Covid-19 outbreak prediction using machine learning algorithm

Student Name: Meheruna Alam

Matric no. : 17202481

Course: Machine Learning

Faculty of Computer Science & Information Technology
UNIVERSITY OF MALAYA
Kuala lumpur

January, 2021

Table of Contents

1: Introduction

Project Background :

Coronavirus (COVID-19) outbreak in Wuhan around last January has significantly impacted us globally throughout the year. The virus has broken almost the entire world by affecting economical, social and physical aspects. It's been almost a year the world is going through this phase. Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. To achieve the goal, different statistical methods have been used to forecast Covid-19 incidence all over the world. Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. However, to predict the future, machine learning algorithms are the most prominent technology to get insights and foresee the different state of the outbreak in terms of number of cases, deaths and recoveries based on different regions and countries. This project has been initiated from the idea of extracting more insights of covid-19 outbreak by generating a number of predictions for active and death cases.

Problem Statement:

So far the confirmed COVID-19 cases across the globe are 1,498,833 and mortality rate approximately 5.8%. Gradually the mortality rate is increasing and it's an alarming factor for the whole world. In such cases, each day covid-19 situation is changing in different regions and the future is quite unpredictable. In order to plan or prepare for the future and to take necessary measures by governments, it is necessary to foresee the situation based on the current knowledge.

Project Overview:

In such, outbreaking situation Machine Learning algorithms can assist to foresee the issue in different aspects and generate prediction for different countries in terms of number of cases, deaths and recoveries. To align with the objective, the project has been executed to focus and experiment on predicting covid-19 confirmed cases and deaths for next 30 days. In this experiment, **polynomial regression** and **SVM**(Support Vector Machine) models are being used to see prediction on confirmed and death cases.

Project Objectives:

The project has two following objectives -

Objective-1

Find the best model with the lowest MAE and MSE comparing polynomial regression and SVM to predict the number of cases for the next 30 days.

Objective-2:

Find the best model with lowest MAE and MSE comparing polynomial regression and SVM to predict the number of deaths for next 30 days.

Covid-19 Data Source:

For this project, the latest dataset operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE) has been used. This is the most cleaned and updated public data source which contains data since 22 January,2020. Data source URL-https://github.com/CSSEGISandData/COVID-19

2. Analysis and Design

Method & System Flow:

In order to achieve the objectives and get the prediction, standard ML process flow has been followed. The following is the flow diagram shows the step by step process to predict results and evaluate the models.

At the beginning of the experiment, various EDA(Exploratory Data Analysis) has been done to get to know the data sets and see different scenarios for different countries. After EDA, the entire datasets are splitted to training by 75 % and test by 25%. With training data are being fitted to the polynomial and SVM models. After fitting the training data to the models, we are built with test data to get the predicted results.

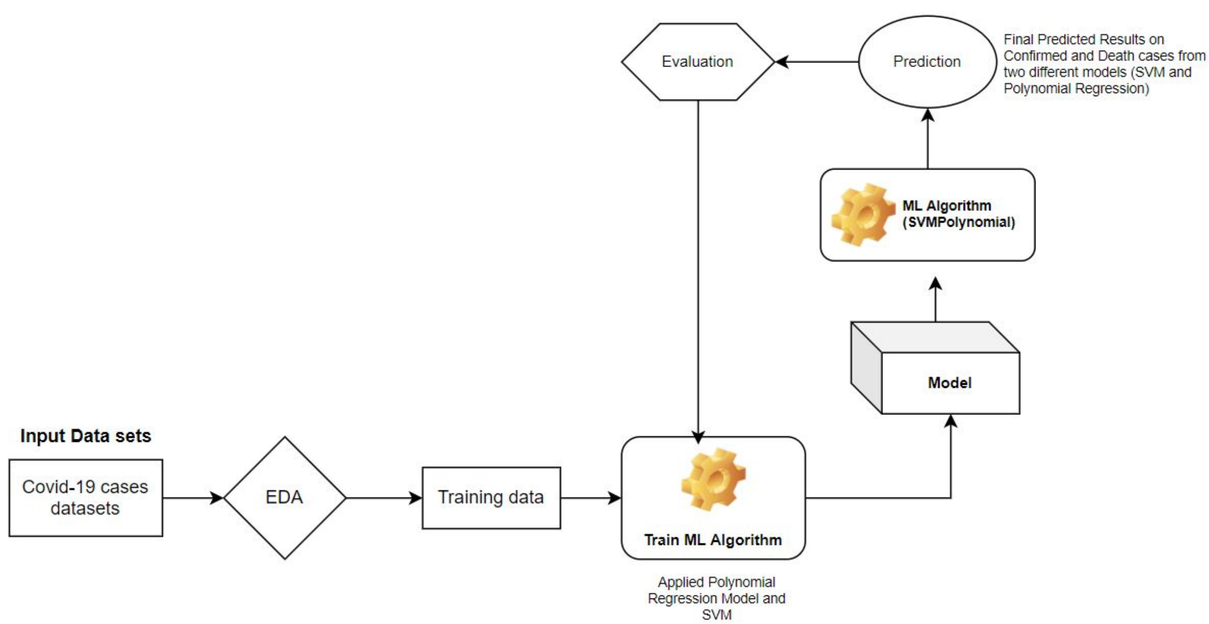

Fig: Flow chart to predict Covid-19 cases

Review of Algorithm

SVM:

The *support vector machine* (SVM) is a predictive analysis data-classification algorithm that assigns new data elements to one of labeled categories

SVM algorithm solves regression problems which is called Support Vector Regression(SVR). In this experiment we have used a Polynomial kernel to the SVM model with a degree of 5.

In comparison to other classifiers, support vector machines produce robust, accurate predictions, are least affected by noisy data, and are less prone to overfitting.

