Fall 2022

Prediction of Covid-19 Cases using Machine Learning and Data Analysis

Anthony Loritz

Prediction of Covid-19 Cases using Machine Learning and Data Analysis

By

Anthony Loritz
B.S., Benedictine University, 1995

GRADUATE CAPSTONE SEMINAR PROJECT

Submitted in partial fulfillment of the requirements

For the Degree of Master of Science,

With a Major in Computer Science

Governors State University
University Park, IL 60484

2022

ABSTRACT

The app for the Covid-19 Machine Learning and Data Analysis Project was developed as a requirement for the Graduate Capstone Seminar Project for the Master of Science Degree with a Major in Computer Science. Machine learning algorithms and models can be used to predict important future Covid case data such as infection rates (outbreaks) and mortality rates. The predicted case data can be used to help control Covid, aid in better treatments and vaccines, or even lead to a cure. This app's main objectives are to forecast Covid-19 global new cases based on previous known new case data, measure the LSTM machine learning model's performance with regression accuracy check metrics based on the actual and predicted global new cases values of the testing dataset, and conduct global and country analysis of Covid-19 new cases and new deaths data. This app's objectives do not aim to replace or enhance any existing Covid-19 Machine Learning and Data Analysis Apps; it is an app developed for academic purposes. The deadline for the Covid-19 Machine Learning and Data Analysis App is 11/28/2022 at 9:00 AM.

Table of Contents

1 Project Description

The Covid-19 Machine Learning and Data Analysis App performs data analysis to obtain interesting and important global and country Covid-19 data statistics and machine learning to make next day predictions on global new cases from previously known global new cases data. Data analysis was done using the total global new cases counts grouped by date and total new cases and new deaths counts grouped by country. Machine learning was completed by using an artificial recurrent neural network called LSTM (Long short-term memory) with ninety percent of the total dataset days for training and the remaining ten percent for testing. Forecasting of the next day (future) global Covid-19 new cases count was predicted using two different methods for both parts one and two to validate method accuracy. Important machine learning metrics were calculated twice using different parameters to measure the algorithm's accuracy and effectiveness. A multitude of graphs (line, scatter, vertical bar, and horizontal bar) were plotted to visualize the data for parts one and two. Part three plots a tree map and daily, weekly, and monthly global Covid-19 case statistics.

1.1 Introduction

The datasets used for this project are obtained from GitHub Covid-19 repositories [1, 2]. The dataset was grouped by date for new cases using different methods for parts one and two to get the total daily global new cases count, change the coding dynamics, and to test for data output reliability. The training dataset was created using ninety percent of the total dataset days, prepared by scaling and conversion, and the LSTM model was constructed, compiled, and trained. The testing dataset was created and prepared by array indexing of scaled values from the most recent (previous) sixty dataset days. The testing dataset was converted and reshaped to and the model's forecasted (predicted) next day new case value was obtained using two different methods. The model's forecasted value was the same for both methods in both parts one and two. A table of values for the actual and predicted global new case counts for ten percent of the total dataset days was created along with line, scatter, v-bar, and h-bar graphical representations. The calculations of regression accuracy check metrics: MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and R-Squared (Coefficient of Determination) were the same for both sets of different parameters for parts one and two.

Data analysis includes but is not limited to counts, mean, standard deviation, minimum, maximum, median (25% and 75% included), and the sums of all valid columns for the original dataset and the datasets grouped by date and by country. Various tabular views and graphical representations of the global and countries new cases and deaths data were created to present the data and results in a variety of different formats.

1.2 Motivation

The main motivation for doing this project was to fulfill the Capstone requirement for the Master's Degree in Computer Science. Additional motivational factors for doing this project were to learn about machine learning and data analysis, the challenge of successfully completing a project with no prior experience in the project topics, and to create/discover machine learning and data analysis results that can be potentially used to help contain and eventually eradicate the Covid-19 virus.

2 Project Technical Description

The project was designed into three parts based on the different data (.csv) files, data analysis methods, and coding logic that was implemented. Attempts to put all the code into one .ipynb file resulted in the program failing to execute to completion. Parts one and two are just derivatives of each other. This was done to learn, compare, and measure the accuracy of different coding styles and techniques. The outputs were the same for both parts one and two. The required dependencies (libraries, packages, modules, etc.) were imported for all three parts. One .csv data file [1] was imported for parts one and two and five .csv data files [2] for part three from the GitHub Covid-19 repositories [1, 2].

In parts one and two, the imported .csv data file [1] was read and statistical analysis of the imported data file was completed to figure out the best solutions for obtaining the proper datasets the app needed to execute correctly. The imported datafile was grouped by date to obtain the global values of all the valid columns. The cumulative global new cases counts were grouped by date and line, scatter, vertical (v) -bar, horizontal (h)-bar graphs were plotted. Statistical analysis and line, scatter, v-bar, h-bar graphs were also completed for the highest and lowest Covid-19 new cases counts by date.

The project team member had no previous courses, background, and experience in machine learning and data science. The project team member consulted with Dr. Yunchuan Liu concerning these issues. Dr. Liu recommended that the LSTM machine learning model be used. This is the reason LSTM was chosen for the Covid-19 Machine Learning and Data Analysis Project. The main advantages and benefits of using the LSTM machine learning model is the provision of a substantial range of parameters

such as learning rates, input biases, and output biases. Fine adjustments are not required. The intricacy to update each weight is decreased to O(1) with LSTM [3].

Using two different methods for the LSTM machine learning model, a new data frame was created with the new cases column, converted to a NumPy array, the number of rows to train the model was set at ninety percent of the total group by date dataset for new cases, a scaled training data set created and split, and the split datasets were converted to NumPy arrays and reshaped to build the LSTM model. The LSTM model was built, compiled, and trained with the split datasets with a batch size of one and an epoch of fifty. After the model was trained the testing dataset consisting of an array of indexed scaled values was created, two testing datasets were created from the indexed scaled testing dataset, converted to a NumPy array, reshaped, accuracy on existing previous sixty days of actual new cases was tested, and the predicted new case values on test data was calculated. The LSTM machine learning model's predicted new cases values were compared to the actual new cases, displayed in tabular format, and plotted (line, scatter, v-bar, and h-bar graphs). The LSTM model's forecasted values were obtained using two different methods. The forecasted values for both methods are identical within each part with different values for each part. Regression accuracy check metrics were calculated using two different sets of parameters. The results for both sets of parameters are identical within each part with different values for each part. See the source code or/and Part 4: Covid-19 Prediction (Figures 6-9 and Figures 13-16) of this document for comparison of the forecasted values and regression accuracy check metrics calculations for within each part and with each other if further clarification is required.

