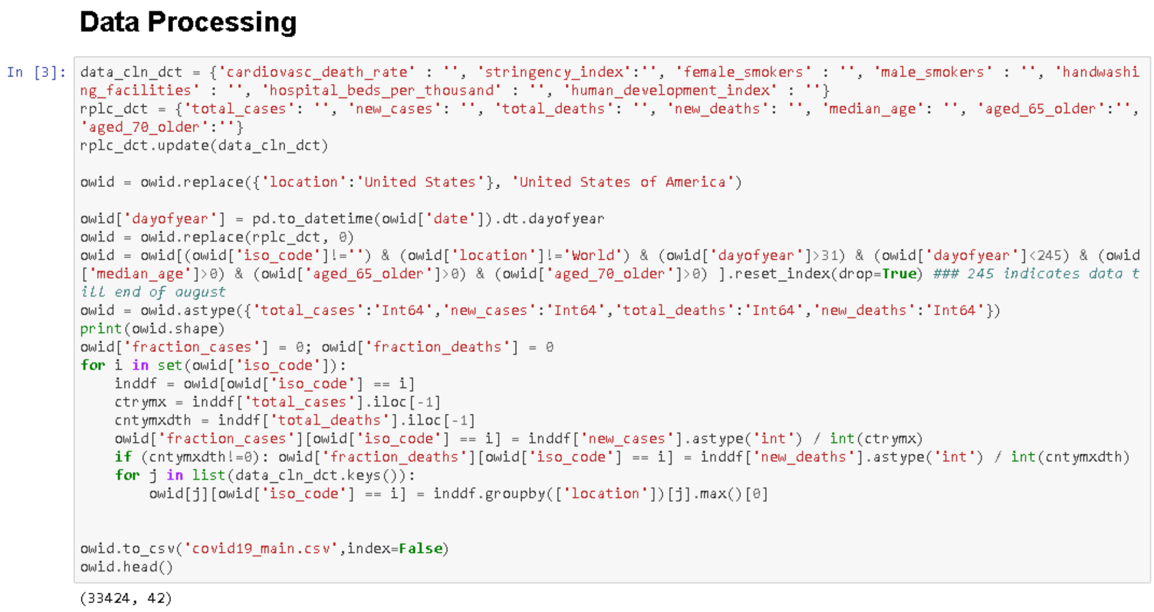
There were lot of missing values from historical records of COVID-19 data, because of lack of record keeping. So I decided to use the data from 1st February, 2020 till 31st August, 2020. There were a several issues with the dataset like numbers stored as string, empty strings instead of zero, missing data.

To analyse data easily, I created a new field where day of the year is stored.

To fill-up the missing data, I indetified the columns whose null values can be replaced with zeros and perform the task for all those columns. This will not only ease the visualisation but also help in model training.

Converted all the numbers to integers wherever float is not possible like number of new cases, number of deaths, number of new deaths, etc

To fill the standard values for a set of records that were mistakely mentioned as zero for few records in the set, I took the maximum values of that set, like taking maximum of human development index of a country and replacing wherever its zero. This process was repeated for few other columns also.



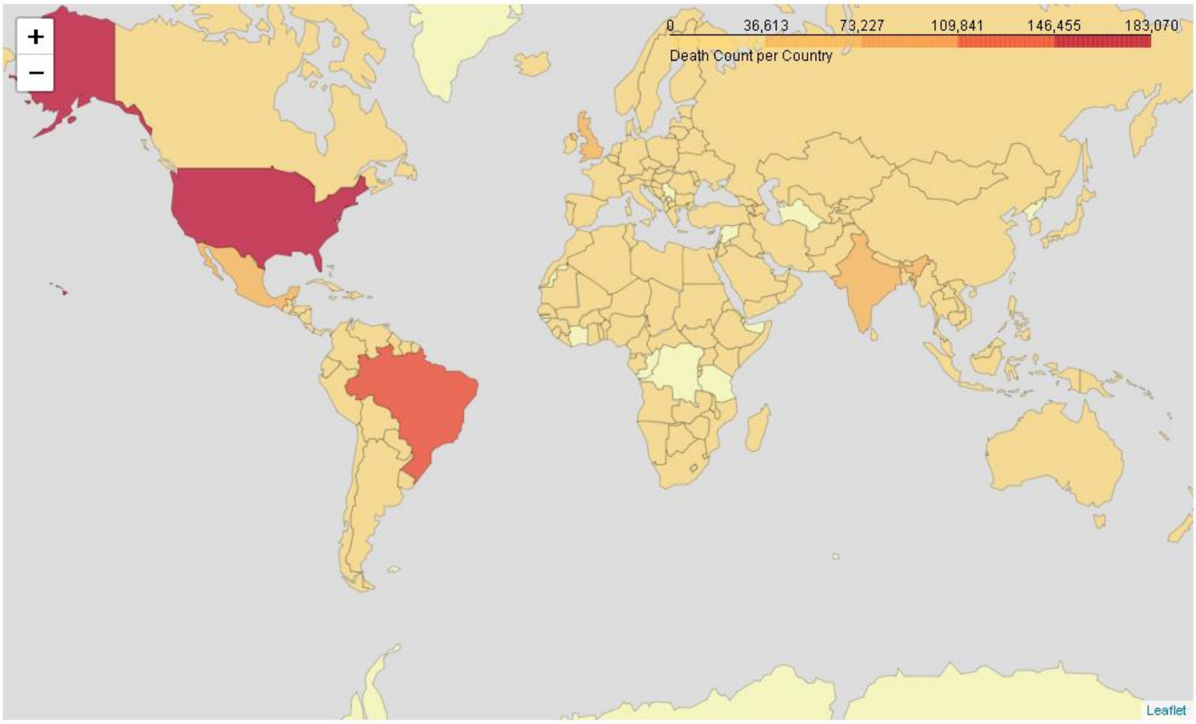
Data Processing

```
In [3]: data_cln_dct = {'cardiovasc_death_rate' : '', 'stringency_index':'', 'female_smokers' : '', 'male_smokers' : '', 'handwashi
ng_facilities' : '', 'hospital_beds_per_thousand' : '', 'human_development_index' : ''}
rplc_dct = {'total_cases': '', 'new_cases': '', 'total_deaths': '', 'new_deaths': '', 'median_age': '', 'aged_65_older':'',
'aged_70_older':''}
rplc_dct.update(data_cln_dct)

owid = owid.replace({'location':'United States'}, 'United States of America')

owid['dayofyear'] = pd.to_datetime(owid['date']).dt.dayofyear
owid = owid.replace(rplc_dct, 0)
owid = owid[(owid['iso_code']!='') & (owid['location']!='World') & (owid['dayofyear']>31) & (owid['dayofyear']<245) & (owid
['median_age']>0) & (owid['aged_65_older']>0) & (owid['aged_70_older']>0) ].reset_index(drop=True) ### 245 indicates data t
ill end of august
owid = owid.astype({'total_cases':'Int64','new_cases':'Int64','total_deaths':'Int64','new_deaths':'Int64'})
print(owid.shape)
owid['fraction_cases'] = 0; owid['fraction_deaths'] = 0
for i in set(owid['iso_code']):
    inddf = owid[owid['iso_code'] == i]
    ctrymx = inddf['total_cases'].iloc[-1]
    cntymxdth = inddf['total_deaths'].iloc[-1]
    owid['fraction_cases'][owid['iso_code'] == i] = inddf['new_cases'].astype('int') / int(ctrymx)
    if (cntymxdth!=0): owid['fraction_deaths'][owid['iso_code'] == i] = inddf['new_deaths'].astype('int') / int(cntymxdth)
    for j in list(data_cln_dct.keys()):
        owid[j][owid['iso_code'] == i] = inddf.groupby(['location'])[j].max()[0]


owid.to_csv('covid19_main.csv',index=False)
owid.head()

(33424, 42)
```

