Prediction and Analyis of Covid-19 using Apache Spark

Balasubramaniyam Muthuramalingam, MSc Big Data Analytics and Artificial Intelligence, L00157060

***Abstract*— A continuous downfall created by Covid-19 led to global pandemic, the competence to analyse the current situation and predict an overview to provide a positive feedback in order to denude the concurrent impendent would prove to be an artifact. Utilizing the vital assets, incoming new cases are forecasted. The report provides an outline of new cases for the next week cycle based on the collected historical data. It is achieved by implementing Machine Learning techniques to predict the New cases in different countries that are administrated by World Health Organization. The objective is to produce an accurate result for the upcoming week using Spark Library. The project demonstrates Covid-19 cases in different categories, Active, Death, Recovered and Confirmed. Using Apache Spark architecture and its corresponding Machine Learning library (MLlib) observations were formed. It's a prediction model so the following regression techniques were implemented, Linear Regression, Decision Tree Regression, Random Forest Regression and Gradient Boost Regression. In the end, using pipeline and Train Validation split the model was tuned for enhanced perception.**

***Index Terms*— Hadoop, Machine Learning, Python, Apache Spark, Linear Regression, Decision Tree Regression, Random Forest Regression, Gradient Boost Regression.**

I. Introduction

The project provides an analysis of Covid-19 by utilizing the given dataset. The prediction of new cases is made by calculating the growth factor of current data. The growth factor is considered as the label column and implemented with the machine learning algorithms. All the other numerical columns that are essential to tabulate the prediction column is converted into feature column. The model is trained with four different ML algorithms, Linear Regression, Decision Tree Regression, Random Forest Regression and Gradient boost Regression. After the model is trained using the training dataset, the test dataset is used to produce the prediction column. Algorithm that is produces the most compatible value is considered to be fit and concluded as best fit technique for the model. The project is implemented on Linux environment using a Virtual box. All the processing is done using Hadoop environment, Apache Spark architecture was used hence it is processed in distributed way. Using PySpark the models are created and MLlib is imported to invoke machine learning algorithms. Features of the project include, Generalization of dataset, pre-processing of dataset, Applying ML techniques to the model and deploying the model by tuning it using train validation split.

A. Covid-19

Coronavirus is distinct type of virus that is easily contagious, most cases affected by the Covid 19 were showing similar symptoms and their feedbacks are running nose, sinuses, dry cough and so on… Most of the corona virus weren't considered as high-level threat but SARS-CoV-2 was a newly developed virus stated by the World Health Organization [WHO]. An enormous outbreak occurred in China on December 2019 due to covid-19. The disease spread like a wildfire across the globe. The newly formed virus SARS-COV-2 can damage your respiratory system by infecting it. There are two possible cases to identify the presence of SARS-CoV-2 in the body, upper respiratory tract and lower respiratory tract, they include nose, throat, windpipe and lungs. The Covid-19 shows similar attributes like other Corona viruses to spread from one host to the other, some are as follows, direct contact with the infected individual [breathing through same air], cough. It is considered to be fatal if the host's immune system is weak and similarly it can also be mild as a fever. Physicians identified that SARS-COV-2 is one of the seven coronaviruses that can enable Middle Respiratory Syndrome which is also terms as MERS and Acute Respiratory Syndrome commonly referred as SARS while the other viruses that are branched under the Covid family doesn't lead to significant threat.

B. Apache Spark

Spark is an engine that allows processing of big data, since it is an open-sourced software it is widely utilized to analyse huge chunks of datasets [big data]. Spark is known for its versatile and speed while computing big datasets. Initially it was introduced as a research project in UC Berkeley's AMPLab in the year,2009. It was officially released as open sourced on March,2010 and later acquired by a software organization called Apache, where it quickly rose to become a top-level project. Machine learning was not scalable and took too long before Spark came along. Spark supports a variety of languages. Spark's scalability and processing speed are two of its most important features. Scala, PySpark, R, and SQL are among the languages supported by Spark. It has a lot of configuration parameters that you can use to fine-tune

the Spark application. Spark is made up of a single driver and a number of executors. Spark can be programmed to run on a single executor or as many as you need to complete the task. Spark has auto scaling capabilities, and you can set a minimum and maximum number of executors. Spark's primitive feature enables to conduct immersive analytics. Without the need of sample, it is adequately fast to run exploratory queries. Convoluted datasets are continuously engaged to interact and visualize the analysed data.