Data analysis of various combinations of data grouped by date include counts, mean, standard deviation, minimum, maximum, median (25% and 75% included), and the sums of all valid columns. Other date data analysis information includes:

1) Global cumulative new cases count for each dataset date.
2) The dates with the top ten new cases and new deaths counts.
3) The dates with the lowest ten new cases and new deaths counts.
4) Global cumulative totals of confirmed cases, new cases, confirmed deaths, and new deaths for each dataset date.
5) Global cumulative totals of new confirmed cases and new confirmed deaths for all countries i.e., the cumulative total of new confirmed cases and new confirmed deaths from the cumulative set of dataset dates.
6) Global mortality rate as a percentage for the cumulative set of dataset dates.
7) Daily average of new confirmed cases and new confirmed deaths.
8) Daily percentage of confirmed cases, new cases, confirmed deaths, and new deaths for each date and the previous (most recent) thirty days.
9) Largest and smallest percentages of new cases for the top ten dates.
10) Largest and smallest percentages of new deaths for the top ten dates.

Line, scatter, v-bar, and h-bar graphs were plotted for the global cumulative new cases count for each dataset date, dates with the top ten new cases and new deaths counts, and dates with the lowest ten new cases and new deaths counts.

Data analysis of various combinations of data grouped by country include counts, mean, standard deviation, minimum, maximum, median (25% and 75% included), and the sums of all valid columns. Other country data analysis information includes:

1) The top ten countries' new cases and daily new cases counts.
2) The top ten countries' new deaths and daily new deaths counts.
3) Mortality rates as a percentage for select countries, the highest sixty countries, and lowest sixty countries.
4) The lowest ten countries' new cases and daily new cases count.
5) The lowest ten countries' new deaths and daily new deaths counts.
6) The percent of confirmed cases, new cases, confirmed deaths, and new deaths for each country.
7) The largest percentages of new cases and new deaths for the top ten countries.
8) The lowest percentages of new cases and new deaths for the top sixty countries.

Line, scatter, v-bar, and h-bar graphs were plotted for the highest ten countries' new cases and new deaths counts and the lowest twenty countries' new cases count.

In parts one and two a copy of the original (live) .csv file is saved and can be used instead of the live file for program execution.

In part three, five imported .csv data files [2] were used to get data statistics, the cumulative totals of active cases grouped by country/region for each date, the cumulative sums of confirmed cases, deaths, recovered cases, and active cases by date, and graphically display:

1) A tree map plot showing each country's contribution (percentages and cumulative totals) to the overall global Covid-19 case numbers (confirmed cases, active cases, recovered cases, deaths, and daily increases).
2) A plot of daily, weekly, and monthly statistics for the global total of confirmed cases, active cases, recovered cases, and deaths.

2.1 Platform

Google Colaboratory (Colab) with Python version 3.7.15 was used for the coding of this project. Colab is easy and fun to use.

The main advantages and benefits of using Colab that I have personally experienced during the coding of this project are as follows:

1) The cost is free. All is required is to create a free Google Account, login, and select the "Keep me signed in" option. If you Choose the "Keep me signed in" option, you will only need to login for the first time and never again on the computer/device you are using.
2) No downloads or configuration required. Colab runs on any web browser on any computer.
3) All files are saved to Google Drive with 15 GB of free storage.
4) File management in Google Drive is very convenient. The configurations and operations are very similar to an email account and file explorer in Windows.
5) All files are saved automatically upon execution, unexpected crashes, and regular time intervals. Additionally, you can easily save your files manually.
6) All files can be uploaded and downloaded.
7) All program executions are processed by the Google servers, not on the local machine, and hence are extremely fast and efficient.
8) Nearly all Python packages are pre-installed. Uninstalled packages are easy to install with the **!pip install <*package name*>** command. I did not need to install any packages for this project.
9) Blocks of code can be executed one cell at a time or all at once, all in a single .ipynb file. This is ideal for machine learning and data science projects.
10) Colab has a plethora of tutorials and with a very reasonable time commitment and a little effort it is easy to learn.
11) Colab is loaded with an abundance of options. My favorites are the Run all (Ctrl + 9) option under the Runtime tab and the Help tab in the menu bar.

Colab has made the scope of this project extremely efficient. I would strongly recommend it for anyone learning and coding machine learning or/and data science projects.

2.2 Data

The data used for this project are from live .csv files read directly from the GitHub Covid-19 repository URLs [1, 2]. The data is updated daily, except on weekends. Every time the programs are executed the latest updates are reflected.
As of 11/9/2022, the date range for the df.csv file of parts one and two are from 7/1/2022 to 11/7/2022 (130 days) for 234 countries.
As of 11/9/2022, the date range for part three are from 1/22/2020 to 11/8/2022 (1,022 days) with 201 countries/regions for the confirmed_df.csv, deaths_df.csv, and recoveries_df.csv files. The latest_data.csv file has 197 countries/regions. The us_medical_data .csv file has 58 states.

The latest (updated on 11/9/2022) data information on the df.csv file used in parts one and two is as follows:

df.info()
RangeIndex: 30420 entries, 0 to 30419
Data columns (total 32 columns):

#	Column	Non-Null Count	Dtype
0	Country	30420 non-null	object
1	Date	30420 non-null	object
2	Confirmed Cases	30420 non-null	int64
3	Confirmed Deaths	30420 non-null	int64
4	New Cases	30420 non-null	int64
5	New Deaths	30420 non-null	int64
6	Income	27690 non-null	object
7	Region	24960 non-null	object
8	Population	30030 non-null	float64
9	Case_Change_7_Day_Rolling_Average	30420 non-null	float64
10	Death_Change_7_Day_Rolling_Average	30420 non-null	float64
11	year	30030 non-null	float64
12	cpm	30030 non-null	float64
13	dpm	30030 non-null	float64
14	new_cases_per_million	30030 non-null	float64
15	new_deaths_per_million	30030 non-null	float64

16	cases_income	30420 non-null	int64
17	deaths_income	30420 non-null	int64
18	cpm_income	27690 non-null	float64
19	dpm_income	27690 non-null	float64
20	Case_7_Day_income	30420 non-null	float64
21	Death_7_Day_income	30420 non-null	float64
22	Case_7_Day_income_per_million	27690 non-null	float64
23	Death_7_Day_income_per_million	27690 non-null	float64
24	cases_region	30420 non-null	int64
25	deaths_region	30420 non-null	int64
26	cpm_region	24960 non-null	float64
27	dpm_region	24960 non-null	float64
28	Case_7_Day_region	30420 non-null	float64
29	Death_7_Day_region	30420 non-null	float64
30	Case_7_Day_region_per_million	24960 non-null	float64
31	Death_7_Day_region_per_million	24960 non-null	float64

