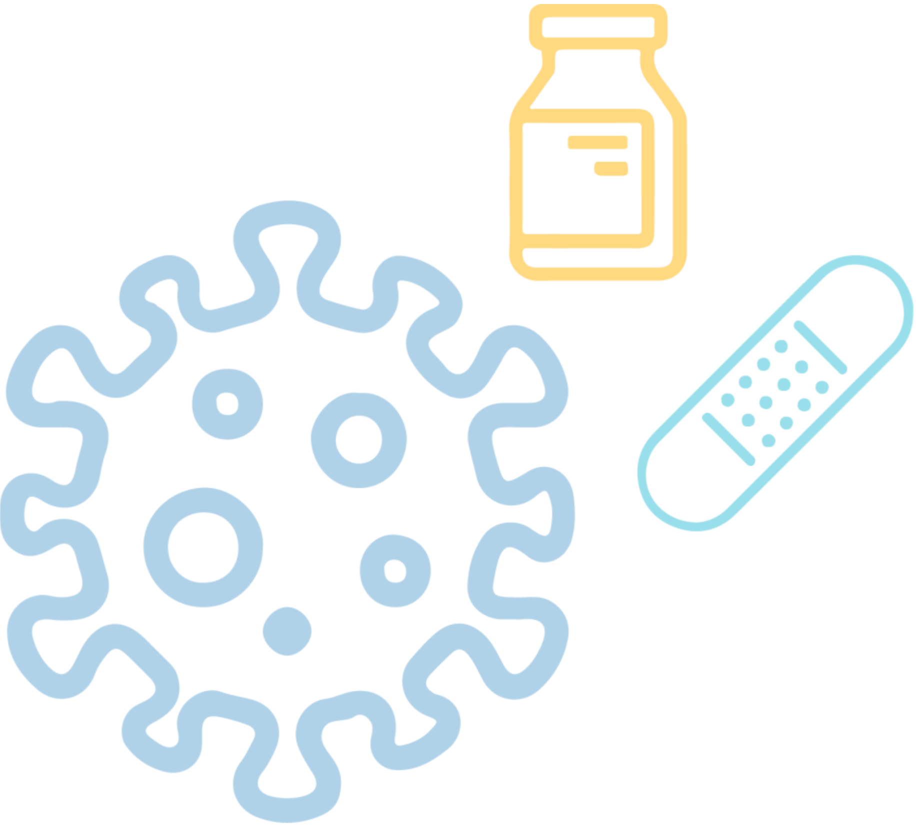
Predict the Amount of Covid-19 Confirm & Death Cases in New York City

Dingjun Liang (dl3472)

Machine Learining for Environmental Engineering and Science

Fall 2021 EAEE 4000 Project

Professor: Pierre Gentine



Contents

1. Introduction

The ongoing novel coronavirus disease 2019 (COVID-19) has already been considered as one of the greatest challenges faced by humanity in the 21st century. On 11 Mar 2020, COVID-19 was officially declared as a global pandemic by the World Health Organization. At present, the number of confirmed COVID-19 cases worldwide has already crossed 190 million, while the number of deaths due to this new virus exceeds 4 million globally.

While vaccinations are gradually deployed worldwide, there are still unanswered questions pertaining to the nature of this complex disease and its variants. First of all, although there are many signs showing that this epidemic disease might have seasonal behavior, we still haven't find out possible relationship between climate conditions (such as temperature and humidity) and the transmission rate of COVID-19. What's more, we also want to see how vaccine could actually help with the job of stemming the spread of the virus. Another specific area of concern is to decide if the social distant and mobility will have any influence on the active case & death rate.

So in this project, I want to use several main climate index, Vaccinated rate among residents and mobility as input variables to predict the increasing new case and death case everyday. In order to eliminate the possible influence of geographical factors on the forecast results and improve the model specificity, I decide to choose a specfic region as my research objective. And finally I choose New York city as my research objective for the following reasons: 1.Most importantly New York is the city where I am living in now and also for the next 2 years; 2. New York is one of the most famous and international cities which brings together people from different countries and different region, its high migration rate makes itself more prone to spread the virus; 3. When covid-19 first outbreak in US, it is one of the most serious cities in America, making itself a typical one to conduct epidemic disease research.

2. Data

2.1 Data Source

While there are plenty of open-data resources pertinent to the study of the spread of the Covid-19 pandemic and its control, I collected and organized my variables data mainly from the following source.

1. Covid-19 Case Data: I downloaded the covid-19 case statistics from New York Times,

it releases a series of data files with cumulative counts of Covid-19 cases in the US, at state and county level, over time. The time-series data are compiled from states, local governments and health departments. Since January 2020, The NY Times has tracked cases of coronavirus in real time as they were identified after testing. Then, these data have been used to power maps and generate reports about the outbreak. The data collection began with the first reported coronavirus case in Washington State, on 21 January 2020. Since then, the NY Times publishes regular updates of data in https://github.com/nytimes/covid-19-data.

2. Climate Data: The weather dataset I am going to use is from the National Aeronautics and Space Administration (NASA), which is one of the most authoritative among related fields. From an open-data perspective, NASA's project prediction of Worldwide Energy Resource (POWER) can be very useful to recollect time series and monthly means of the most relevant weather and climate variables for a given location. For my project, I used the coordinates(Latitude 40.785091; Longitude -73.968285) which is location the New York Central Park and find my weather data.

3. Vaccinated Rate: The public Vaccinated rate data of my research is from Johns Hopkins University. JHU experts in global public health, infectious disease, and emergency preparedness have been at the forefront of the international response to Covid-19. JHU provides a daily update of many covid-19 related data, including vaccination, disease rate, which can be found at https://coronavirus.jhu.edu/map.html.

4. Mobility Data: I obtained my mobility dataset from Apple's website (https://www.apple.com/covid19/mobility), it developed Covid-19 Community Mobility Reports, in which each report is broken down by location. It can be obtained filtering by country and contains mobility of three dimensions, which are Driving Mobility, Walking Mobility and Transit Mobility.

Type	Variable	Abbreviation	Source
Case Statistics	Total Accumulated Active Case	cases	New York Times
	Total Accumulated Death Case	deaths	
Climate Index	Temperature (°C)	TS	NASA
	Precipitation (mm)	PRECTOTCORR	
	Specific Humidity at 2 Meters (g/kg)	QV2M	
	Relative Humidity at 2 Meters (%)	RH2M	

	Surface Pressure (kPa)	PS	
	Wind Speed at 10 Meters (m/s)	WS10M	
Vaccinated Rate	People Vaccinated per Hundred	PVPH	JHU
	People Fully Vaccinated per Hundred	PFVPH	
Mobility Index	Driving Mobility	driving	Apple
	Walking Mobility	walking	
	Transit Mobility	transit	

Table 1: Summary of my Data Source

2.2 Data Visualization

My dataset includes information of 610 days recording the disease's development of New York city from 2020/3/1 to 2021/10/31, which is the latest one when I started to conduct this research.

