Data 100 Final Project: Modeling and Predicting the Spread of COVID-19 in the United States

Robin (Jae-Hee) Yoo, Judy Moon, Jae Hyun Kim

I. Introduction and Abstract

Less than two months ago, the United States had less than 1,000 confirmed cases of COVID-19. Now, the epicenter of the pandemic is the United States, which has more cases than any country in the world with more than 1.4 million confirmed cases. Given the drastically rising confirmed cases and death tolls across one of the biggest nations in the world, we hope to address some of the questions that have been lingering around us for the past two months.

Thus, in this final project, we will investigate the spread of COVID-19 in the United States from January of 2020. We hope to answer the following two questions: does the spread of COVID-19 display different trends across various regions of the United States? How can we forecast the epidemiological outbreak of COVID-19 within the United States by predicting the number of confirmed cases using regression techniques? Through this project, we hope to demonstrate our abilities to work with data at different levels of granularities, identify the type of data collected (including missing values, anomalies, etc), and explore characteristics and distributions of individual variables through exploratory data analysis, data cleaning, and provide multiple visualizations to model our given data. We will be using scikit-learn as a tool for our predictive analysis, and we will aim to create a simple Linear Regression model to be able to forecast the number of new confirmed COVID-19 cases for the near future.

II. Description of the Data

The datasets we chose for the analysis are 4 csv files: 4.18states.csv, time_series_covid19_confirmed_US.csv, time_series_covid19_deaths_US.csv, and abridged_couties.csv. However, to get more accurate and updated data, we have decided to use the updated daily report and time series data sets from JHU CSSE COVID-19 Dataset in the Github (https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data). Therefore, the data we used in this project contains the most recently updated data until May 8th 2020.

III. Exploratory Data Analysis and Data Visualizations

A. Geospatial Hexbin Plot Visualization

Our first method was to use a geospatial hexbin plot to identify the factors that can be beneficial in predicting the number of confirmed COVID-19 cases throughout different regions of the United States. From this method, we hoped to find the relationship between specific regions across the United States and the rate of COVID-19 confirmed cases and death numbers.

To start, we used the following code to modify data from the confirmed dataframe and the death dataframe. The purpose of this cell was to identify the accumulated values of the confirmed cases and the deaths from different US counties. We then merged new confirmed and death dataframe to create a dataframe called **summary** to combine Total_deaths and Total_confirmed columns into one single dataframe. The **summary** dataframe dropped the irrelevant columns like iso2, iso3, and code3, County_Region, and renamed some of the columns that might give confusion. Then, we counted the values of each.

```
summary.head(10)
```

	UID	FIPS	CountyName	Province_State	Lat	Long	Combined_Key	Population	Total_deaths	Total_confirmed
0	16	60.0	NaN	American Samoa	-14.271000	-170.132000	American Samoa, US	55641	0	0
1	316	66.0	NaN	Guam	13.444300	144.793700	Guam, US	164229	5	149
2	580	69.0	NaN	Northern Mariana Islands	15.097900	145.673900	Northern Mariana Islands, US	55144	2	15
3	630	72.0	NaN	Puerto Rico	18.220800	-66.590100	Puerto Rico, US	2933408	102	2031
4	850	78.0	NaN	Virgin Islands	18.335800	-64.896300	Virgin Islands, US	107268	4	66
5	84001001	1001.0	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	55869	3	61
6	84001003	1003.0	Baldwin	Alabama	30.727750	-87.722071	Baldwin, Alabama, US	223234	5	205
7	84001005	1005.0	Barbour	Alabama	31.868263	-85.387129	Barbour, Alabama, US	24686	1	51
8	84001007	1007.0	Bibb	Alabama	32.996421	-87.125115	Bibb, Alabama, US	22394	0	44
9	84001009	1009.0	Blount	Alabama	33.982109	-86.567906	Blount, Alabama, US	57826	0	44

In the **summary** data, each county had a UID (or county FIPS) with FIPS(or state FIPS). If UID is 84001001 and FIPS is 1001, then this represents Autauga County, Alabama. However, UID and FIPS here have extra digits compared to the **counties** data set. So we cleaned our data sets by converting to string and renamed UID as County FIPS and FIPS as State FIPS.

To merge **summary** and **countries**, we renamed the FIPS values of **counties** and checked the County FIPS.unique() to find the unnecessary rows. Then we left-merged **counties** (including State FIPS, County Name, Province State, Census Division Name, Rural-Urban Continuum Code of 2013) and **summary** based on CountyName, State FIPS, and Province_State.

```
[ ] summary=summary.merge(counties[['State FIPS','CountyName','Province_State','CensusDivisionName','Rural-UrbanContinuumCode2013']],
              on = ['State FIPS','CountyName','Province_State'], how='left')
    summary.head(6)
```

	County FIPS	State FIPS	CountyName	Province_State	Lat	Long	Combined_Key	Population	Total_deaths	Total_confirmed	CensusDivisionName	Rural-UrbanContinuumCode2013
0	16.0	60.0	NaN	American Samoa	-14.271000	-170.132000	American Samoa, US	55641	0	0	NaN	NaN
1	316.0	66.0	NaN	Guam	13.444300	144.793700	Guam, US	164229	5	149	NaN	NaN
2	580.0	69.0	NaN	Northern Mariana Islands	15.097900	145.673900	Northern Mariana Islands, US	55144	2	15	NaN	NaN
3	630.0	72.0	NaN	Puerto Rico	18.220800	-66.590100	Puerto Rico, US	2933408	102	2031	NaN	NaN
4	850.0	78.0	NaN	Virgin Islands	18.335800	-64.896300	Virgin Islands, US	107268	4	66	NaN	NaN
5	1001.0	1.0	Autauga	Alabama	32.539527	-86.644082	Autauga, Alabama, US	55869	3	61	East South Central	2.0

```
summary.CountyName.value_counts().index.tolist
```

```
<bound method IndexOpsMixin.tolist of Index(['Unassigned', 'Washington', 'Jefferson', 'Franklin', 'Lincoln',
       'Jackson', 'Madison', 'Montgomery', 'Clay', 'Union',
       ...
       'Out of ID', 'Cape May', 'Kittson', 'Juab', 'Wasco', 'Ionia', 'Spokane',
       'Upton', 'Riverside', 'Kit Carson'],
      dtype='object', length=1901)>
```

From this list, we noticed ‘Unassigned’, ‘Out of (state name)’, and NaN values in the CountyName column. These were invalid country names that were likely used to code invalid values. In the following table, we constructed a pivot table that counts the number of states that have invalid county names. Here, each state had ‘Unassigned’ and ‘Out of (state name)’ and only some Islands have the NaN value in the CountyName column.