Data types: float64 (20), int64 (8), object (4)

The latest (updated on 11/9/2022) data information on the .csv files used in part three are as follows:

1) confirmed_df.info()
   RangeIndex: 289 entries, 0 to 288
   Columns: 1026 entries, Province/State to 11/8/22
   Data types: float64 (2), int64 (1022), object (2)

2) deaths_df.info()
   RangeIndex: 289 entries, 0 to 288
   Columns: 1026 entries, Province/State to 11/8/22
   Data types: float64 (2), int64 (1022), object (2)

3) recoveries_df.info()
   RangeIndex: 274 entries, 0 to 273
   Columns: 1026 entries, Province/State to 11/8/22
   Data types: float64 (2), int64 (1022), object (2)

4) latest_data.info()
   RangeIndex: 3986 entries, 0 to 3985
   Data columns (total 14 columns):

#	Column	Non-Null Count	Dtype
0	FIPS	3257 non-null	float64
1	Admin2	3262 non-null	object
2	Province_State	3810 non-null	object
3	Country_Region	3986 non-null	object
4	Last_Update	3986 non-null	object
5	Lat	3903 non-null	float64
6	Long_	3903 non-null	float64
7	Confirmed	3986 non-null	int64
8	Deaths	3986 non-null	int64
9	Recovered	3986 non-null	int64
10	Active	3986 non-null	int64
11	Combined_Key	3986 non-null	object
12	Incidence_Rate	3903 non-null	float64
13	Case-Fatality_Ratio	3921 non-null	float64

Data types: float64 (5), int64 (4), object (5)

5) us_medical_data.info()

RangeIndex: 58 entries, 0 to 57

Data columns (total 21 columns):

#	Column	Non-Null Count	Dtype
0	Province_State	58 non-null	object
1	Country_Region	58 non-null	object
2	Last_Update	58 non-null	object
3	Lat	56 non-null	float64
4	Long_	56 non-null	float64
5	Confirmed	58 non-null	int64
6	Deaths	58 non-null	int64
7	Recovered	48 non-null	float64
8	Active	48 non-null	float64
9	FIPS	58 non-null	float64
10	Incident_Rate	56 non-null	float64
11	Total_Test_Results	10 non-null	float64
12	People_Hospitalized	37 non-null	float64
13	Case_Fatality_Ratio	0 non-null	float64
14	UID	58 non-null	float64
15	ISO3	58 non-null	object
16	Testing_Rate	54 non-null	float64
17	Hospitalization_Rate	37 non-null	float64
18	Date	58 non-null	object
19	People_Tested	56 non-null	float64
20	Mortality_Rate	57 non-null	float64

Data types: float64 (14), int64 (2), object (5)

2.3 Project design

Step 1: Import the required packages.

Step 2: Read data from .csv datafile locations.

Step 3: Group the data by date for global Covid-19 statistical totals.

Step 4: Group the data by country for Covid-19 statistical totals for each country.

Step 5: Perform data analysis on the grouped data.

Step 6: Split grouped by date dataset into ninety percent training and ten percent testing.

Step 7: Build, compile, and train the LSTM machine learning model.

Step 8: Get the LSTM's predicted global new cases values and compare the predicted values with the actual values of the testing dataset.

Step 9: Perform regression accuracy check metric (MAE [Mean Absolute Error], MSE [Mean Squared Error], RMSE [Root Mean Squared Error], and R-squared [Coefficient of Determination]) calculations using sklearn.metrics.

Step 10: Get the LSTM's predicted global new cases forecasted (predicted) value.

Step 11: Visualize the machine learning and the grouped data statistics via line, scatter, v-bar, and h-bar graphs for parts one and two.

Step 12: Graph a tree map plot showing each country's contribution (percentages and cumulative totals) to the overall global Covid-19 case numbers (confirmed cases, active cases, recovered cases, deaths, and daily increases) and a plot of daily, weekly, and monthly statistics for the global total of confirmed cases, active cases, recovered cases, and deaths.

2.4 Evaluation Metrics

The four regression accuracy check metrics used to evaluate the LSTM machine learning model's performance in terms of accuracy and error rate in parts one and two are as follows:

1) MAE (Mean Absolute Error) represents the difference between the original and predicted values extracted by averaging the absolute difference over the dataset [4].

   The mathematical formula for MAE is as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

2) MSE (Mean Squared Error) represents the difference between the original and predicted values extracted by squaring the average difference over the dataset [4].

   The mathematical formula for MSE is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

3) RMSE (Root Mean Squared Error) is the error rate by the square root of MSE [4].

   The mathematical formula for RMSE is as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

4) R-squared (Coefficient of Determination) represents the coefficient of how well the values fit compared to the original values. The value from 0 to 1 is interpreted as percentages. The higher the value is, the better the model is [4].

   The mathematical formula for R-squared is as follows:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ $-$ *predicted value of* y

$\bar{y}$ $-$ *mean value of* y

The regression accuracy check metrics were calculated in parts one and two by using sklearn.metrics.

