COVID-19 Time Series Analysis

Priyanka Sahoo, *MSc. Big Data Analytics,Letterkenny Institute of Technology*, *L00151175@student.lyit.ie*

***Abstract*— "Novel Corona virus, Pandemic, Covid-19, Lockdown" are the only buzzwords in the past year across the planet. Within a few months this contagious corona virus spread from country to country universally and disintegrated almost every industry considered as pillar of world's economy e.g. health-care, transportation, aerospace, agriculture, entertainment, education, manufacturing, hospitality and so on leaving a trail of crumbled humanity behind. This research paper will mainly focus on dynamics of COVID-19 pandemic worldwide with time series analysis and its future predictions. There is a discussion on the trend analysis of the disease spread on the world map in addition to the visualizations of new cases, confirmed, recovered and death in the most affected countries with various graphs. The most important features of this exploration are established with segregating the world population into confirmed, death, recovered and new cases corresponding to different countries. Various machine learning regression models are developed and trained with the World Health Organization (WHO) and ArcGIS (Geographic Information System) steadfast data. The study will demonstrate a life cycle of data science project prototype using BigData Analytics and architecture. Notable results are observed with truthful cumulative predictions country wise. Moreover, there are also discussions on comparison of the root mean squared values from different models in the conclusion part with regards to validation of machine learning model's credibility and the future statistics of COVID-19 could be helping hand for government, NGOs, hospitals etc. for the battle against present pandemic.**

***Index Terms*— Databricks, Apache Spark, Time-series Analysis, Regression, MLlib, Random Forest, K-means clustering, Gaussian Mixture, Forecasting.**

I. Introduction

COVID-19 is the foremost question in the year 2020. World health organization declared this as global pandemic during the first quarter last year, impacted more than 90% of the continent's population. It is spreading exponentially daily and become a never-ending process. Few medical studies even revealed that this corona virus will reside inside human body eternally, only antibodies within us can conquer it gradually with time. As per today's date, human race is trying to balance off between the corona virus vaccines and the new covid-19 strains emerging from different realms. In this research paper, there will be analysis on the pattern of spread using the features like new and confirmed case, mortality rate and recovered records in most affected countries with respect to time.

Time series analysis can be defined as the study done on a data set where each data point is observed under particular time instance [1]. As part of this venture,various analytics process will be able to appease further down queries.

1) *Stats of total deaths and recovered daily/weekly/monthly across all over the world.*
2) *COVID-19 variations and trends over time.*
3) *COVID-19 cases spread Visualization across countries in 2019-2020.*
4) *Predictions for the new confirmed cases, death and recovered numbers through machine learning regression models.*
5) *Forecasting the future of pandemic through deep learning.*

In view of, variation of COVID-19 impact with time, forecasting for count of new confirmed cases and deaths for near future done using deep learning. Apart from COVID-19 trends visualization, the study presents comparison between 7 types of regression models and their integrity. The research paper is organized in different units coined as dataset overview, system architecture, exploratory data analysis, ML model development, forecasting, results, and conclusion. Additionally, the whole analytics process is done in Databricks community edition that provides clusters that run on top of AWS and adds a convenience of already hosted notebook system. It supports variety of spark APIs as well and has its own storage like DBFS, S3, Hive etc.

II. DATASET OVERVIEW

The dataset used for the study is extracted from 2 different websites, ArcGIS and WHO.

ArcGIS is a geographic information system (GIS) provided by Environmental Systems Research Institute (ESRI) for developing maps and maintaining analyzed topographical information through spatial analytics software. The data file is in Javascript Object Notation (JSON) format, i.e. schema-less, text-based representation of structured data based on key-value pairs and ordered lists [2].

Moreover, the JSON file used in the analysis is in nested format with key column as *"features"*, mentioned in figure 2. This file contains aggregated data for 764 territories around the globe. There are other key columns available as well in

Under the supervision of Dr. Shagufta Henna, Letterkenny Institute of Technology, Letterkenny, CO. Donegal

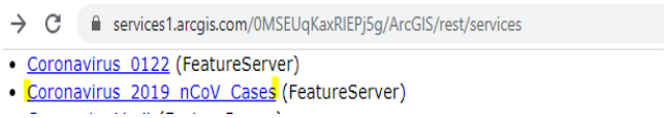

Fig. 1: *Snapshot of the url from where the live covid-19 stats drawn.*

the raw dataset when directly extracted from the url, such as *"fields"*, *"geometryType"*, *"globalIdFieldName"*, *"objectIdFieldName"*, *"spatialReference"*, *"uiqueIdField"*, displayed in figure 3. Most of the columns contains null values in this dataset which will be removed in the first instance of analytics process displayed in figure 3.

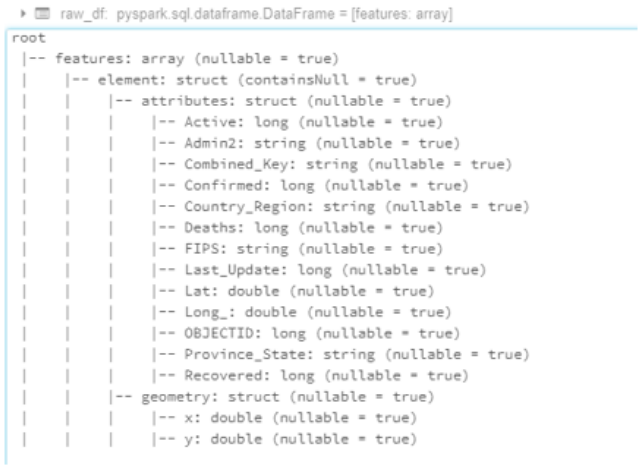

Fig. 2: *Snapshot of the json datafile in nested format and the 'feature' column.*

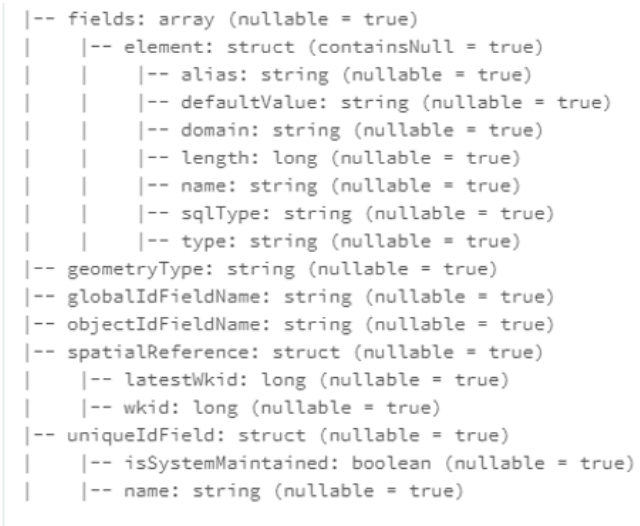

Fig. 3: *Snapshot of the json raw datafile in nested format*

The JSON data set utilized in developing the COVID-19 spread visualizations. The "feature" key comprises the subsequent columns.