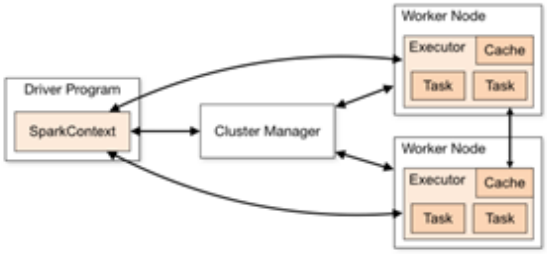

Fig 1: Architecture of Apache Spark

C. Machine Learning in Spark

Machine learning (ML) is a subset of artificial intelligence that was created in 1959 by Arthur Samuel. ML can be described as a computer that tries to learn from its previous experiences. Users gain experience by feeding data to the computer, which uses predefined formulas to understand particular problems and produce a valid result. Nevertheless, the accuracy of the result may not be accurate or sufficient, so regular testing and tweaking of mathematical models will increase accuracy. As a result, the computer learns to adjust to the problem based on the feedback it receives. Training data is a term used to describe the learning process. Since there is no need to add or change code specifically for various sets of inputs, ML is commonly used in many applications. The field of machine learning is constantly evolving, and it has yet to enter any domain and application. It also shows potential for expanding its breadth of use and exploring new areas. Machine learning capabilities are another of Apache Spark's many applications. Spark allows unified framework for continuous analysis that prompts users to compute consecutive queries on finite datasets while also processing ML algorithms. MLlib is one of the Spark's framework components and can be used for different types of big data tasks, including clustering, sorting and dimensionality reduction. It is also used for accurate perception and sentiment analysis. Spark's machine learning capabilities can also be used to improve network security. Inspections of malicious behaviour in data packets is conducted in real time using different components present in spark stack.

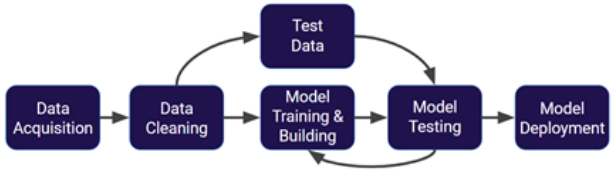

Fig 2: Implementation of ML Model

II. Dataset Overview

The dataset used to build Covid 19 model is in the form of .csv format. It is taken from Kaggle website and it consists of two different datasets: (a) country_wise_latest.csv and (b) worldmeter_data.csv. The primary dataset country_wise_latest is used to build the model and apply the machine learning algorithm. The secondary dataset is used to analyse the relationship using matplotlib.

Link for the primary and secondary dataset:

1) https://www.kaggle.com/imdevskp/corona-virus-report?select=country_wise_latest.csv

2) https://www.kaggle.com/imdevskp/corona-virus-report?select=worldometer_data.csv

The primary dataset country_wise_latest.csv consists of following attributes:

- Country/Region: It displays all the country affected by Covid-19.
- Confirmed: Records of confirmed covid cases.
- Deaths: Total number of deaths occurred due to Covid-19.
- Recovered: Cases that have recovered from the disease.
- Active: Number of Cases going through treatment and yet to recover from Covid-19.
- New Cases: Number of cases that have been added recently.
- New Deaths: Cases that have not recovered from Covid-19 that has been added recently.
- New Recovered: Newly added cases that have been recovered and yet to be updated in the main column.
- Death/100 Cases: Total number of deaths for every 100 cases.
- Recovered/100 Cases: Total number of Recovery's for every 100 cases.
- Deaths/100 Cases: Total number of Deaths for every 100 cases.
- Confirmed last week: Cases that have been confirmed last week which is converted into growth factor.
- 1 week Change: The fluctuation of number new cases for each week.
- 1 week % increase: Growth Factor, rate in increase of number of newly added cases in 1 week.
- WHO Region: Determines the WHO region for every country.

