Understanding the Impact: A COVID-19 Data Analysis

Ana Fonseca
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up202002019@edu.fe.up.pt

Gonçalo Rocha
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up201707455@edu.med.up.pt

Mohamad Sis
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up202300529@edu.fe.up.pt

Pooja Muneswarappa
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up202308623@edu.fe.up.pt

Sara Romão
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up199402624@edu.fe.up.pt

Wasif Mushtaq
M. Data Science and Engineering
Faculty of Engineering, UP
Porto, Portugal
Email: up202300682@edu.fe.up.pt

***Abstract*—In this project we mainly focus on PySpark's MLlib and SQL capabilities to analyse a comprehensive COVID-19 dataset[1].We begin with pre-processing of our dataset using PySpark, data exploration and handling missing values. Then by employing SQL queries, we extract relevant subsets of data to delve into various dimension's to get insights of COVID-19 pandemic, such as geographical spread with total cases and deaths, vaccinations, reproduction rate, excess mortality etc. Utilizing PySpark's MLlib, we employ various machine learning algorithms to perform predictive analytics, including prediction of deaths, new cases and total vaccinations. Lastly the results are analysed and its performances are compared with different number of nodes on google dataproc.**

I. Introduction

Covid 19 was a worldwide pandemic that impacted the societies way of living. Therefore the analysis and interpretation of relevant data became relevant for decision making of public policies, private sector path and above all peoples lives. The goal of this project is twofold: FIRST: This work aim to extract valuable insights from a large COVID-19 dataset through a combination of pyspark SQL queries and Machine Learning techniques. Precisely, we seek to uncover patterns, trends, and correlations related to various aspects of the pandemic, including new cases, mortality rates, hospitalizations, testing efforts, and vaccination progresso as stated previously. SECOND: This work intents to analyze the runtime performance and scalability of our data analysis tasks. This involves measuring the execution time and resource utilization of different operations, such as data preprocessing, exploratory data analysis, statistical analysis, and machine learning model training, while varying the number of computational nodes. By evaluating the performance of our tasks under different configurations, we can assess the efficiency and scalability of our approach in handling large-scale COVID-19 data analysis tasks. For that, Colab. from Google and DataProc as well from the same corporation was used as tools in our work.

II. Related work

To rigorously examine the related work, our study involves a comprehensive analysis of three systematic reviews. Each review provides critical insights into distinct aspects of machine learning applications in COVID-19 research, which integral to understanding the current landscape and identifying gaps in the application of machine learning in the pandemic. **Saleem's [2]***"Machine Learning, Deep Learning, and Mathematical Models to Analyse Forecasting and Epidemiology of COVID-19: A Systematic Literature Review"* is a comprehensive review on the General Utilization of Machine Learning in COVID-19 Datasets that provides an extensive analysis of varied methodologies and outcomes across different COVID-19 datasets. This review evaluates the effectiveness, challenges, and opportunities presented by machine learning techniques within an epidemiological framework, documenting significant findings across 57 studies. It highlights the predominance of applications like Automatic Detection and the quantification of COVID-19 cases, noting the frequent use of CNNs (Deep Learning) and SVMs (Machine Learning) algorithms. In the realm of Machine Learning for Forecasting New COVID-19 Cases, the review by **Palermo**[3] called "*Tracking machine learning models for pandemic scenarios: a systematic review of machine learning models that predict local and global evolution of pandemics*" synthesizes the findings from 45 studies to assess the predictive capabilities of machine learning models. This review meticulously evaluates the accuracy of these models and the variables they incorporate, emphasizing their applicability in public health planning and response. Palermo concludes that models such as Linear Regression, Long Short-Term Memory Networks, Logistic Models, and Ensemble Methods demonstrate superior performance, underlined by robust certainty measures. While Saleem's comprehensive analysis sheds light on a variety of ML techniques, it lacks a deep dive into the specific challenges of data heterogeneity in global datasets, a gap that Palermo begins to address through his examination of local vs. global model effectiveness. The results corroborate the possibility of using ML prediction to serve the healthcare authorities for decision-making and preventive actions toward saving lives. **Aslam's[5] "***Analysis of COVID-19 Death Cases Using Machine Learning***"** focuses on the application of various regression machine learning models to determine the relationship between