Data Visualisation over World Map

To analyse the COVID-19 deaths, I used folium to display the country level data over map. I fetched the world country polygon data as json and loaded into map. Colour bar as legend shows the death count per country.

As per the map, its clear that in terms of number of death United States is the worst impacted country and Brazil is the next to United States. India, Mexico and United Kingdom are next to these top two countries in the list of worst impacted countries in terms of deaths.



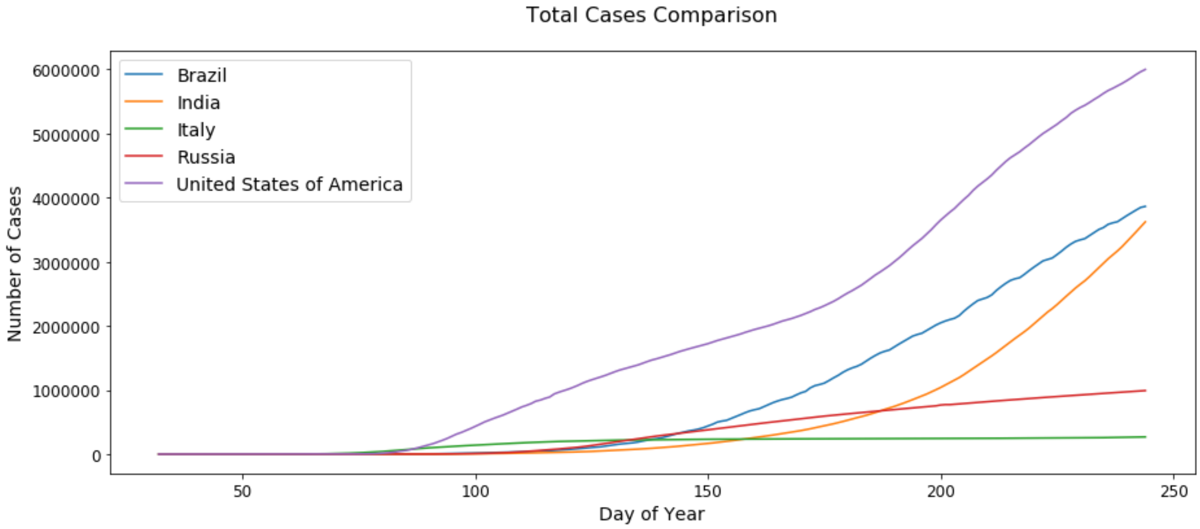
Data Visualisation over Graph

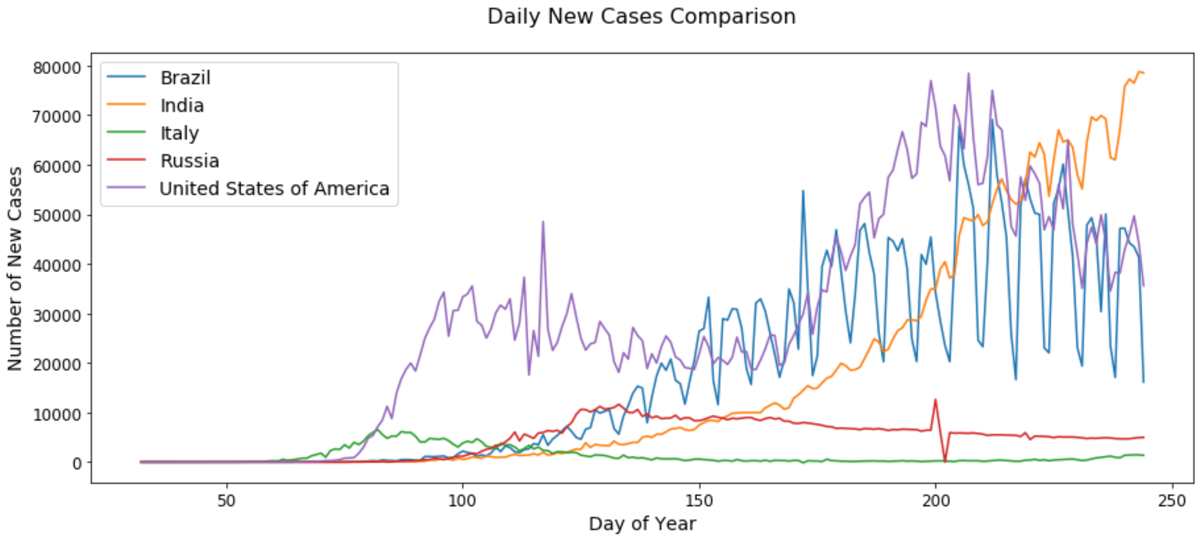
To perform the comparison of various parameters over time, we will be using the line graph for data visualisation. For this visualisation, we will consider few of the top impacted countries like United States, Brazil, India, Italy, Russia.

In this section, I performed the analysis using various columns like total cases, new cases, cases per million, deaths per millions, etc. This will show the trend/growth of cases in different regions. This can show us the which country controlled the growth best and worst, but as per the topic we will focus on the trend of the few selected countries only.