Polynomial Regression

In statistics, polynomial regression is a form of regression analysis in which the relationship between the independent variable *x* and the dependent variable *y* is modelled as an *n*th degree polynomial in *x*. In this experiment , a polynomial regression model has been used to predict both confirmed and death cases for the next 30 days.

Evaluation Methods of the models

In this experiment, MSE, MAE and R-square has been implemented to see the magnitude of the error of the models.

MSE

The Mean Squared Error (or MSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of error.

MAE

The Mean Absolute Error (or MAE) is the average of the absolute differences between predictions and actual values. It gives an idea of how wrong the predictions were

R^2 matric(Coefficient of Determination):

The R^2 (or R Squared) metric provides an indication of the goodness of fit of a set of predictions to the actual values. In statistical literature, this measure is called the coefficient of determination.

This is a value between 0 and 1 for no-fit and perfect fit respectively.

3. Experiment Results

Inputs data sets:

In the data repository for the 2019 Novel Coronavirus Visual Dashboard, operated by the Johns Hopkins University Center for Systems Science and Engineering (JHU CSSE), there are several types of file containing different sets of information. Out of which I have picked up the following file as part of the project and started exploring those:

Time_series_covid19_confirmed_global.csv, time_series_covid19_deaths_global.csv and 12-23-2020.csv

From the data source the following dataset(csv files) has been downloaded for EDA and to feed into the model.

1. **Confirmed cases by country/region from January 22, 2020 to December 25, 2020 -**

```
#exporting confirmed cases dataset(CSV file) into confirmed_cases
confirmed_cases=pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c
```

```
confirmed_cases.head()
```

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	12/16/20	12/17
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	49161	49
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	50637	51
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	93507	93
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	7446	7
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	16407	16

5 rows × 343 columns

Fig1 : Input dataset - Confirmed cases by country

2. **Number of deaths by country/region from January 22, 2020 to December 25, 2020 -**

```
[85]: #exporting confirmed cases dataset(CSV file) into confirmed_cases
deaths_reported=pd.read_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/c
```

```
[86]: deaths_reported.head()
```

[86]:

	Province/State	Country/Region	Lat	Long	1/22/20	1/23/20	1/24/20	1/25/20	1/26/20	1/27/20	...	12/16/20	12/17
0	NaN	Afghanistan	33.93911	67.709953	0	0	0	0	0	0	...	2011	2
1	NaN	Albania	41.15330	20.168300	0	0	0	0	0	0	...	1040	1
2	NaN	Algeria	28.03390	1.659600	0	0	0	0	0	0	...	2631	2
3	NaN	Andorra	42.50630	1.521800	0	0	0	0	0	0	...	79	
4	NaN	Angola	-11.20270	17.873900	0	0	0	0	0	0	...	379	

5 rows × 343 columns

Fig2 : Input dataset - Death reported by country

3. **Latest data(confirmed cases, deaths, recovered) by country and date-**

```
[10]: latest_data.head()
```

[10]:

	FIPS	Admin2	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	Combin
0	NaN	NaN	NaN	Afghanistan	2020-12-24 05:23:04	33.93911	67.709953	50433	2117	39692	8624.0	Afgh
1	NaN	NaN	NaN	Albania	2020-12-24 05:23:04	41.15330	20.168300	54317	1117	29799	23401.0	
2	NaN	NaN	NaN	Algeria	2020-12-24 05:23:04	28.03390	1.659600	96549	2696	64401	29452.0	
3	NaN	NaN	NaN	Andorra	2020-12-24 05:23:04	42.50630	1.521800	7669	82	7106	481.0	/
4	NaN	NaN	NaN	Angola	2020-12-24 05:23:04	-11.20270	17.873900	16931	393	9729	6809.0	

Fig3 : Input dataset - latest dataset of confirmed,death, recovered, active cases by country/date

Exploratory Data Analysis

At beginning of the exploration, observe the statistical summary on the 'latest_data' on which the experiment will be conducted for both confirmed and death cases and found the summary as below -

```
latest_data.describe()
```

	FIPS	Lat	Long_	Confirmed	Deaths	Recovered	Active	Incident_Rate	Case_Fatality_Ratio
count	3263.000000	3890.000000	3890.000000	3.976000e+03	3976.000000	3.976000e+03	3.973000e+03	3890.000000	3933.000000
mean	32405.273675	36.004323	-72.036928	1.980952e+04	435.321177	1.116544e+04	8.200346e+03	5381.673934	1.871041
std	18007.161500	12.968793	53.506558	9.944779e+04	2541.776549	7.784494e+04	5.650906e+04	3077.386200	3.430190
min	66.000000	-52.368000	-174.159600	0.000000e+00	0.000000	0.000000e+00	-1.005248e+06	0.000000	0.000000
25%	19052.000000	33.274355	-96.591968	6.477500e+02	9.000000	0.000000e+00	5.420000e+02	3248.643067	0.940185
50%	30069.000000	37.953523	-86.854220	1.799500e+03	29.000000	0.000000e+00	1.559000e+03	5359.231433	1.509573
75%	47040.000000	42.217931	-77.504500	7.178250e+03	104.000000	0.000000e+00	4.662000e+03	7234.553323	2.321186
max	99999.000000	71.706900	178.065000	2.503693e+06	61614.000000	1.901307e+06	2.278325e+06	26150.800198	182.562175

Fig4: Statistical summary of latest dataset

However, before going further for the exploration, NULL values are ensured as below image for the focused features that is, dates, confirmed cases and deaths. It was found that data set was quite clean and no null values available for the concerned features.

```
latest_data.isna().sum()
```

```
FIPS                   713
Admin2                 708
Province_State         170
Country_Region           0
Last_Update              0
Lat                     86
Long_                   86
Confirmed                0
Deaths                   0
Recovered                0
Active                   3
Combined_Key             0
Incident_Rate           86
Case_Fatality_Ratio     43
dtype: int64
```

Fig5 : Summary on Null values over 'Latest_dataset'

From the dataset, following trends for total deaths, recovery and confirmed cases are observed since January 22, 2020. It shows active cases have been rising followed by recovery cases since January, 2020. However, the number of deaths seems significantly low and in a flat state compared to confirmed cases till now.