- *Country_Region, is the countries all over the world*
- *Last_Update, is the last updated dates of the facts in the website*
- *Lat, is the latitude of the corona affected region*
- *Long, is the longitude of the corona affected region*
- *Confirmed, is the number of people has confirmed affected with covid-19*
- *Recovered, is the number of people having recovered from covid-19*
- *Deaths, is the number of people died due to covid-19*
- *Active, is the number of people affected with corona virus and not yet in good health.*

Similarly, the dataset extracted from World Health Organization (WHO) is a comma separated file containing around 99,500 records from the period Jan-2020 till date. The key features from this dataset are countries and their cumulative figures for COVID-19 new cases and deaths.

Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
1/4/2021	IE	Ireland	EURO	4961	101887	7	2259
1/5/2021	IE	Ireland	EURO	6110	107997	6	2265
1/6/2021	IE	Ireland	EURO	5325	113322	17	2282
1/7/2021	IE	Ireland	EURO	7832	121154	17	2299
1/8/2021	IE	Ireland	EURO	6503	127657	8	2307
1/9/2021	IE	Ireland	EURO	8227	135884	20	2327
1/11/2021	IE	Ireland	EURO	6886	147613	8	2344

Fig. 4: *Snapshot of .csv raw datafile used for Time-Series Analysis and forecasting.*

This file is mainly used for time series analysis as the statistics are captured on each day of the year and later the same would be split into training and test dataset for analytics using machine learning models. Key columns are following,

- *Date_reported, is the date when the covid positive declared*
- *Country_code, is the specific code for the countries*
- *Country, is the name of countries all over the world*
- *New_cases, is the number of new cases of covid positive people*
- *New_deaths, is the number of people died due to corona virus*
- *Cumulative_cases, is the cumulative summation of the new cases.*
- *Cumulative_deaths, is the cumulative summation of the death cases.*

Further down the process, these datasets would be cleaned by filtering the dimensions which may not be more efficient or not holding valuable information. The key performance indicators will be well-thought-out for better analytics and foresight with respect to date dimension.

III. BIG DATA ANALYTICS, ARCHITECTURE AND TOOLS

In continuation with dataset overview, this section will put the spotlight on architecture and tools utilized down to line for big data analytics. Big data is about 5 Vs, ie, volume, variety, velocity, variability & value. Another way of defining big data is, when we need more than 1 computer to process the dataset then it could be observed as big data. This type of data set requires a distributed cluster computing framework like spark also known as analytics operating system. Describing the key element of spark, distributed layer and analytics layer will shield between different applications built on the top and the hardware and actual operating system as shown below [19].

The study is done, exploiting Python Scikit learn and MLlib

Fig. 5: *Snapshot of the distributed architecture*

Spark libraries. With the help of below snap it would be easy to be understand this research's literature related to spark, i.e. *Pyspark = Spark + MLlib + python*

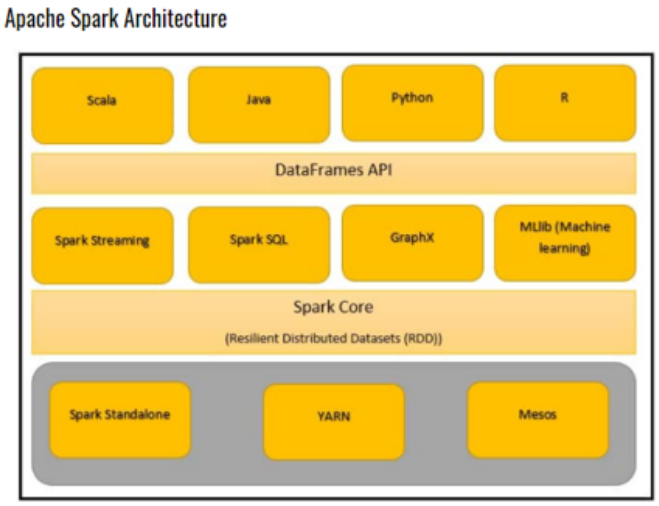

Fig. 6: *Snapshot of Apache SPARK architecture*

In this seek, Databricks is used due to its optimized usage cost, readily available notebook system and compatibility with python plus spark. The procedure is conducted on cluster of 16 GB and 2 cores processor having configuration of DBR 7.5, Spark 3.0.1, Scala 2.12. Moreover, databricks filesystem (DBFS) is used to store the data. DBFS is distributed file storage system, installed on Databricks work space.

A. Objective

The aim of the practical is to analyze the confirmed, death, recovered, and corresponding cumulative cases counts w.r.t countries and dates. There will be a time-series analysis on daily counts and visualization of spread across the globe. The focus would be on impact of new cases verses death counts using regression and clustering methods. Implementation of Linear regression, K-Means clustering, Gaussian mixture algorithm, Random forest and analysis with *'root mean squared error'* and *'effective variance'*.

B. Materials & Methods

- Python3, Databricks Notebook.
- Apache Spark version 3.0.2 installed from url- "https://www-us.apache.org/dist/spark/spark-3.0.2/spark-3.0.2-bin-hadoop3.2.tgz"
- Datasets sourced from ArcGIS and WHO websites, "https://services1.arcgis.com/FeatureServer/1/query" and "https://covid19.who.int/WHO-COVID-19-global-data.csv"
- OLS, LASSO, Ridge, Elasticnet, Bayesian, OMR Regression models imported from python Scikit learn package.
- Linear regression, K-Means clustering, Gaussian mixture and Random forest models imported from MLLib PySpark.
- Other libraries used for data processing, visualization and forecasting are pyspark, pandas, numpy, matplotlib, plotly, folium, fbprophet etc.
- Help for code development taken from the internet site "https://scikit-learn.org/stable/ " and "http://spark.apache.org/docs/latest/" [13] [21].

IV. DATASET EXPLORATION

Till now, exploring the dataset used for study and big data analytics tools impression is accomplished. The very next step could be considered as reconnoitering the information in json file pulled from ArcGIS and visualizing in the 2-dimensional displays. Python and its vast libraries are exploited for slicing and dicing. The total counts of COVID-19 new, confirmed, recovered and death cases is plotted against date and countries.