Time-series analysis of Total Cases and New Cases

As we can see, United States is the worst impacted country in terms of total number of cases. The rate of increase for United States is high and for India it started increase near 200th day of the year. United States and Brazil are the countries so far, which is worst impacted with the total number of cases, however as per the rate of increase of cases of India, India will soon cross Brazil in terms of total number of Cases. As of now, India's new cases are increasing at highest rate.

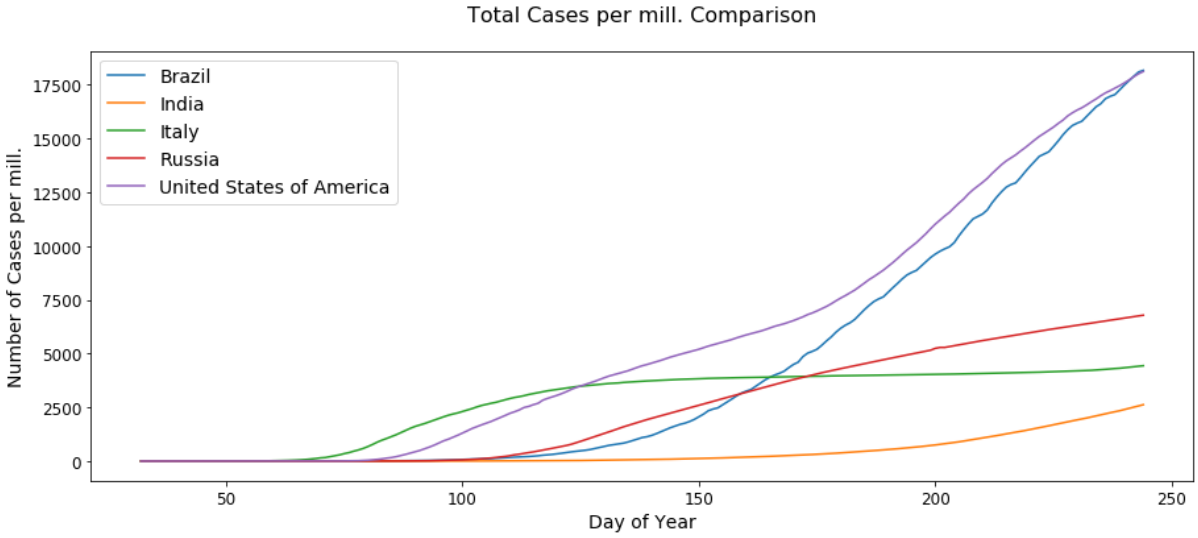


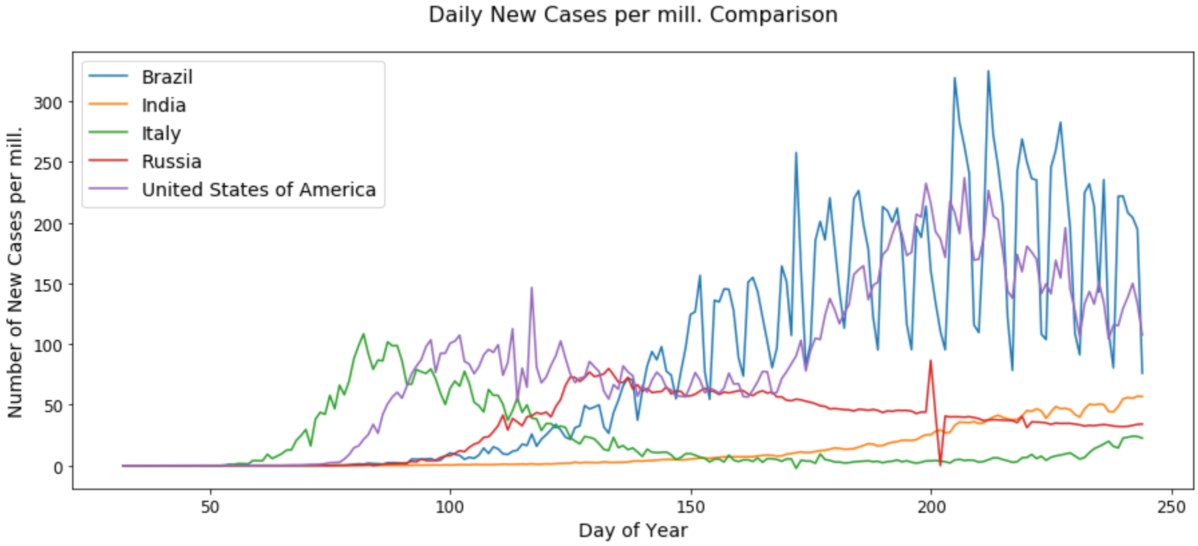


Time-series analysis of Total Cases and New Cases per million of population

As we know countries population plays an important role in the growth/increase of COVID-19 cases as it can be transmitted from one person to other. If we focus on total cases per million of population, we see some different trends for Italy and India.

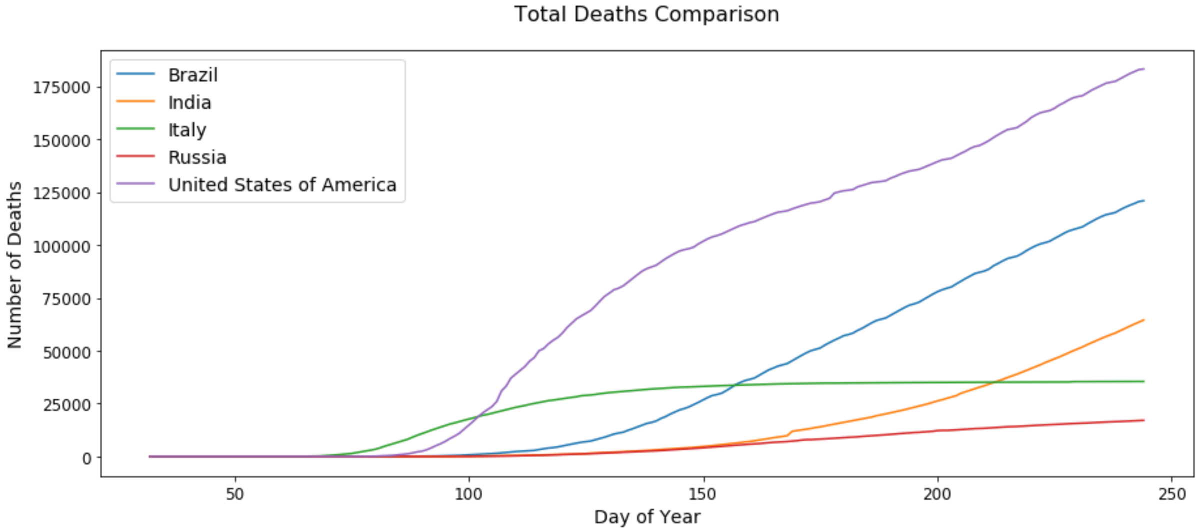
As per the trend, the number of cases in Italy increased suddenly in initial days however rate of increase in cases reduced. However, in India, line is at lowest side as India's population is highest as compared to other mentioned countries, so if the number of cases is on higher side in India as compared to Russia but Total cases per million is on the lower side. But in the terms of new cases per million, India already crosses Russia and Italy. So, as per total cases per million, India is least impacted and US/Brazil are the worst impacted.