3 *Key Observations*

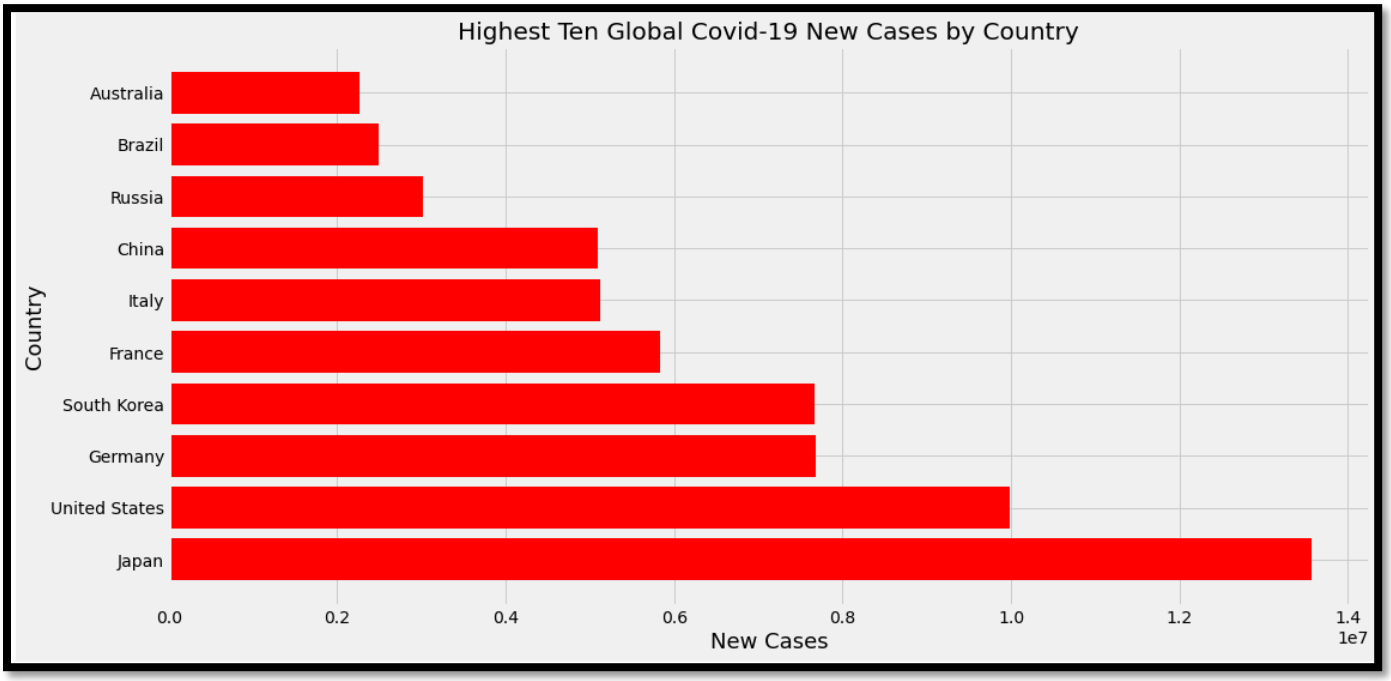

Figure 1: Horizontal bar graph showing the countries with the highest total of Covid-19 new cases from 7/1/2022 to 11/9/2022.

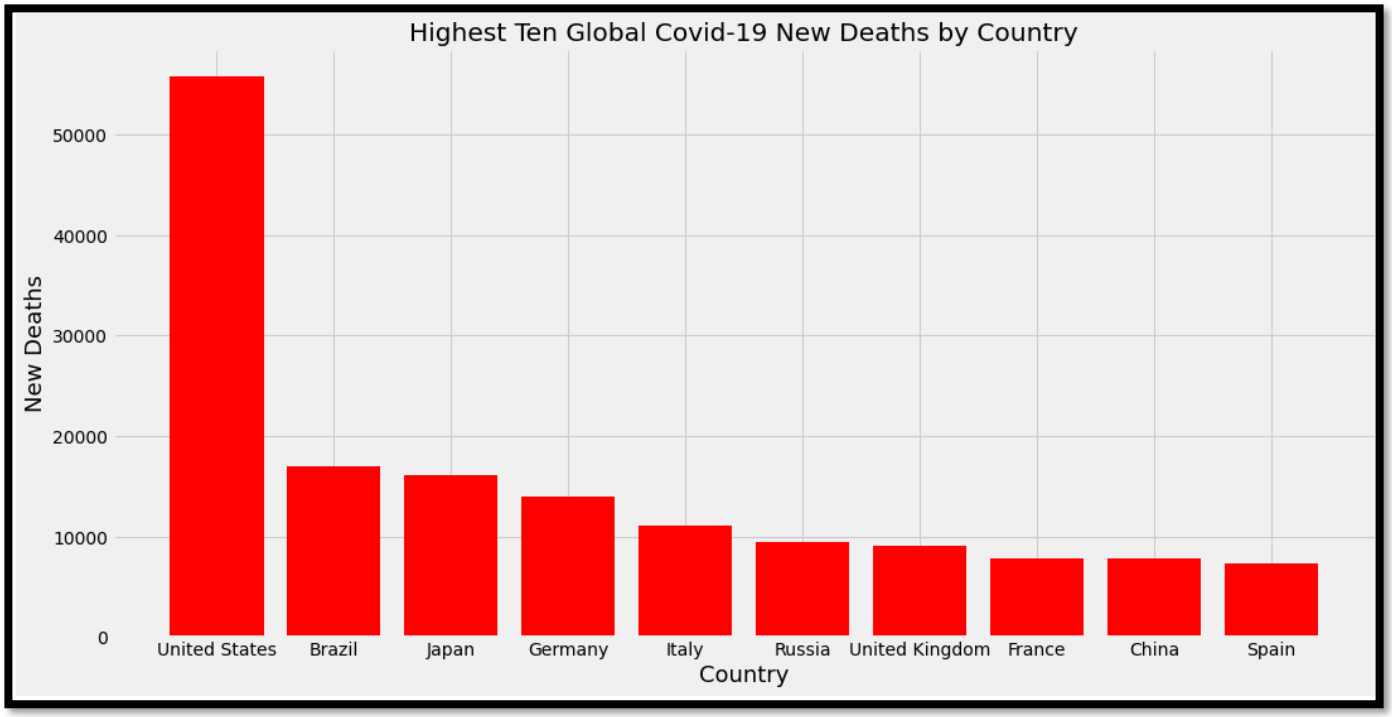

Figure 2: Vertical bar graph showing the countries with the highest total of Covid-19 new deaths from 7/1/2022 to 11/9/2022.

The graphs of Figures 1 and 2 are from Part1_CovidCapstone.ipynb and Part2_CovidCapstone.ipynb. The Covid-19 data analysis completed in parts one and two calculates that the United States had from 7/1/2022 to 11/9/2022 on average 4.288% (335,942,003) of the total world's population (7,834,144,567).

The United States is second in the world in new Covid-19 cases during this time with 11.602% (~75,594 new cases per day [9,978,371 in 132 days]), shockingly Japan leads the world with 15.78% (~102,807 new cases per day [13,570,503 in 132 days]). The global daily average of new cases is ~651,510 (85,999,294 in 132 days).

The United States leads the world in Covid-19 new deaths with 22.744% (~423 deaths per day [55,773 in 132 days]) of the global total (~1,858 deaths per day [245,223 deaths in 132 days]). The United States has a .558% Covid-19 mortality rate (one fatal

case per 179.211 new infections). The global mortality rate during this timeframe is .285% (one fatality per 350.877 new infections).

4 Covid-19 Prediction

Figures 3 to 9 are from Part1_CovidCapstone.ipynb.