The json data file is pulled directly from ArcGIS url to get the latest information for visualization. Main aim is to get the interactive graph plots displaying the results for the same day as the code executed without any lag in latest statistics. As discussing in the above section dataset overview, first column *'features'* is sliced out for further examination. Excavating deep through the facts, the figure 7 is a snapshot of the raw, normalized data from json file. Column *'states'* corresponding to *'Countries'* were filled with null values initially, which was replaced with spaces. The column *'Last_Update'* holds date values like *[1.614418e+12]* which is formatted to actual timestamp value e.g. *[2021-02-27 00:23:52]* as part of data preprocessing. Popular python libraries 'Plotly Express' and 'Plotly Graph Objects' are used for plotting the features. It accepts the Pandas data frames and description about the graph to scatter the data with a simple syntax as , *px.scatter(data,*

	State	Country	Last Update	Lat	Long	Confirmed	Recovered	Deaths	Active
0		Afghanistan	1.614418e+12	33.93911	67.709953	55696	49285	2442	3969
1		Albania	1.614418e+12	41.15330	20.168300	105229	68007	1756	35466
2		Algeria	1.614418e+12	28.03390	1.659600	112805	77842	2977	31986
3		Andorra	1.614418e+12	42.50630	1.521800	10822	10394	110	318
4		Angola	1.614418e+12	-11.20270	17.873900	20759	19307	504	948

Fig. 7: *Snapshot of raw json data after flattening*

="column_name", y="column_name") [18].

Here, the maximum number of corona virus confirmed cases are aggregated by equivalent countries, organized in descending order. Figure 8 displays the 10 countries having the highest confirmed cases, where US, India and Brazil are leading the queue. The countries are designed on x-axis and count of confirmed cases on y-axis. Speaking about the facts, US has crossed 28 million, India with 11 million, Brazil with 10 million, Russia and United Kingdom with 4 million and so on. These counts in million are nothing but human lives turned down by a mere virus. Isn't it startling?

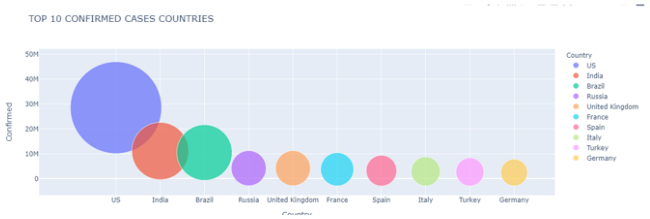

Fig. 8: *Visualization of most affected countries*

The next visualization is done on the death counts across the countries, followed by recovered and active counts, refer to figure 9,10,11. The exact numbers can be read by just hovering the mouse over the plot observed during practical. This feature is an embedded version of *'Plotly'* for developing more informative and interactive graphs.

In the figure 9, its clearly visible the top 10 countries of the planet having the highest number of deaths counts due to COVID-19. In US alone, more than 5 lakhs people died and accompanying 28 million people are still fighting for their life and the numbers are still increasing with stint worldwide, refer to figure 10.

But still, figure 11 is philanthropic that there are quite good number of patients who recovered. And the plots are clearly showing India, Brazil, Russia and Turkey leading the column with highest recovered figure.

Moreover, breaking down the cases country wise corresponding to the provinces, surely these visuals will leave

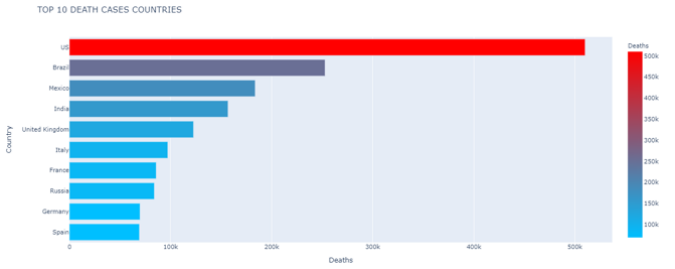

Fig. 9: *Visualization of top 10 countries having highest death counts*

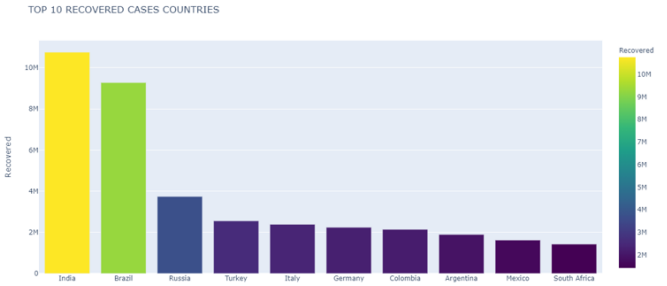

Fig. 10: *Visualization of top 10 countries having highest recovered counts*

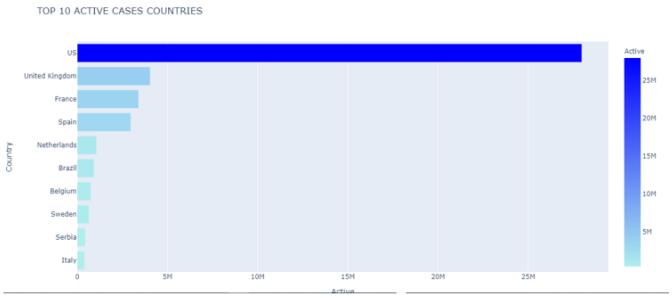

Fig. 11: *Visualization of top 10 countries having highest active cases*

the readers alarmed. Figure 12 shows that California, Texas, Florida, New York, and Illinois are the most affected states of USA. As these plots are created interactive during hands-on, so while the mouse is moved on top, easily can be read 3.58 million of active cases marked under blue block, 3.53 million confirmed cases marked under red and green shows 53.6 thousand died in California.

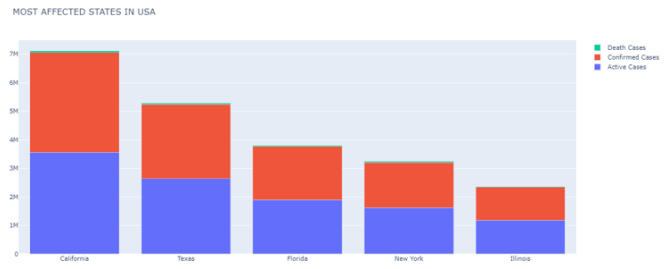

Fig. 12: *Visualization of most affected states of USA*

Maharashtra, Kerala, Karnataka, Andhra Pradesh, and Tamil Nadu are badly affected states in India, refer to figure 13. Similarly, active, confirmed and death cases tallies are displayed

in figure 14,15 visuals for topmost affected countries Brazil and United Kingdom, respectively. The interactive graphs developed during the research practical will give the exact counts for each states once the mouse hovered over them.