diverse factors and COVID-19 mortality rates, with a specific focus on "New deaths per million" as a target metric. Utilizing a dataset comprising 165,870 observations with 67 attributes across several countries, this analysis employs models such as XGBoost, Random Forest, and SVM tailored to datasets from the US, India, Italy, and continents like Asia, Europe, and North America. Notably, the Random Forest model exhibited the best performance, achieving an R-squared value of 0.86 in the North American dataset. The results show that the models can be used to forecast the death cases for the near future in case of an epidemic like Coronavirus. These systematic reviews collectively provide a critical academic foundation for advancing machine learning applications in tracking, predicting, and managing the COVID-19 pandemic effectively.

III. Dataset Preprocessing

The dataset used for this project is COVID-19, which is sourced from OWID (Our World In Data) website[4]. The total number of records in this dataset are 390606, with each record corresponding to 67 features(columns).The schema of our dataset is shown in fig 1.

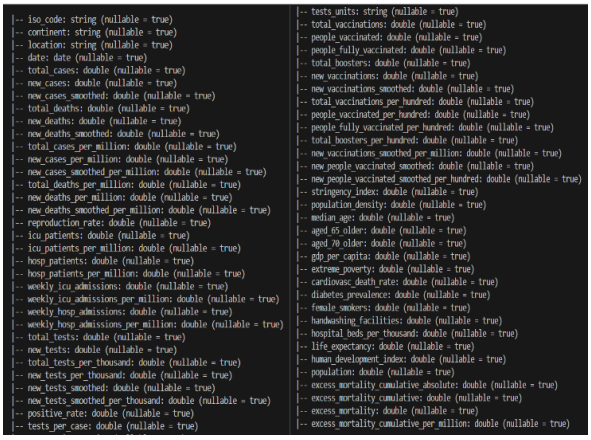


Fig. 1: Complete Schema of our dataset

Our dataset has maximum numeric columns. There are 62 double columns and five non-numeric columns, in that four are string columns and a date field. We decided to replace all nulls with zero's in all 62 numeric columns and in remaining five non-numeric columns, we found nulls in continent and tests_-units columns only other non-numeric columns did not had null values.

We observed that all null continent rows had data related to OWID like e.g, OWID_AFR in ISO_code column for null in continent, which shows that these rows were OWID related data. OWID is the website from where we took our dataset they might have inserted rows for some analysis purpose. So we decided to replace OWID with all nulls in continent column .As we did not had any information about nulls in tests_units column, we decided to replace these nulls with no_info. Now our final dataset is ready for processing.

The data exploration of our dataset is done in descriptive analysis and with matrix of scatter plots. which is shown in fig 2 and 3 only on some features.

	0	1	2	3	4
summary	count	mean	stddev	min	max
iso_code	390606	None	None	ABW	ZWE
continent	371938	None	None	Africa	South America
location	390606	None	None	Afghanistan	Zimbabwe
total_cases	351749	7445786.1887482265	4.4252547714527797	1.0	7.75251765E8
...	...	...	...	...	...
population	390606	1.2898179467722462E8	6.624013283877282E8	47.0	7.975105024E9
excess_mortality_cumulative_absolute	13199	55758.781738847287	155884.42084739966	-37726.098	1345330.4
excess_mortality_cumulative	13199	9.770028032426692	12.017298067707678	-44.23	78.08
excess_mortality	13199	10.972189559815135	24.576696580592294	-95.92	377.83
excess_mortality_cumulative_per_million	13199	1776.5086368395541	1991.9984622009647	-2936.4531	10293.515