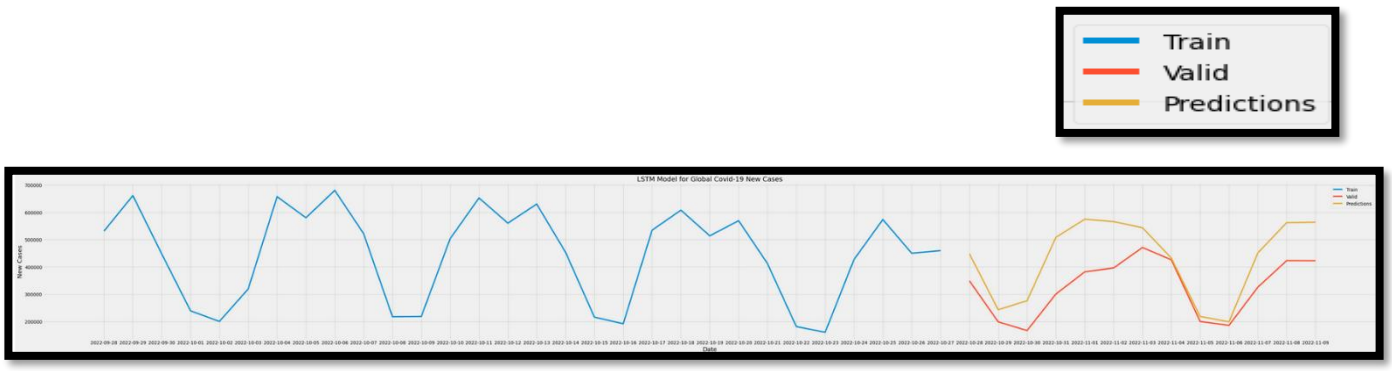

Figure 3: Line graph showing the LSTM model (training, testing, and predictions) for global Covid-19 new cases.

Please, increase the magnification of Figure 3 to view clearly. The last thirty days of the training dataset (blue line) is displayed prior to the full testing dataset (valid; orange line) and the LSTM model's predictions (yellow line) of the testing dataset. The original graph displaying all the training days was too large to view clearly in this document.

In Figure 4 below, the LSTM model's prediction results are compared to the actual results from the testing dataset. The testing dataset was set at ten percent of the total dataset length (days). This is an attempt to obtain some forecast results. Due to the real-world data, time constraints, and with the project team member having no prior coursework, background, and experience in machine learning, it is very difficult to achieve a perfect result.

Date	Actual Confirmed New Cases	Predicted Confirmed New Cases
2022-10-28	349,367	448,303
2022-10-29	198,897	243,307
2022-10-30	167,086	276,348
2022-10-31	300,907	508,628
2022-11-01	382,118	574,833
2022-11-02	396,323	566,343
2022-11-03	471,259	543,859
2022-11-04	426,414	432,820
2022-11-05	200,343	218,763
2022-11-06	185,964	198,998
2022-11-07	325,986	451,335
2022-11-08	423,187	562,553
2022-11-09	422,628	564,148

Figure 4: The LSTM model's predicted global Covid-19 new cases values compared to the actual global new cases values of the testing dataset.

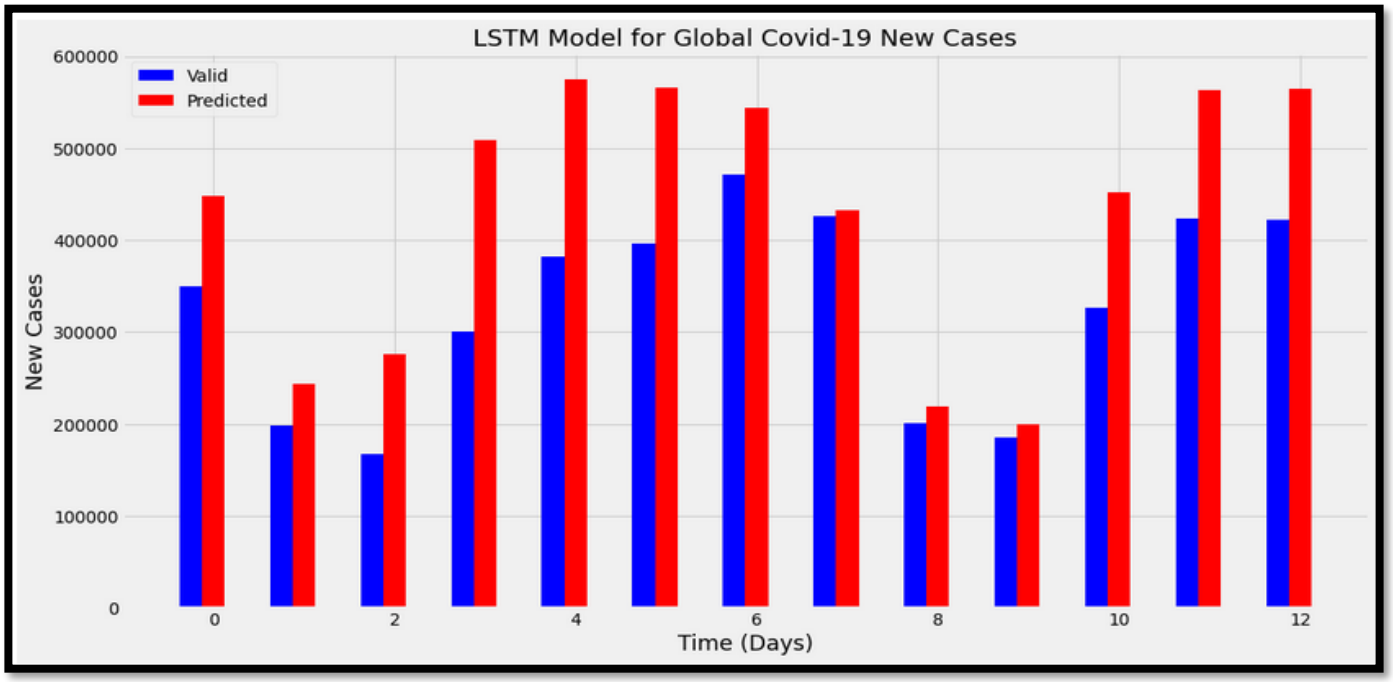

Figure 5: Vertical bar graph of the actual vs. predicted results of Figure 4.

```
# Calculate the Regression Accuracy Check metrics by sklearn.metrics.
mae = metrics.mean_absolute_error(y_test, predictions)
mse = metrics.mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse) # OR rmse = mse**(0.5)
r2 = metrics.r2_score(y_test, predictions)

print("Results using sklearn.metrics:")
print("MAE (Mean absolute error):",mae)
print("MSE (Mean Squared Error):", mse)
print("RMSE (Root Mean Squared Error):", rmse)
print("R-Squared (Coefficient of determination):", r2)
```

```
Results using sklearn.metrics:
MAE (Mean absolute error): 103058.34495192308
MSE (Mean Squared Error): 14913585594.247276
RMSE (Root Mean Squared Error): 122121.19224052505
R-Squared (Coefficient of determination): -0.4252250717393684
```