The secondary dataset records total population of each country to calculate the total number of confirmed cases using total number of active cases. It consists of following columns:

- Country/Region: Displays the country affected by Covid-19 in alphabetical order.
- Continent: Records the continent for the specified country.
- Population: Entries of total number population overall in each country.
- Total Cases: Total cases affected by covid-19.

- New Cases: Newly added covid-19 cases that are yet to be added with total cases.
- Total Deaths: Number of cases have failed to recover from Covid-19.
- New Deaths: Number of deaths that occurred recently.
- Total Recovered: Number of cases that have recovered from the Covid-19.
- New Recovered: Number of newly added cases that have been recovered from Covid-19.
- Active Cases: Total number of cases that are still active and yet to recover.
- Serious/Critical: Cases that are in critical or fata condition.
- Total Cases/1m Population: Total number of cases that have been affected by covid-19 for every 1M population.
- Death Cases/1m Population: Cases that have failed to recover from covid-19 for every 1M population.
- Total Tests: Total number of tests that have been conducted to confirm the case.
- Tests/1m: Total number of tests that have been taken for every 1 million population. region: Determines the WHO region for every country.

The total number of rows and column in the primary dataset is (187,15) and for the secondary dataset it is around (209,15).

III. Data Exploration

At the beginning, both the primary and secondary dataset is loaded into the environment using spark. In the primary dataset, there are 15 columns, all the respective columns are displayed to get an overview of the dataset. The dataset is grouped in ascending order based on the column Country/Region.

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	1 week % increase	WHO Regi
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.50	69.49	5.04	35526	737	2.07	East Mediterran
1	Albania	4880	144	2745	1991	117	6	63	2.95	56.25	5.25	4171	709	17.00	Eur
2	Algeria	27973	1163	18837	7973	616	8	749	4.16	67.34	6.17	23691	4282	18.07	Af
3	Andorra	907	52	803	52	10	0	0	5.73	88.53	6.48	884	23	2.60	Eur
4	Angola	950	41	242	667	18	1	0	4.32	25.47	16.94	749	201	26.84	Af

Fig 3: Overview of the dataset

Overview of the dataset is observed by printing the schema of the dataset. It displays the column and datatype of the respective column.

```
root
 |-- Country/Region: string (nullable = true)
 |-- Confirmed: integer (nullable = true)
 |-- Deaths: integer (nullable = true)
 |-- Recovered: integer (nullable = true)
 |-- Active: integer (nullable = true)
 |-- New cases: integer (nullable = true)
 |-- New deaths: integer (nullable = true)
 |-- New recovered: integer (nullable = true)
 |-- Deaths / 100 Cases: double (nullable = true)
 |-- Recovered / 100 Cases: double (nullable = true)
 |-- Deaths / 100 Recovered: string (nullable = true)
 |-- Confirmed last week: integer (nullable = true)
 |-- 1 week change: integer (nullable = true)
 |-- 1 week % increase: double (nullable = true)
 |-- WHO Region: string (nullable = true)
```

Fig 4: Schema Of the dataset

Similar process is made for the secondary dataset in order to pre-process the dataset. Using matplotlib library, total number of confirmed covid cases is visually presented. In the figure, the X-axis is total number of confirmed cases and the Y-axis is the total number of confirmed covid 19 cases. The graph illustrates the countries affected by Covid-19 in alphabetical order.

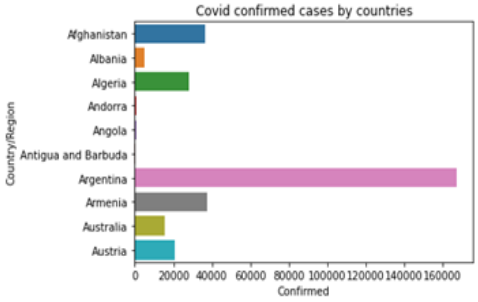

Fig 5: Confirmed Covid Cases grouped by Countries

In the second figure, the X-axis is country/region column, and the Y-axis is the Total number of cases due to covid-19. It defines different features of the dataset, confirmed case, number of deaths, recovered, active cases and New cases.