Figure 1 shows a histogram of the count of the number of new confirmed case in New York for the period I choose. As we can see, the daily new case datas range from 0 to 20184 and heavily lean towards fewer new case. About half of the datas are in the range of 0 to 3000 new cases.

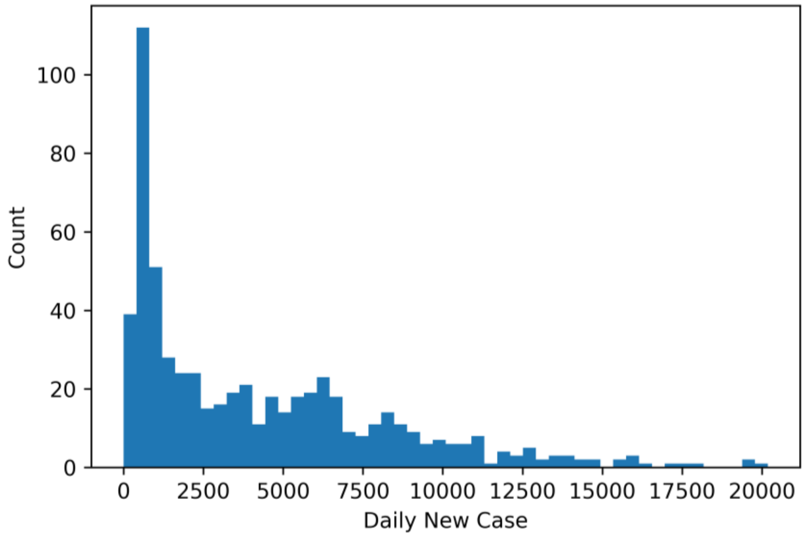


Figure 1: Histogram of Daily New Confirmed Case

While Figure 2 shows a histogram of the count of the number of daily new denth caused by covid-19 in New York. We can see that this data seems even more slant than the daily new case data. With total range from 0 to 1036 but 75% of the datas are under 100.

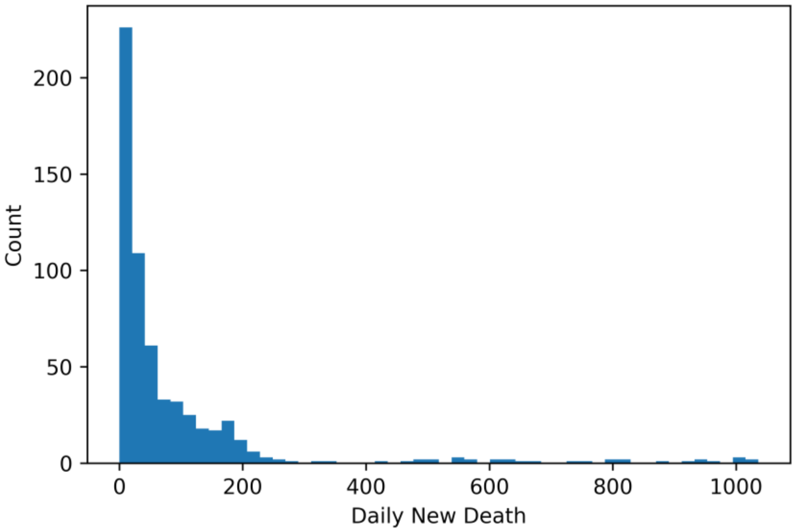
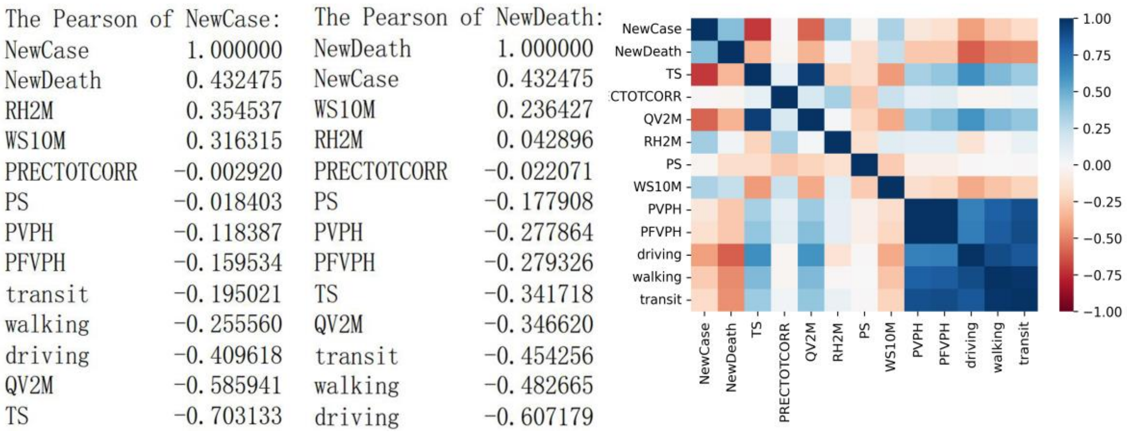


Figure 2: Histogram of Daily New Death

Before I build models to make a prediction, I also want to check if the numder of daily new confirmed case and daily new death are really correlating with my variables of three dimensions. If it turned out to be no relationship between weather and covid-19 case, or between vaccination and covid-19 case, this project would be dispensable and unmeaningful. The first thing I do is to calculate the Pearson correlation of each variable, and the results are shown in the following figure.



```
The Pearson of NewCase:          The Pearson of NewDeath:
NewCase          1.000000        NewDeath          1.000000
NewDeath         0.432475        NewCase           0.432475
RH2M             0.354537        WS10M             0.236427
WS10M            0.316315        RH2M              0.042896
PRECTOTCORR     -0.002920        PRECTOTCORR      -0.022071
PS              -0.018403        PS               -0.177908
PVPH            -0.118387        PVPH             -0.277864
PFVPH           -0.159534        PFVPH            -0.279326
transit         -0.195021        TS               -0.341718
walking         -0.255560        QV2M             -0.346620
driving         -0.409618        transit          -0.454256
QV2M            -0.585941        walking          -0.482665
TS              -0.703133        driving          -0.607179
```

Figure3: Pearson Correlation of Variables

As we can see that, a number of weather variables obviously correlated with the number of covid-19 cases, so as mobility variables. The most outstanding and noteworthy one is temperature, with -0.70 between temperature and daily new confirmed case and -0.34 between temperature and daily new death, representing a strong seasonality of this infectious disease. Below scatter figure, which shows the relationship between temperature and disease case, can also futher prove this viewpoint.