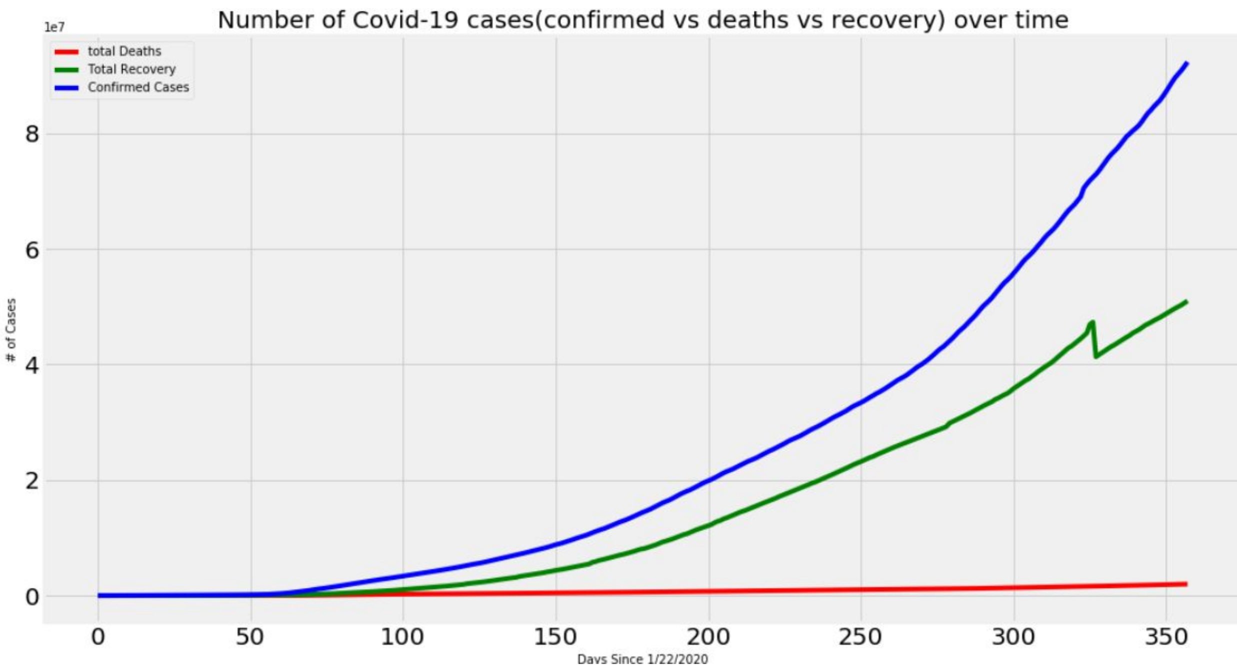

Fig6 : Confirmed cases, Recovery and death trend since January, 2020

To get into more deeper analysis, mortality rate vs recovery rate was plotted since January, 2020 as below. From the graph it has been observed that the mortality trend is pretty low compared to recovery rate. However, between day 50 to 75, there was quite a drop in recovery rate, which eventually improved by days.-

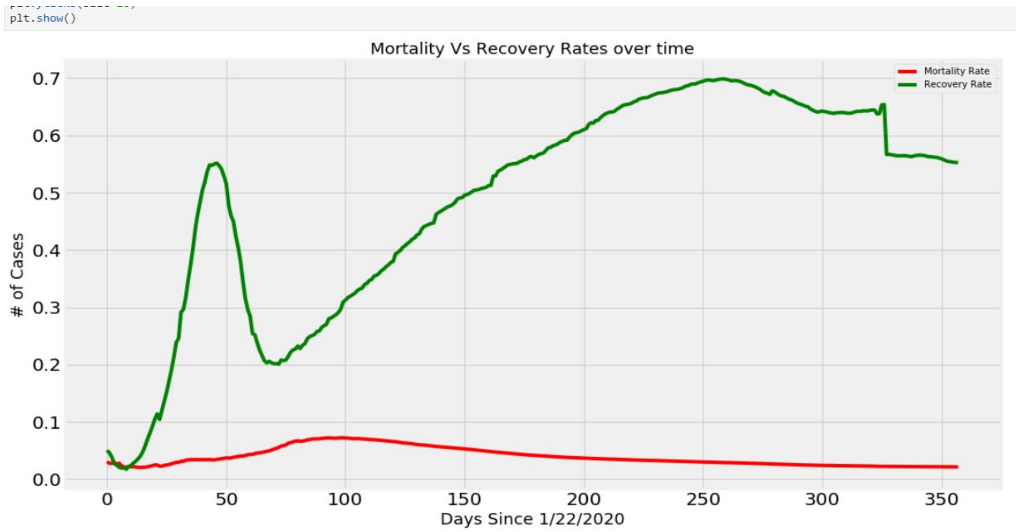

Fig7 : Trend on Mortality rate and recovery rate

To get country specific trends on the confirmed cases, three countries were focused - Bangladesh, Malaysia, and China. From the below graph, it can be seen that Bangladesh started to rise its cases after around 100th days, whereas Malaysia confirmed cases were quite stable since recently it started to rise from the end of last year. On the other hand, China cases rose at the beginning and remain stable since then.