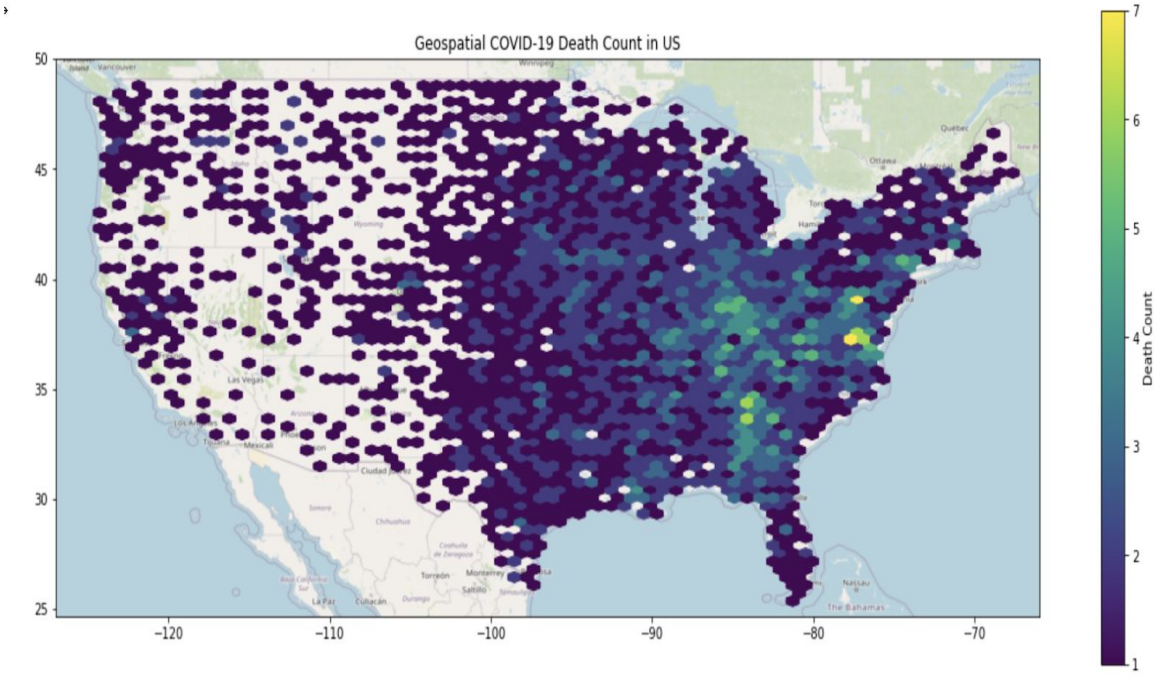
Missing County	False	True	Total
Province_State			
Texas	254	2	256.0
Georgia	159	2	161.0
Virginia	133	2	135.0
Kentucky	120	2	122.0
Missouri	116	2	118.0
Kansas	105	2	107.0
Illinois	102	2	104.0
North Carolina	100	2	102.0
Iowa	99	2	101.0
Tennessee	95	2	97.0
Nebraska	93	2	95.0
Indiana	92	2	94.0
Ohio	88	2	90.0
Minnesota	87	2	89.0
Michigan	85	2	87.0
Mississippi	82	2	84.0
Oklahoma	77	2	79.0
Arkansas	75	2	77.0
Wisconsin	72	2	74.0
Pennsylvania	67	2	69.0
Alabama	67	2	69.0
Alabama	67	2	69.0
Florida	67	2	69.0
South Dakota	66	2	68.0
Louisiana	64	2	66.0
Colorado	64	2	66.0
New York	62	2	64.0
California	58	2	60.0
Montana	56	2	58.0
West Virginia	55	2	57.0
North Dakota	53	2	55.0
South Carolina	46	2	48.0
Idaho	44	2	46.0
Washington	39	2	41.0
Oregon	36	2	38.0
Utah	35	2	37.0
New Mexico	33	2	35.0
Alaska	29	2	31.0
Maryland	24	2	26.0
Wyoming	23	2	25.0
New Jersey	21	2	23.0
Nevada	17	2	19.0
Maine	16	2	18.0
Arizona	15	2	17.0
Massachusetts	15	2	17.0
Vermont	14	2	16.0
New Hampshire	10	2	12.0
Connecticut	8	2	10.0
Rhode Island	5	2	7.0
Hawaii	5	2	7.0
Delaware	3	2	5.0
District of Columbia	1	2	3.0
American Samoa	0	0	NaN
Diamond Princess	0	0	NaN
Grand Princess	0	0	NaN
Guam	0	0	NaN
Northern Mariana Islands	0	0	NaN
Puerto Rico	0	0	NaN
Virgin Islands	0	0	NaN