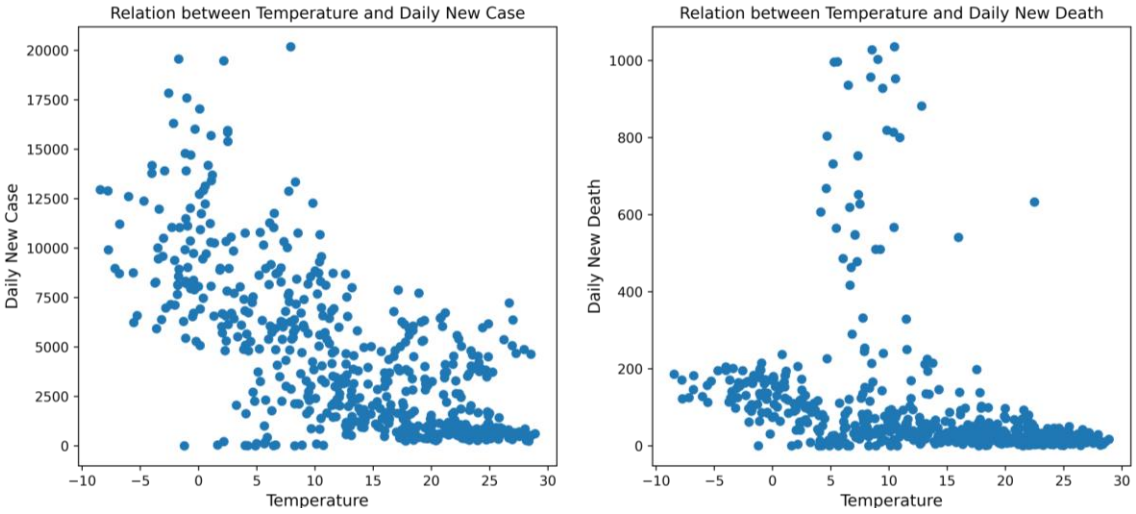


Figure4: Relationship between Temperature and Disease Case

Another dimension caught my eyes is the mobility related variables. From the data we can preliminary conclude that all three mobility variables state a quite strong correlation with the number of disease cases, which means that social distant and resident exposure rate did have influence on the spread of the covid-19 virus in some degree. It makes sense based on my initial assumption of how mobidity would impact covid-19 cases. Lastly, I am very surprised to see that there seems no strong correlation between Vaccinated percentage and new case/death since base on our knowledge, vaccination is a super outstanding and crucial weapon to fight against disease virus.

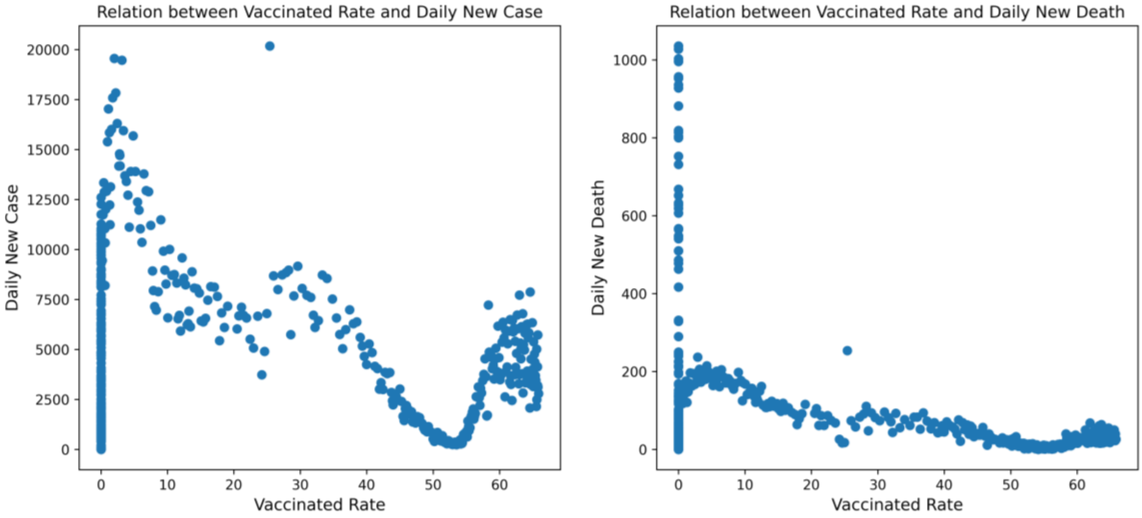


Figure5: Relationship between Vaccinated Rate and Disease Case

To futher check why such ridiculous outcome shows up, I draw the above scatter to show relationship between vaccinated rate and disease case. It is clear that this picture is exactly in compliance with what we expected because it states that the number of daily new case/death drop sharply as more and more people are vaccinated. In my viewpoint, it is the outlier points, which are the points before the certified vaccine first comes out(on the left side of both figures), lower the correlation between vaccinated rate and disease cases. So we have to find method to improve the importance of this significant variable in our further research,

2.3 Cleaning and Preparing

In order to avoid any errors whenI run my prediction models and get the most accurate results, I thoroughly cleaned the data before starting.When cursory look at my dataset, I found that there are a few missing records for the variables 'Vaccinated Rate' and 'Mobility Index'. What I do to the missing 'Vaccinated Rate' is filling the blank ones with the mean of the day before them and the data after them. Since the Vaccinated rate of a certain region normaly increase by time and would not change sharply in a short period, I think this interpolation method would be quite helpful in filling th NA value.

But when it comes to the missing 'Mobility Index', I found that the data for mobility is quite random without a certain pattern change over time, and the mobility is sometimes even very different from the nearby dates. So finally I decided to drop all records with missing 'Mobility Index'. What's more, while checking the data I was also very surprised to find some negative value for the daily new case & daily new death records, which maybe caused by some faults in the process of data collection. I just drop all these records because I find it against our common sense and I can not find any discriptions about such abnormal datas in its intruction.

Finishing all the above work for dealing with missing and abnormal data, the number of records decreases to 602 rows now. I also realized that before using models to fit my data, it is critical to standardize them, given the different scales of the variables may make gradient descent process much slower. So I scaled all the numerical data by using them to substract their mean and then dividing them by their standard deviation to ensure their values are all in a similar interval.