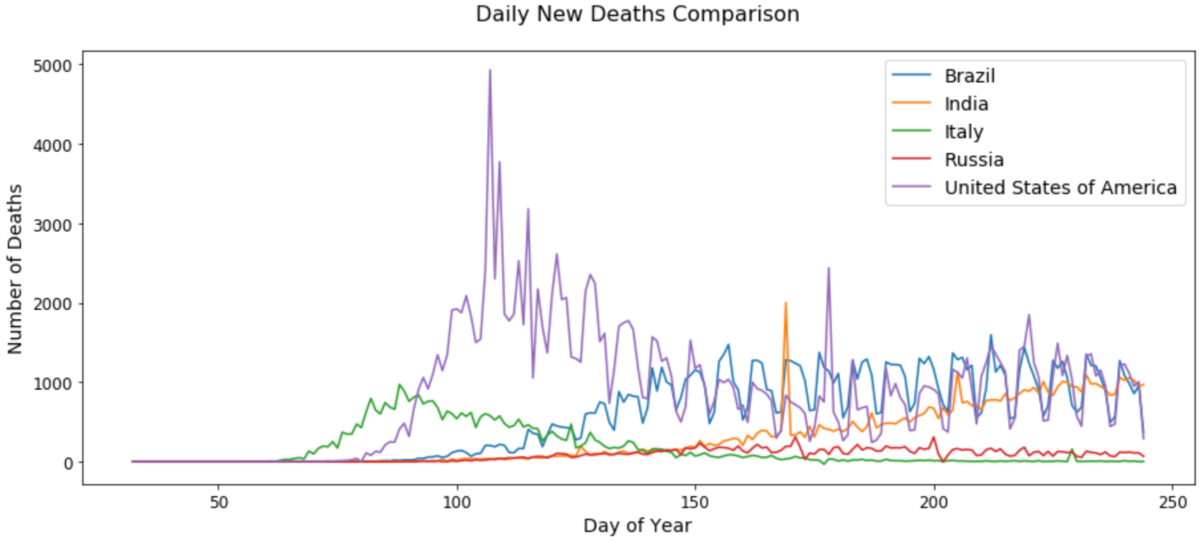


Time-series analysis of Total Deaths and New Deaths

As per trend, Italy was the first to see the death increase and then United States rate of death increased suddenly and US is the worst impacted country in terms of total number of deaths. However, Italy showed the increased initially but controlled the rate of increase later on. Brazil is coming next to United States with almost similar increase trend. However for India, we see delayed growth in rate of deaths.



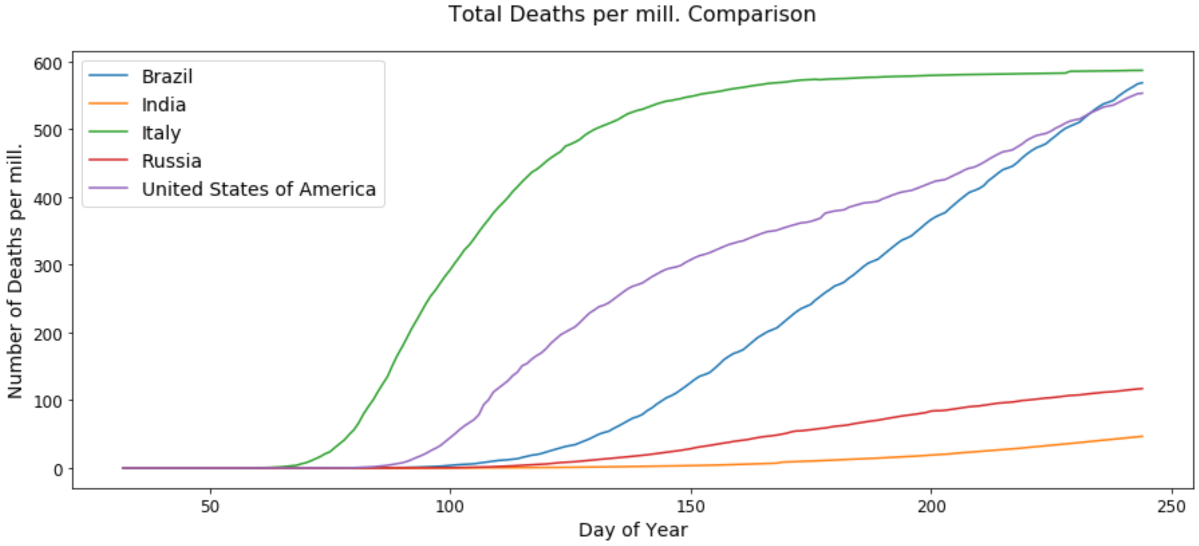
As of now, India is having almost similar number of daily deaths as compared to United States and Brazil.