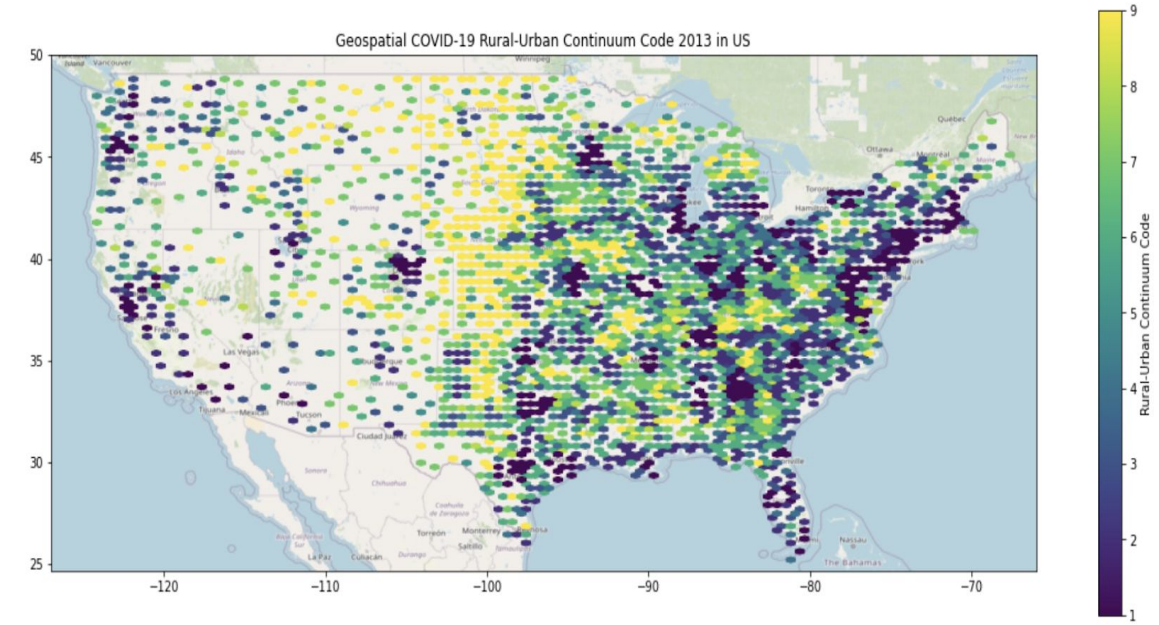
These invalid values ('Out of state' and 'Unassigned') have zero latitude and longitude, or no population size. With this in mind, we will now check the **valid** values by creating the Geospatial Hexbin Plot, which will visualize the total death count (using the Total_death value) for states within the mainland of the United States. First, we assigned the 'Missing County' value by checking if the 'Lat', 'Long', and 'Population' values were zero or not. This will help to create the **valid** values.

```
Inaproporiate =summary[summary['Missing County']==True]
Inaproporiate.head()
```

	County FIPS	State FIPS	CountyName	Province_State	Lat	Long	Combined_Key	Population	T
3147	70002.0	NaN	Dukes and Nantucket	Massachusetts	41.406747	-70.687635	Dukes and Nantucket,Massachusetts,US	0	
3149	80001.0	80.0	Out of AL	Alabama	0.000000	0.000000	Out of AL, Alabama, US	0	
3150	80002.0	80.0	Out of AK	Alaska	0.000000	0.000000	Out of AK, Alaska, US	0	
3151	80004.0	80.0	Out of AZ	Arizona	0.000000	0.000000	Out of AZ, Arizona, US	0	
3152	80005.0	80.0	Out of AR	Arkansas	0.000000	0.000000	Out of AR, Arkansas, US	0	

Then, based on the 'Missing County', we sorted only the 'Missing County' == False values (no Island and odd data sets). Since we were looking for the mainland of the United States Geospatial Hexbin, we set the boundary of the latitude and longitude based on Wikipedia (top = 50, bottom =24.7, left = -127, right=-66). Then created the hexbin using the plt.hexbin based on the 'Total_deaths' and 'Rural-Urban Continuum Code' values sizes. Here, the 'Total_deaths' is based on the size and 'Rural-Urban Continuum Code' is based on the mean.



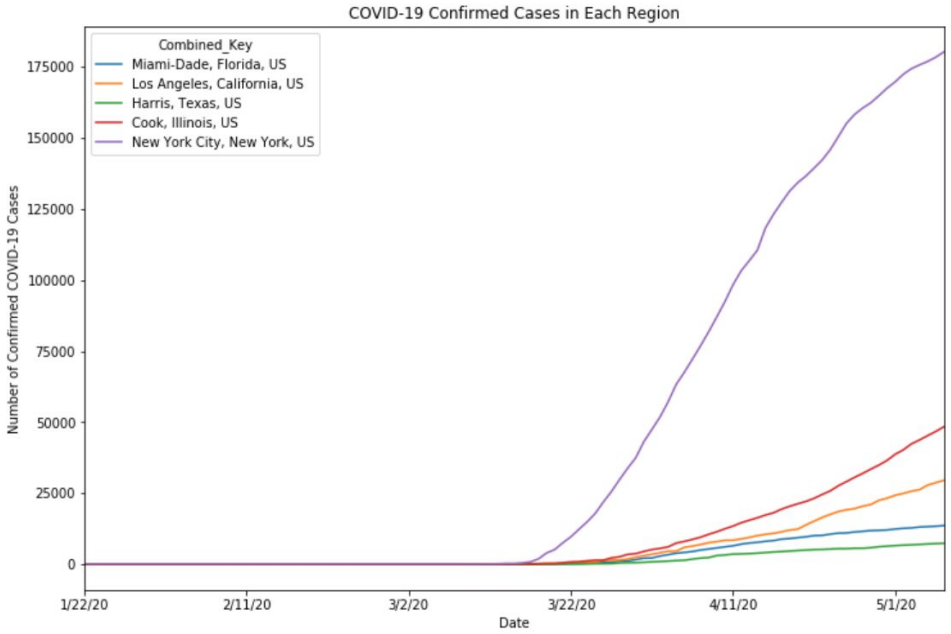


For the death count, most deaths tend to be more frequent in the East coast, and less heavily concentrated in the Midwestern states. States that seem to have the most densely concentrated death are New York, New Jersey, Virginia, Louisiana, and Rhode Island. Here we found interesting facts that about 33 states have at least one county with no death and confirmed COVID-19. For the Rural-Urban Continuum Code, the countries in the metro area with more population if the code is higher and the countries are urbanized if the code is lower. Compared with the death count hexbin, we figure out that more urbanized areas (where not adjacent to a metro area) have large populations of total death due to COVID-19.

B. Line Plot Visualization

For our second visualization, we wanted to compare the number of confirmed COVID-19 cases across a time span from late January to early May of 2020 in different regions of the United States. We wanted to compare the trend of coronavirus cases across different regions of the country, and how the pattern of the increase in the number of confirmed cases is similar or different based on location. We decided to use the most recently updated data from the COVID-19 Data Repository by the CSSE at Johns Hopkins University to plot a line graph of the number of confirmed COVID-19 cases for five chosen states across the span of four consecutive months. We divided up the five regions based on relative location across the mainland of the United States. We chose the five counties based on population -- after doing some research, we found out that **Miami-Dade, Los Angeles, Harris, Cook, and New York City** were the most populous counties in each state. The states were chosen from the West coast, East coast, Gulf Coast, Midwest, and Rocky Mountain regions of the US. We also dropped columns that were not relevant to plotting our graph.