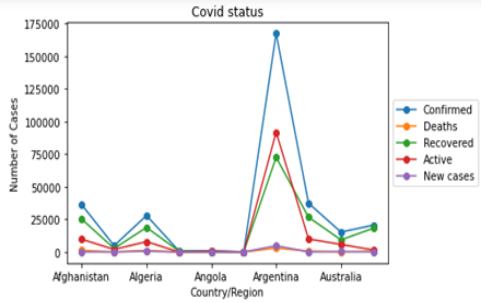

Fig 6: Status of covid-19

IV. Data Pre-Processing

The primary dataset is cleaned to negate all the invalid entries and reform the dataset which makes it compatible to use it with machine learning models.

1) Using the print Schema function, the datatypes of the columns are displayed and "Deaths/100 Recovered" is in string type, but the values are numerical, hence the column datatype is changed to integer.
2) Primarily, changing the dataset column name that is relevant to the problem. In this case, since growth factor is considered as the label column, the name of "1 week
3) Checking for Null values in the "growth_factor" column, if so, the entire row is dropped to filter out the dataset.
4) Again, checking for Null values in the other column, if so, the rows consisting of Null/NaN values are filled with the value "0".
5) Considering all the column with "growth_factor" value of greater than 0 and saving it in a new memory. Thus the data is cleaned and processed and can be applied with machine learning algorithms.

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recovered / 100 Cases	Deaths / 100 Recovered	Confirmed last week	1 week change	growth_factor	
0	New Zealand	1557	22	1514	21	1	0	1	1.41	97.24	1.0	1555	2	0.13	We
1	Guinea-Bissau	1954	26	803	1125	0	0	0	1.33	41.10	3.0	1949	5	0.26	A
2	Mauritius	344	10	332	2	0	0	0	2.91	96.51	3.0	343	1	0.29	A
3	Ireland	25892	1764	23364	764	11	0	0	6.81	90.24	7.0	25766	126	0.49	Eu
4	Estonia	2034	69	1923	42	0	0	1	3.39	94.54	3.0	2021	13	0.64	Eu

Fig 7: Processed Dataset

A. Splitting Dataset

The primary dataset that is processed will be split into two different sections, Training dataset and Testing Dataset. In order to proceed, the multiple numerical columns with finite values are grouped together and converted into vector format using vector assembler and the column is called as "features". The label column is chosen to be "growth_factor" column and train and test split is executed. The ratio chosen for this project 0.8 and 0.2 for training and testing.

```
+--------------------+-------------+
|            features|growth_factor|
+--------------------+-------------+
| [1557.0,2.0,1555.0]|         0.13|
| [1954.0,5.0,1949.0]|         0.26|
|   [344.0,1.0,343.0]|         0.29|
|[25892.0,126.0,25...|         0.49|
|[2034.0,13.0,2021.0]|         0.64|
|[246286.0,1662.0,...|         0.68|
|   [289.0,2.0,287.0]|          0.7|
|[5059.0,39.0,5020.0]|         0.78|
|[7398.0,58.0,7340.0]|         0.79|
|[1854.0,15.0,1839.0]|         0.82|
|[9132.0,98.0,9034.0]|         1.08|
|[4599.0,51.0,4548.0]|         1.12|
|[8904.0,104.0,880...|         1.18|
|[86783.0,1161.0,8...|         1.36|
|[3297.0,47.0,3250.0]|         1.45|
|[2513.0,38.0,2475.0]|         1.54|
|[67251.0,1038.0,6...|         1.57|
|[301708.0,4764.0,...|          1.6|
|[79395.0,1347.0,7...|         1.73|
|[207112.0,3787.0,...|         1.86|
+--------------------+-------------+
```