Figure 6: The code used to calculate the regression accuracy check metrics from the actual and predicted values of Figure 4 (Method 1 of 2) with the outputs.

```
# Calculate the Regression Accuracy Check metrics by sklearn.metrics.
mae = metrics.mean_absolute_error(valid['New Cases'] , valid['Predictions'])
mse = metrics.mean_squared_error(valid['New Cases'] , valid['Predictions'])
rmse = np.sqrt(mse) # OR rmse = mse**(0.5)
r2 = metrics.r2_score(valid['New Cases'], valid['Predictions'])

print("Results using sklearn.metrics:")
print("MAE (Mean absolute error):",mae)
print("MSE (Mean Squared Error):", mse)
print("RMSE (Root Mean Squared Error):", rmse)
print("R-Squared (Coefficient of determination):", r2)
```

```
Results using sklearn.metrics:
MAE (Mean absolute error): 103058.34495192308
MSE (Mean Squared Error): 14913585594.247276
RMSE (Root Mean Squared Error): 122121.19224052505
R-Squared (Coefficient of determination): -0.4252250717393684
```

Figure 7: The code used to calculate the regression accuracy check metrics from the actual and predicted values of Figure 4 (Method 2 of 2) with the outputs.

```
# Get the forecasted new cases count.
newCases_count = date_grp

# Create a new dataframe.
new_df = newCases_count.filter(['New Cases'])

# Get the last 60 days of new case values and convert the dataframe to an array.
last_60_days = new_df[-60:].values

# Scale the data to be values between 0 and 1.
last_60_days_scaled = scaler.transform(last_60_days)

# Create an empty list.
X_test =[]

# Append the past 60 days.
X_test.append(last_60_days_scaled)

# Convert the X_test data set to a numpy array.
X_test = np.array(X_test)

# Reshape the data.
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

# Get the predicted scaled new cases count.
pred_count = model. predict(X_test)

# Undo the scaling.
pred_count = scaler.inverse_transform(pred_count)
print('Forecasted New Cases count (next day):', pred_count)
```

```
1/1 [==============================] - 0s 23ms/step
Forecasted New Cases count (next day): [[546540.7]]
```

Figure 8: The code used to get the LSTM model's projected global Covid-19 new cases count for 11/10/2022 (Method 1 of 2) with the forecasted value.

```
# Predict the next day.
#real_data = [model_inputs[len(model_inputs) + 1 - prediction_days:len(model_inputs+1), 0]]
real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs+1), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"Predicted next day New Cases count: {prediction}")
```

```
1/1 [==============================] - 0s 20ms/step
Predicted next day New Cases count: [[546540.7]]
```

Figure 9: The code used to get the LSTM model's projected global Covid-19 new cases count for 11/10/2022 (Method 2 of 2) with the forecasted value.

Figures 10 to 16 are from Part2_CovidCapstone.ipynb.

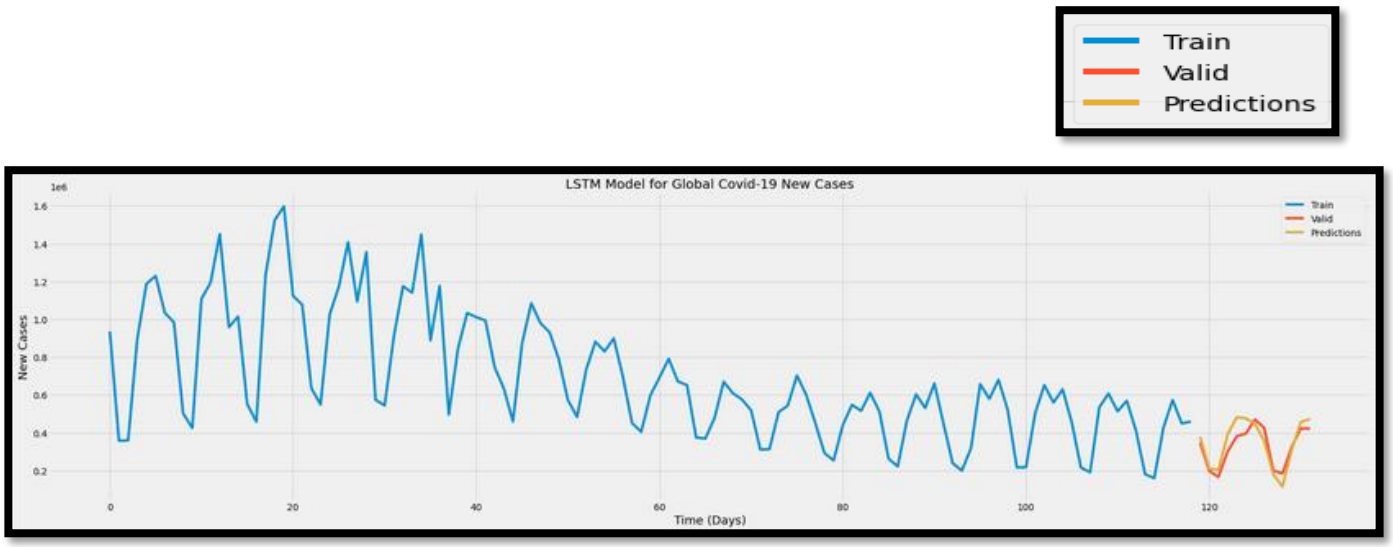

Figure 10: Line graph showing the LSTM model (training, testing, and predictions) for global Covid-19 new cases.

Please, enlarge Figure 10 to view clearly.

In Figure 11 below, the LSTM model's prediction results are compared to the actual results from the testing dataset. The testing dataset was set at ten percent of the total dataset length (days). This is an attempt to obtain some forecast results. Due to the real-world data, time constraints, and with the project team member having no prior coursework, background, and experience in machine learning, it is very difficult to achieve a perfect result.

Index	Date	Actual Confirmed New Cases	Predicted Confirmed New Cases
119	2022-10-28	349,367	378,975
120	2022-10-29	198,897	212,047
121	2022-10-30	167,086	202,919
122	2022-10-31	300,907	391,081
123	2022-11-01	382,118	483,139
124	2022-11-02	396,323	476,616
125	2022-11-03	471,259	447,911
126	2022-11-04	426,414	357,561
127	2022-11-05	200,343	179,772
128	2022-11-06	185,964	115,344
129	2022-11-07	325,986	310,040
130	2022-11-08	423,187	458,205
131	2022-11-09	422,628	473,125

Figure 11: The LSTM model's predicted global Covid-19 new cases values compared to the actual global new cases values of the testing dataset.

