**Abstract**

The source of data used in this study comprised of different open sources that contain information about COVID-19 all over the globe. These are daily changes in infection rates, number of people recovering from the disease, and death tolls in various districts and nations. Some major characteristics of the data include date-specific data records which permit a real-time analysis of the progression of the pandemic. The main goal of analyzing this dataset is exploratory data analysis and visualization to find insights and patterns of the virus spread. In this project, it is planned to use statistical and machine learning approaches to forecast future trajectories of COVID19 and assess the impact of quarantine measures by employing linear regression or time series analysis.

**Introduction**

The data set used in this report was obtained from the records of daily COVID-19 cases across the globe that are publicly available. It incorporates statistics on number of confirmed cases, new and active patients, recovery and death rate per region. The first goal is to predict deaths and evaluate recovery rates on using machine learning methods on the collected data. The algorithms used for predictions are for regression and classification and they include Random Forest, support vector machines for regression, and additional statistical measures include F1 scores and R-squared. It gives knowledge about the pandemic situation and supports the prediction of future traits to make policies and strategies.

**Data Preparation**

Firstly, data preparation involves data collection, and in this case, we collected data using Jupyter notebook.

1. Load Data and Preliminary Preprocessing:

The data is read in the form of a CSV file and then imported to the Pandas Dataframe. The first step, which will be completed in this step, is to run the first several rows to get an idea of the structure and scale of the data and to make sure that all the columns of interest are imported correctly.

1. Cleaning and Validation:

It is important to check for missing values in this rather large dataset: the notebook contains one example of a missing value check. How it can be done depends to the extent and, necessity of them missing. Perhaps assign or best of all, simply leave out.

Data Types: Some caution is needed to ensure that ‘Date’ would be formatted to datetime because the research is based on time series: one of the crucial preliminary stages that, if omitted, makes further temporal-based analysis impossible.

1. Feature Engineering: Hypothetical or Suggested

Since analyzing this dataset, if not done by others, it would be useful to perform a form of data manipulation where daily rates fresh cases, fatalities, and recovered cases are derived from the raw cumulative figures offered. This involves the analysis of differences fabricated between two successive days in order to understand fluctuations within a day.

Other features could be the calculation or verification of ‘Active’ cases for instance in case there is an indication of irregularities in some set data they are produced.

1. Data Exploration:

Basic statistical descriptions, such as data.describe(), are done to understand the numerical features: positive cases, mortality and number of people who have regained their health. This will give an impression of outliers, other forms of statistical patterns and an overall picture of the data and its dispersion.

Some of the initial visualizations may be based on Matplotlib, and Seaborn tool which plots distributions or trends characteristic of the dataset features, for example, geographic distribution of COVID-19 thereby helping in identifying the regions affected.

1. Advanced Data Visualization:

Data can sometimes be depicted dynamically on a map. The notebooks can have some complicated geospatial visualizations created by libraries like Basemap or the alternative GeoPandas. These are especially relevant for datasets having geographical coordinates and regions such as those defined by WHO for their regions.

1. Preparing Data for Modeling (If Possible):

Most probably, it would then have reached the stage of modeling whereby the data would normally have been split into training and test sets to assess the predictive accuracy of the models. This process confirms that the model is built upon part of the data and applied with the remaining unseen part to analyze how predictive it will be and generalizing conditions.   
Dataset consists of following columns:  
Province/State: Sub-national division to give details of the location of the units under consideration.

Country/Region: Country or the continent of the data.

Latitude, Longitude: Geographical coordinates of the region.

Date: The specific date on which the data was reported was also recorded.

Confirmed: Total cases of COVID-19 infections and diseases.

Deaths: Total Number of deaths by COVID-19 each day.

Recovered: Total new cases of COVID-19 patient who have received medical attention and got out of danger.

Active: Number of actually active COVID-19 cases.

WHO Region: World Health Organization regional classifications that enables one understand the regional response approach and the severity of the outbreak.

Summary:

The analyses above show that the Jupyter notebook deals with loading, cleaning, basic explorations, and possibly preparatory steps for more advanced analysis or modeling. These steps are very critical in ensuring that the dataset will stand strong against a more detailed, reliable analysis whether for trend understanding or future predictions in the particular context of COVID-19.