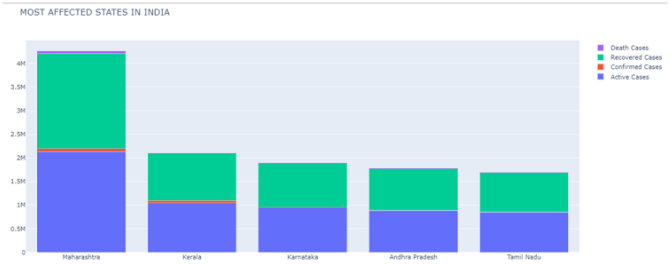

Fig. 13: *Visualization of most affected states of INDIA*

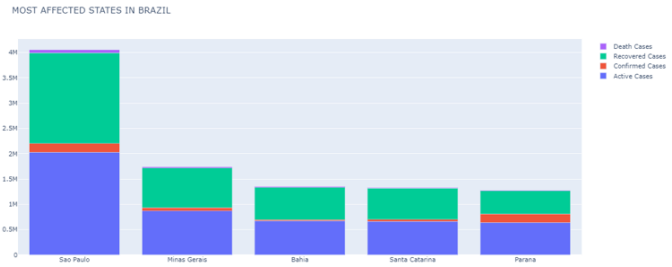

Fig. 14: *Visualization of most affected states of BRAZIL*

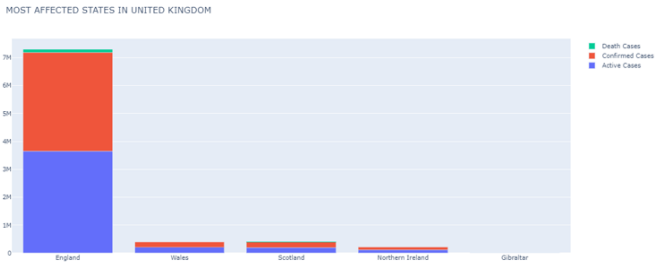

Fig. 15: *Visualization of most affected states of United Kingdom*

V. EXPLORATORY DATA ANALYSIS

In this section, there will be more focused discussion on the time series analysis on up-to-the-minute WHO COVID data stored in databricks file system read as spark data frames as well as pulled directly from API for including today's data also.

At the outset, the comma separated files are warehoused to the databricks distributed file system. Then the stored data files are read and constructed spark data frames for further data wrangling. The dataframe contains features as *'New_cases'*, *'Cumulative_cases'*, *'New_deaths'*, and *'Cumulative_deaths'* that is aggregated by *'Date_reported'*. And then by using the functionality of *'Plotly Graph Objects'* cumulative news cases and death are scattered in 2-dimension [14].

A. Time Series Analysis

Figure 16 and 17, displays dates values started from 3-Jan-2021 to till date labeled on X-axis and the total counts are labeled on Y-axis. A linear growth is observed for both new cases and death with respect to days and months.

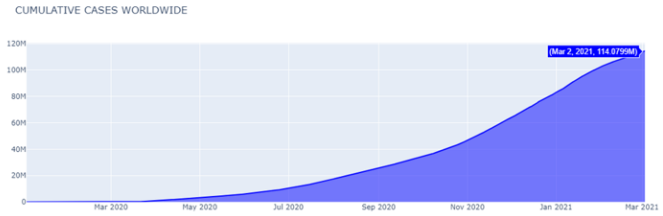

Fig. 16: *Visualization of Cumulative cases worldwide*

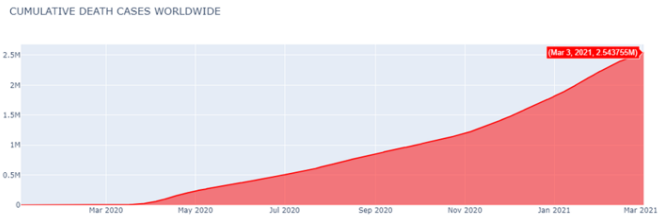

Fig. 17: *Visualization of Cumulative death cases worldwide*

However, when the similar graph is plotted with daily counts of new cases and deaths the sketch is no more linear. There are multiple spikes observed with respect to date reported, e.g. highest number of cases, 842K reported on 20-Dec-2020 , followed by a drop in counts, 541K on 22-Dec-2020 and then a sudden rise by 643K on 24-Dec-2020. In case of death counts maximum numbers are observed in the month of January 2021. This could be assumed that the chilling winters worsen the effect and symptoms by corona virus. Refer to figure 18,19.

Down the line, if the cumulative figures are broken down country wise then time series for most affected countries are observed. As per the graph plotted in Figure 20, USA has been observed polynomial regression whereas Brazil shows a linear line. And the same scenario is with time series of most affected countries with respected to cumulative death cases,

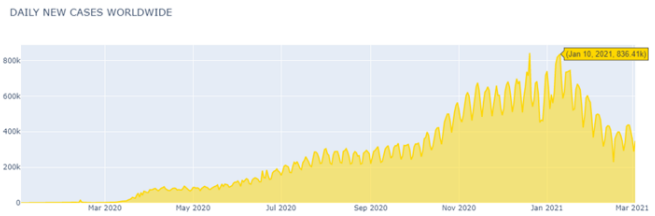

Fig. 18: *Visualization of daily new cases worldwide*

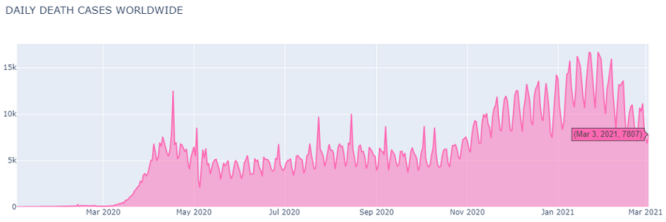

Fig. 19: *Visualization of daily death cases worldwide*

refer to figure 21. However, in the time series analysis for the most affected countries on daily data, the graph is fluctuating like series sines and cosines wave approximately exemplified in Fourier series as shown in figure 22 and 23.

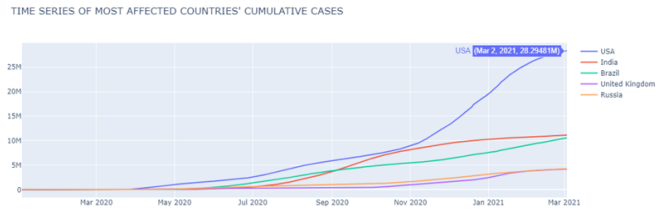

Fig. 20: *Time series of most affected countries cumulative new cases*

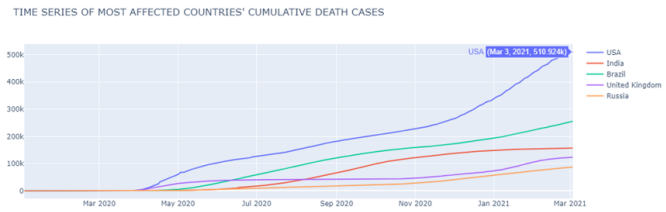

Fig. 21: *Time series of most affected countries cumulative death cases*

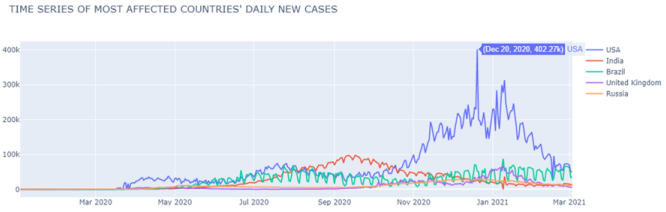

Fig. 22: : *Time series of most affected countries daily new cases*

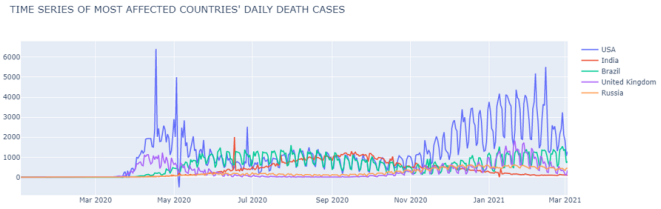

Fig. 23: *Time series of most affected countries daily death cases*

Lastly, as one of the goal of this study was, visualizing the COVID-19 spread in the world map. Here all the corona virus affected countries are highlighted in 2-dimensional world map in figure 24. This has been achieved by using *'folium'* library in python. The countries in the world map, encircled in red, can be visualized effectively by zooming in/out option in the graph created during the research practical.