Fig. 2: Descriptive Analysis

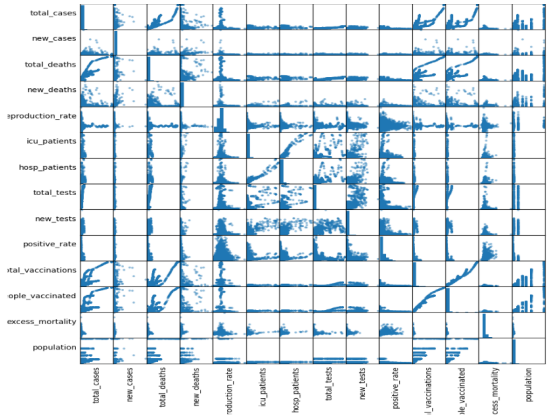


Fig. 3: Scatter Plot Matrix

IV. Query and Learning Tasks on the Dataset

In this section we will be using various pyspark queries and machine learning algorithms that can provide valuable insights to analyse our dataset.

A. Task description

1) Querying : The 'pyspark.sql' is a module in pyspark that is used to perform SQL like operations and 'pyspark.sql.functions' has list of standard built-in functions. In this section we will be using these functions and perform querying operations on our dataset.
   - Hospitalized Patients and ICU Admissions : Here we will visualize how many patients are hospitalized globally and in that how many are ICU admitted. For this as a sample we have choose a set of date for three months in 2020 (2020-05-28 to 2020-08-28). The small code snippet is shown in fig 4.

```
# Create new dataframe with record only between 2020-05-28 and 2020-08-28
dates = ("2020-05-28", "2020-08-28")
covid_dfd = covid_dff.where(F.col('date').between(*dates))

# Create a list with only dates
dates_frame = covid_dfd.select("date").distinct().orderBy('date').collect()
dates_list = [str(dates_frame[x][0]) for x in range(len(dates_frame))]
print(dates_list)

covid_dt = covid_dfd.orderBy("date", ascending=True).groupBy("date")

# Create a list with sum of hosp_patients - with normal beds
hosps = covid_dt.agg(F.sum("hosp_patients")).collect()
hosps = [hosps[i][1] for i in range(len(hosps))]
print(hosps)

# Create a list with sum of icu patients
icus = covid_dt.agg(F.sum("icu_patients")).collect()
icus = [icus[i][1] for i in range(len(icus))]
print(icus)
```

Fig. 4: Code snippet for hospitalized and ICU patients

- Total percentage of population has taken vaccination and not : Here we will visualize how many of total percentage of population in a country are vaccinated and what is the percentage of population are yet to get vaccination.As a sample we have shown for USA and Portugal countries. The small snippet code is shown in fig 5.

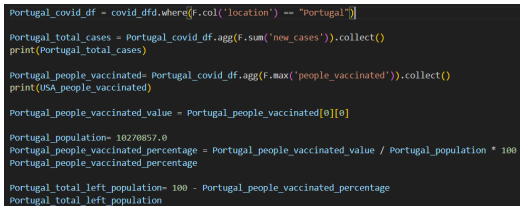


Fig. 5: Code snippet for vaccination percentage

- Reproduction Rate with respect to continent : Here we will visualize how is the growth of reproduction rate across the continent between 2020-07-03 to 2022-08-05 dates as sample. we have excluded OWID from continent which we replaced in place of nulls in our preprocessing steps.
- Excess Mortality reasons : Here we have tried to see what might be the reason for excess mortality. The excess mortality cumulative absolute value is difference between the reported number of deaths since 1 January 2020 and the projected number of deaths for the same period based on previous years. We tried to see whether the habits like smoking or disease like diabetes, cardiovascular might have there impact on excess mortality, we took there mean values and tried to compare its value with high mortality countries smokers and these disease values and we came to an conclusion that at least one of these are responsible for excess mortality as there values were more than mean values.The small snippet code is shown fig 6.

Fig. 6: Code snippet for excess mortality reasons

- 20 cases showing average CFR : Presents the medical indicator “Case Fatality Rate” , that measures the proportion of people who have been diagnosed with a certain disease and end up dying of it (disease lethally).The code contemplates the 20 cases taking country, continent, and date separated by month and year for a better division. Note: We considered “Total Cases” superior to 0 and “Total Deaths” not null.As showed on fig 7.