**Exploratory Data Analysis**

1. Data Inspection and Descriptive Statistics:

The notebook starts the EDA by using data.head() that gives some information on the basic structure of the dataset; the types of the data columns included and their names such as ‘Confirmed’, ‘Deaths’, ‘Recovered’, ‘Active’ and ‘WHO Region’.

The data.describe() function is used for descriptive Summary statistics where they use, count, mean, standard deviation, minimum and maximum and quartiles for numeric variables.

A screenshot of a computer

Description automatically generated  
A table with numbers and symbols

Description automatically generated

1. Missing Value Analysis:

Using data.isnull().sum(), an analysis to determine if there are any missing values in certain columns is done. This assist in determining completeness of data set.

1. Visualization Techniques Used in the Notebook:
2. Global Scatter Plot on Basic Map:

It also has a scatter plot of cases where latitude and longitude data were used to present the COVID-19 spread across the globe. Bigger dots on the map reflect each point for the region that has a higher number of confirmed cases. This visualization plays a crucial role in making an understanding of the virus distribution area.

A graph of a number of covid-19 cases

Description automatically generated

1. Scatter Plots for Outlier Detection:

Scatter plots are used to identify outliers in the confirmed cases. This visualization helps detect regions with unusually high or low case counts compared to the general distribution.

A map of the world with blue dots

Description automatically generated

1. Histograms for Case Distribution:

In the case of COVID-19, histograms are used to represent the COVID-19 cases distribution data with a view of appreciated how cases are spread or could be densely populated.

A graph of a number of people in different colors

Description automatically generated

1. Time Series Plot of Cases Over Time:

A time series plot describes the COVID-19 cases, where daily or cumulative cases can be represented, to determine the existence of trends, rises or falls, necessary when considering the evolution of the pandemic or the efficiency of measures taken.

A graph showing the number of cases

Description automatically generated

1. Correlation:

Visualization for the correlation of the dataset is produced to show the underlying patterns of variables like Confirmed and Deaths cases are related. This can be used also in model feature selection and hypothesis generation regarding the existence of causation between attributes.  
A graph with a red line

Description automatically generated

Summary:

The EDA in the Jupyter notebook is systematic including statistical tests alongside various visualizations that both complement each other and leave no stone unturned with respect to the COVID-19 dataset. Every action is deliberate in these steps and visualization to help in analyzing the data and preparing for further analysis or modeling. By integrating data inspection, cleaning advice, and initial visualization, the notebook provides an all-encompassing starting point for a wide variety of analytical queries concerning the COVID-19 crisis.

**Extract-Load-Transform**

Step 1: Data Extraction:

In earlier sections of the notebook, the data was loaded from a CSV file into a Pandas DataFrame as the initial input tables (as discussed in Part 1 of this notebook).

Step 2: Data Transformation:

Date Conversion: For analysis, the ’Date’ column is then converted to datetime format using the code; pd.to\_datetime() to enable time series analysis Date is a date column.

Handling Missing Values: There are also some empty spaces in the ‘Province/State’ column which are replaced by “Unknown” by applying fillna(“Unknown”, inplace= True). This makes the data standardized as it equals the lost data points and allows no data to be omitted in the subsequent analysis due to missing state or province value.

Removing Duplicates: Cleaning the dataset is done on rows where data is duplicated to avoid the repetition of records, which are achieved by data.drop\_duplicates(inplace=True).

Calculating New Columns: New columns – ‘Case\_Fatality\_Rate’ and ‘Recovery\_Rate’ are added. CFR stands for case fatality rate and a statistical percentage that calculates the number of deaths due to a virus divided by the total number of confirmed cases of the same virus; The recovery rate or the chance of recovery is calculated by the number of persons recovered from the virus divided by the number of persons infected with the virus.

Filtering Data: Only the cases where ‘Confirmed’ is greater than zero is selected by filtering the dataset (data = data[data[‘Confirmed’> 0]). This brings the analysis to bear on areas affected by the virus and eliminates any other area that may not be affected and hence cannot be used for spread or impact analysis.

Step 3: Loading the Transformed Data:

Saving Transformed Data: Finally, after all transformations, the Covid-19 data is cleaned and saved to a new CSV file as shown below to\_csv(“transformed\_covid\_data.csv”, index=False). This operation saves the transformed data set for future use, so that, none of the preprocessing operations are to be performed again and again and also provides a safer data set for the analysis purpose or even for sharing.