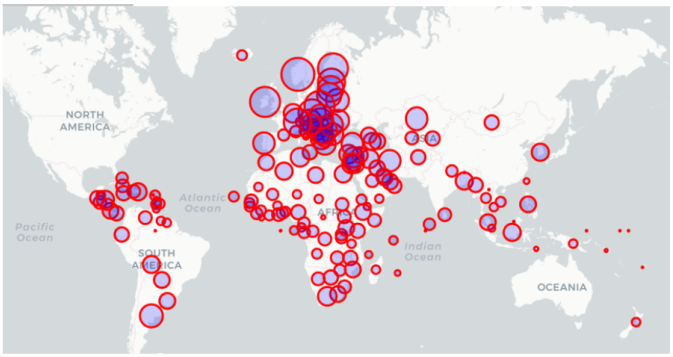

Fig. 24: *Visualization of Cumulative cases worldwide*

VI. MACHINE LEARNING MODELS

Machine learning (ML) is defined as the subgroup of Artificial Intelligence (AI) that mainly focuses on development of applications that learn from existing data, past experiences etc. and then improvising itself in decision making, predictions and forecasting. The algorithms are prepared to identify various patterns, trained to select specific characteristics and attributes in vast and diversified data sets. Various mathematical and statistical models are also used as part of ML. A classic approach followed in big data analytics starts from learning the training data, implementing into models and the provides the output or execute any assignment.

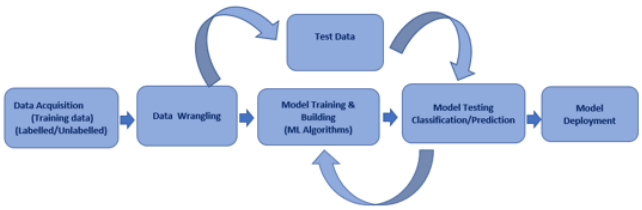

Fig. 25: *Classic approach in Big Data Analytics*

This section will demonstrate the implementation of below ML regression models by using Sklearn and Pyspark MLlib.

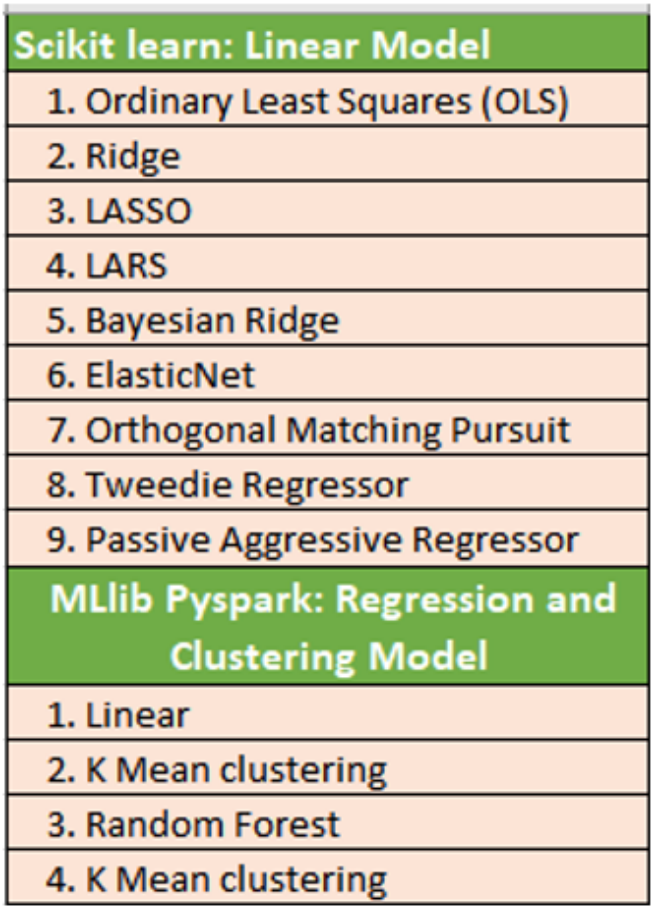

Fig. 26: *ML models used in this study*

A. Linear Regression models by Scikit learn

Regression based algorithms analyze the input data features and corresponding continuous numeric output values to provide predictions.

We analyzed the relationship between new COVID-19 new cases and the new death reported using various linear regression models. The value of a dependent variable with respect to variation in the independent variable feature is determined.

Ordinary least-squares is another name for the linear regression model. It follows the equation of a line, y = mx + b where y is the dependent variable and x is the independent variable. Here, the relation between new cases and corresponding impact on new deaths will be observed. However, there are multiple regularization techniques used with regression models.

Discussion more about regularization, ridge regression is applied when the data suffers from multi collinearity (independent variables are highly correlated). In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. Similarly, least absolute shrinkage and selection operator

	Models	EVS	R2
0	OLS	1.000000	1.000000
1	Ridge	1.000000	1.000000
2	Lasso	0.999926	0.999925
3	Bayesian	1.000000	1.000000
4	ElasticNet	0.999926	0.999925
5	Lasso LARS	1.000000	1.000000
6	OMR	1.000000	1.000000
7	Tweedie	0.997821	0.997170

Fig. 27: *Effective Variance and R-square values for different models*

(LASSO) penalizes the absolute size of regression coefficients, shrinking to zero that helps in feature selection. And Elastic-Net is hybrid of both Ridge and LASSO. It is useful when there are multiple correlated features [22]. Bayesian model represents the uncertainties in predictor variables and evaluates the probability distribution for model parameters [23]. And Tweedie regression is a case of exponential dispersion used for generalized linear models distribution.

Figure 27 displays the effective variance score(EVS) and R-squared errors for different regularized regression models. The data set is split into 75:25 ratio for training and test data. The square root of the average of the squared differences between the predictions and the actual values is calculated for all the regression models. OLS, Ridge, Bayesian and OMR models are perfect fit with used dataset.