Fig. 7: 20 cases showing average CFR

- Top 20 Lowest Test Vaccination Metrics : The following query provides insights into the testing and vaccination efforts against COVID-19 across different continents over time, helping to assess the effectiveness of these measures and identify areas of improvement. For that is presented Total Tests Conducted (total_tests), Total Cases Reported (total_cases), the Average Positivity rate (avg_positive_rate), the Total Vaccinations Administered (total_vaccinations), the Average Vaccinations per Hundred People (avg_vaccinations_per_hundred). As well, the Test Positivity (test_positivity_status), that categorizes the average positivity rate into ”Low Positivity”, ”Medium Positivity”, or ”High Positivity” based on certain thresholds. By last, Vaccination Status (vaccination_status) Categorizes the average vaccinations per hundred people into ”Low Vaccination”, ”Medium Vaccination”, or ”High Vaccination” based on certain thresholds as well. Average Vaccinations per Hundred: The average number of COVID-19 vaccine doses administered per hundred people in the population during the given month and year. It provides a measure of vaccination coverage.As showed on fig 8.

Fig. 8: Top 20 Lowest Test Vaccination Metrics

- 20 cases for Mortality Metrics : Hospitalizations,ICU patients, Deaths/million:The following query takes 2 points within: The Total Excess of Mortality (total_excess_mortality), as the sum of excess mortality reported for each month. A well, the Average of Mortality Excess per million of the country’ population (avg_excess_mortality_per_million). These provide us insights into the overall mortality impact and the mortality rate per million people attributed to COVID-19 or related factors. For Hospitalization Metrics Calculation: Similar to mortality metrics, the code groups the data by year and month. It calculates two metrics: Average ICU Patients (avg_icu_patients), as the average number of patients in intensive care units (ICUs) reported for each month. Average Hospital Patients (avg_hosp_patients), as the average number of hospitalized patients reported for each month. These metrics give a perspective into the bur-

den on healthcare systems of countries, particularly in terms of ICU occupancy and general hospitalization rates. As showed on fig 9.

```
# Calculate mortality metrics
mortality_metrics = covid_dff.groupBy(f.year("date").alias("year"), F.month("date").alias("month")) \
    .agg(f.sum("excess_mortality").alias("total_excess_mortality"),
         F.avg("excess_mortality_cumulative_per_million").alias("avg_excess_mortality_per_million"))

# Calculate hospitalization metrics
hospitalization_metrics = covid_dff.groupBy(f.year("date").alias("year"), F.month("date").alias("month")) \
    .agg(f.avg("icu_patients").alias("avg_icu_patients"),
         F.avg("hosp_patients").alias("avg_hosp_patients"))

# Join mortality_metrics and hospitalization_metrics
combined_metrics = mortality_metrics.join(hospitalization_metrics,
                                          ["year", "month"],
                                          "inner") \
                                    .orderBy("year", "month")

# Show the result
combined_metrics.show()
```

Fig. 9: 20 cases for Mortality Metrics: Hospitalizations,ICU patients, Deaths/million

2) Learning : The pyspark MLlib supports several popular machine learning algorithms for classification, regression, clustering, and for collaborative filtering. Below are few algorithms which we have used to predict the number of new cases, mortality and vaccination rate.
   - Linear Regression : Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation to observed data. It assumes a linear relationship between the input variables and the output, making it simple and interpretable.
   - Linear Regression Bayesian: Bayesian Linear Regression is an extension of traditional linear regression that applies Bayesian inference techniques to estimate the parameters of the linear model. It allows for the incorporation of prior knowledge or beliefs about the parameters, providing a framework for uncertainty quantification.
   - Decision Trees Regression : Decision Trees Regression is a non-parametric supervised learning method used for regression tasks. It works by recursively partitioning the input space into regions and fitting a simple model (e.g., constant value) in each region. Decision trees are easy to interpret and can handle non-linear relationships.
   - Random Forest: Random Forest is an ensemble learning method that builds multiple decision trees during training and outputs the average prediction of individual trees for regression tasks. It improves predictive accuracy and reduces over-fitting by combining predictions from multiple trees.
   - Gradient Boosting: Gradient Boosting is an ensemble learning technique that builds a series of weak learners (typically decision trees) sequentially, where each new model corrects the errors made by the previous one. It aims to minimize a loss function by iteratively adding new models to the ensemble.