Reloading and Reviewing Transformed Data: The dataset is then written back to the current working directory using to\_csv() and the file is loaded again back into the notebook using pd.read\_csv() and the head of the imported data is printed to ensure that the previous transformations were done right.

A screenshot of a computer

Description automatically generated

Conclusion:

This ETL process also proves to be useful in transforming COVID-19 dataset for the subsequent steps in analysis by cleaning and structuring the data properly. When missing values, duplicates, and other aberrations are handled, and new metrics are added, it is not only a more manageable dataset, but new columns derived from them will encourage further exploration and enable richer data analysis and presentation. It is hygienic to see such careful procedures to prepare the data to show how crucial ETL is to support the subsequent findings derived from analytics.

**Data Modeling and Analysis**

The model building section on the Jupyter notebook centres on two main models of COVID-19 related outcomes in the form of number of deaths given confirmed case numbers, recoveries and active cases. The selected algorithms are Random Forest Regressor algorithm and Support Vector Machine (SVM).

Model Selection:

Random Forest Regressor: Selected due to its suitability for processing non-linear data and for feature importance assessment tools, at the time of creating the stopped model. This ensemble model is rather accurate and space invariant, which is suitable for the datasets with interaction between variables.

Support Vector Machine (SVM): Chosen due to the presence of the kernel methods which are capable of modeling non-linear decision functions. Specifically, this paper adopted SVM for the high-dimensional space can be manage well by it.

Model Building Process:

Random Forest Model:

Feature Selection: ‘Confirmed’, ‘Recovered’, ‘Active’ features as a dependent variable point to the target variable ‘Deaths’.

Training and Testing: Cross validation is used so as to create training and test sets so that the model is correctly tested.

Model Training: A Random Forest Regressor is built with particular hyperparameters such as n\_estimators = 100, etc.

Performance Evaluation: The model’s accuracy is analyzed using Mean Squared Error while variance analysis is done using coefficients of determination the R².

Feature Importance Analysis: The significance of features is as calculated from the model, which showcases which of them is most useful for the prediction of the total mortality rate.

Support Vector Machine (SVM):

Data Standardization: StandardScaler is applied onto features in order to remove the scale dependency during the usage of the SVM.

Model Training: An SVM with a RBF Kernel is built, and the parameters C and epsilon are tuned for getting the best results.

Prediction and Evaluation: The SVM model predicts deaths on the test set after the training. Thus, the findings of the ACE model are measured by MSE and R-Squared to measure the performance of the model.

Binarization for Classification: This makes it easier to perform the evaluation in binary classification form by converting the continuous prediction made by the SVM to binary classes per a defined threshold to compute the F1 measure.

Model Performance Analysis:

1. Random Forest Regressor Performance:

A screenshot of a computer

Description automatically generated

Mean Squared Error (MSE): 557105.5875677752. This measure shows the average squared deviations of the observed actual results from the predicted by the model. The MSE will be smaller if the prediction of the model is accurate; however, it is important to note, that this coefficient is quite high, which means that there is significant variation between the predicted and real outcomes.

R² Score: 0.9901338545512457. This score shows the extent of the variation in the dependent variable (Deaths) that can be explained by the independent variables (Confirmed, Recovered, Active). Such an R² equals to 1 indicates that the model has indeed a large impact on explaining the data variance.

Feature Importances:

Active Cases: 0.494170 prematurely aged skill Most influential in predicting deaths.

Confirmed Cases: 0.452454 − Significant in the model as well, though not as highly as odometer reading.

Recovered Cases: 0.053376 – Least influential but not excluded from the models’ calculations.

Random Forest Regressor F1 Score: 0.9179793590493979. This score is particularly used to evaluate the performance of a model on a dataset and is based on precision and recall weights. An F1 score near the ideal score of 1 is a good sign of the model’s efficiency in classifying the labels of the class.

1. Support Vector Machine (SVM) Performance:

A computer screen shot of a number

Description automatically generated

Mean Squared Error (MSE): 5505613.12568545. The variance of the SVM is however, higher than that of the Random Forest, so this means the average squared difference between the predicted and actual values are higher in the case of SVM.

R² Score: 0.025072868154463224. Finally, the models R² score of 0.0005 for the SVM model shows that the model does not account for much of the variation in the dependent variable from the predictors, which in this sense equals poor performance of the model.

Feature Importances (Permutation Importance):

Confirmed Cases: 0.006465 – Associated with a relatively small effect in the model’s assessments.