B. K-Means Clustering using PySpark MLlib

Clustering algorithms are used to structure/group the data set. The data set for new cases and death cases are implemented over KMeans clustering techniques. Generally, there was use of elbow method to get the value of optimal cluster numbers for the data set segregation and Inertia defined as the sum of squared distances of samples to their closest cluster center in implemented. Within Cluster Sum of Squares (WCSS), which measures the squared average distance of all the points within a cluster to the cluster centroid. No of cluster is the value selected where wcss graph is reduced significatly. Silhouette core metric determine the efficiency of clustering procedure [24]. Its value ranges from -1 to 1, where 1 signifies clusters are distinguished, 0 means distance between clusters is not significant and -1 shows clusters are assigned not in correct way.

$$\text{WCSS} = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Fig. 28: *Within clusters sum of squares, WCSS*

In this study, the Silhouette values with squared euclidean distance = 0.8605849166486804 It can be assumed that the clusters are well apart from each other as the silhouette score is closer to 1. Further, cluster centers are calculated as below.

Cluster Centers: [1.53848763e+05 1.21830817e+07 3.93647697e+03] [5.64136129e+05 8.14667517e+07 1.11559224e+04]

C. Gaussian Mixture Model using PySpark MLlib

A Gaussian Mixture Model represents a composite distribution whereby points are drawn from one of k Gaussian sub-distributions, each with its own probability [13]. The assumption considered here that there are certain number of clusters defined by Gaussian distributions [25], and this model group the data points belonging to a single distribution together. Below figure demonstrate the mean and co-variance calculated by GMM.

```
Gaussians shown as a DataFrame:
+-----------------------------------------+-----------------------------------------
---------------+
|mean                                     |cov
|
+-----------------------------------------+-----------------------------------------
---------------+
|[167189.33595096352,1.4159609501875635E7]|2.300949580416308E10  2.405138497982195E12
2.405138497982195E12  3.002765937924109E14      |
|[562825.7551337301,8.206240450289537E7]  |1.6845400903131672E10  -7.412261202165865E11
-7.412261202165865E11  4.303977643848395E14   |
+-----------------------------------------+-----------------------------------------
```

Fig. 29: *Mean and Co-Variance by Gaussian model*

D. Linear Regression Regularization (LASSO, RIDGE, ELASTICNET) using PySpark MLlib

As there is always possibility of sparsity within the dataset, training data may tend to ill-posed. This could be accurate by regularization of the regression models and avoid overfitting.

- *Ridge: This method diminishes the coefficients of correlated values*
- *LASSO: This method selects one variable, while dropping the others*
- *ElasticNet; This is a combination of above methods. It also eliminates corrupted and unpredictable behavior due to extreme correlations.*

The above theory is referred from [17]. Figure 30, 31 and 32 shows the variation in 'regParam' and 'elasticparam' values to implement L1, L2 , L1+L2 regularization and root mean squared error calculation.

lasso_model [rootMeanSquaredError = 3.4876973388182693]

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{n} \|\mathbf{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{\beta}\|_1$$

When $\lambda > 0$ (i.e. `regParam` > 0) and $\alpha = 1$ (i.e. `elasticNetParam` $= 1$), then the penalty is an L1 penalty.

Fig. 30: *LASSO*

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{n} \|\mathbf{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2 + \lambda \|\boldsymbol{\beta}\|_2^2$$

When $\lambda > 0$ (i.e. `regParam` > 0) and $\alpha = 0$ (i.e. `elasticNetParam` $= 0$) , then the penalty is an L2 penalty.

Fig. 31: *Ridge*

$$\min_{\beta \in \mathbb{R}^n} \frac{1}{n} \|\mathbf{X}\boldsymbol{\beta} - \boldsymbol{y}\|^2 + \lambda(\alpha \|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_2^2), \alpha \in (0,1)$$

When $\lambda > 0$ (i.e. `regParam` > 0) and `elasticNetParam` $\in (0,1)$ (i.e. $\alpha \in (0,1)$) , then the penalty is an L1 + L2 penalty.

Fig. 32: *ElasticNet*

ridge_model [rootMeanSquaredError = 3.5192293262510668]

elastic_model [rootMeanSquaredError = 3.5002505823896928]

E. Random Forest Regression using PySpark MLlib

Random Forest algorithm is an extension of Decision tree classification ML model. It improvises the weakness of decision tree as it tends to overfit. In simple terms, it is an ensemble learning method means expanding several decision trees at a single instance and finally choosing the regression outcome as average value [5]. It will randomly choose the observations and data features to build multiple trees.

Here, the model is first trained with training data and then its performance is evaluated using test data. The prediction error is measured through Root Mean Squared Error(RMSE), which is nothing but the average difference between observer known value and the values predicted by the model [16]. And importantly lower RMSE values defines better model.

'Root Mean Squared Error (RMSE) on test data = 3.53081'
'Root Mean Squared Error (RMSE) on train data = 2.15716'

Further, the prediction values are plotted against the date reported, clearly shows an flat line having a prediction value of 11.7.

prediction	newcaselog	features
11.706049478408307	0.0	(420,[0],[1.0])
11.706049478408307	0.0	(420,[2],[1.0])
11.706049478408307	0.0	(420,[4],[1.0])
11.706049478408307	0.0	(420,[5],[1.0])
11.706049478408307	0.0	(420,[12],[1.0])
11.706049478408307	1.791759469228055	(420,[14],[1.0])
11.706049478408307	4.418840607796598	(420,[17],[1.0])
11.706049478408307	6.677083461247136	(420,[24],[1.0])
11.706049478408307	7.488293515159428	(420,[25],[1.0])
11.706049478408307	7.866338923046544	(420,[30],[1.0])
11.706049478408307	8.089175678837561	(420,[32],[1.0])
11.706049478408307	8.221747728346623	(420,[34],[1.0])
11.706049478408307	8.13593277200489	(420,[36],[1.0])
11.706049478408307	7.913155185928068	(420,[43],[1.0])
11.706049478408307	7.597897950521784	(420,[46],[1.0])
11.706049478408307	6.3784261836515865	(420,[52],[1.0])
11.706049478408307	7.385230923066573	(420,[58],[1.0])
11.706049478408307	7.711996507047669	(420,[60],[1.0])

Fig. 33*: Snapshot of prediction newcaselog and feature [date_reported]*

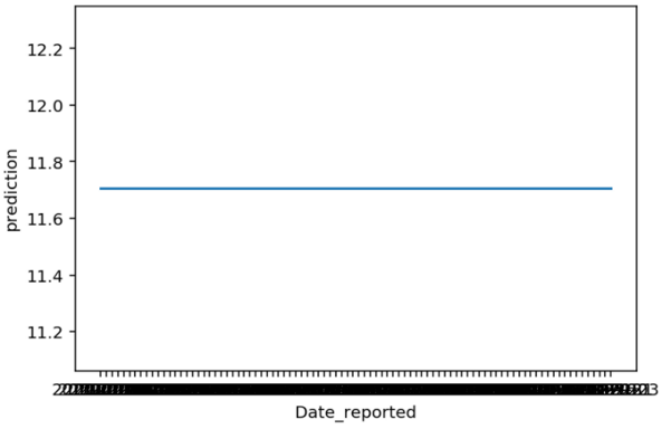

Fig. 34*: prediction vs date_reported plotting*

VII. FORECASTING WITH DEEP LEARNING

The above observations from ML regression models evidently shows good fit with the current dataset. Now, let's proceed with discovering concepts of Deep Learning for forecasting the stats of new cases and deaths in coming months. Here, Prophet is used for time series prediction. It decomposes the time series into trends, seasonality, and holidays. It has inbuilt easily tuned, instinctual hyper parameters [6].