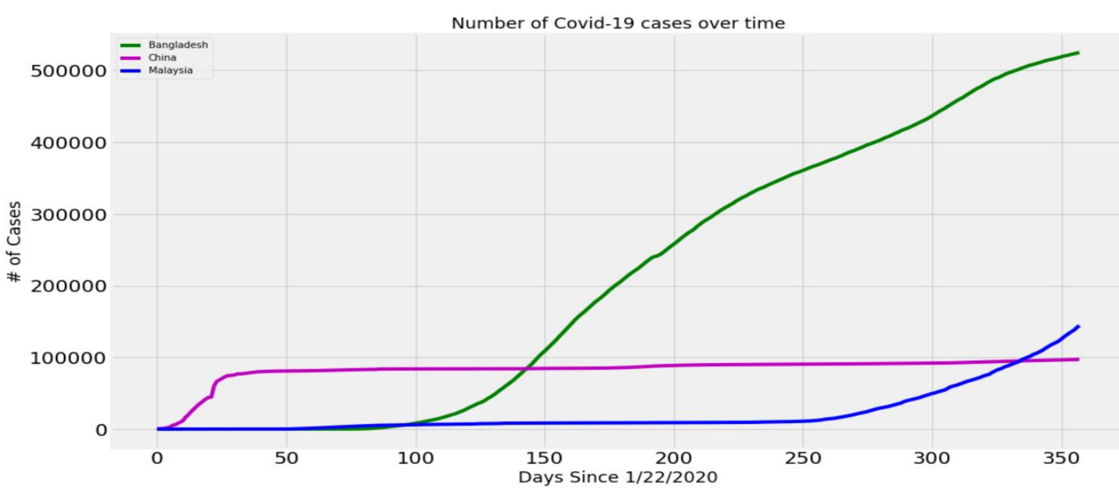

Fig8 : Trend on confirmed cases for Bangladesh, Malaysia and China

Output Data

1. Confirmed Cases

At the beginning of the modeling, string formatted date values are converted into date format and appended next 30 dates into forecast_dates list as this model will predict for next 30 days confirmed cases. After that, the confirmed case data set is split into testing and training data through the train_test_split function of sklearn package by 75%:25% ratio.

1.1 Polynomial Regression prediction-

After training the model following result was obtained for test vs predicted data through polynomial regression model -

```
#Plotting test data vs Polynomial predicted data
plt.plot(y_test_confirmed)
plt.plot(test_linear_pred)
plt.legend(['Test Data','Poloynomial Reggression Predictions'])
```

```
<matplotlib.legend.Legend at 0x2b7e5256d48>
```

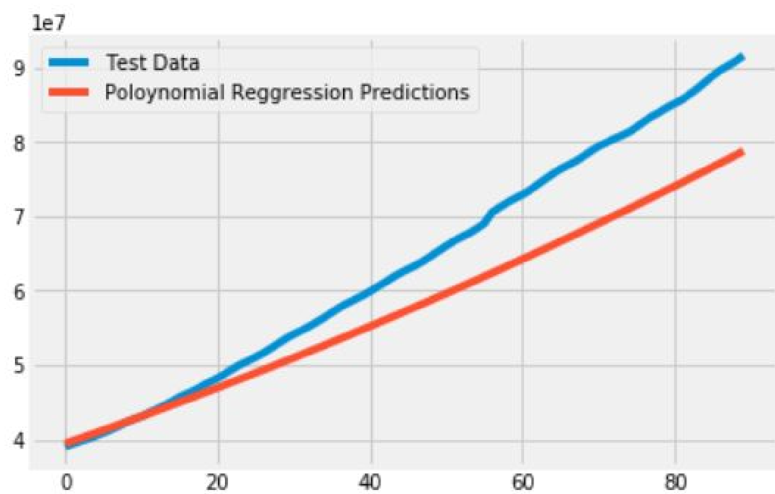

Fig9 : Test Vs predicted trend on polynomial regression model

1.1.2 Polynomial Regression evaluation:

To evaluate the model, MAE, MSE and R^2 matrix functions were implemented to see the magnitude of the errors-

1. Following MAE and MSE values were observed for the polynomial regression model -

Polynomial Regression Model Evaluation

```
#Calculate MAE and MSE
print('MAE:', mean_absolute_error(test_linear_pred, y_test_confirmed))
print('MSE:', mean_squared_error(test_linear_pred, y_test_confirmed))
```

```
MAE: 5754969.287654901
MSE: 50365668396381.63
```

Fig10 : MAE & MSE scores for polynomial regression model

From evaluating the model, it was found that the model has around 7.9% mean absolute percentage error as below-

```
#Defining MAPE function
def MAPE(Y_actual,Y_Predicted):
    mape = np.mean(np.abs((Y_actual - Y_Predicted)/Y_actual))*100
    return mape
# Using MAPE error metrics to check for the error rate and accuracy level
LR_MAPE= MAPE(y_test_confirmed,test_linear_pred)
print(LR_MAPE)
```

```
7.920414140335636
```

Fig11 : MAPE score for polynomial regression model

2. Coefficient of Determination (R2) score was found to be around 0.79 as the image below-

```
]: r2=r2_score(y_test_confirmed, test_linear_pred)
   print(r2)

   0.7857954955821307
```

Fig12 : R^2 score for polynomial regression model

1.2. SVM prediction-

To verify the experiment with different models, at this stage the SVM model was implemented to see the prediction results. Where the test and train data were fitted to SVM. After modeling SVM with kernel ‘poly’ and degree =5, following result was obtained for test vs predicted data, which has quite a lot variance than real -

```
#svn_confirmed cases for next 30 Days = svm_search.best_estimator

svm_confirmed= SVR(shrinking=True, kernel='poly',gamma=0.01, epsilon=1,degree=5,C=0.1)
svm_confirmed.fit(X_train_confirmed, y_train_confirmed)
svm_pred=svm_confirmed.predict(future_forecast)
```