After selecting the five regions we want to compare, we then composed a line plot graph comparing the number of confirmed COVID-19 cases from 01/22/2020 to 05/04/2020 amongst the five counties. The resulting graph is shown below:



Upon observing the line plot we made above, we can assume that the number of cases for the New York county was much higher than any other state, followed by Cook, Los Angeles, Miami-Dade, and Harris. It is surprising to see how exponentially fast the graph rises from late Mid-Late March and steadily increases with the same exponential rate, even till the most recent date, May 4th. Compared to the strikingly high numbers reached in New York, the number of COVID-19 cases seem to be rising relatively at a much slower rate in other regions of the U.S. We can assume that most of the confirmed cases so far have largely been densely populated in the North Eastern regions of the country and much less amongst the Gulf coast states.

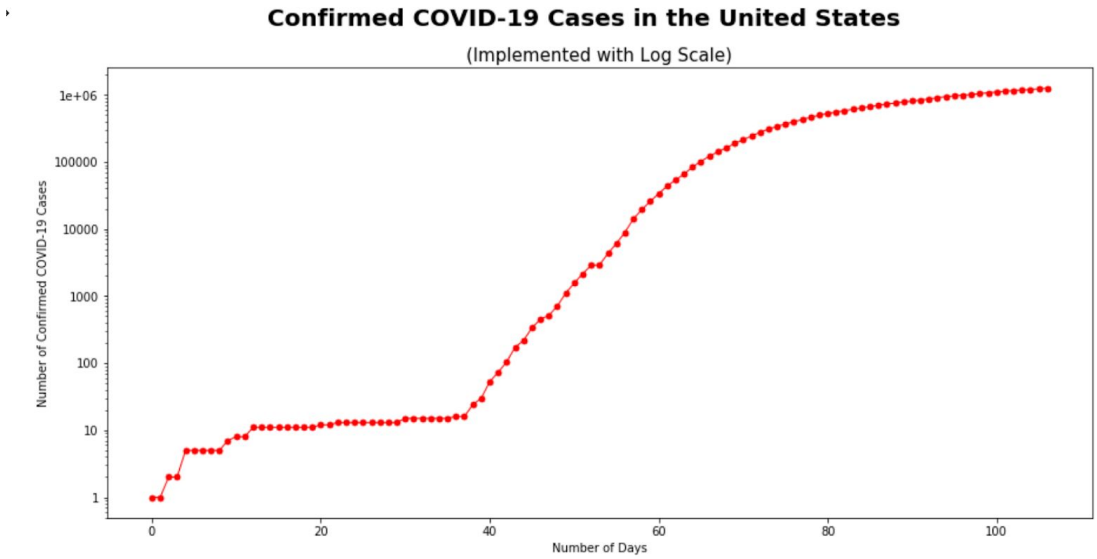
IV. Prediction Methods -- Linear Regression

For the next part of our final project, we wanted to predict the trend of COVID-19 confirmed cases within the U.S. for the next 30 days by using the current trends given the dataset of confirmed cases. We are using a linear regression model to describe the relationship between the number of COVID-19 confirmed cases

and the days passed since the first recorded confirmed case back in January. As a brief summary of what we would be doing, we created an instance of the model, fit the model by passing it the X and Y data, made some predictions and saved them back to the original data frame, then visualized the model as before.

We first decided to use the most recently updated csv file given from the JHU CSSE to construct our prediction model. After loading the **updated_time_series_covid19_confirmed_US** csv file, we converted our new dataframe into a long format where we could separate the dates in chronological order of each index. Then, we dropped unnecessary columns that we were interested in analyzing. Because we were interested in predicting the increase of confirmed COVID-19 cases within the region of the United States, we aggregated our values by date. We then wanted to check if our data contained any missing values. We wanted to replace any NaN values with appropriate filler values, then finally checking that there are no missing values in our new dataset. We then dropped the latitude and longitude columns, as we are not interested in exploring the effects of latitude/longitude with the prediction of the increase of confirmed cases throughout the nation.

Next, we wanted to create a list for the number of days that has passed since the first recording of confirmed COVID-19 cases, which was January 22, 2020. We knew from above that the **confirmed_US_date** table has 106 rows, thus we saw that our code outputted 105 days from 01/22/2020 (the code is outlined in our jupyter notebook). We then wanted to produce a line plot displaying confirmed cases with our finalized dataframe above. We implemented a logarithmic scale to produce a more linear shape with our plot.

