COVID-19 Pandemic Data Analysis

Vanitha Varadharajan (019076803E)

Abstract – As we all know that the COVID-19 pandemic gets severe every day and affecting almost all of our lives financially and mentally. Due to its widespread, people have started reading statistical analytics of its growth and sharing their thoughts with their family, friends, and social media, etc. So, the data analysis of the pandemic is very important for people to know about the current situation and to take necessary preventive measures accordingly. In this report, the study of how frequently the coronavirus confirmed cases increases worldwide, the death rate, and which are all the countries taking more preventive measures to decrease the confirmed cases and increases the recovering rate has been done. The main idea is to find out the confirmed cases for the future dates based on the pattern the given data follows using machine learning algorithms. The results of the prediction have been plotted and shown.

Problem Definition

The problem statement is to find the confirmed corona cases for the future dates with the given time series data set by different parameter setting. For that, I have done analyzing and visualizing the relationship between each data and found that the pattern is in linear trend. The plots of the analysis are shown below. Since the pattern is in linear, the prediction of future cases would be possible and would give more accurate results. That is the reason I chose regression. Here, I am intending to get the accurate prediction value for the test data given for the evaluation of the model and the accurate predicted confirmed cases for the future dates (it should be in linear trend)

Approach

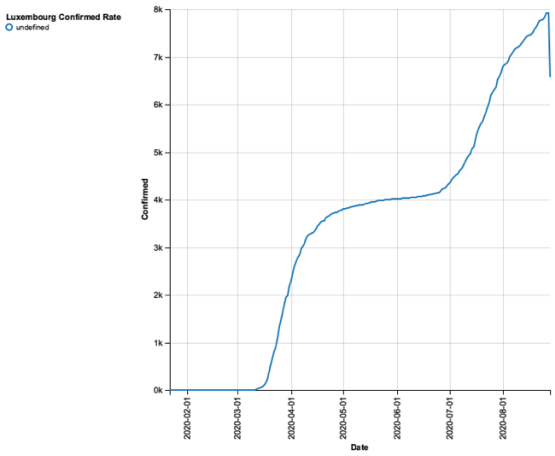
To compare the results, I have tried both Random forest regression and Linear regression for the prediction. From the results which I got from both the algorithms, the Random forest regression could not extrapolate the linear trend and predicting the accurate new values. Since the provided data are in a perfectly linear trend, the Linear Regression model would be a good choice for detecting the trend and making accurate predictions.

For the parallelization, I have used the HPC infrastructure (Username: vvaradharajan) and for plotting the graph, I used my local machine's spark-shell since I could not do plotting in the cluster. The frameworks which I have used for the project are Spark ML libraries, Spark SQL libraries, and Vegas libraries (plotting) and I have used data frame structure throughout the project. I have taken the COVID-19 time series (Confirmed, Death, and Recovered) datasets for my analysis since it has all over the country's data on a daily basis.

Evaluation & Results

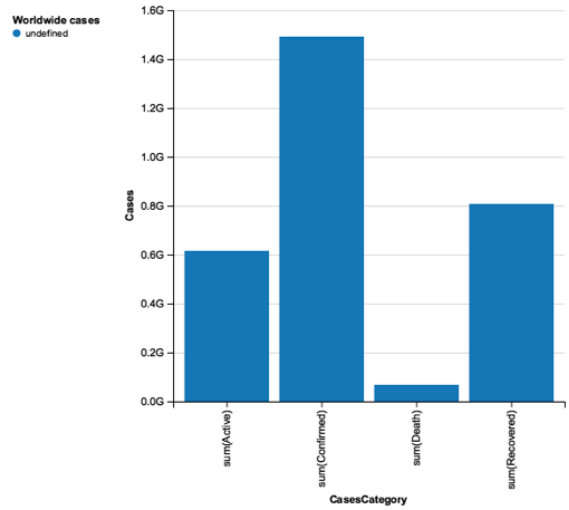
Before doing regression, I have done some data analysis and have plotted the trends of the datasets

i) Luxembourg confirmed cases Trend (Date Vs Confirmed cases)
The below figure shows the trend of confirmed cases increase from start date to current date