Plotting test and predicted data using SVM Model

```
svm_test_pred = svm_confirmed.predict(X_test_confirmed)
plt.plot(y_test_confirmed)
plt.plot(svm_test_pred)
plt.legend(['Test Data', 'SVM Prediction'])
```

```
<matplotlib.legend.Legend at 0x2b7e55e2c08>
```

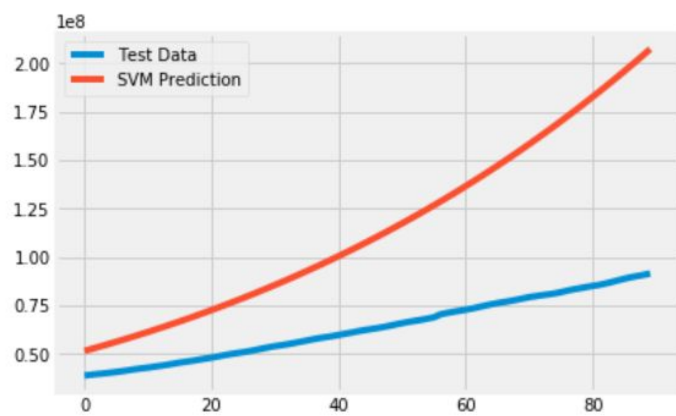

Fig13 :Test Vs Predicted trend through SVM

1.2.1 SVM evaluation:

As from the above SVM plot, the difference seems to be significant between real and predicted data. Here, MAE and MSE values were observed for the model as below, which also showed a high number of errors. Even the MAPE score show almost 92% error for the model.-

SVM Model Evaluation

```
print('MAE:', mean_absolute_error(svm_test_pred, y_test_confirmed))
print('MSE:', mean_squared_error(svm_test_pred, y_test_confirmed))
```

```
MAE: 51734104.92227845
MSE: 3554471293232899.0
```

Fig14 : MAE and MSE score for SVMl

Mean Absolute Percentage Error

```
# Using MAPE error metrics to check for the error rate and accuracy level
LR_MAPE= MAPE(y_test_confirmed,svm_test_pred)
print(LR_MAPE)
```

```
92.91745522786275
```

Fig15 : MAPE for the SVM result

3. Coefficient of Determination (R2) score was found to be around 0.79 as the image below-

```
r2=r2_score(y_test_confirmed, svm_test_pred)
print(r2)
```

```
-14.015942107121802
```

Fig16 : R^2 matric score for SVM model

1.3 Final output on confirmed cases prediction:

1.3.1 Polynomial regression prediction on confirmed cases:

Following are the final predicted vs confirmed cases using both polynomial and SVM models for next 30 days -

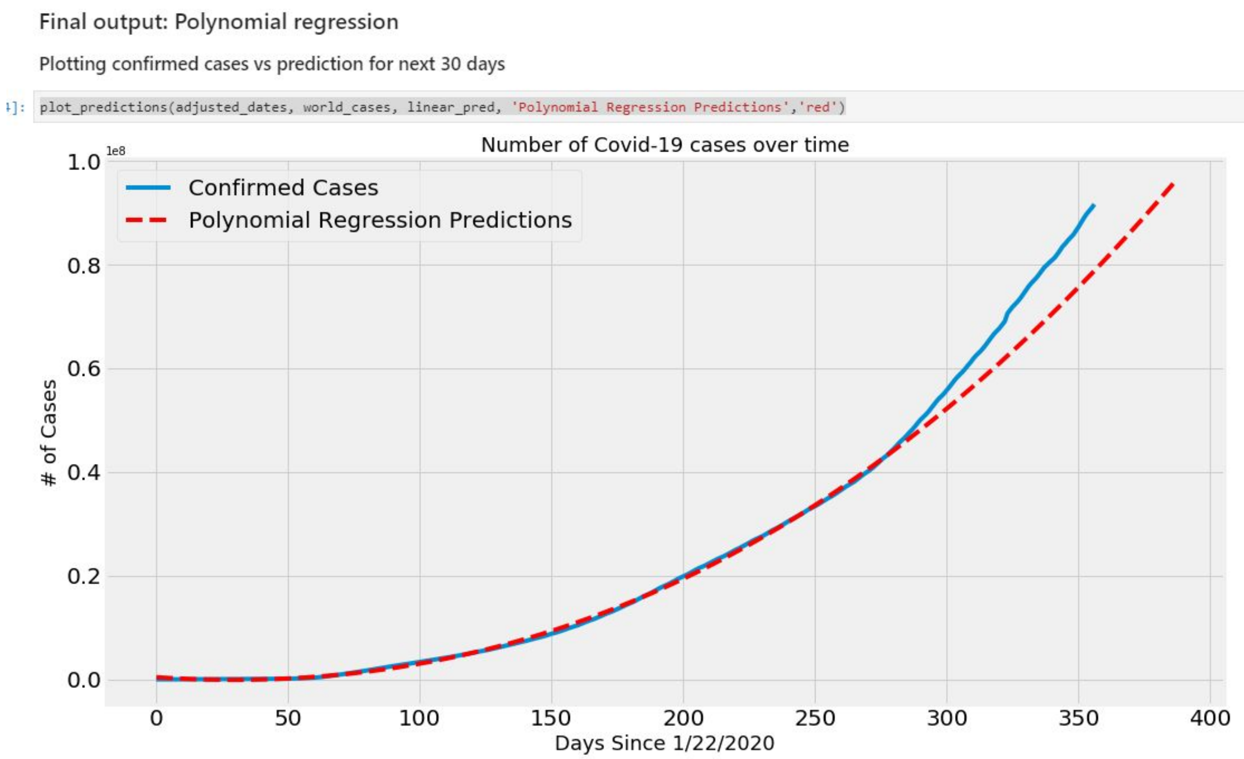

Fig17 : Final predicted Confirmed cases trend on polynomial regression model

1.3.2 SVM Prediction on Confirmed Cases:

Final output: SVM

Plotting Confirmed cases vs prediction for next 30 days

```
[110]: plot_predictions(adjusted_dates, world_cases, svm_pred, 'SVM Predictions','green')
```