Fig 8: Label and Feature Column

V. Big Data Architecture and Tools

Using a virtual box with LinuxOS, in Hadoop Environment, complete installation of Apache Spark is made. After downloading all the tools from the terminal, a new notebook is created to build the model. Pyspark library along MLlib for machine learning is imported. MLlib contains various ML model to train, test and deploy the model. The other secondary and tertiary libraries are Sickit, MatPlotlib, Seaborn, Plotly are imported. All the required libraries are installed using the Pip3 command executed via terminal and were given specific path to connect to the Jupyter Notebook which is the interface for building the model.

VI. Machine Learning Models

The following Machine Learning models are implemented to build the model.

A. Linear Regression

A mathematical model is linear regression. It examines the linear relationship between two variables in the presence of a number of dependent and independent variables. A linear relationship between variables is defined as when the value of one or more independent variables rises or falls, the value of the dependent variable rises or falls in lockstep. Y = mX+ b is another way to express it mathematically. A positive or negative linear relationship exists between two variables. Two attributes are assigned to the X and Y coordinates in this example: X represents Years and Y represents Male. The slope and intercept for the model are determined using a function. In simple terms, classic linear regression objective is to reduce the vertical distance present between all data points.

B. Decision Tree Regression

The supervised learning method of decision tree (DT) is used to solve classification and regression problems. It's a tree with each node representing a class name. There is a root node and a branch node in it. Any Boolean function on discrete values can be represented by DT. DT fails to select a root node for each branch after generating each collection of branch nordes. Information Gain and the Gini Index are two methods for determining the root node for branch nodes. When a node is used in DT to divide the training instances into smaller subsets, the randomness of the information shifts, which is referred to as entropy. It is known as a change in entropy; the lower the change, the easier it is to find an appropriate root node.

C. Random Forest Regression

Trees make up a random forest (RF), and more trees mean a more stable forest. Similarly, it constructs decision trees from datasets, obtains predictions from each node, and then votes to pick the best node. It's an ensemble approach that's better than a single decision tree because it averages the results to minimize over-fitting.The first one is called Boosting, and it involves using statistical parameters like weighted averages to transform a weak learner into a good learner – a concept similar to teamwork. Models run in order, with the previous model dictating which function the next model should concentrate on.The overall output is improved since the following model learns from the previous one. The second one is known as bootstrap aggregation or bagging, and it involves the machine performing a random sampling. This helps the machine to better comprehend the dataset's variance. Each model operates independently, and the final result is the sum of all of the models' outputs.

D. Gradient Boost Regression

Prediction and optimization are frequently used to optimize large and complex datasets. Gradient boosting obtains the best solution through an iterative process, combining the next best possible model with the older model to minimize prediction error. Depending about how much the data forecast has changed and how it affects the total prediction errors, target outcomes are set and estimated for each. The goal value is set to high if a small change creates a significant load on error. Otherwise, if the model shows minor changes that do not result in an error, the outcome for the next case is set to zero. The error rate does not decrease when the estimate is modified. Creating a Gradient model for training and testing data and

assigning it to a new variable that is treated as an entity to refer to the gradient model. We also compute the R value and Mean squared value to determine the model's error rate and accuracy.

VII. Evaluation Metrics

Using the method evaluation every continuous value present in the dataset can be used to predict the outcome but it is not possible in categorical data as it a Classification. Some of the evaluation metrics involved to evaluate the model are called accuracy, perception or recall but it cannot be used if the data is continuous in nature. Hence the most common metrics are Mean Absolute Error, Mean Squared Error, Root Mean Square Value and R Squared Value.

- MAE is defined as mean of the absolute value of errors.

$$\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE Formulae

- MSE is defined as mean of the Squared Error, larger error is recorded more than MAE resulting usage of MSE popular.

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

MSE Formulae

- RMSE is similar to MSE but it takes the root of mean of the Squared Error.