B. Results

1) Querying : The results of the above querying task descriptions are shown in this section.
   - Hospitalized Patients and ICU Admissions result is shown in fig 10.

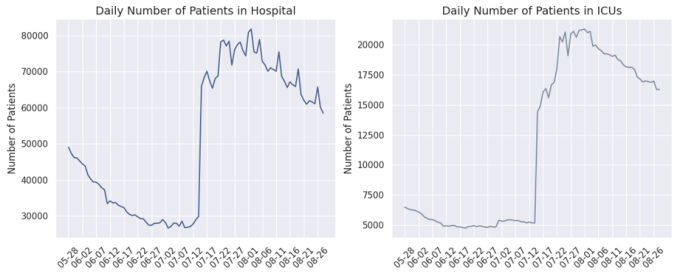


Fig. 10: Result for total patients hospitalized and in ICUs

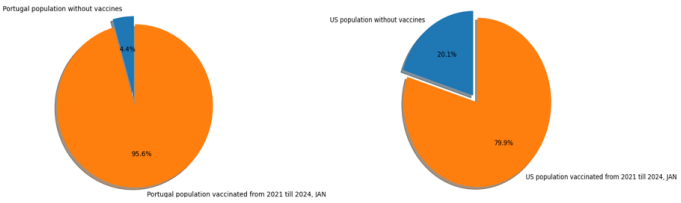


Fig. 11: Result for percentage of total vaccination

   - Total percentage of population has taken vaccination and not result is shown in fig 11.
   - Reproduction Rate with respect to continent result is shown in fig 12.

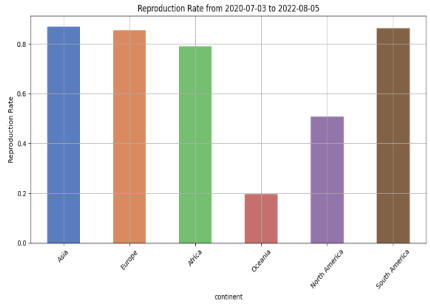


Fig. 12: Result of Reproduction Rate with respect to continent

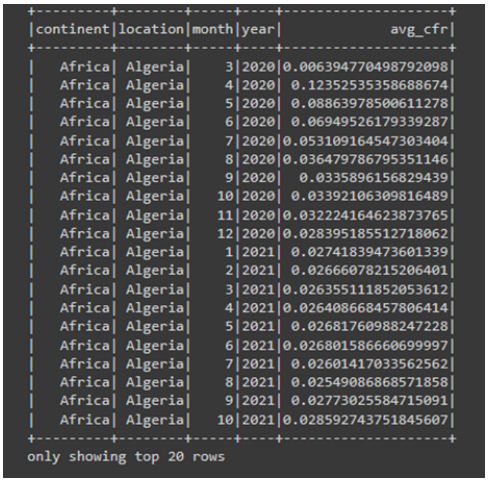
   - Excess Mortality reasons result is shown in fig 13.

Based on data up to 2021-02-20, the mean of female smokers is 10.79.
The mean of male smokers is 32.91.
The mean of people suffering from diabetes (aged 20-79) is 8.56.
The mean number of deaths per 100,000 people due to cardiovascular conditions is 264.27.