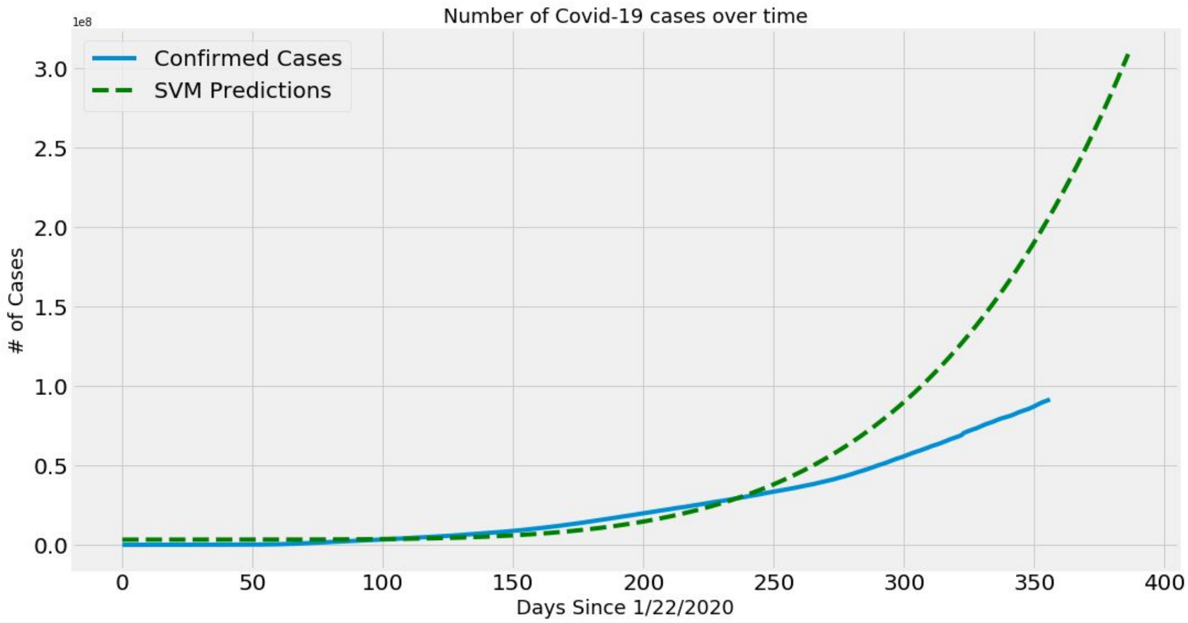

Fig18 : Final predicted Confirmed cases trend on SVM

2. Number of death prediction

2.1 Polynomial regression prediction:

To predict the number of deaths for next 30 days, polynomial regression algorithm was implemented. Following result was obtained for test vs predicted data, which shows quite opposite prediction -

Plotting Polynomial regression for test vs predicted data

[79]:
```
plt.plot(y_test_deaths)
plt.plot(test_linear_pred)
plt.legend(['Test Data for # of Deaths','Poloynomial Reggression Predictions for Deaths'])
```

[79]: <matplotlib.legend.Legend at 0x2b7e5764bc8>

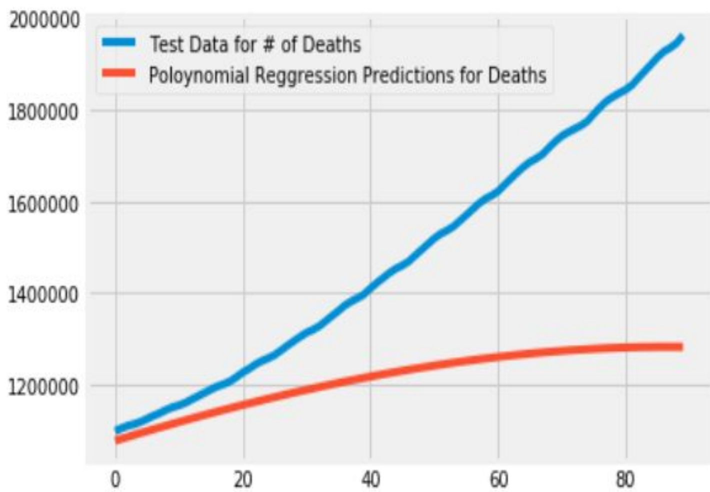

Fig19 : Test Vs Predicted trend on Polynomial Regression Model

2.1.1 .Polynomial Regression evaluation:

To evaluate the model, MAE, MSE and R^2 matric values were implemented to see the magnitude of the error-

1. Following MAE and MSE values were observed for the model -

From evaluating the model, it was found that the model has around 16% mean absolute percentage error with below MAE and MSE scores, which are significantly high.