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE Formulae

- R squared value is also known as co-efficient of determination is a statistical measure of the regression model. It provides a measure of how much variance the model accounts. R squared can enhance the understanding of model or can be used to compare the model to check which suits the dataset.

VIII. Results

Using regression evaluation all the models are compared with each other and best fit model is chosen to deploy the model. According to the model, Random Forest Regressor produced a better result with maximum fit compared to Linear Regression and Decision Tree Regression and Gradient Boost Regression. The RMSE value for both Decision Tree and Gradient Boost Regressor yielded a similar result 26.03 and 27.73 but the value was a lose fit for the model. In linear regression model, the RMSE value is 12.260 which is comparatively lower than Random Forest Regressor's RMSE value 14.508 but the co-efficient of determination resulted in negative 0.12, hence Random Forest Regressor is the optimal algorithm that is compatible with the dataset and fits the model. The RMSE Values for each algorithm is listed below

1) Linear Regression

```
RMSE for Linear Regression
12.260658176138577
R Squared Value for Linear Regression
-0.1257931587872998
```

2) Decision Tree Regression

```
Root Mean Squared Error (RMSE) for Decision Tree on test data
26.303860347810872
```

3) Random Forest Regressor

```
Root Mean Squared Error (RMSE) for Random Forest on test data
14.580710779426923
```

4) Gradient Boost Regressor

```
Root Mean Squared Error (RMSE) for Gradient Boost on test data
25.729553865265853
```

The prediction table produced by random forest regressor for the Second week cycle of Covid-19 using the growth factor.

prediction	growth_factor	features
4.022140619623712	0.26	[1954.0,5.0,1949.0]
8.170291656515932	0.29	[344.0,1.0,343.0]
13.595798383479439	0.7	[289.0,2.0,287.0]
3.8687863980166606	0.78	[5059.0,39.0,5020.0]
5.506505082946465	1.12	[4599.0,51.0,4548.0]

Fig 9: Prediction Column for Random Forest

A. HyperParameter Tuning

The built model is the tuned using hyperparameter tuning and the method chosen is Train Validation Split. By implementing Linear Regression algorithm, the model is tuned to produce optimized value and better perception.In general, to improve the prediction accuracy of the trained model, hyperparameter tuning is used.

features	label	prediction
[246286.0,1662.0,...]	0.68	9.241321131457799
[9132.0,98.0,9034.0]	1.08	14.336751651536426
[2513.0,38.0,2475.0]	1.54	14.475933105974457
[301708.0,4764.0,...]	1.6	8.558165394245743
[462.0,11.0,451.0]	2.44	14.517500925236522
[1132.0,27.0,1105.0]	2.44	14.505253860410901
[907.0,23.0,884.0]	2.6	14.509621400458085
[14203.0,387.0,13...]	2.8	14.275203958824356
[227019.0,6447.0,...]	2.92	10.566486739971866
[50299.0,1528.0,4...]	3.13	13.667154199699686
[116458.0,3533.0,...]	3.13	12.536674567540102
[701.0,24.0,677.0]	3.55	14.514484979615759
[2305.0,94.0,2211.0]	4.25	14.491044737615399
[24.0,1.0,23.0]	4.35	14.52559249112532
[1167.0,60.0,1107.0]	5.42	14.510580622108927
[853.0,44.0,809.0]	5.44	14.514743243676852
[114.0,6.0,108.0]	5.56	14.524476183406323
[265.0,14.0,251.0]	5.58	14.522531127284006
[50838.0,2803.0,4...]	5.84	13.891429601715728
[39482.0,2546.0,3...]	6.89	14.101640130619082

Fig 10: Prediction Column using Train Validation Split[Tuned Model]

IX. Conclusion and Future Work

The report presents the Covid-19 Analysis, model implementation done on Apache Spark MLlib and PySpark as its interface. After processing the dataset it was split into train and test dataset, further four different model were implemented to find the best fit model. Out of which Random Forest Regressor produced lower RMSE value compared to other models and Linear Regression also produced lower value in comparison with Random Forest but the R squared value was negative , hence Random Forest Regressor was chosen as its best fit. The trained model is further tuned by hyper parameter tuning to improve the accuracy of the model, hence by train validation split method the model was tuned using linear regression.