3. Methodology

It is clear that covid-19 confirm & death cases prediction is a regression problem and my aim is to forcast a continuous value. In this project, I decided to use Polynomial Regression, Neural Network, Gaussian Process and XGBoost as my prediction models to fit the train data and make a prediction of the test data. By doing so, I will compare the results of these four models to decide which one achieves the best results and then keep improving the model's performance by adjusting their parameters.

3.1 Polynomial Regression

Polynomial Regression is an extension of normal linear regression by constructing polynomial features from the coefficients. This extension is pretty helpful while we are trying to use linear models trained on nonlinear functions of the data. This approach maintains the generally fast performance of linear methods, while allowing them to fit a much wider range of data.

The first step to use Polynomial Regression is to separate data into training set and testing set according to a certain percentage. In my project I set the test size = 0.2 and apply the random state = 100 for all the models I am going to use for the convenience of comparison. In my first try I used the polynomial degree = 2 to fit my training set and then forecast the new confirmed cases of my testing set. To determine how well this algorithm performs, I calculated the mean absolute error and R^2 value of my model and the result was that it had a MAE of 1367 and a R^2 score of 0.7489. I also drew the below figures to show the range of the forecast error as well as how close the predicted values was to the true ones. We can conclude both numercally and visually that the prediction result was actually quite good as most values were very similar to their real numbers except for few outliers, and most of these outliers normally represented a relative huge number of new cases.

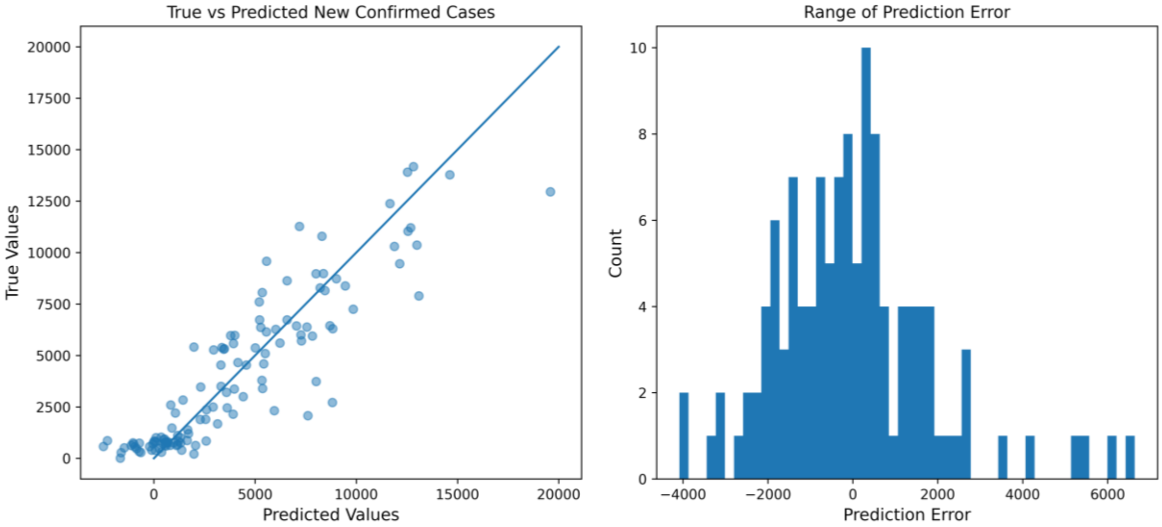


Figure6: Prediction Result Visualization of Polynomial Regression

To decided whether polynomial degree = 2 is an ideal parameter, I tried different degree while keeping other parameter remain unchanged and calculated the R^2 score %

MAE for different models to find out the optimal one, the result is shown in the below table.

Polynomial Degree	R^2 Score(Training set)	R^2 Score(Testing set)	MAE(Testing set)
1	0.5818	0.6015	1860
2	0.8437	0.7489	1367
3	0.9770	-6.7127	3842
4	1.0000	-175.6942	12683

Table 2: Results of Different Polynomial Degree

As we can see, when increasing the degree, the R^2 score and MAE for testing dataset went down at first and then boosted rapidly, while the R^2 score for training dataset kept increasing and almost reached 1.0 after degree = 3. It represented a overfitting phenomenon for my prediction when the polynomial degree is equal to or bigger than 3. The model had a highst R^2 score and lowest MAE when degree = 2. So obviously, polynomial degree = 2 is exactly the perfect value for my Polynomial Regression model.

3.2 Neural Network

The second model I am going to use is Neural Network. Widely applied in regression problems, Neural Network is like a black box, it imitates how real brain of humanity works, trains the neurons of each hidden layer by giving specific weights to the neurons of the former layer and then updates the weights and every neurons in the process of gradient descent. When the loss function finally reaches a smallest value, the model ends it’s training step and gives us a quite efficient model for prediction.

In this part I want to use Keras, which is a open source Python library for building deep learning models, to create my Neural Network prediction model. After spliting the training set and testing set just as same as what I did to Polynomial Regression, I have to define the basic structure of the network model. I firstly tried 2 hidden layers, each with 16 neurons using the Relu activation function, and set the model’s optimizer as Adam for my initial model. I also set the patience = 20, which would stop the training process if the loss fail to decrease in 20 continuously epoches. Next, I trained the model using training data and created a chart with the epochs on the x-axis and loss on the y-axis, in order to figure out what is going on during training process.

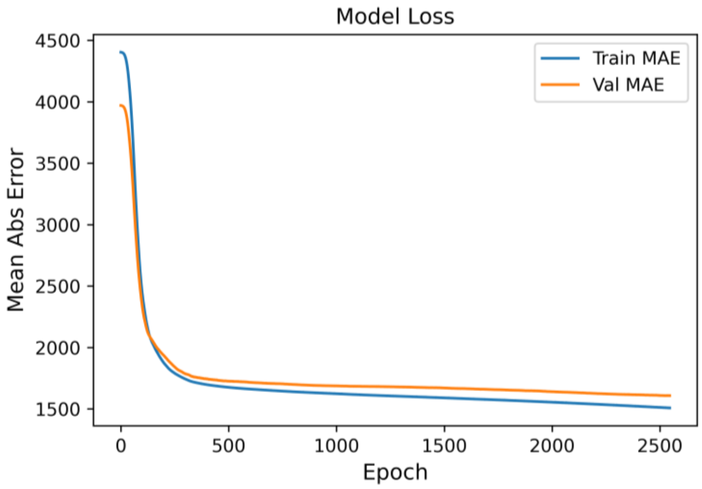


Figure7: Model Loss of each Epoch of Neural Network

Lastly, it was time to bring in the testing data and evaluate the model performance. The below figures show comparison between the true values and predicted values for the number of daily new confirmed cases and the range of model forecast error. I also calculated the score of my model and got a mean absolute error = 1462 and an R^2 = 0.6914, representing similar performance with Polynomial Regression.