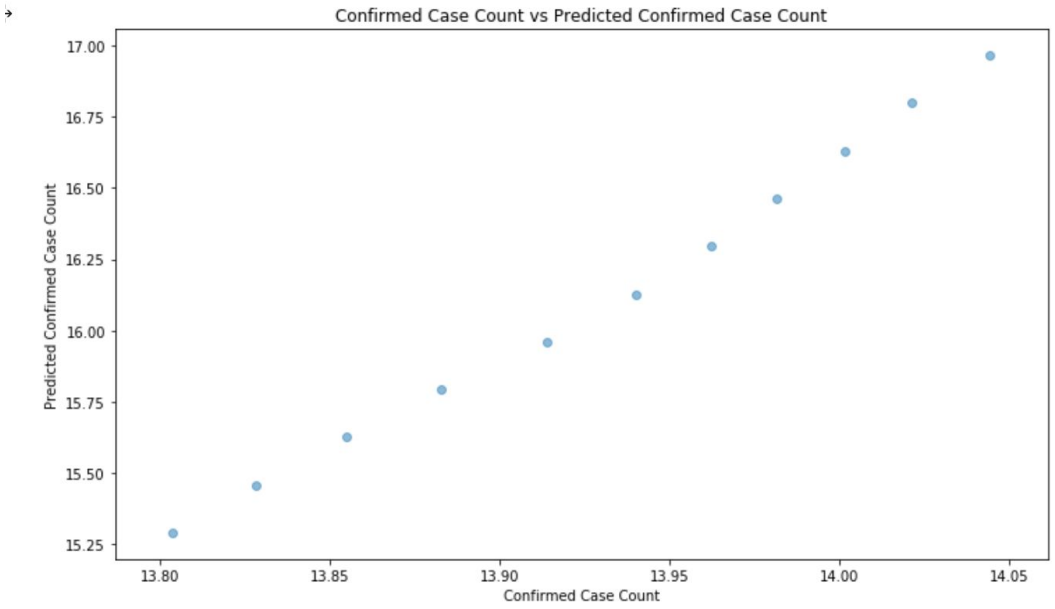


As we can see here in the plot above, the number of confirmed COVID-19 cases remained relatively stagnant for the first ~35 days since January 22nd. The number of cases exponentially increased from the 40th day mark, then steadily increased with a steep slope until the 80th day mark since the first recorded confirmed case, then plateaued early May. Note that our x-axis is labeled “Number of Days”, but this implies that we are using the number of days since the first recorded confirmed case, which was January 22nd, 2020.

In order to proceed with our predictive analysis and implementation of a linear regression model, we first made a list of the number of confirmed COVID-19 cases from our **us_confirmed_data**, using the **tolist()** function. Then, we defined the feature we want to use in order to predict the number of COVID-19 cases in the upcoming weeks and months. We will be defining X, our feature, to days (105 days). Because we implemented a log scale on the y-values of our initial data visualization, we implemented the same logarithmic scale onto

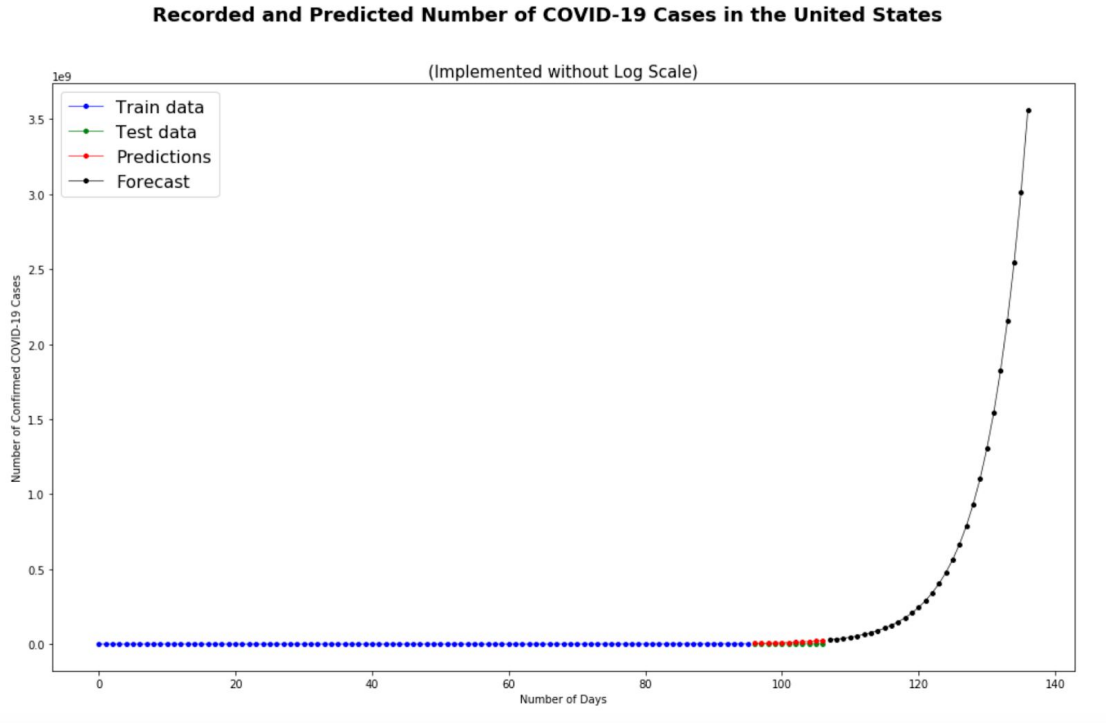
the list of confirmed COVID-19 cases that we made above. The code for these three steps are provided in our ipynb file.

The training data we had up to this point was all the data we had available for both training models and testing the models that we wanted to train. We therefore needed to split the training data into separate training and testing datasets. We utilized the **train_test_split** function to split out 10% of the data for testing. We then fit our linear model with our training data to derive our prediction data. We then decided to create a scatter plot for our predictions vs. the true number of COVID-19 cases to roughly visualize the accuracy of our model.



In order to check our model's error, we computed the root mean squared error (RMSE) of our predicted confirmed cases. After computing the RMSE for our predictions on both the training data X_train and the test set X_test, we got a Training Error: 1.176 and Test Error: 2.243. Our training error is lower than the test error. This could be happening from "overfitting" as the model is overfit to the training data, thus explaining why it has a higher root mean square error (the test data). We then used our trained linear regression model to forecast the number of confirmed COVID-19 cases in the near future. We calculated the coefficient and intercept of our linear model by running **linear_model.coef_** and **linear_model.intercept_**, then predicted the y values (confirmed cases) using our linear regression model.

As shown above, we visualized our predictions for the number of confirmed cases in the US for the next 30 days. Going back to the line plot we made above, we implemented a logarithmic scale onto our y-values. Now, because we wanted to know the actual number of confirmed cases (without the scale), we needed to transform our train and test data back to the original scale. After changing the scale from log by calculating **e^x** for each value of x in our input array, we created a line plot displaying the recorded and predicted number of COVID-19 cases in the United States using our train data, test data, predictions, and forecasted number of cases.