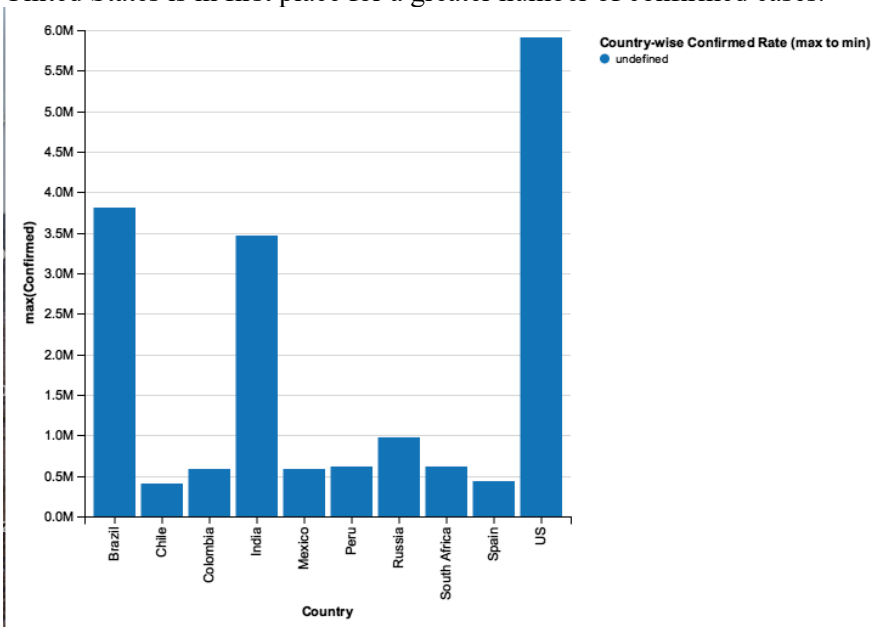
ii) Worldwide analysis

The below figure shows the total number of confirmed, death, recovered and active cases for worldwide.



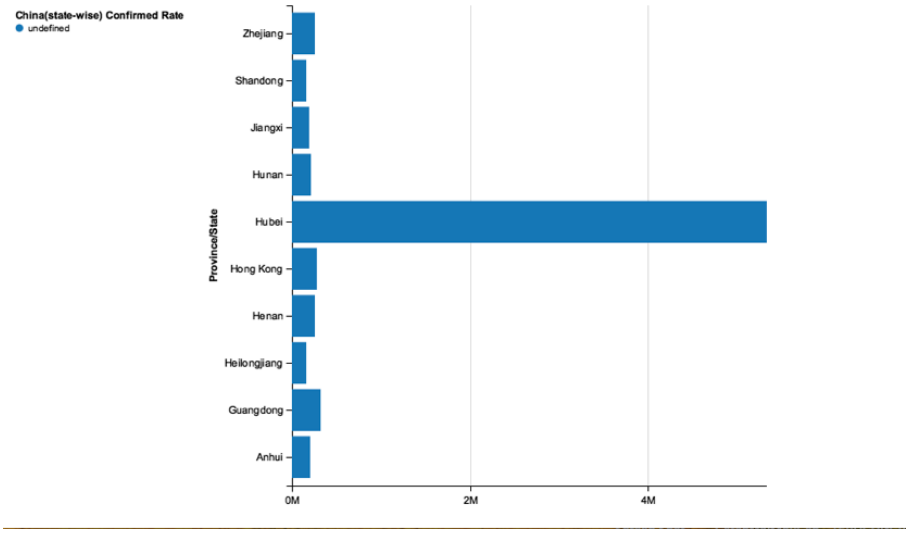
iii) Top 10 Countries which registered more Confirmed cases

United States is in first place for a greater number of confirmed cases.



iv) State wise confirmed rate in China

Top 10 states which registered more cases in China in current date



In Random Forest Regression: (Prediction of confirmed cases)

The result shown below is for "Luxembourg" country data for different dates. The prediction from the evaluation of the model using the test data is shown below. The prediction values are less than the actual values. There is not much changes in the results even adjusting the numTrees to 5 ,10 and 15.