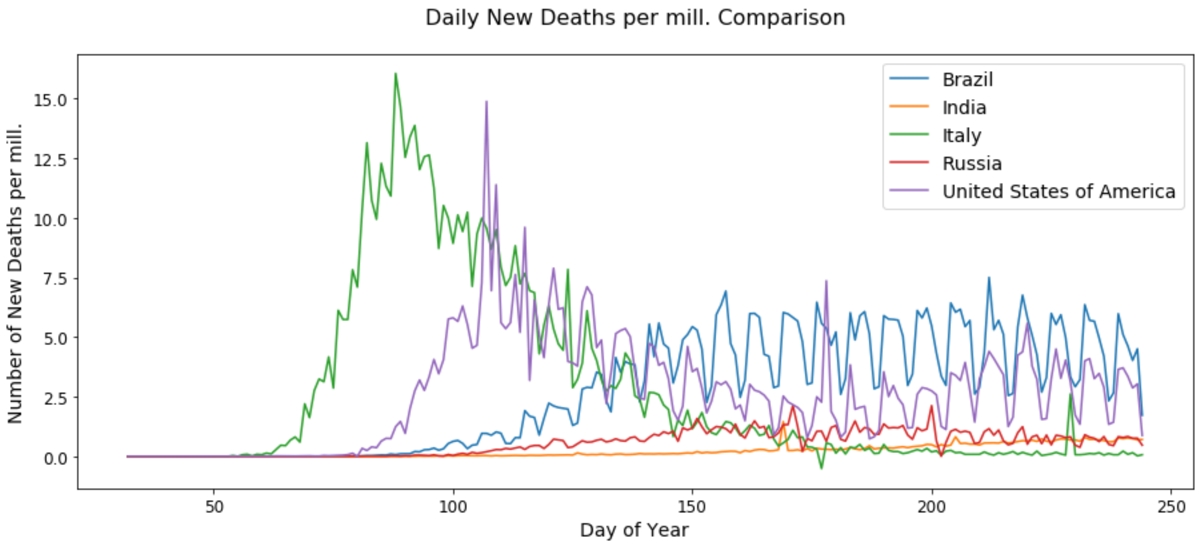


Time-series analysis of Total Deaths and New Deaths per million of population

If we focus on the trend of Total deaths per million of population, we see totally different picture for Italy. As in this case, Italy is the worst impacted country with the deaths per million of the population, so Italy had maximum of population percentage.

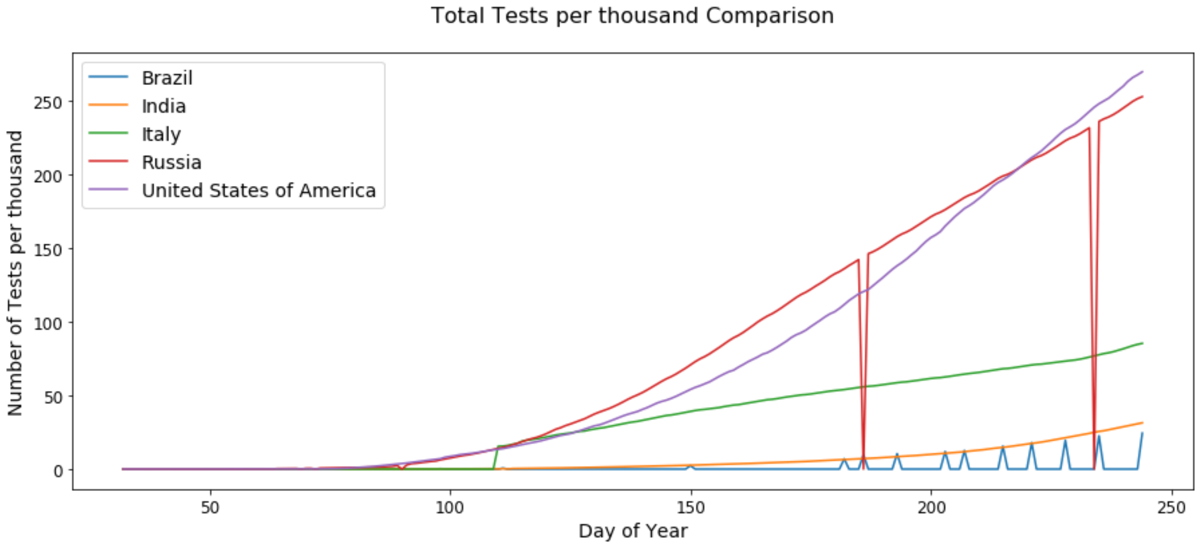
However for India, we see least number of deaths per population of country, as India's population is highest as compared to considered countries.



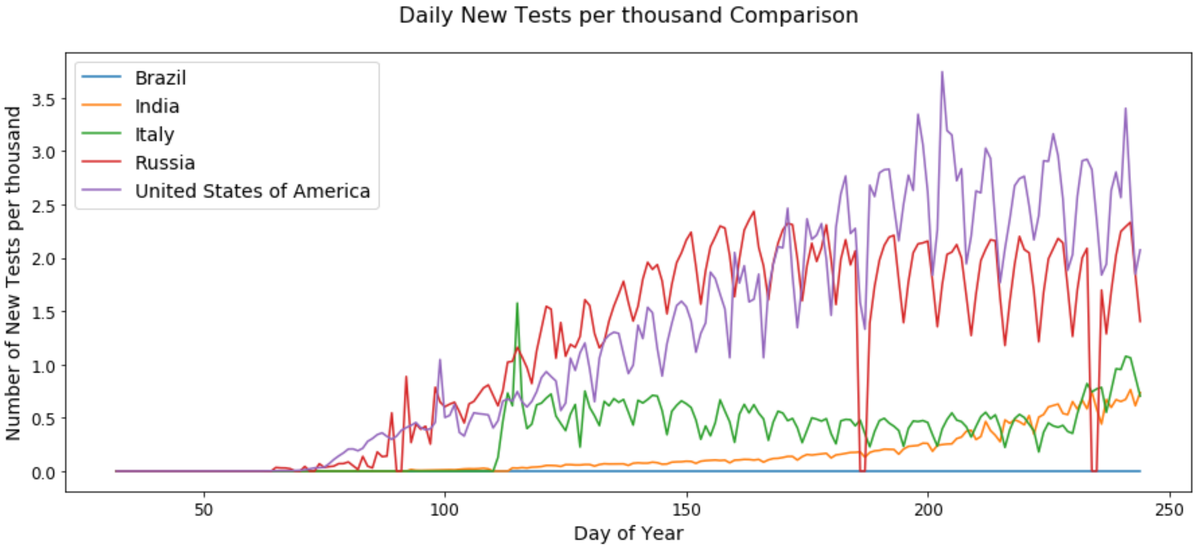


Time-series analysis of Total Tests and New Tests per thousand of population

United States and Russia are the topmost countries with the COVID-19 daily testing. As per the trend, looks like Brazil is not recording the testing done, so we will ignore the Brazil trend for this analysis. However, India is at the lowest in the tests per thousand of population, but this raises a major concern behind the lowered trend of India's cases and deaths.