Location	excess_mortality_cumulative_per_million	female_smokers	male_smokers	diabetes_prevalence	cardiovasc_death_rate
Armenia	3842.74851	1.51	52.11	7.111	341.011
Mexico	3673.11551	6.91	21.41	13.061	152.7831
Belarus	3241.5371	10.51	46.11	5.181	443.1291
Russia	3174.9961	23.41	58.31	6.181	431.2971
Albania	2872.7261	7.11	51.21	10.061	304.1951

Fig. 13: Result of Excess Mortality reasons

   - 20 cases for Mortality Metrics:Explanation: In April 2020, the average CFR increased to approx. 12.35 p.p., indicating a higher proportion of deaths among COVID-19 in Algeria in April.
   - Top 20 Lowest Test Vaccination Metrics: Explanation: In Africa, in 01.2020, no reported cases, no tests made, no vaccines were administrated. Likewise the Positive tests out of total Tests was 0 p.p., so no positive cases. Since no vaccination and testing, the positive_status and vaccination_status is ”low positivity” , ”low vaccination” respectively.As show on figure 15:



continent	location	month	year	avg_cfr
Africa	Algeria	3	2020	0.006394770498792098
Africa	Algeria	4	2020	0.12352535358688674
Africa	Algeria	5	2020	0.08863978500611278
Africa	Algeria	6	2020	0.06949526179339287
Africa	Algeria	7	2020	0.053109164547303404
Africa	Algeria	8	2020	0.036479786795351146
Africa	Algeria	9	2020	0.0335896156829439
Africa	Algeria	10	2020	0.03392106309816489
Africa	Algeria	11	2020	0.032224164623873765
Africa	Algeria	12	2020	0.028395185512718062
Africa	Algeria	1	2021	0.02741839473601339
Africa	Algeria	2	2021	0.02666078215206401
Africa	Algeria	3	2021	0.026355111852053612
Africa	Algeria	4	2021	0.026408668457806414
Africa	Algeria	5	2021	0.02681760988247228
Africa	Algeria	6	2021	0.026801586660699997
Africa	Algeria	7	2021	0.02601417033562562
Africa	Algeria	8	2021	0.02549086868571858
Africa	Algeria	9	2021	0.02773025584715091
Africa	Algeria	10	2021	0.028592743751845607

only showing top 20 rows

Fig. 14: Results of the 20 cases for Mortality Metrics

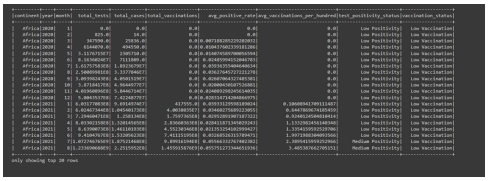
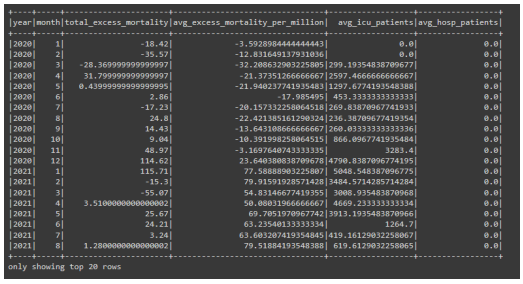


Fig. 15: Explanation TOP lowest vaccination and testing

- 20 cases for Mortality Metrics: Hospitalizations,ICU patients, Deaths/million. Explanation:: For the monthly breakdown of excess mortality data for the year 2020. It indicates that in January, there was a negative excess mortality of 18.42, implying a mortality rate lower than expected, and an average of -3.59 excess deaths per million population, with no reported ICU or hospital patients.