During the implementation phase, using Seaborn and PyPlot analysis were made to find the top countries lead highest Confirmed cases, Death, Recovered and Active cases. This library helps to visualize the dataset to provide more content through graphical representation.

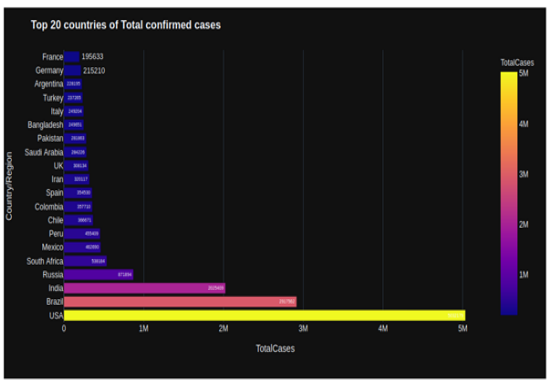

Fig 11: Top 20 countries with total confirmed Cases

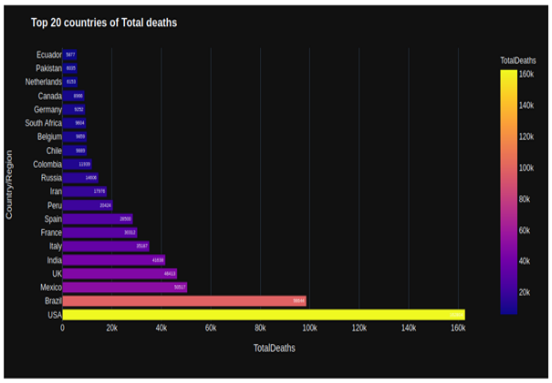

Fig 12: Top 20 countries with total deaths

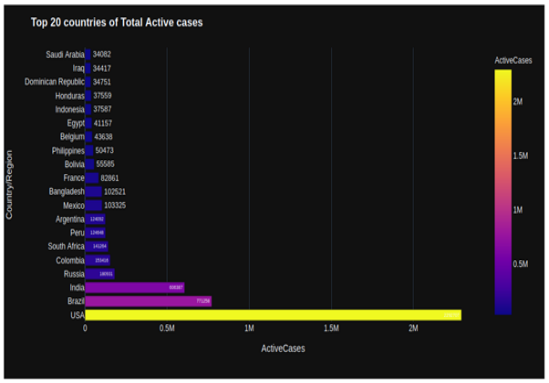

Fig 13: Top 20 countries with total active cases

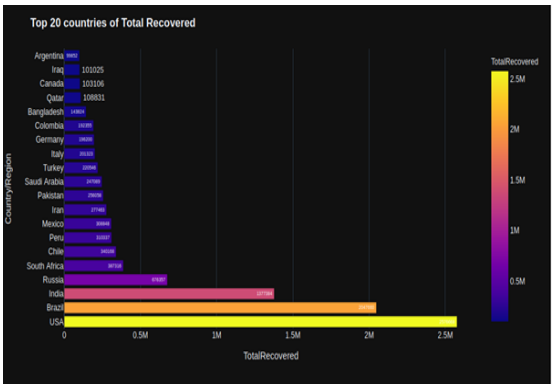

Fig 14: Top 20 countries with total recovered cases

A future work is to implement the model using other regression models to improve its efficiency and execute it in a high end system to increase its performance that consumes lower work load and increased computational speed.

X. References

1) https://spark.apache.org/docs/3.1.1/ml-guide.html
2) Ekta Gambir, Rikita Jain and Uma Tomer.(September,2020).Regression Analysis of COVID-19 using Machine Learning Algorithms.
3) Wenxing Hong, Ziang Xiong, Nannan Zheng.(September,2019).A Medical-History-Based Potential Disease Prediction Algorithm.
4) https://stackoverflow.com/
5) https://towardsdatascience.com/