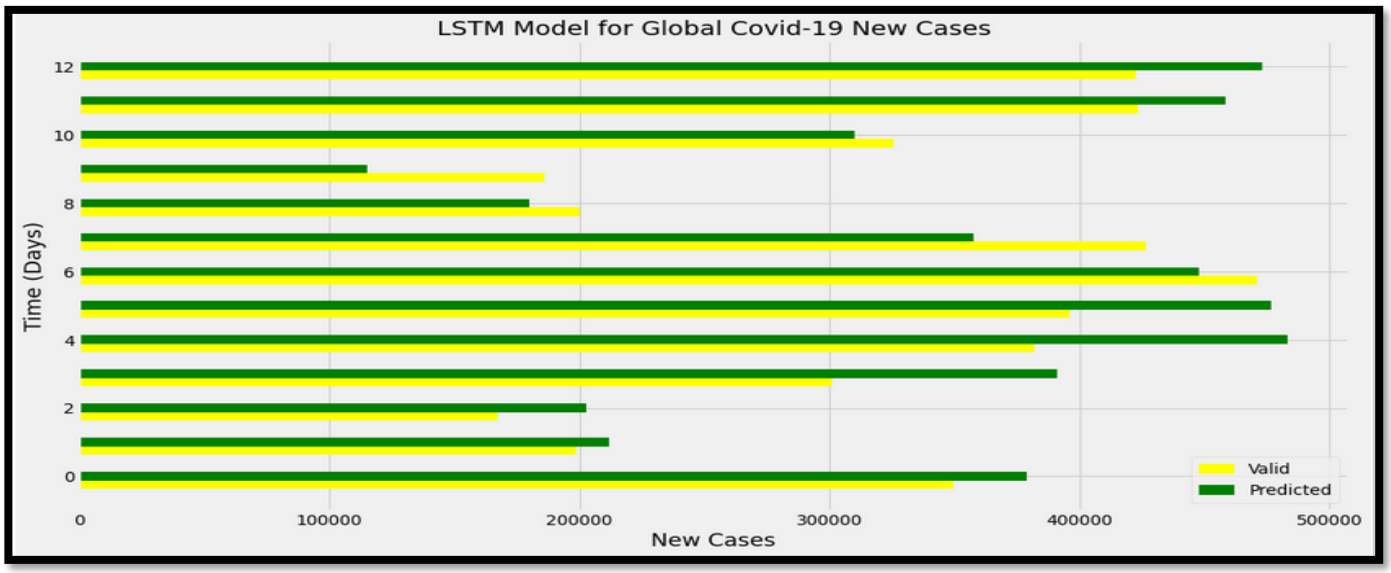

Figure 12: Horizontal bar graph of the actual vs. predicted results of Figure 11.

```
# Calculate the Regression Accuracy Check metrics by sklearn.metrics.
mae = metrics.mean_absolute_error(y_test, predictions)
mse = metrics.mean_squared_error(y_test, predictions)
rmse = np.sqrt(mse) # OR rmse = mse**(0.5)
r2 = metrics.r2_score(y_test, predictions)

print("Results using sklearn.metrics:")
print("MAE (Mean absolute error):",mae)
print("MSE (Mean Squared Error):", mse)
print("RMSE (Root Mean Squared Error):", rmse)
print("R-Squared (Coefficient of determination):", r2)
```

```
Results using sklearn.metrics:
MAE (Mean absolute error): 48840.997596153844
MSE (Mean Squared Error): 3218771011.329214
RMSE (Root Mean Squared Error): 56734.213763206535
R-Squared (Coefficient of determination): 0.6923963646070576
```

Figure 13: The code used to calculate the regression accuracy check metrics from the actual and predicted values of Figure 11 (Method 1 of 2) with the outputs.

```
# Calculate the Regression Accuracy Check metrics by sklearn.metrics.
mae = metrics.mean_absolute_error(valid['New Cases'] , valid['Predictions'])
mse = metrics.mean_squared_error(valid['New Cases'] , valid['Predictions'])
rmse = np.sqrt(mse) # OR rmse = mse**(0.5)
r2 = metrics.r2_score(valid['New Cases'], valid['Predictions'])

print("Results using sklearn.metrics:")
print("MAE (Mean absolute error):",mae)
print("MSE (Mean Squared Error):", mse)
print("RMSE (Root Mean Squared Error):", rmse)
print("R-Squared (Coefficient of determination):", r2)
```

```
Results using sklearn.metrics:
MAE (Mean absolute error): 48840.997596153844
MSE (Mean Squared Error): 3218771011.329214
RMSE (Root Mean Squared Error): 56734.213763206535
R-Squared (Coefficient of determination): 0.6923963646070576
```

Figure 14: The code used to calculate the regression accuracy check metrics from the actual and predicted values of Figure 11 (Method 2 of 2) with the outputs.

```
# Get the forecasted new cases count.
newCases_count = date_grp
#newCases_count = pd. read_csv('Covid_Processed.csv')

# Create a new dataframe.
new_df = newCases_count.filter(['New Cases'])

# Get the last 60 days of new case values and convert the dataframe to an array.
last_60_days = new_df[-60:].values

# Scale the data to be values between 0 and 1.
last_60_days_scaled = scaler.transform(last_60_days)

# Create an empty list.
X_test =[]

# Append the past 60 days.
X_test.append(last_60_days_scaled)

# Convert the X_test data set to a numpy array.
X_test = np.array(X_test)

# Reshape the data.
X_test = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))

# Get the predicted scaled new cases count.
pred_count = model. predict(X_test)

# Undo the scaling.
pred_count = scaler.inverse_transform(pred_count)
print('Forecasted New Cases count (next day):', pred_count)
```

```
1/1 [==============================] - 0s 27ms/step
Forecasted New Cases count (next day): [[453078.47]]
```

Figure 15: The code used to get the LSTM model's projected global Covid-19 new cases count for 11/10/2022 i.e., index (day) 132 (Method 1 of 2) with the forecasted value.

```
# Predict the next day.
#real_data = [model_inputs[len(model_inputs) + 1 - prediction_days:len(model_inputs+1), 0]]
real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs+1), 0]]
real_data = np.array(real_data)
real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

prediction = model.predict(real_data)
prediction = scaler.inverse_transform(prediction)
print(f"Predicted next day New Cases count: {prediction}")
```

```
1/1 [==============================] - 0s 25ms/step
Predicted next day New Cases count: [[453078.47]]
```

Figure 16: The code used to get the LSTM model's projected global Covid-19 new cases count for 11/10/2022 i.e., index (day) 132 (Method 2 of 2) with the forecasted value.