year	month	total_excess_mortality	avg_excess_mortality_per_million	avg_icu_patients	avg_hosp_patients
2020	1	-18.42	-3.5928984444444443	0.0	0.0
2020	2	-35.57	-12.831649137931036	0.0	0.0
2020	3	-28.369999999999997	-32.208629030225805	299.1935483870967	0.0
2020	4	31.799999999999997	-21.373512666666667	2597.4666666666667	0.0
2020	5	0.43999999999999995	-21.940237741935483	1297.6774193548388	0.0
2020	6	2.86	-17.985495	453.3333333333333	0.0
2020	7	-17.23	-20.157332258064518	269.8387096774193	0.0
2020	8	24.8	-22.421305161290324	236.3870967741935	0.0
2020	9	14.43	-13.643108666666667	260.03333333333336	0.0
2020	10	9.04	-10.391998258064515	866.0967741935484	0.0
2020	11	48.97	-3.1697640743333335	3283.4	0.0
2020	12	114.62	23.640308387096778	4790.838709677419	0.0
2021	1	115.71	77.58888903225807	5048.548387096775	0.0
2021	2	-15.3	79.91591928571428	3484.5714285714284	0.0
2021	3	-55.07	54.83146677419355	3008.935483870968	0.0
2021	4	3.5100000000000002	50.80031966666667	4669.233333333334	0.0
2021	5	25.67	69.70519709677742	3913.1935483870966	0.0
2021	6	24.21	63.23540133333334	1264.7	0.0
2021	7	3.24	63.603207419354845	419.16129032258067	0.0
2021	8	1.2800000000000002	79.51884193548388	619.6129032258065	0.0

only showing top 20 rows

Fig. 16: Explanation 20 cases for Mortality excess Germany

2) Learning : The results of MLlib algorithms are shown in this section.
   a) Result Comparison: Mortality
      - Linear Regression has the highest testing R-squared value, followed closely by Bayesian Linear Regression.
      - Random Forest performs slightly better than Decision Trees Regression in terms of testing R-squared.
      - Decision Trees Regression also performs well, but its testing R-squared value is slightly lower compared to the other models.

Fig. 17: Result of R2 for mortality prediction

b) Result Comparison: New Cases
   - Linear Regression shows an extremely high testing R-squared value, indicating a very close fit of the model to the testing data.
   - Random Forest exhibits a relatively high RMSE on test data and a moderate testing R-squared value. It seems to capture less variance in the testing data compared to Linear Regression.
   - Decision Trees Regression performs better than Random Forest with a higher testing R-squared value, suggesting that a single decision tree may generalize better to the testing data in this case.
   - Gradient-Boosted Trees with maxIter = 5 also show a decent testing R-squared value but lower than Decision Trees Regression.

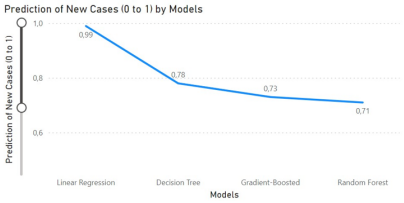


Fig. 18: Result of R2 for new_cases prediction

c) Result Comparison: Total Vaccination
   - Linear Regression seems to be the best-performing model, followed by Decision Trees Regression.
   - Random Forest performs relatively worse compared to the other models, with a lower testing R-squared value.

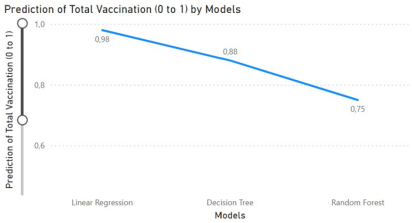


Fig. 19: Result of R2 for total vaccination prediction

V. Runtime and Scalability Analysis

Google Cloud Dataproc is a fully managed and highly scalable service for running Apache Hadoop, Apache Spark, Apache Kafka etc and Dataproc is a managed Spark and Hadoop service which allows us to take advantage of open source data tools for batch processing,

querying, streaming, and machine learning. We have ran our code on this Dataproc and calculated the run time performance on different nodes.

A. Methodology

- We chose appropriate machine learning models for predicting COVID-19 outcomes, considering the nature of the data and the task at hand
- Initially we implemented the selected machine learning models on N2 series with 2 worker nodes in the distributed computing environment. We also distributed the training data across the N2 nodes to leverage parallel processing capabilities for faster model training.
- Evaluation of the trained models was acquired by using appropriate evaluation metrics such as mean squared error, mean absolute error, or accuracy, depending on the prediction task..
- Once the models were trained and optimized on N2 Series with 2 worker nodes, we compared their performance on the N1 Series with 3 worker nodes as well. Distributed the validation dataset across the N1 nodes and assess the models' predictive accuracy and generalization ability in comparison to N2.