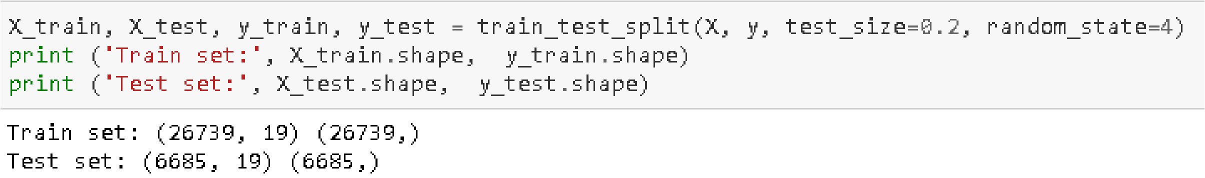


Based on the trend of India, it looks like testing facilities are worse than other countries, so this might be the major reason behind the lesser number of cases as there might be higher number of COVID-19 positive cases which are still hidden.



Machine Learning Models

To predict the deaths, we will now use the various available parameters in the dataset to train the model and test them on the test set. Before we start our training, we will divide the dataset into train and test set with 80-20 ratio.



```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

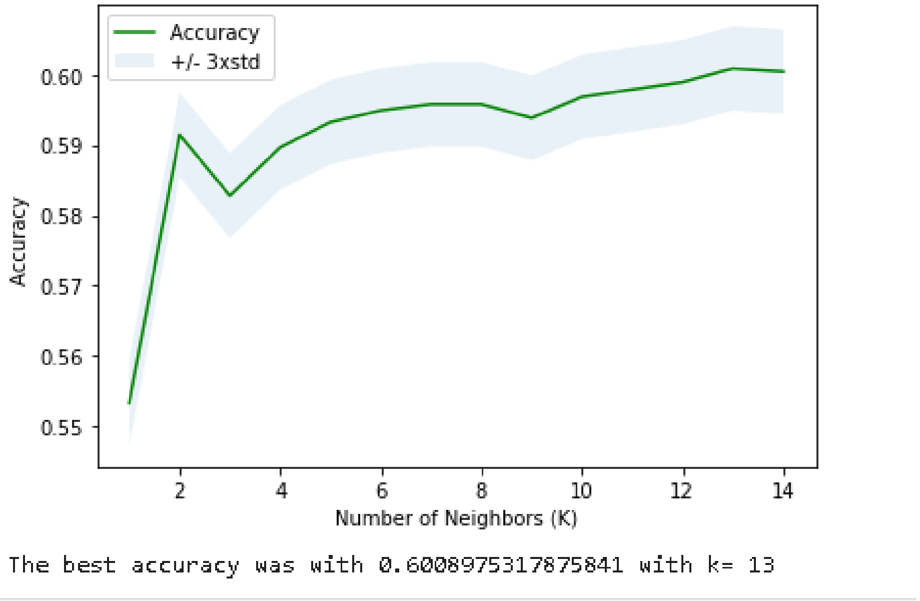
```
Train set: (26739, 19) (26739,)
Test set: (6685, 19) (6685,)
```

We will be using below mentioned Machine Learning models:

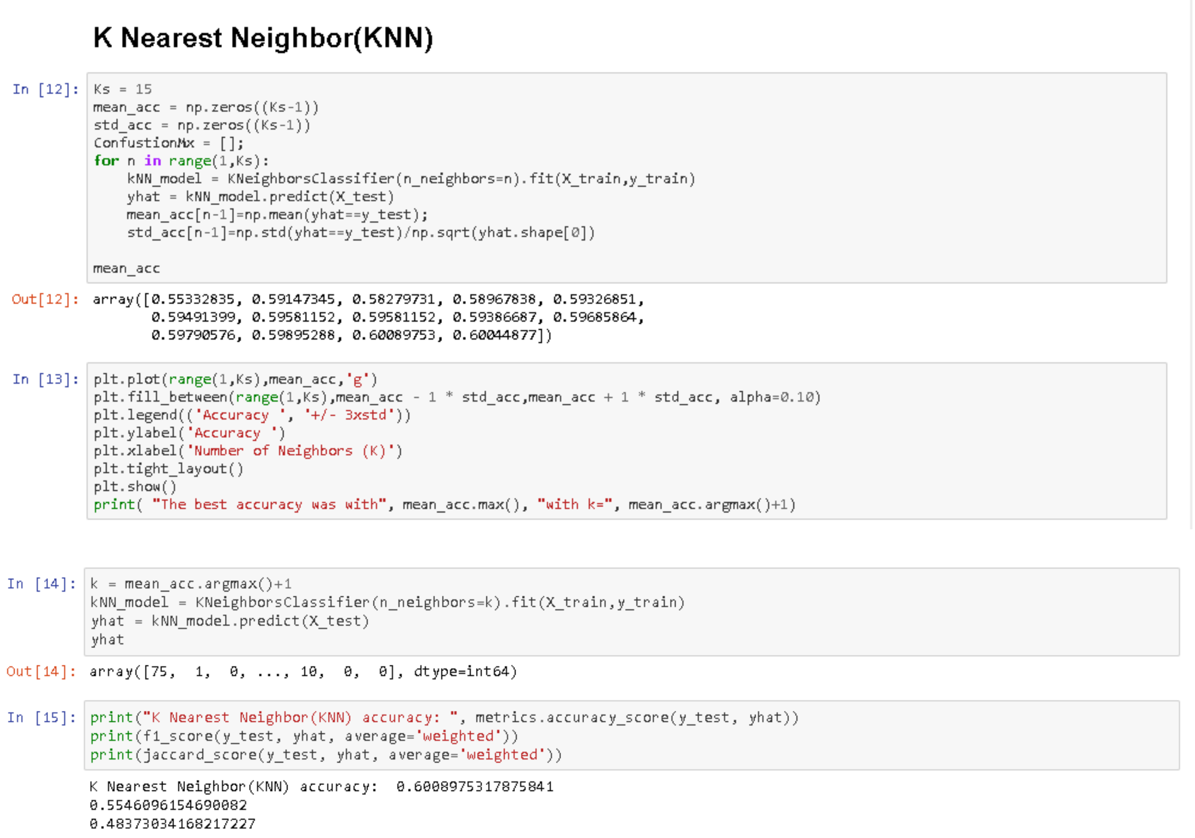
- K Nearest Neighbour (KNN)
- Decision Tree
- Support Vector Machine
- Logistic Regression

K Nearest Neighbour (KNN)

K-NN will help us predict the number of deaths by finding the most similar to point within k distance. k-Nearest Neighbor classifier was used from the scikit-learn library to run the k-Nearest Neighbor machine learning classifier on the COVID-19 data. The best K, as shown below, for the model exists at 13 with accuracy of 0.60.