Please, see the source code for a full complement of line, scatter, vertical bar, and horizontal bar graphs.
The interpretation of the regression accuracy check metrics are as follows:

1) MAE (mean absolute error): The closer to zero this value is the more accurate the machine learning model is.
   a) Part one's value is ~ 103,059.
   b) Part two's value is ~ 48,841.
2) MSE (mean squared error): The lower this value is, the better the model is, a value of zero means the model is perfect.
   a) Part one's value is ~1,4913,585,594.
   b) Part two's value is ~ 3,218,771,011.

3) RMSE (root mean squared error): The lower this value is, the better the model is, a value of zero is perfection.
   a) Part one's value is ~ 122,121.
   b) Part two's value is ~56,734.
4) R-squared (coefficient of determination): The closer to one this value is the better the model is fitted i.e., trained. A value of one means the model is perfect and a value of zero or less means the model will perform poorly on an unknown dataset.
   a) Part one's value is ~ -0.425.
   b) Part two's value is ~0.692.

All of the regression accuracy check metrics calculations, except the R^2 value of part two for this particular run, indicate the LSTM model used in this project performs poorly in predicting new case data for Covid-19. The R^2 value in part two indicates a relatively high level of correlation between the actual new case (test) values and the model's predictions of these test values. The LSTM's new case predictions in Figures 4 and 11 and new case forecasts in Figures 8, 9, 15, and 16 are different every time the programs are executed. As a result, sometimes there is a considerable improvement in the regression accuracy check metrics. Overall, the regression accuracy check metrics results might seem disappointing, however, they could be given the project's dataset inputs, dataset length, dataset value magnitude, constraints, circumstances, and prospects be sufficient and acceptable, even expected and good.
The LSTM model's forecasted value for new confirmed Covid-19 cases on 11/10/2022 was 546,541 for part one and 453,078 for part two. The actual new confirmed Covid-19 cases on 11/10/2022 were checked after this date and reported at 454,792. This is an accuracy rate of ~79.83% (an absolute case difference of 91,749 between the actual and predicted value) for part one and ~99.62% (an absolute case difference of 1,714 between the actual and predicted value) for part two. In my opinion, this is a very good result for part one and a phenomenal result for part two on this particular run. Overall, I am very pleased with how well the project's LSTM machine learning model is fitted and predicts unknown Covid-19 new case data.

5 Open Issues

Everything about the Covid-19 virus is a real mess from infections, hospitalizations, treatments, vaccines, and deaths to the reporting and record keeping of this data. The reporting and record keeping of Covid-19 data is irregular, inaccurate, and inconsistent at best. For example, during the Covid-19 project data analysis in parts one and two, I found while calculating the highest global mortality rates for the top sixty countries, Egypt had a mortality rate calculated as 'inf' or infinite. Upon further investigation, it turns out that Egypt had zero reported confirmed new cases and seventy-five confirmed new deaths during the project's dataset dates. Maybe, there were seventy-five confirmed new deaths from confirmed new cases prior to the project's dataset dates. However, this is very highly improbable, no matter how one chooses to think about this. Additionally, there are ten countries (~4.3%) that have zero confirmed new cases and forty-three countries (~18.4%) that have zero confirmed deaths out of two-hundred-thirty-four countries. This is just preposterous, except maybe in the case of Antarctica unless we are also collecting Covid data for polar bears and penguins. For the other countries, factors such as war, famine, social-economic hardships, isolation, etc. may be the reason for the absence, under reporting, and poor record keeping of valid and reliable Covid data. On an interesting note, fifty-two countries (22.2%) have confirmed new case counts of under one-thousand and one-hundred-ninety-six countries (~83.76%) have new confirmed death counts of under one-thousand. These are just a few examples of the skewed Covid-19 data used for this project. The list goes on and on. The current state of Covid-19 data reporting and record keeping is a major problem. The quality of the Covid data must be improved and reporting/record keeping done accurately and correctly to be able to effectively use tools like data analysis, machine learning, and artificial intelligence to combat Covid. The Covid-19 datasets used for this project [1, 2] are sourced from Johns Hopkins University, World Bank, United Nations, and the World Health Organization.

The Covid stats for the United States are even worse than the reporting and record keeping. Refer to "Section 3: Key Observations" of this document. It is a shame that with all the advanced technology, treatments, vaccines, and other resources accessible to the United States population, many U.S. citizens refuse to comply with and fail to fully utilize the effective methods to control Covid. Unless we all get our act together and follow the FDA and CDC Covid-19 protocols there is going to continue to be variant after variant and cycle after cycle of infections, severe Covid, hospitalizations, long-term Covid, and fatalities. This also goes for most countries. One of these days, we will not be as fortunate as we are today. There will probably be sooner rather than later, "SUPER" Covid variants with very high and uncontrollable transmission, severe infection, hospitalization, mortality, and mutation rates for which there are no treatments or vaccines. This is totally avoidable. Proper behavior is the best way to control and eventually eradicate Covid. Wear the masks or face shields (if masks make breathing and communication difficult), as much as possible and everyone who is medically able must stay updated on the latest vaccines. It is that simple. Once we, not just the United States, as a global community achieve this we can, because Covid is now under control, maximize the now newly additional available resources to find a cure and use data analysis, machine learning, and artificial intelligence technology with valid and reliable Covid datasets to effectively assist in providing groundbreaking data for the extinction of Covid. This groundbreaking data involves much more than just predicting future Covid statistical data, it involves providing and predicting error free data used in the development of the actual formulas for, production of, and administration of the treatment drugs and vaccines (cures), as well as developing effective strategic protocols for improving and successfully implementing permanent Covid-19

monitoring, regulation, management, and extermination plans. We all have a very long way to go. It is still not too late to embark on this long and demanding journey together. Every day is a new beginning.

6 Acknowledgements

I would like to give special thanks to Dr. Yunchuan Liu for all of his help, time, effort, and support in teaching me how to successfully complete this Graduate Capstone Seminar Project. I would not have been able to complete this project without him.

7 References

[1] Johns Hopkins University, World Bank, United Nations, and the World Health Organization. "KFFData /COVID-19-Data". GitHub, Inc. 2022. https://github.com/KFFData/COVID-19-Data/tree/kff_master/Country%20Trend%20Data.

[2] COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. "CSSEGISandData / COVID-19". GitHub, Inc. 2022. https://github.com/CSSEGISandData/COVID-19.

[3] Sinha, Rinkesh Kumar. "Understanding of LSTM Networks". June 25, 2021 https://www.geeksforgeeks.org/understanding-of-lstm-networks/#:~:text=LSTMs%20provide%20us%20with%20a,)%2C%20which%20is%20an%20advantage.

[4] Yorkinov, Otabek. "DataTechNotes. A blog about data science and machine learning". DataTechNotes. 2016-2022. https://www.datatechnotes.com/2019/10/accuracy-check-in-python-mae-mse-rmse-r.html.