B. Results

Resource Utilization: We monitored resource utilization on both sets of worker nodes during training and analyze if there are any differences in resource utilization between n2 with 2 worker nodes and n1 with 3 worker nodes nodes and how it impacts training performance.

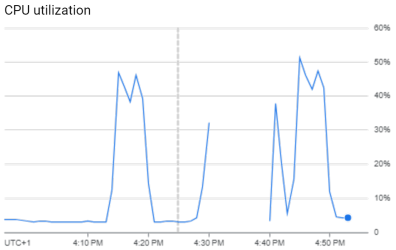


Fig. 20: CPU utilization in N2 with 2 worker nodes

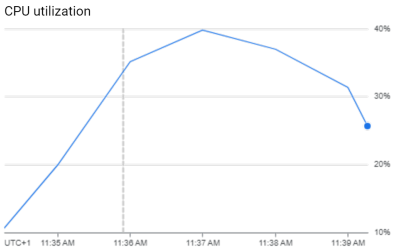


Fig. 21: CPU utilization in N1 with 3 worker nodes

VI. Conclusion

In the present work, we aimed to develop a deep analysis and predictive framework throughout COVID-19 data. The work explored critical aspects of the COVID-19 pandemic, including new case magnitudes, mortality rates, and vaccination efforts. Leveraging PySpark's powerful analysis techniques, we uncovered key patterns, trends, and metrics across various countries and continents. Our approach involved comprehensive data preprocessing and descriptive statistics to understand the underlying data characteristics of the respective dataset. We further accelerated our analysis tasks through parallel processing and distributed computing techniques, utilizing tools like Google Cloud Dataproc. By harnessing SQL queries, we efficiently extracted relevant information to uncover significant patterns within the data. Utilizing Colab. notebooks and Google Cloud Dataproc, we applied machine learning models to forecast COVID-19 trends and compared their performance with varying numbers of computing nodes. Nevertheless, there is still opportunity for further exploration and analysis! The COVID-19 dataset is continually updated (we took data until the day the dataset was presented to start the analysis) and offers a wealth of information that can be leveraged for deeper insights. Future work could extend the analysis to include additional aspects of the pandemic, such as government intervention policies, virus strain variations, and the impact of increased awareness of preventive measures. Moreover, exploring different cluster configurations and machine types in Google Cloud Dataproc could yield valuable insights into optimizing performance and scalability. Additionally, expanding the number of computing nodes and testing more machine learning models would contribute to ongoing evaluation of performance, runtime, and predictive accuracy. Overall, our study provides a solid foundation for continued exploration and analysis of COVID-19 data to inform decision-making and resource allocation in combating the pandemic.

References

[1] https://github.com/owid/covid-19-data/blob/master/public/data/owid-covid-data.csv.

[2] Saleem F, Al-Ghamdi ASA, Alassafi MO, AlGhamdi SA. Machine Learning, Deep Learning, and Mathematical Models to Analyze Forecasting and Epidemiology of COVID-19: A Systematic Literature Review. Int J Environ Res Public Health. 2022 Apr 22;19(9):5099. doi: 10.3390/ijerph19095099. PMID: 35564493; PMCID: PMC9099605.

[3] Palermo MB, Policarpo LM, Costa CAD, Righi RDR. Tracking machine learning models for pandemic scenarios: a systematic review of machine learning models that predict local and global evolution of pandemics. Netw Model Anal Health Inform Bioinform. 2022;11(1):40. doi: 10.1007/s13721-022-00384-0. Epub 2022 Oct 11. PMID: 36249862; PMCID: PMC9553296.

[4] https://ourworldindata.org/coronavirus.

[5] Hina Aslam and S. Biswas. Analysis of covid-19 death cases using machine learning. *SN Computer Science*, 4:403, 2023. Accessed: 2024-05-07.