Polynomial regression evaluation for death prediction

```
print('MAE:', mean_absolute_error(test_linear_pred, y_test_deaths))
print('MSE:', mean_squared_error(test_linear_pred, y_test_deaths))
```

```
MAE: 271090.6697476223
MSE: 114255509000.87125
```

```
# Using MAPE error metrics to check for the error rate and accuracy level
LR_MAPE= MAPE(y_test_deaths,test_linear_pred)
print(LR_MAPE)
```

```
16.398601497751876
```

2. Coefficient of Determination (R2) score was found to be around -0.68 as the image below-

```
'5]: r2=r2_score(y_test_deaths, test_linear_pred)
     print(r2)
```

```
-0.6793161433663191
```

Fig20 : MAE, MSE and R^2 matric score on Polynomial Regression Model

2.2 SVM prediction on number of deaths

Next, SVM model was implemented to predict death cases for the next 30 days. After modeling SVM with kernel ‘poly’ and degree =5, following result was obtained for test vs predicted data -

Plotting SVM for test vs predicted data for number of deaths

```
[81]: svm_test_pred = svm_deaths.predict(X_test_deaths)
      plt.plot(y_test_deaths)
      plt.plot(svm_test_pred)
      plt.legend(['Test Data for #of Deaths', 'SVM Prediction(Deaths)'])
```

```
[81]: <matplotlib.legend.Legend at 0x2b7e56ec548>
```

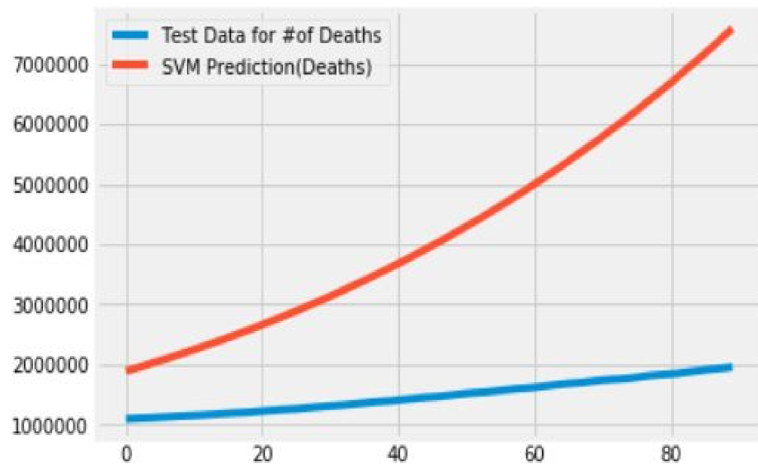

Fig21 : Test Vs Predicted trend on SVMl

2.1 SVM evaluation:

As from the above SVM plot, the difference seems to be significantly high between real and predicted data, MAE and MSE values were observed for the model as below, which also showed high number of errors.Even the MAPE score show unexpectedly high by 298% error for the model.-

SVM evaluation for death prediction

```
print('MAE:', mean_absolute_error(svm_test_pred, y_test_deaths))
print('MSE:', mean_squared_error(svm_test_pred, y_test_deaths))
```

```
MAE: 4268945.636868819
MSE: 22227981242771.63
```

```
# Using MAPE error metrics to check for the error rate and accuracy level
LR_MAPE= MAPE(y_test_deaths,svm_test_pred)
print(LR_MAPE)
```

```
298.6293605859902
```

Fig22 : MAE and MSE Score for SVM

4. Coefficient of Determination (R2) score was found to be around 0.79 as the image below-

```
r2=r2_score(y_test_deaths, svm_test_pred)
print(r2)
```

```
-325.70466450020797
```

Fig23: R^2 Score for the SVM

2.2 Final output on death prediction:

Following are the final predicted vs number of deaths using both polynomial and SVM models for next 30 days -

Polynomial Regression prediction -

```
]: plot_predictions_deaths(adjusted_dates,total_deaths, linear_pred_death, 'Polynomial Regression','red')
```

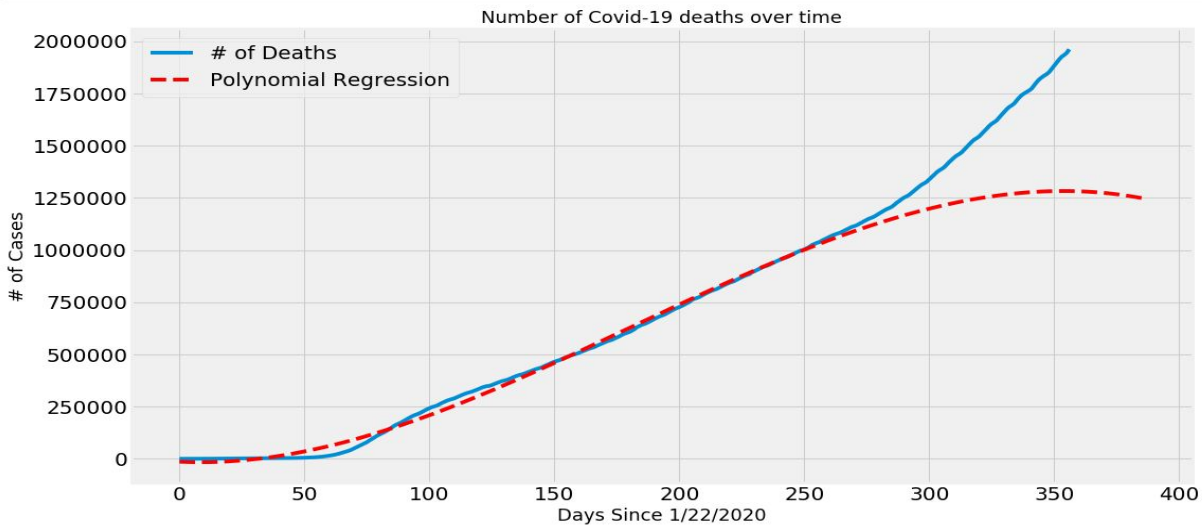

Fig24 : #of Death prediction using Polynomial Regression Model

SVM prediction:

Plotting #of Deaths vs prediction for next 30 days

```
: plot_predictions_deaths(adjusted_dates,total_deaths, svm_pred_death, 'SVM','red')
```