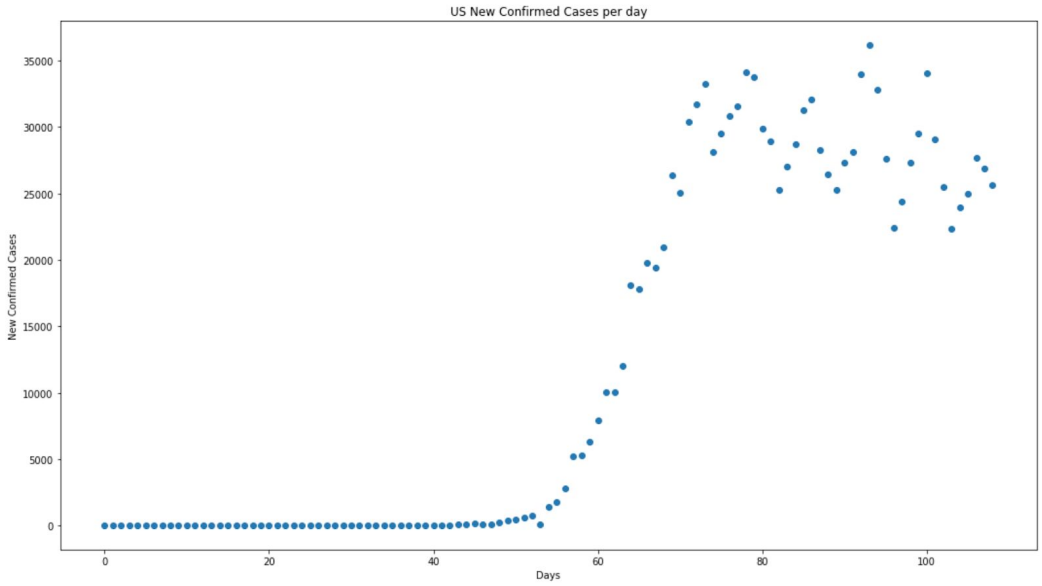


As seen in the graph above, we can see that after around 120 days of the first recorded confirmed case (01/22/2020), we expect to see the number of COVID-19 cases surpass one million, and in around 140 days, we would expect to see that number surge past the 2 million mark. Based on this regression model, we can expect to see the situation in the United States get progressively worse. Looking at the rapidly exponential rate of the spread of the virus across the country, we do not expect the numbers of confirmed cases to flatten anytime soon.

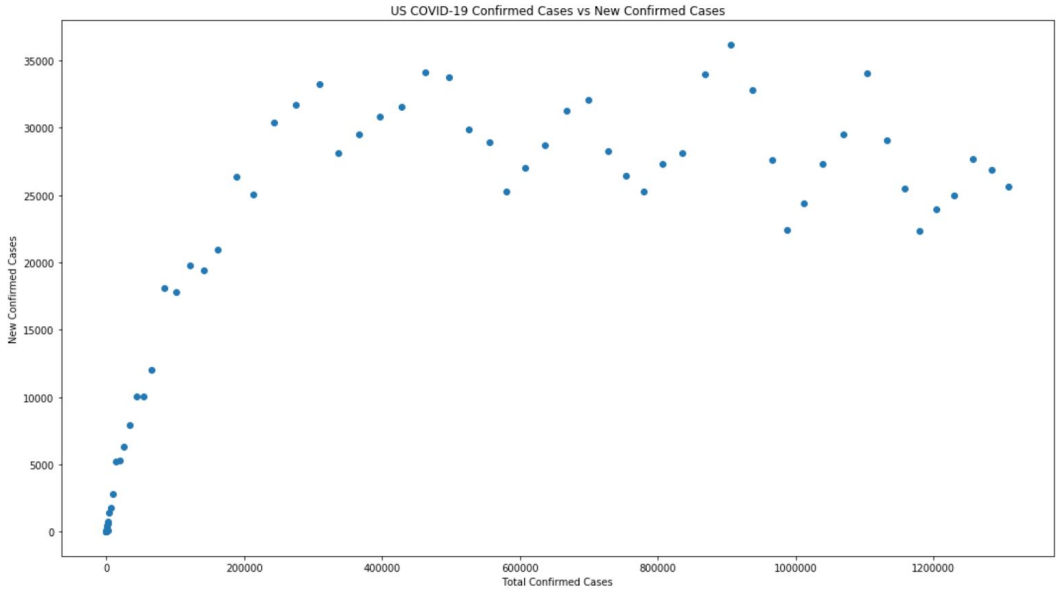
V. Cross Validation and Overfitting

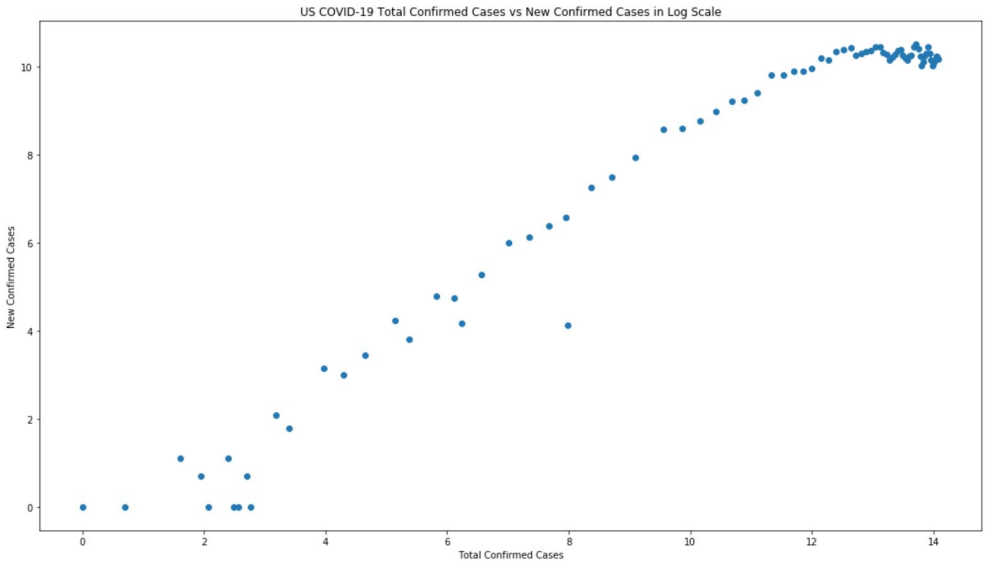
As we pointed out before, our training error is lower than the test error. To avoid "overfitting" and to increase our model's accuracy by adding more features to our model. Moreover, just by looking at the cumulative confirmed cases in the data, it is apparent that it is hard to predict the future since the predicted total confirmed case is growing exponentially. So it is now better to use the new daily cases in the model to see the rate of growth. In addition to the daily confirmed cases, we added one more feature, the daily death cases, in the model to increase the accuracy.