Actual confirmed cases: 4345
Predicted: 2895
Elapsed time: 204846599 ns

Result for test data (Date >= "01/07/20")

```
scala> println("Elapsed time: " + (t1 - t0) + "ns")
Elapsed time: 204846599ns

scala> predictions.show()
+--------+---------+---------+-----------------+-------------------+------------------+
|Latitude|Longitude|Confirmed|dateType_timestamp|       featureVector|        prediction|
+--------+---------+---------+-----------------+-------------------+------------------+
| 49.8153|   6.1296|     4345|       1593554400|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4395|       1593640800|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4447|       1593727200|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4476|       1593813600|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4522|       1593900000|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4542|       1593986400|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4603|       1594072800|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4650|       1594159200|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4719|       1594245600|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4777|       1594332000|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4842|       1594418400|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4925|       1594504800|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     4956|       1594591200|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5056|       1594677600|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5122|       1594764000|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5285|       1594850400|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5409|       1594936800|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5483|       1595023200|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5605|       1595109600|[49.8153,6.1296,1...|2895.2031284563654|
| 49.8153|   6.1296|     5639|       1595196000|[49.8153,6.1296,1...|2895.2031284563654|
+--------+---------+---------+-----------------+-------------------+------------------+
only showing top 20 rows
```

In Linear Regression: (Prediction of confirmed cases)

The results which are shown below are for “Luxembourg”. The Linear regression algorithm has predicted the confirmed cases accurately.

Result for test data (Date >= “01/07/20”)

```
scala> println("Elapsed time: " + (t1 - t0) + "ns")
Elapsed time: 247132883ns

scala> results.show()
+------------------+---------+------------------+
|dateType_timestamp|Confirmed|        prediction|
+------------------+---------+------------------+
|        1593554400|     4345| 5218.315122247441|
|        1593640800|     4395|  5254.91924475634|
|        1593727200|     4447| 5291.523367265123|
|        1593813600|     4476| 5328.127489773906|
|        1593900000|     4522| 5364.731612282805|
|        1593986400|     4542| 5401.335734791588|
|        1594072800|     4603| 5437.939857300371|
|        1594159200|     4650|  5474.54397980927|
|        1594245600|     4719| 5511.148102318053|
|        1594332000|     4777| 5547.752224826952|
|        1594418400|     4842| 5584.356347335735|
|        1594504800|     4925| 5620.960469844518|
|        1594591200|     4956|5657.5645923534175|
|        1594677600|     5056|   5694.1687148622|
|        1594764000|     5122| 5730.772837370983|
|        1594850400|     5285| 5767.376959879883|
|        1594936800|     5409|5803.9810823886655|
|        1595023200|     5483| 5840.585204897448|
|        1595109600|     5605| 5877.189327406348|
|        1595196000|     5639| 5913.793449915131|
+------------------+---------+------------------+
only showing top 20 rows
```

Result for future Dates (from 01/09/2020 to 11/09/2020)

```
scala> println("Elapsed time: " + (t1 - t0) + "ns")
Elapsed time: 235739168ns

scala> val finalResult = resultsForFutureDates.withColumn("dateType_timestamp",col("dateType_timestamp").cast("timestamp"))
finalResult: org.apache.spark.sql.DataFrame = [dateType_timestamp: timestamp, Latitude: double ... 2 more fields]

scala> finalResult.show()
+-------------------+--------+---------+------------------+
| dateType_timestamp|Latitude|Longitude|        prediction|
+-------------------+--------+---------+------------------+
|2020-09-01 00:00:00|  49.815|   6.1296| 7564.390806082054|
|2020-09-02 00:00:00|  49.815|   6.1296| 7601.667281613336|
|2020-09-03 00:00:00|  49.815|   6.1296| 7638.943757144734|
|2020-09-04 00:00:00|  49.815|   6.1296| 7676.220232676133|
|2020-09-05 00:00:00|  49.815|   6.1296| 7713.496708207531|
|2020-09-06 00:00:00|  49.815|   6.1296| 7750.773183738929|
|2020-09-07 00:00:00|  49.815|   6.1296|7788.0496592703275|
|2020-09-08 00:00:00|  49.815|   6.1296| 7825.326134801726|
|2020-09-09 00:00:00|  49.815|   6.1296| 7862.602610333124|
|2020-09-10 00:00:00|  49.815|   6.1296|7899.8790858645225|
|2020-09-11 00:00:00|  49.815|   6.1296| 7937.155561395804|
+-------------------+--------+---------+------------------+
```