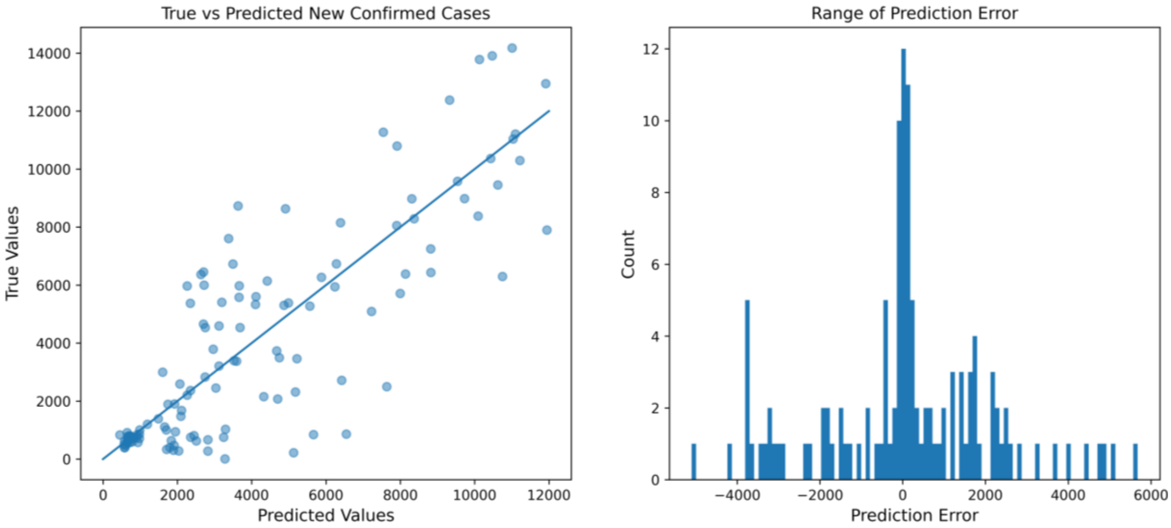


Figure8: Prediction Result Visualization of Neural Network

In the following step, I am going to change the architecture of the model by adjusting the number of hidden layers and the number of neurons of each layers to see I can boost the model performance. I decided to start with adding or dropping some layers and record the results in the following table.

Hidden Layers	MAE	MSE	RMSE	R^2 Score
1	1491	4293667	2072	0.6875
2	1462	4240632	2059	0.6914

3	1446	3990354	1997	0.7096
4	1395	3749226	1936	0.7271
5	1488	4222357	2054	0.6927
6	1541	4371510	2090	0.6818

Table 3: Results of Different Number of Hidden Layers

It seems that the model results just changed in a tini range while changing the number of hidden layers. When we set 4 hidden layers, the model have a best performance with MAE = 1395 and R^2 score = 0.7271. So in the next step I am going to fix the number of hidden layers and then change the amount of neurons for all layers to see it can help to improve the results.

Neurons of Each Layers	MAE	MSE	RMSE	R^2 Score
4	1554	4618805	2149	0. 6639
8	1487	4424166	2103	0.6780
16	1395	3749226	1936	0.7271
32	1425	3958691	1989	0.7119
64	1412	4040055	2009	0.7060
128	1469	4228149	2056	0.6923
256	1479	4370349	2090	0.6819

Table 4: Results of Different Amount of Neurons

Situation is similar to changing the number of hidden layers, the prediction accuracy could only improve In a slight range by altering the amount of neurons all layers. But we can still concluded from the table that with 16 neurons for every hidden layers, the Neural Network have a best performance. So combining with the information we got before, the best Neural Network structure for the prediction of the number of daily new confirmed case is 4 hidden layers and each layers containing 16 neurons.

3.3 Gaussian Process

The next model I am going to use is Gaussian Process. Gaussian Process (GP) is a kind of stochastic process in probability theory and mathematical statistics. It is a combination of a series of random variables that obey a normal distribution in an index set. Gaussian process is completely determined by its mathematical expectation and covariance function, and inherits many properties of the normal distribution One advantage of this regression algorithm is that it can compute empirical confidence intervals during fitting process.

The only parameter I can change while using Gaussian Process is its kernal. I tried every common kernals and found that the model had a best performance while applying dot product kernel, which is defined as ‘kernal = ConstantKernel(0.1, (0.01, 10.0)) * (DotProduct(sigma_0=1.0, sigma_0_bounds=(0.1, 10.0)) ** 2)’, with MAE = 1227 and R^2 score = 0.7750, implying a little improvement compared to the models I used before. The prediction outcome and its 95% confidence interval are shown in the below figure.

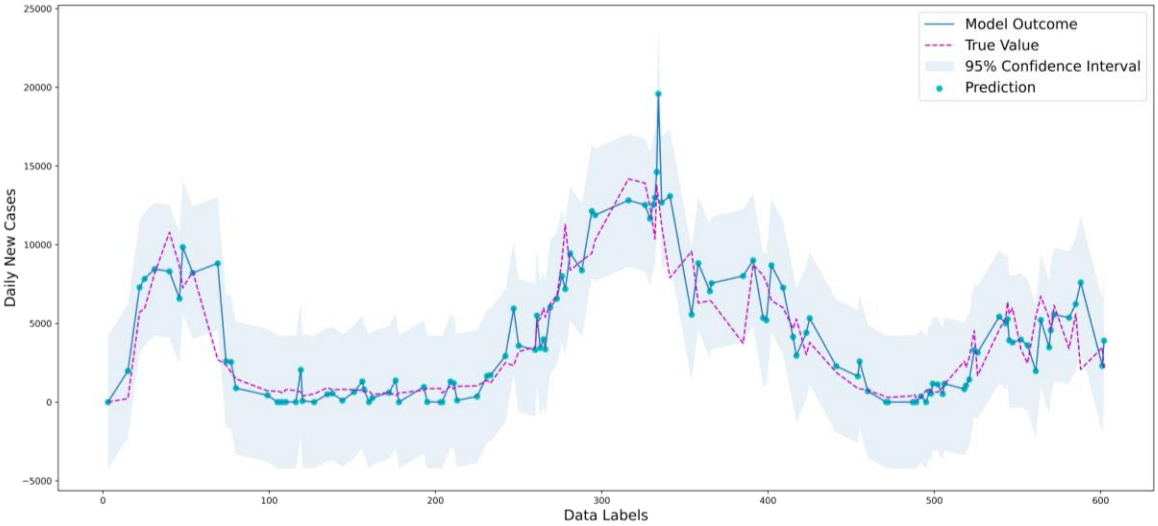


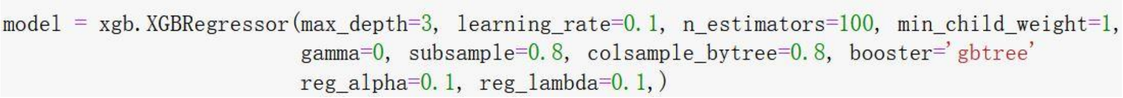
Figure9: Prediction Result Visualization of Neural Network

We can see from the picture that the fiting outcome is quite close to the true one, revealing good performance of Gaussian Process while applying to my dataset.