Since we created the plot that reflects the total confirmed cases as time passes, we wanted to see the trend of the daily confirmed cases. Before we started, we calculated the daily confirmed cases from the confirmed dataframe which only has cumulative confirmed cases. Then, we could get the values of the daily confirmed cases to see the rate of the growth of the confirmed cases. The code that we used to accomplish this is shown in our ipynb file. Then we plotted the scatter plot that reflects the rate of the growth of the confirmed cases.



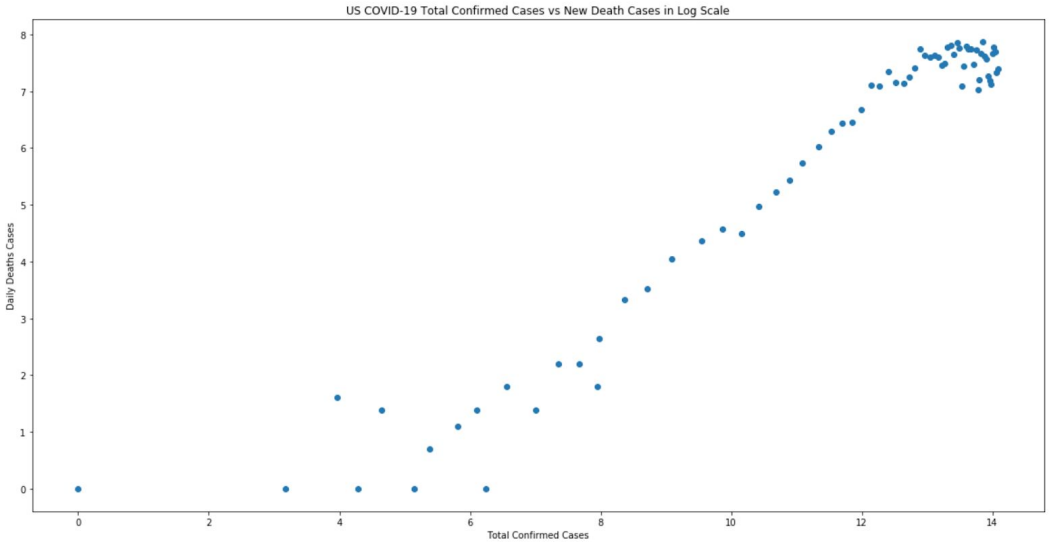
From the US New Confirmed Cases per day plot, we noticed that as time passes, the rate of increase in the new confirmed cases increased rapidly, and then gradually decreased. So we created another plot that shows the relationship between the total confirmed cases and the new confirmed cases. And to more easily visualize a representation of the relationship, we also created the plot on those variables in a log scale.





The scatter plot between the total confirmed cases and the new daily confirmed cases shows a clear trend between those two variables with a gradually decreasing sinusoidal pattern. We can infer that as the total confirmed case increases, the new confirmed case will slowly decrease in a sinusoidal pattern from that plot. To see the relationship between those two variables more clearly, we applied the logarithmic scale on those variables and plotted the plot. From that scatter plot, we noticed a linear relationship between total confirmed cases in log scale and daily confirmed cases in log scale. But as total confirmed cases increases, the data of new confirmed cases are getting more concentrated instead of keep increasing.

We also created a plot that shows the relationship between the total confirmed cases and new daily death cases. So before we do that, we followed the similar step from the previous cells and computed out the daily death cases data. The code for this step can be seen in our code file. And then plotted the scatter plot with total confirmed cases in log scale and daily death cases in log scale as we did before.



The scatter plot showing the relationship between the total confirmed cases in log scale and the daily death cases in log scale. These two variables have very similar relationship as the previous variables. From

these plots, it is clear that the logarithmic scale of daily death cases and daily confirmed cases has a linear relationship with the logarithmic scale of total confirmed cases. So this results indicates we might be able to use the daily death cases and daily confirmed cases to estimate the total confirmed cases in a logarithmic scale. Therefore, we used daily death and daily confirmed cases in log scale as our features to create a model for the total confirmed cases. Then, we carried out the **k-fold** validation with two features: daily confirmed cases and daily death cases. We first created a dataframe containing all the variables that we need to create a model and carry out the cross validation. The resulting data frame is shown below.

	Total Confirmed Log	Daily Confirmed Log	Daily Deaths Log	Days
0	3.970292	3.135494	1.609438	40
1	4.290459	2.995732	0.000000	41
2	4.644391	3.433987	1.386294	42
3	5.147494	4.219508	0.000000	43
4	5.379897	3.806662	0.693147	44
5	5.817111	4.779123	1.098612	45
6	6.109248	4.736198	1.386294	46
7	6.242223	4.158883	0.000000	47
8	6.562444	5.267858	1.791759	48
9	7.007601	5.983936	1.386294	49
10	7.350516	6.113682	2.197225	50

We then split training and test sets from our data with the 3:1 ratio with only one feature, 'Daily Confirmed Log' and imported LinearRegression from scikit-learn to create a linear model for the training sets. Then we computed the training error (RMSE) with the rmse function that we created previously. We got about 0.6718 for the RMSE, as shown in our code.