'Prophet time series = Trend + Seasonality + Holiday + error'

The non-periodic changes in the value of time series are modelled by *''Trend'*. Any periodic change like daily, weekly, or yearly is called as *''Seasonality'*. *''Holiday'* effect which occur on irregular schedules over a day or a period of days and any term which is not explained by the model is defined as *''Error'*.

Prophet accepts the pandas dataframes with minimum 2 parameters, datetime value and respective feature (must be numeric value) which needs to be predicted. Hence, this could be considered as a suitable implementation for forecasting the new covid cases and deaths counts with upcoming dates. The values of *'new_cases,cumulative_cases'*, *'new_deaths'*, and *'cumulative_deaths'* are plotted against *'Date_reported'* in 2-dimension.

A. Forecasting New and Cumulative cases

In figure 35, 36, 37, the light blue lines, for the period March to May 2021, forecasts that there will be decrease in the count of new cases while a linear regression trend will be observed. A very important observation here that the new cases drastically increased, reported over the weekends. That's obvious due to lack of social distancing among people during weekends. If there is discussion on yearly forecasting, the counts will rise by 40 percent during year end.

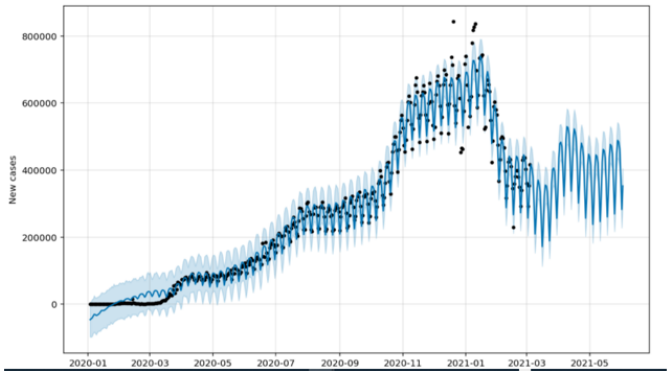

Fig. 35*: New cases vs. Date_reported forecasting*

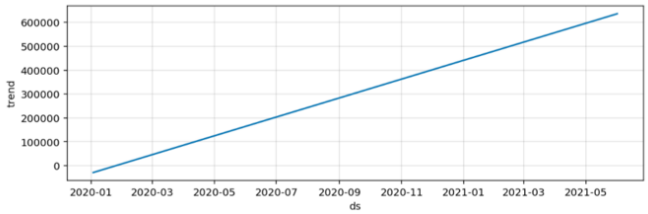

Fig. 36*: New cases vs. Date_reported forecasting trends*

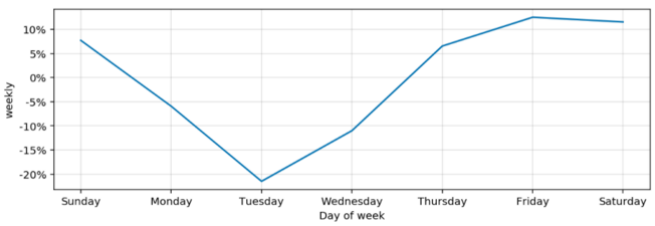

Fig. 37*: New cases vs. Date_reported forecasting weekly*

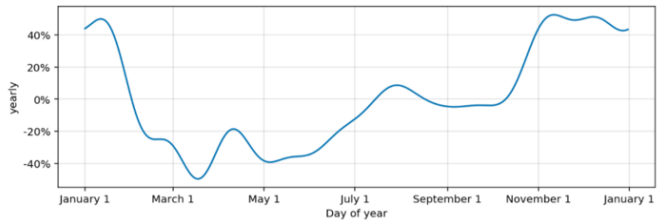

Fig. 38: *New cases vs. Date_reported forecasting yearly*

B. Forecasting Death and Cumulative cases

Similarly, the below figures demonstrate the forecasting of death cases will be increase with time. Figure 39 and 40 predicts a range between 15000 to 5000 death cases could be observed in coming months. Even the death cases are surprisingly maximum during weekends.

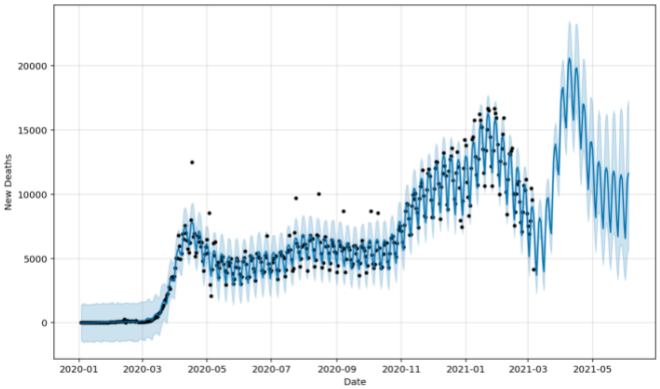

Fig. 39: *Death cases vs. Date_reported forecasting*

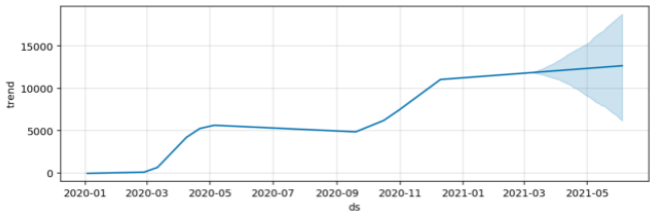

Fig. 40: *Death cases vs. Date_reported forecasting trends*

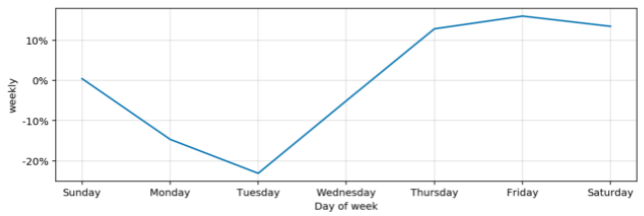

Fig. 41: *Death cases vs. Date_reported forecasting weekly*

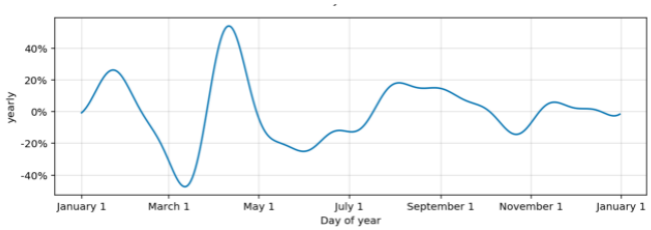

Fig. 42: *Death cases vs. Date_reported forecasting yearly*

VIII. OUTCOME DISCUSSION/ EXPERIMENTAL RESULTS

Following the findings, in this research paper, an observation could be made for the pattern in COVID-19 spread globally. US, India, Brazil, Russia, UK, France, Spain, Italy, Turkey, and Germany are the topmost affected countries in the world. The month of December 2020 displays the maximum spread of virus in USA, which could be assumed that this may happened due to Lockdown restriction liberty during Christmas. India has a spike in confirmed cases in the month of September, after which Indian Government declared Level 5 restriction and here it shows quite a good decrease by October. The time series analysis between the cumulative new cases and deaths generates a polynomial curve of nth degree for various countries.