3.4 XGBoost

The last model I am trying to run in my dataset is XGBoost. Based on gradient boosting algorithms, XGBoost combines a lots of weak models to formula a relatively strong one. During the training process, XGBoost fits the residuals of the prediction values, so that the prediction residuals keep decreasing, and the forecast values keep approaching the true values.

In my first try I just randomly set some initial parameters to build a baseline model, which could help me preliminarily judge the fitting performance of this algorithm. I used the following parameters while building my first XGBoost model.



```
model = xgb.XGBRegressor(max_depth=3, learning_rate=0.1, n_estimators=100, min_child_weight=1,
                         gamma=0, subsample=0.8, colsample_bytree=0.8, booster='gbtree'
                         reg_alpha=0.1, reg_lambda=0.1,)
```

The result of this model is very outstanding with a mean absolute error of 1060 and a R^2 score of 0.8094, better than all the previous models. The fococast performance and prediction residuals are shown in the below figures.

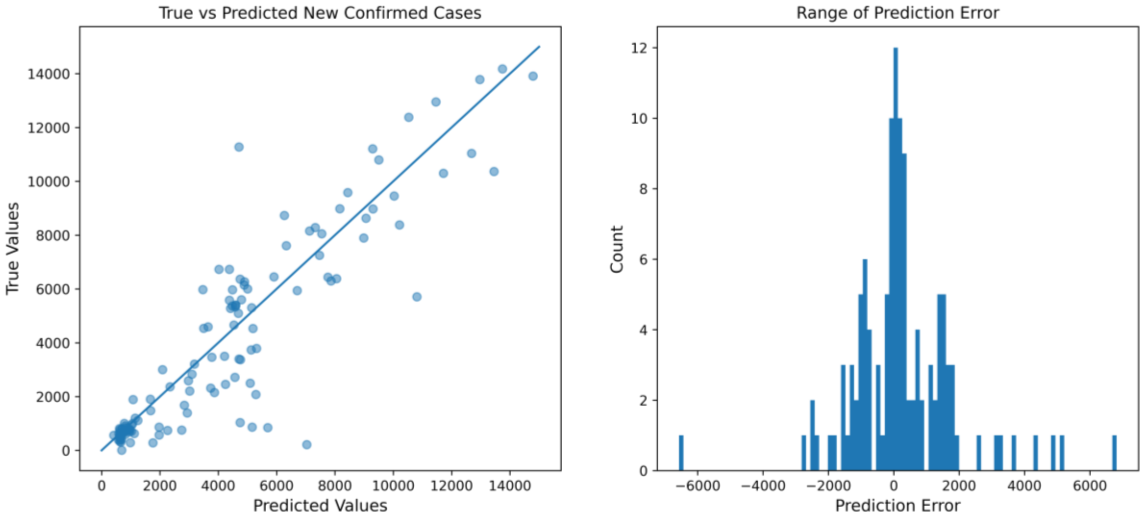


Figure10: Prediction Result Visualization of XGBoost

Next step is trying to tune the parameters. I am going to adjust the hyperparameters one by one to decide the optimal values for all adjustable parameters. The library sklearn provides us a wonderful function, sklearn.model_selection.GridSearchCV, to complete this task. For example, when I try to find the best ‘n_estimators’ value, I can keep all other parameter remain unchanged and then give a cv_parms = {‘n_estimators’:[100, 200, 300, 400,500]} to the function, it would output the performance of each model and show you the rank of each value, just like the table below.

n_estimators	Mean R^2 Score	Rank
100	0.7790	1
200	0.7779	4
300	0.7780	3
400	0.7777	5
500	0.7780	2

Table 5: Results of Different n_estimators

After tuning all the adjustable parameters, I finally got an optimal parameters assemble, the specific parameter values are as follows: {‘n_estimators’ : 100; ‘max_depth’ : 6; ‘min_child_weight’ : 1; ‘gamma’ : 0.1; ‘learning_rate’ : 0.1; ‘reg_alpha’ : 2; ‘reg_lambda’ : 0.1; ‘colsample_bytree’ : 0.8; ‘subsample’ : 0.8}. I then applied this parameter combinations to my original base model, fitted my training data and then made a prediction for my testing data, the result is fabulous, with MAE = 937 and R^2 score = 0.8630, showing a obvious improvement compared to the base model.

Another advantage of XGBoost algorithm is that it is able to show you the feature importance and make a rank for them. So I drew a bar graph to visualize my feature significance, the result is quite revealing. It is clear the three most important features are Temperature(°C), People Vaccinated per Hundred and Specific Humidity at 2 Meters (g/kg), which is not much different from my previous assumption as well as the Pearson correlation we got before.