Recovered Cases: 0.000604 – Similar low effect as reported cases.

Active Cases: 0.004289 - least effect on decision of model.

SVM F1 Score: 0.6901694915254237. This score means that the SVM model has roughly average accuracy and effectiveness of class label recognition and is higher than the previous model.

Conclusion:

The comparison between the Random Forest Regressor and the SVM highlights significant differences in performance:

The Random Forest Regressor indicates good predictive performance based on a high R² and a reasonably good F1; which points to its ability to account for the relations in the data.

The SVM as a classifier is generally resistant for classification problems, however in this regression type the code gives much poorer results with increased standard deviation, as well as much lower coefficient of determination (R²) and F1 score.

These performance metrics will give clear information of each model for the prediction of COVID-19 related deaths; this will depend with the data used in the entire process. The metrics of feature importance also provide the information regarding which input features matter most in the predictions of all models.

**Predicting COVID-19 Patient Outcomes Using Machine Learning: A Random Forest Approach**

Random Forest Classifier model for the COVID-19 cases prediction as the outcome from the confirmed as well as active cases. The goal is to determine whether a case will result in recovery or death, employing a binary classification where:

Class 0 (Death Dominant): Outbreaks where the daily toll of deaths is equal to or exceeds the number of reported recoveries.

Class 1 (Recovery Dominant): Some of the examples of conditions that have more recoveries than deaths.

Model Training and Evaluation:

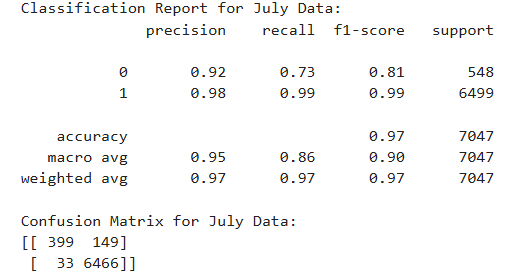
Training Data:

The model is then trained between January and June and selects characteristic features associated with the pandemic and its current activity; ‘Active’ and ‘Confirmed’ cases.

Target Definition:

A binary target is defined as follows; The values 1 and 0 are assigned to cases that result in more recoveries than deaths, and equivalently the values that are attributed to cases where deaths equal or surpass the recoveries.

Testing and Validation:

The model is tested with data from July to validate its predictive power and to ensure that the model generalizes well over time.  


Classification Metrics:

Precision: Measures the level of correctness in classification of positive predictions. The accuracy of the model for class 1 in predicting recovery-dominant cases is demonstrated by Class 1 having a precision of 0.98.

Recall: Measures how all positive samples are being located. Class 1 shows recall of 0.99, implying that good cases of recovery have been well identified.

F1-Score: An average of two estimated values of precision and recall in which the value of precision is weighed higher than that of recall. Class 1 attains an F1-score of 0.99, showing high accuracy in the recovery prediction; recall, Class 0 has an F1-score of 0.81, which is also reasonable for death-dominant prediction but improvable.

Confusion Matrix:

True Positives (Recoveries correctly predicted): 6,466

True Negatives (Deaths correctly predicted): 399

False Positives (Recoveries incorrectly predicted as deaths): 149

False Negatives (Deaths incorrectly predicted as recoveries): 33

Insights:

Having an over all accuracy of 97 %,the model clarity indicates that its performance is very good and it can effectively predict the outcome of COVID-19 cases from the given data. It is the most accurate at identifying cases where the recovery will occur, specifically for Class 1, our F1-score is close to 1. Class 0 performs as well but slightly lower, however it remains quite strong; this translates the model into a powerful tool for interpreting patients’ outcomes and, possibly, distribution of medical resources.

Suggestions for Improvement:

In order to improve the model’s performance and forecast the occurrence of death-dominant cases (Class 0), other characteristics, including patient attributes (age, presence of comorbidities), hospitalization rates or health care capabilities within the area of interest may be incorporated in the model. These could give further information on the risk factors for the deaths and make the model more sensitive in the future for the more severe events.

**References**

*Confusion matrix*. Confusion Matrix - AI Wiki. (n.d.).

<https://machine-learning.paperspace.com/wiki/confusion-matrix>

Keita, Z. (2022, September 21). *Classification in machine learning: A guide for beginners*. DataCamp. <https://www.datacamp.com/blog/classification-machine-learning>