Finally, the following function computes the cross validation (KFold) rmse with 5 splits of the training sets. We got 0.6769 for the KFold validation, which is very similar to the training error.

```
from sklearn.model_selection import KFold
from sklearn.base import clone

def cross_validate_rmse(model, X, y):
    model = clone(model)
    five_fold = KFold(n_splits=5)
    rmse_values = []
    for tr_ind, va_ind in five_fold.split(X):
        model.fit(X.iloc[tr_ind,:], y.iloc[tr_ind])
        rmse_values.append(rmse(y.iloc[va_ind], model.predict(X.iloc[va_ind,:])))
    return np.mean(rmse_values)
```

```
cv_error = cross_validate_rmse(lin_model, X_training, Y_training)
print("KFold Validation RMSE: {}".format(cv_error))
```

```
KFold Validation RMSE: 0.6769044615951709
```

To increase the model accuracy, we added one more feature, **'Daily Death Log,'** and tested the RMSE on the test sets. The result we got is about 1, which is higher than the value of the training error that we got before with only one feature. The RMSE is still pretty low, but to make sure whether adding the feature helped increase the accuracy, we computed out the RMSE for the training set with the final model. We got about 0.4717 for the final training model error, and this indicates the new feature helped improve our final model.

```
from sklearn.linear_model import LinearRegression
final_model = LinearRegression()

X_training_f = training_set[['Daily Confirmed Log', 'Daily Deaths Log']]
Y_training = training_set['Total Confirmed Log']
X_test_f = test_set[['Daily Confirmed Log', 'Daily Deaths Log']]
Y_test = test_set['Total Confirmed Log']

final_model.fit(X_training_f, Y_training)
Y_Predicted_final = final_model.predict(X_test_f)
Y_Real_fianl = Y_test
print("Final Test Model Error (RMSE):", rmse(Y_Real_fianl, Y_Predicted_final))
```

```
Final Test Model Error (RMSE): 1.0090503879997554
```

```
Y_Predicted_final = final_model.predict(X_training_f)
Y_Real_fianl = Y_training
print("Final Training Model Error (RMSE):", rmse(Y_Real_fianl, Y_Predicted_final))
```

```
Final Training Model Error (RMSE): 0.4716927340167074
```

Using the final linear model to predict our results, we got 1.009 for the test error (RMSE), which is pretty low. Moreover, this low RMSE shows that the two features (daily confirmed log and daily death log) helped increase the accuracy of the model. So this tells us that there can be a linear relationship between daily death cases and daily confirmed cases in log scale and might be able to use the model to predict the number of future total confirmed cases. However, looking at the concentrated plots as the total confirmed cases in log scale increases, we can infer that at some point the total confirmed and death cases in log scale will fall rapidly. This shows that there will be a gradual decrease in both confirmed and death cases at some value of the total confirmed case. The increase in total confirmed cases will slow down and eventually stop. To see this, we have to create the curve fitting model with the quadratic linear regression, but since the plots are very clustered with not much of a pattern and we don't have more data regarding the moment of the rapid fall, it is hard for us to create an accurate forecasting model.

VI. Summary & Answer to final seven questions

When constructing our predictive model to answer our overarching question of: How can we forecast the epidemiological outbreak of COVID-19 within the United States by predicting the number of confirmed cases using regression techniques? We first used the number of days that had passed since the first recorded confirmed case and the trend of confirmed cases to forecast the increase of cases in the United States for the next 30 days. However, after fitting our model and training it, we ended with a training error of around 1.176 and test error of 2.2436. We found that using only this one feature was not that effective for accurately predicting the number of confirmed cases; thus, we wanted to decrease our test error by using a different set of features that could result in the smallest RMSE. We decided to use two different features -- new daily cases in the model to see the rate of growth and the daily death cases for the model to increase our accuracy. We got about 0.4717 for the final training model error, and this indicates the two new features helped improve our final model.

Some challenges we faced with our data was that the four original datasets that we were given only contained data until April 18th, and thus did not provide us an updated set of data points to accurately analyze and create a predictive model with higher testing accuracy. Another challenge we faced was after we created our first linear regression model -- we figured that it was hard to predict the near future because the predicted *total* number of confirmed cases was growing exponentially and rapidly fast, and thus using one feature wasn't enough for accurate predictions. After developing our second model with two more features (new daily

confirmed cases and deaths), some limitations that we faced were that, even after we noticed that at some point in the future, the total confirmed and death cases in the log scale will fall rapidly, because we did not have more data, we could not take this into account before finalizing our model. Additional data, such as how trends have changed in the number of confirmed cases and deaths in other countries that have shown similar trends may have helped in fully developing our model and adding more features to see how trends may change in the near future.

Some ethical dilemmas we faced were that we did not know exactly how and through what measures the organization that we derived our datasets from collected the data we analyzed. Although our datasets were from a credible source, we did not collect the data ourselves, thus there were limitations as to how much knowledge we had in the process of compiling all the statistics and numbers from different multiple different sources that was done by the JHU CSSE. Furthermore, some ethical concerns that we might encounter in studying this issue is the question of how and to whom we should share our findings with other individuals, and to what extent we should trust the predictions we have made. If we were part of the real-world healthcare industry developing a similar model, we would ask ourselves whether or not those in the public have a right to know the potential disastrous future in our near future -- and handling a good balance between citizens' right to know scientific research and the panic that could arise amongst the general public if such information were to be disclosed.

VII. Discussion

Surprising Discoveries

We saw through our visualizations and graphs that the rate of increase in the total confirmed cases of COVID19 in the United States was strikingly rapid. Furthermore, one surprising trend that we noticed was that the number of new confirmed cases are decreasing daily in a sinusoidal trend. Another discovery that we made that was not so obvious was in the log scaled values of the total confirmed cases, daily confirmed cases, and the daily death cases. The linear relationships between those values weren't expected. And another more surprising outcome was that as the rate of increase in the confirmed cases started to slow down, the log scaled relation plots also started to cluster at the point where the rate of increase slows down. This is a very important discovery since we can see the US trend when the new daily COVID19 confirmed cases start decreasing and we can determine which stage we are at compared to other countries. Are we still struggling with the skyrocketing number of the confirmed cases or are we at the stage where the situation is stabilizing?

Evaluation of Approach and Limitations

Our approach validly shows the relationship of the total confirmed cases with the daily death and confirmed cases in log scale with the clear visualization. And this approach will be very informative with an appropriate model. From our model, it is apparent that there is a linear relationship, and even though our RMSE was very low, it is obvious that there will be rapid fall at some point. And our model was linear, which will only predict the upward trend, and this doesn't make sense too. So our model is not very informative at this stage. We needed more future data to create a model that will clearly predict the period when the rate of the increase in the total confirmed cases becomes nearly zero. Moreover, if we have the same dataset from other countries like South Korea or China where they are at the stabilization stage, we can compare the trend of the data with the trends of those countries and possibly can create a model based on them. Lastly, since we are not plotting the variables against the time, it is very hard to see the timeframe of the data and it is doubtful whether we can forecast the number of confirmed cases in a certain time-based period from the model.