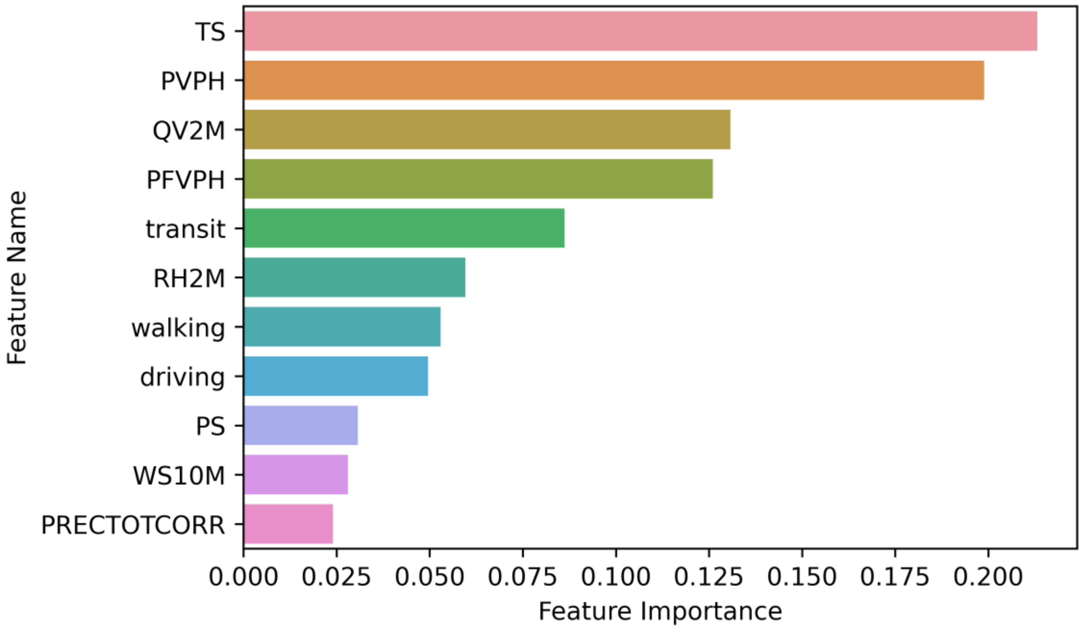


Figure11: Features Importance for New Confirmed Case

3.5 Models Comparison

After fitting my data as well as making a prediction, it is crucial to compare the performance of all the models to decide a best one. To show how well they did in a more intuitive way, I again put their outcome together in a single table.

Model Name	**Mean Absolute Error**	**R^2 Score**
Polynomial Regression	1367	0.7489
Neural Network	1395	0.7271
Gaussian Process	1227	0.7750
XGBoost	937	0.8630

Table 6: Model Comparison

It is very obvious that XGBoost is the best model to forecast the number of daily new confirmed cases, since it achieves a lowest MAE and the highest R^2 score. In the next step I am trying to see if I can further improve its performance by applying some change to the feature.

3.6 Attempts for Further Improvement

3.6.1 Alter Features

In the initial data visualization stage, I notice that there are many data points whose vaccination rate is equal to zero due to vaccination hadn't been widely applied for New York residents since 2020/12/20. So I think adding a new feature representing whether vaccination had been invented or not might make a difference to the result. I also try to conduct the same method to rainfall, because precipitation <= 3mm happens in about 70% of the data. In that case, adding a feature deciding if it is rainy maybe also a good choice. Furthermore, randomly droping some of the features I used is also worth trying, because I might possibly pick some excess features at the beginning which could be useless to the prediction.

I tried the above three strategies as well as their combination but the results was proved to be quite upseting, the scores for these models are all slightly lower than the initial one. I finally gave up this attempt since they turn out to be bad methods.

3.6.2 Separate Model

After trying to add, drop and alter the features, I thought it would be interesting to see how models would preform if I separate it into two models. As we look at the below figures showing the count of new case & death change over time, we could easily detect that there is an unusual sudden augment for amount of both new case and new death happened in around 2021/03/24, which might caused by the mutated new coronavirus(Delta virus) and the unconsciousness of residents. I realized that my model would become less efficient while facing mutated virus because I didn't put any feature about mutation ditection, so I think cutting the data before and after the mutation and build two modes to make a prediction for them seperately might be helpful.

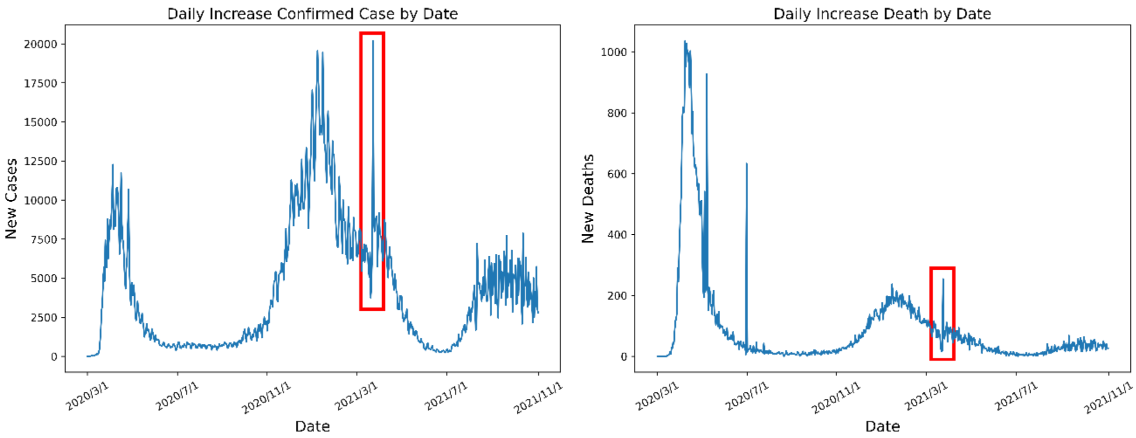


Figure12: Amount of New Case & Death Over Time

Stage	MAE	MSE	RMSE	R^2 Score
Not Seperate	937	1882927	1372	0.8630
Stage 1	1040	2247559	1499	0.8949
Stage 2	659	1071020	1034	0.8196

Table 7: Models Performance After Seperated

The results after seperating the data according to their stage(before and after mutation happened) is pretty revealing. As we look at the stage 1(before mutation), although the value of MAE increases slightly, there is an apparent augment for the R^2 score, almost reaches 0.9, indicating a better prediction performance. For stage 2(after mutation), there is obvious decrease for the R^2 score, but we should also notice that after dividing the datas there are only 176 rows of recording used to fit the stage2-model. Such a small amount of data is normally difficult to make the model have a good prediction performance. So in general, I believe that seperating the model according to the time when new mutation happened does good to the model effect.

3.7 Prediction for Amount of New Death

All the above work we have done is just for the prediction of amount of new confirmed cases, we still have another dimension to predict is the number of new death In New York city. Since the research process is totally the same with what we did to the comfirmed cases prediction, I'd like to skip the details of this round and just write down the main results in this paper.

Model Name	Mean Absolute Error	R^2 Score
Polynomial Regression	64	0.5184
Neural Network	32	0.7573
Gaussian Process	42	0.6616
XGBoost	20	0.9164

Table 8: Model Comparison

The scores table shows that while applying these four models to forecast the number of new death, XGBoost is still the best one. It's performance is even better than when it is applied to the prediction of new confirmed cases, with R^2 score = 0.9164 itself compared to R^2 score = 0.8630 of the latter one. What's more, the other three estimator tacitly perform different degree of worse effect, the most violently decreasing one among which is the Polynomial Regression, dropping from R^2 score = 0.7489 to R^2 score = 0.5184. However, seperating the model according to when mutation happened seems failed to improve the model performance for prediction of the amount of new death. The models' results are shown in the following table.