```
The best accuracy was with 0.6008975317875841 with k= 13
```



K Nearest Neighbor(KNN)

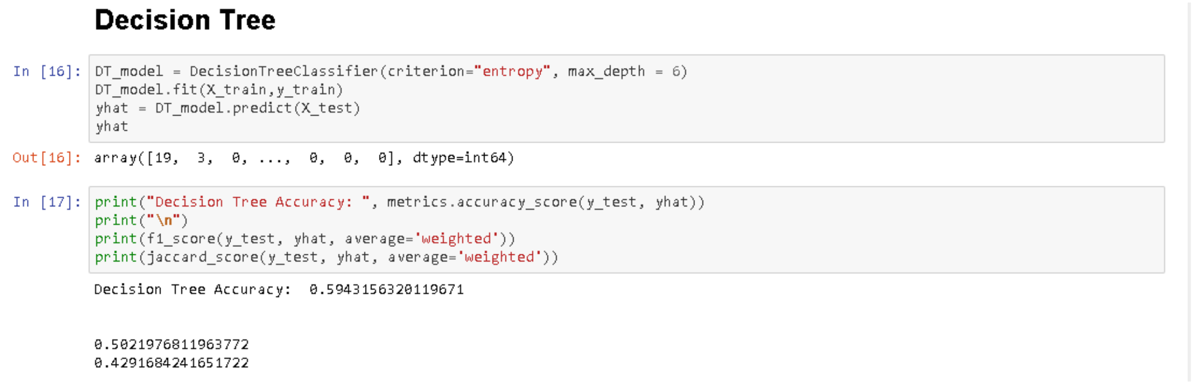
```
In [12]: Ks = 15
         mean_acc = np.zeros((Ks-1))
         std_acc = np.zeros((Ks-1))
         ConfustionMx = [];
         for n in range(1,Ks):
             kNN_model = KNeighborsClassifier(n_neighbors=n).fit(X_train,y_train)
             yhat = kNN_model.predict(X_test)
             mean_acc[n-1]=np.mean(yhat==y_test);
             std_acc[n-1]=np.std(yhat==y_test)/np.sqrt(yhat.shape[0])

         mean_acc

Out[12]: array([0.55332835, 0.59147345, 0.58279731, 0.58967838, 0.59326851,
                0.59491399, 0.59581152, 0.59581152, 0.59386687, 0.59685864,
                0.59790576, 0.59895288, 0.60089753, 0.60044877])
```

```
In [13]: plt.plot(range(1,Ks),mean_acc,'g')
         plt.fill_between(range(1,Ks),mean_acc - 1 * std_acc,mean_acc + 1 * std_acc, alpha=0.10)
         plt.legend(('Accuracy ', '+/- 3xstd'))
         plt.ylabel('Accuracy ')
         plt.xlabel('Number of Neighbors (K)')
         plt.tight_layout()
         plt.show()
         print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
```

```
In [14]: k = mean_acc.argmax()+1
         kNN_model = KNeighborsClassifier(n_neighbors=k).fit(X_train,y_train)
         yhat = kNN_model.predict(X_test)
         yhat

Out[14]: array([75,  1,  0, ..., 10,  0,  0], dtype=int64)

In [15]: print("K Nearest Neighbor(KNN) accuracy: ", metrics.accuracy_score(y_test, yhat))
         print(f1_score(y_test, yhat, average='weighted'))
         print(jaccard_score(y_test, yhat, average='weighted'))

         K Nearest Neighbor(KNN) accuracy:  0.6008975317875841
         0.5546096154690082
         0.48373034168217227
```

Decision Tree

Decision tree is a graph that uses a branching method to illustrate every possible outcome of a decision. Once we train our Decision Tree model using the train and selected fields, we got 0.59 accuracy, 0.5 F1 score and 0.43 as Jaccard score. We used max depth as 6 for this model.

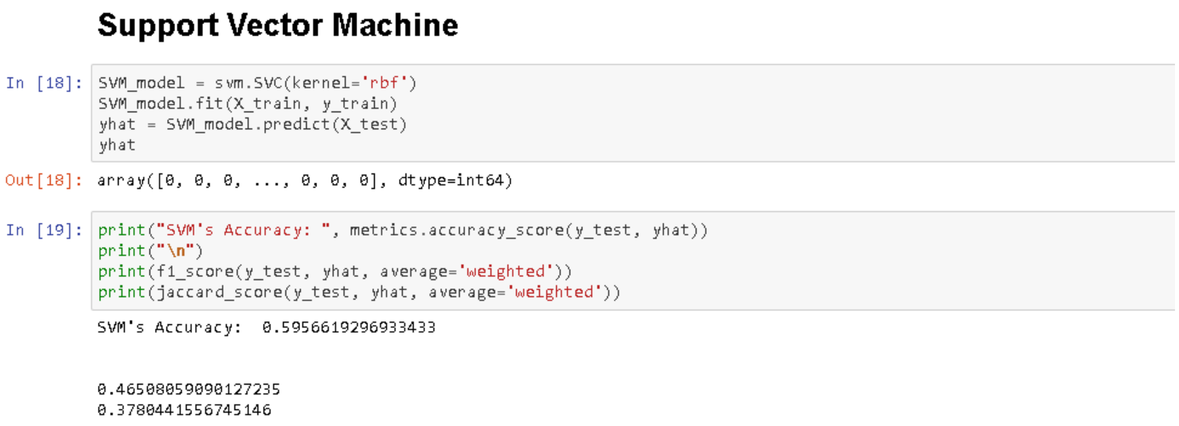


Decision Tree

```
In [16]: DT_model = DecisionTreeClassifier(criterion="entropy", max_depth = 6)
         DT_model.fit(X_train,y_train)
         yhat = DT_model.predict(X_test)
         yhat

Out[16]: array([19,  3,  0, ...,  0,  0,  0], dtype=int64)

In [17]: print("Decision Tree Accuracy: ", metrics.accuracy_score(y_test, yhat))
         print("\n")
         print(f1_score(y_test, yhat, average='weighted'))
         print(jaccard_score(y_test, yhat, average='weighted'))