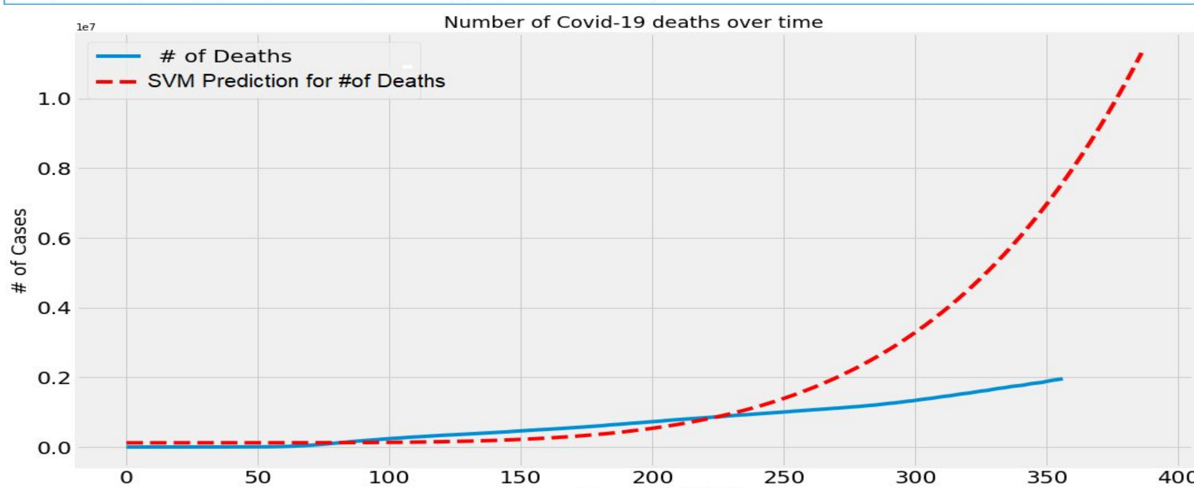

Fig25 : #of Death prediction using SVM

4.Discussions

Method Evaluation and Performance

1. Model evaluation for ‘Confirmed Cases’:

   In this experiment, two models - Polynomial regression and SVM are being used to calculate next 30 days predicted confirmed cases. Through evaluating these models for confirmed cases, following evaluation scores are found for two models -

Model	MAE Score	MSE Score	MAPE	R^2 matric score
Polynomial Regression	5965791.891618436	53441280372402.6	8%	0.78
SVM	51734104.92227845	3554471293232899.0	93%	-14.015

Table 1 : Polynomial regression Vs SVM evaluation on Confirmed Cases

From the above table, considering MAE , MSE, MAPE and R^2 matric score to evaluate two models, it can be said that Polynomial regression model performed well compared to SVM models in terms of confirmed cases. Polynomial regression model holds less error in all off the error matric than SVM.

Therefore, the Polynomial regression model works better to predict covid-19 confirmed cases for next 30 days.

2. Model evaluation for Death cases:

   In this experiment, two models - Polynomial regression and SVM are being used to calculate next 30 days predicted confirmed cases. Through evaluating these models for confirmed cases, following evaluation scores are found for two models -

Model	MAE Score	MSE Score	MAPE	R^2 matric score
Polynomial Regression	271090.6697476223	114255509000.87125	16%	0.67
SVM	4268945.636868819	22227981242771.63	298%	-325.7

Table 2 : Polynomial regression Vs SVM evaluation on Death Cases

From the above table, considering MAE , MSE, MAPE and R^2 matric score to evaluate two models, it can be said that the Polynomial regression model performed well compared to SVM models in terms of confirmed cases. Polynomial regression model holds less error in all off the error matrices than SVM.

Therefore, the Polynomial regression model works better to predict covid-19 confirmed cases for next 30 days.

Final Results

From the above two evaluation tables, it has been observed that polynomial regression algorithm works better on prediction than SVM. In both confirmed and death prediction, polynomial regression had the significantly lowest error rate i.e 8% in confirmed cases and 16% in death cases, which justifies that Polynomial Regression model servers the better result.

Future Improvements

To get better prediction in future Autoregressive Integrated Moving Average (ARIMA) algorithm can be tested. The ARIMA model is popular because of its known statistical properties and the well-known Box–Jenkins methodology in the modeling process. It is one of the most effective linear models for seasonal time series forecasting. The ARIMA model has advantages in its well-known statistical properties and effective modeling process. It can be easily realized through mainstream statistical software. The model can be used when the seasonal time series are stationary and have no missing data

5. Conclusion

The cases of COVID-19 are rapidly increasing day by day. Machine Learning (ML) and Cloud Computing can be deployed very effectively to track the disease, predict growth of the epidemic and design strategies and policies to manage its spread. Similarly, this experiment is being applied to foresee the growth using ML models. Supervised ML models like Polynomial Linear Regression and Support Vector Machine (SVM)have been implemented in this project to forecast and determine the outbreak of Covid-19. The models were trained with 70% training data and tested with the remaining 30% of the data. To forecast Confirmed and death cases the model implemented using polynomial regression happened to be the best model than SVM terms of MAE (%) value which are 0.8% and 16% respectively.

6. References

1. Coronavirus: All You Need to Know in 500 Words. https://www.aljazeera.com/news/2020/9/20/coronavirus-all-you-need-to-know-in-under-500-words. Accessed 5 Jan. 2021.
2. H, Wie Kiang. “COVID-19 Outbreak Prediction Using Machine Learning Algorithm.” Medium, 1 Aug. 2020, https://towardsdatascience.com/covid-19-outbreak-prediction-using-machine-learning-algorithm-ce5641bd55bf.
3. Total Death of Coronavirus in the World - Google Search. https://www.google.com/search?q=total+death+of+coronavirus+in+the+world&rlz=1C1GCEU_enBD882BD882&oq=total+death+&aqs=chrome.1.69i57j0i20i263i457j0i67j0l5.3706j0j4&sourceid=chrome&ie=UTF-8. Accessed 4 Jan. 2021.WHO Director-General’s Opening Remarks at the Media Briefing on COVID-19 - 11 March 2020.
4. https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020. Accessed 5 Jan. 2021.