Stage	MAE	MSE	RMSE	R^2 Score
Not Seperate	20	1715	41	0.9164
Stage 1	39	8954	94	0.6660
Stage 2	8	134	11	0.6941

Table 9: Models Performance After Seperated

I also drew the bar graph of the features importance for forecasting the amount of new death. It is surprising that the three mobility indexes, transit, walking, driving turn out to be the most decisive ones, contributing together around 85% to the results prediction.

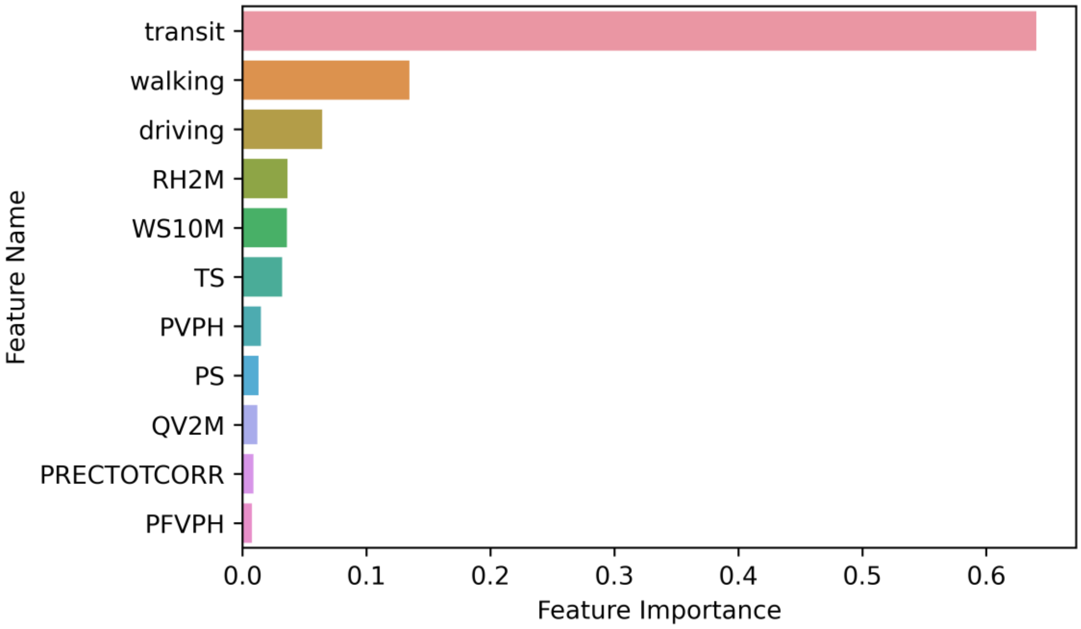


Figure13: Features Importance for New Death

4. Conclusion & Discussion

The goals of this porject is to determine how weather, vaccination and mobility indexes influence the amount of daily new confirmed cases and death in New York city, as well as to build a model to make a prediction. After many attempts for several common models used for regression problems, XGBoost is proved to be the most outstanding one while being applied in my dataset, with MAE = 937 cases & R^2 score = 0.8630 for predicting new cases, and MAE = 20 cases & R^2 score = 0.9164 for forecasting new deaths. What's more, segregating the model after new mutation took place could slightly improve the performance for forecasting the amount of new confirmed cases, but seems not so effective when it comes to predicting the number of new deaths.

Considering the pretty good predictive performance of my model, I think all the features I choosed are related and helpful for the results. Weather and Vaccination are definitely vital conditions for preventing the spread of the virus. We can see a clear pattern that the number of daily new confirmed cases dropped rapidly with more and more people getting partially vaccinated or fully vaccinated, and it's also obvious for the seasonality of covid-19 virus as we can detact that with apparently more cases in winter and fewer cases in summer. The influence of mobility is could not be ignore, as it is the most important feature while predicting the amount of new cases. And it is also consistent with our common sense that with less contact, the chance for reaching potential patients is getting lower, and thus less likely for people to be confirmed sicked.

Actually creating an extremely accurate predictive models for covid-19 cases/deaths is a very hard task because the world is keeping changing, so as the virus. The mutations of covid-19 virus occur from time to time and the process is unpredictable. Considering medical scientists and professors haven't find a superfine way to predict the occurrence of new mutation, which would possibly make the existing vaccine invalid and increase the new cases drastically, it is even more difficult for us to seek for a feature representing the chance of new mutation.

Overall, I learned a lot from this project and I believe this would not be successful without the guidance of Professor Gentine, the help of TA and the thoughts of other classmates. Actually I knew nothing about machine learning before participating in this course but now I can code and build simple model by myself. Again thank you all guys, I am super grateful for the support and ideas you guys provided.

References:

[1] Ml A , Tr B , Cg C . Predictive models for COVID-19 cases, deaths and recoveries in Algeria[J]. Results in Physics, 2021.

[2] Haghighat F . Predicting the trend of indicators related to Covid-19 using the combined MLP-MC model - ScienceDirect[J]. Chaos, Solitons & Fractals, 2021.

[3] Iloanusi O N , Ross A . Leveraging Weather Data for Forecasting Cases-to-Mortality Rates Due to COVID-19[J]. Chaos Solitons & Fractals, 2021:111340.

[4] Chew A , Wang Y , Zhang L . Correlating Dynamic Climate Conditions and Socioeconomic-Governmental Factors to Spatiotemporal Spread of COVID-19 via Semantic Segmentation Deep Learning Analysis[J]. Sustainable Cities and Society, 2021(6501):103231.

[5] Yadav M , Perumal M , Srinivas M . Analysis on Novel Coronavirus (COVID-19) Using Machine Learning Methods[J]. Chaos Solitons & Fractals, 2020:110050.

[6]A study of the possible factors affecting COVID‐19 spread, severity and mortality and the effect of social distancing on these factors: Machine learning forecasting model[J]. International Journal of Clinical Practice, 2021.

[7] Alamo T , Daniel Gutiérrez, Mammarella M , et al. Covid-19: Open-Data Resources for Monitoring, Modeling, and Forecasting the Epidemic[J]. Electronics, 2020, 9(827).

[8] Abdulkareem A B , Sani N S , Sahran S , et al. Predicting COVID-19 Based on Environmental Factors With Machine Learning[J]. Intelligent Automation and Soft Computing, 2021, 28(2):305-320.