         Decision Tree Accuracy:  0.5943156320119671


         0.5021976811963772
         0.4291684241651722
```

Support Vector Machine

Support Vector Machine from the scikit-learn library was used to run the SVM model on the COVID-19 data. The kernel used for SVM is ‘rbf’. Once we train our Support Vector Machine model using the train and selected fields, we got 0.59 accuracy, 0.47 F1 score and 0.38 as Jaccard score.

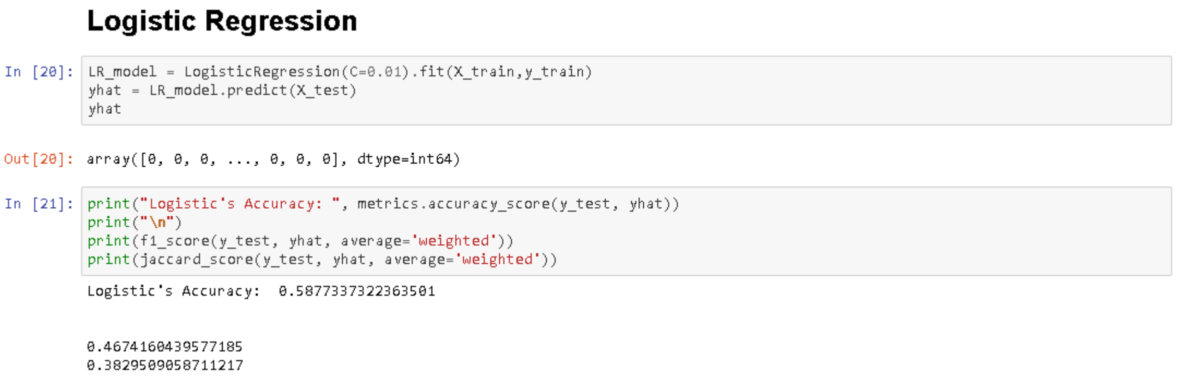


Support Vector Machine

```
In [18]: SVM_model = svm.SVC(kernel='rbf')
         SVM_model.fit(X_train, y_train)
         yhat = SVM_model.predict(X_test)
         yhat

Out[18]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [19]: print("SVM's Accuracy: ", metrics.accuracy_score(y_test, yhat))
         print("\n")
         print(f1_score(y_test, yhat, average='weighted'))
         print(jaccard_score(y_test, yhat, average='weighted'))

         SVM's Accuracy:  0.5956619296933433


         0.46508059090127235
         0.3780441556745146
```

Logistic Regression

Logistic Regression from the scikit-learn library was used to run the Logistic Regression model on the COVID-19 data. The C used for regularization strength was ‘0.01’. Once we train our Logistic Regression model using the train and selected fields, we got 0.58 accuracy, 0.47 F1 score and 0.38 as Jaccard score.



Logistic Regression

```
In [20]: LR_model = LogisticRegression(C=0.01).fit(X_train,y_train)
         yhat = LR_model.predict(X_test)
         yhat

Out[20]: array([0, 0, 0, ..., 0, 0, 0], dtype=int64)

In [21]: print("Logistic's Accuracy: ", metrics.accuracy_score(y_test, yhat))
         print("\n")
         print(f1_score(y_test, yhat, average='weighted'))
         print(jaccard_score(y_test, yhat, average='weighted'))

         Logistic's Accuracy:  0.5877337322363501


         0.4674160439577185
         0.3829509058711217
```

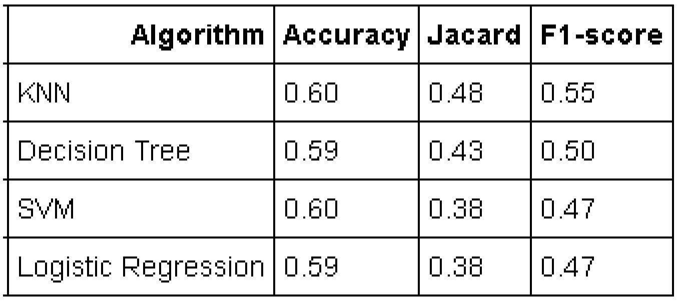
Results

To find the best model to predict the number of death cases, we first selected few columns like new cases, country's health condition, population density, number of smokers, etc. Then we performed the cleaning of data and standardised it for the visualisation and model training.

During visualisation, we came up with various trends in which we found that number of cases and deaths were initially high in Italy, however now its high for US and Brazil. However, India is in position of delayed increase in cases. There are high chance of hidden cases in India as the number of tests per thousand of population in India is very low.

During model training and testing, KNN model and SVM model worked best in terms of accuracy. However, Jaccard score is best for KNN only and same in the case of F1-score, KNN is having 0.55 score.

KNN model is the best suited model to predict the COVID-19 death counts with accuracy of 0.6, Jaccard score of 0.48 and F1-score as 0.55.



Algorithm	Accuracy	Jacard	F1-score
KNN	0.60	0.48	0.55
Decision Tree	0.59	0.43	0.50
SVM	0.60	0.38	0.47
Logistic Regression	0.59	0.38	0.47

Discussion

As per the analysis and visualisation of data, there are few parameters which are missing for few countries like for Brazil, the daily test cases data is not available. Models can be trained well and better picture will come out if the data is records for all the scenarios.

However, based on the available data, its clear that Italy faced the early increased trend of increase in number of deaths. After Italy, US and Brazil death count increased and India comes in third phase where we can see delayed increasing trend in death count. As of now, India is at the highest new deaths per day.

All these increased trends can be slowed down by providing the proper health-care facilities that we observed in Italy and better facility in US/ Brazil.

These models could have performed better if a few more things were present and possible.

- A balanced dataset for the target variable
- More number of records for countries like North Korea, China where data collection is not precisely done.
- Less missing values within the dataset for variables such as few countries do not provide new case count on daily basis.

Conclusion

Predictions performed using developed models worked with accuracy of 0.6 in case of k-nearest neighbors (k-NN) and Support Vector Machine (SVM) model. However, to predict the number of deaths due to COVID-19 we require more accurate parameter with less missed values and frequently updated data.

However, in countries like India, more COVID-19 tests should be performed in order to get a better picture of number of COVID-19 positive cases and data should be recorded properly in countries like North Korea, China.

All this can be done in better way with the deepening of the understanding of COVID-19 by governments and scientific research institutions in various countries, improving medical methods or implementing stricter control policies will greatly affect the changes in the number of infected people and this will directly impact the number of deaths also. It is a more sensible study to classify the trend of the number of infected people under the influence of different factors, and then modify the policy according to the trend.