The result of the confirmed cases registered in Luxembourg for the last few days:

```
scala> luxData.show
+--------------+--------------+-------+------+-------+---------+
|Province/State|Country/Region|    Lat|  Long|   Date|Confirmed|
+--------------+--------------+-------+------+-------+---------+
|          null|    Luxembourg|49.8153|6.1296| 8/3/20|     6864|
|          null|    Luxembourg|49.8153|6.1296| 8/4/20|     6917|
|          null|    Luxembourg|49.8153|6.1296| 8/5/20|     7007|
|          null|    Luxembourg|49.8153|6.1296| 8/6/20|     7073|
|          null|    Luxembourg|49.8153|6.1296| 8/7/20|     7113|
|          null|    Luxembourg|49.8153|6.1296| 8/8/20|     7169|
|          null|    Luxembourg|49.8153|6.1296| 8/9/20|     7205|
|          null|    Luxembourg|49.8153|6.1296|8/28/20|     6580|
+--------------+--------------+-------+------+-------+---------+
```

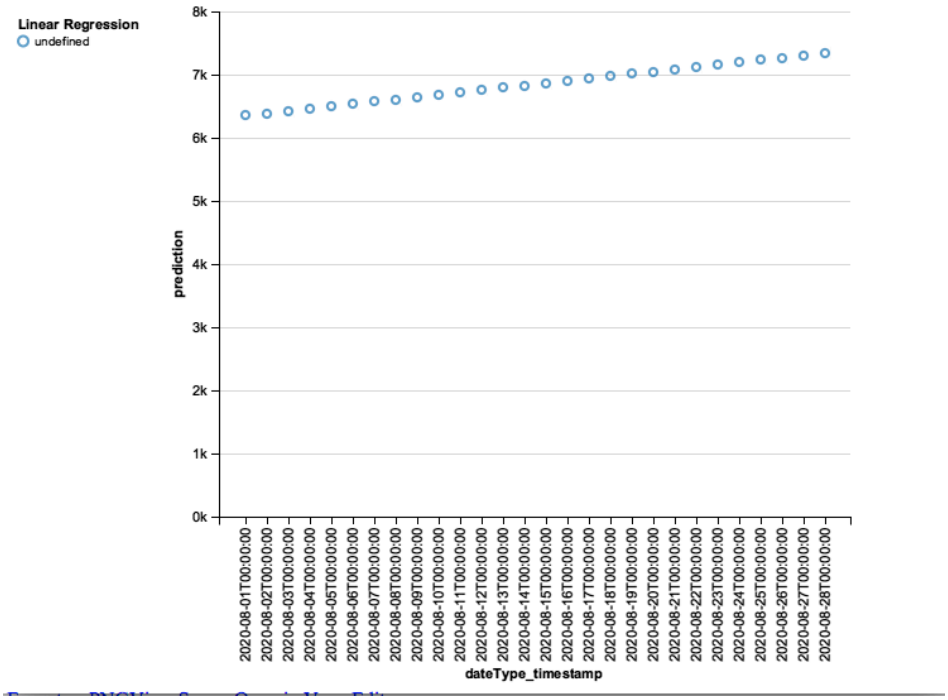
we can see that the prediction for future dates is perfect when comparing it with the last few days cases shown above.

On “28/08/20” the confirmed cases in Luxembourg is 6580. Based on its pattern (the cases have been reduced for the last two days from 7900 to 6580), the algorithm has predicted the future cases. The evaluation does not take longer duration.

Elapsed time = 235739168 ns

The plot for Linear Regression (Luxembourg):

The graph below shows the trend of Linear regression for the prediction of confirmed cases.



Conclusion

From the evaluation of the two algorithms, I found that the better solution for the given data set is Linear Regression and experimented the same. The problem statement of predicting future values has been completed and shown in the screenshot. From this use case study, I have learned how to analyze the data from different perspectives and which techniques would be suitable to find a better solution for the problem. Also, I have learned some basics of Vegas and breeze plotting which is better since the plotting libraries are lagging for Scala.

References:

1. Linear Regression- http://spark.apache.org/docs/latest/ml-classification-regression.html#linear-regression
2. Spark by Examples- https://sparkbyexamples.com/
3. Vegas Tutorial - http://datasmarts.net/a-tour-to-vegas/
4. Library-Vegas-spark- 0.3.11 (https://mvnrepository.com/artifact/org.vegas-viz/vegas-spark_2.11/0.3.11)
5. Library-Vegas-0.3.11- https://mvnrepository.com/artifact/org.vegas-viz/vegas
6. Library-Vegas-macros-0.3.11- https://mvnrepository.com/artifact/org.vegas-viz/vegas-macros