Discussing over the effective variance and root mean squared error experienced from OLS, Ridge, Lasso, Baeyesian, ElasticNet, Lasso Lars, OMR and Tweedie regressor the values are approximately same as 1.0. This clearly shows an linear increase in Cumulative death cases with respect to new cases. And the same is forecasted as well using Fbprophet time series predictions.

The comparison between the least R- squared values calculated using linear regression and n-degree polynomial regression model proved the later one is good approach. However, higher order of polynomial could result in overfitting where the curve is going out of its way to accommodate outliers. A high r-squared simply means the curve fits training data well, but it may not be a good predictor [20]. Additionally, The K-means clustering model shows a defined clusters with silhoutte score of 0.86. The inter cluster distance called as centroids is also calculated to check the cluster centers. However, k-means only considers the mean to update the centroid while Gaussian Mixture model takes into account the mean as well as the variance of the data [25]. The values observed in GMM are mote distinct as compared to Kmean clustering.

Random forest regressor algorithm is also used to do predictions for new cases with respect to dates. There is a calculation of R-square on both train data and test data have values as 2.15716 and 3.53081 respectively. Larger R-square values signifies better regression model that fits to the observations. Here it shows test data performs better that trained data in predicting the new cases with respect to date. The trained dataset is fitted in the model and got an average prediction of 11.7.

IX. CONCLUSION

This paper observed the importance of Big data Analytics with this COVID-19 science. It is so much required to predict the future possibilities in such random and unique occurrence of the pandemic around the globe. In this data driven era, analysis of the existing dataset greatly impact decision making process. There was an analysis on the highest count of active, recovered and death cases in countries. Then a time series analysis and forecasting done on the daily/cumulative new case and death cases worldwide with visualization of the affected countries on the globe.

Statistical forecasting helps to prepare researchers and scientists with the deep aftereffects of this pandemic and future preparedness. Also, organizational, and social entities would get course to handle tough situation in the country. Hence, various regression and clustering models implemented, their root mean squared error is compared to preform accurate predictions.

On the closing note, would only say that accurate results mostly depend on various features used. Sometimes, there may be unforeseen scenarios which may not fit with a specific technique for the first time. As said earlier, big data analytics with machine learning will get trained and become more mature with the experience. More comprehensive studies to describe the context of a broader type of time series analysis with covid data are currently being considered by larger research communities.

REFERENCES

[1] Time series analysis of COVID-19 cases — Emerald Insight." https://www.emerald.com/insight/content/doi/10.1108/WJE-09-2020-0431/full/html (accessed Feb. 19, 2021).

[2] J. Freeman, "What is JSON? A better format for data exchange," InfoWorld, Oct. 25, 2019. https://www.infoworld.com/article/3222851/what-is-json-a-better-format-for-data-exchange.html (accessed Feb. 25, 2021).

[3] "API Reference — scikit-learn 0.23.2 documentation." https://scikit-learn.org/stable/modules/classes.html (accessed Nov. 29, 2020).

[4] "10. Regularization — Learning Apache Spark with Python documentation." https://runawayhorse001.github.io/LearningApacheSpark/reg.html (accessed Mar. 06, 2021).

[5] "https://scikit-learn.org/stable/ "

[6] R. Khandelwal, "Time series prediction using Prophet in Python," Medium, Nov. 17, 2019. https://towardsdatascience.com/time-series-prediction-using-prophet-in-python-35d65f626236 (accessed Mar. 06, 2021).

[7] "Time Series Analysis of the Covid-19 Datasets - IEEE Conference Publication." https://ieeexplore.ieee.org/document/9298390 (accessed Jan. 18, 2021).

[8] *A. Bansal, A. Bhardwaj, and A. Sharma, "Forecasting the Trend of Covid-19 Epidemic," in 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC), Nov. 2020, pp. 406–409, doi: 10.1109/PDGC50313.2020.9315795.*

[9] S. Bodapati, H. Bandarupally, and M. Trupthi, "COVID-19 Time Series Forecasting of Daily Cases, Deaths Caused and Recovered Cases using Long Short Term Memory Networks," in 2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Oct. 2020, pp. 525–530, doi: 10.1109/ICCCA49541.2020.9250863.

[10] S. Roy, M. N. Pal, S. Bhattacharyya, and S. Lahiri, "Implementation of an Informative Website – 'Covid19 Predictor', Highlighting COVID-19 Pandemic Situation in India," in 2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Sep. 2020, pp. 1–6, doi: 10.1109/IEMTRONICS51293.2020.9216352.

[11] S. Sengupta and S. Mugde, "Covid-19 Pandemic Data Analysis and Forecasting using Machine Learning Algorithms," Public and Global Health, preprint, Jun. 2020. doi: 10.1101/2020.06.25.20140004.

[12] N. Adithyan, "COVID-19 Analysis With Python," Medium, Jan. 06, 2021. https://medium.com/codex/covid-19-analysis-with-python-b898181ea627 (accessed Feb. 07, 2021).

[13] "http://spark.apache.org/docs/latest/"

[14] https://www.aptech.com/blog/introduction-to-the-fundamentals-of-time-series-data-and-analysis/

[15] https://ieeexplore.ieee.org/document/9363824

[16] http://www.sthda.com/english/articles/35-statistical-machine-learning-essentials/140-bagging-and-random-forest-essentials

[17] https://runawayhorse001.github.io/LearningApacheSpark/reg.html

[18] https://medium.com/plotly/introducing-plotly-express-808df010143d

[19] https://dwgeek.com/apache-spark-architecture-design-and-overview.html/

[20] http://media.sundog-soft.com/Udemy/DataScienceSlides.pdf

[21] https://scikit-learn.org/stable/

[22] https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/

[23] https://towardsdatascience.com/bayesian-linear-regression-project-in-python-forecast-water-consumption-under-the-impact-of-cea62c2693e4

[24] https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c

[25] https://www.analyticsvidhya.com/blog/2019/10/gaussian-mixture-models-clustering/