

**COLLEGE CODE: 5113**

# APPLIED DATA SCIENCE

Project No.5- COVID -19 VACCINE ANALYSIS

BATCH MEMBERS:

1. K.HARISH-au511321104028- [harishkumar251603@gmail.com](mailto:harishkumar251603@gmail.com)

2.ASVINDHAN-au511321104008- [asvindhanelangovan@gmail.com](mailto:asvindhanelangovan@gmail.com)

3.HEMNATH AJAY-au511321104030- hemnathajay51@gmail.com

**INTRODUCTION:**

Analyzing COVID-19 vaccines is a critical component of the global response to the ongoing pandemic caused by the novel coronavirus, SARS-CoV-2. These vaccines have been developed at an unprecedented pace and are essential tools in controlling the spread of the virus and reducing the severity of the disease. Vaccine analysis involves a multifaceted approach that encompasses various aspects, including efficacy, safety, distribution, public acceptance, and their impact on the pandemic One of the primary aspects of analyzing COVID-19 vaccines is assessing their efficacy and effectiveness.

**ABOUT THE DATA:**

Where did we get the dataset?

Kaggle:

The dataset provided on Kaggle, <https://www.kaggle.com/datasets/gpreda/covid-world-vaccinationprogress>,

offers a valuable resource for our project aimed at forecasting covid-19 vaccine analysis.

**Dataset Details:**

The data (country vaccinations) contains the following information:

Country- this is the country for which the vaccination information is provided;

Country ISO Code - ISO code for the country;

Date - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;

Total number of vaccinations - this is the absolute number of total immunizations in the country;

Total number of people vaccinated - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;

Total number of people fully vaccinated - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;

***BEGINNING WITH THE PROJECT***

To begin building a project for air quality analysis and prediction, we first need to load the dataset.

We have a dataset file in a common format like CSV, here are the steps to load the dataset:

**1.Importing the required Libraries(data.csv):**

In this step, we import the necessary Python libraries and modules to work with our data and perform various data processing and machine learning tasks.

## import pandas as pd

## import numpy as np

## from sklearn.preprocessing import Imputer

## from sklearn.model\_selection import train\_test\_split

## from sklearn.preprocessing import StandardScaler

## from sklearn.preprocessing import OneHotEncoder

**2.Importing the data set(read data set; create matrix ):**

This step involves loading our dataset into memory. We use libraries like pandas to read data from a CSV file or other formats. After loading, we create a feature matrix (often denoted as X) and a target vector (often denoted as Y).

## dataset = pd.read\_csv('"C:\Users\haris\OneDrive\Documents\country\_wise\_latest.csv"')

## X = dataset.iloc[:, :-1].values

## Y = dataset.iloc[:, -1].values

**3.Handling the Missing Data.(sklearn.preprocessing library contains class called imputer, helps in missing data):**

Datasets often have missing values. The sklearn.preprocessing.Imputer class is used to address this issue. You can specify a strategy for imputing missing values, such as replacing them with the mean, median, or mode of the column.

## imputer = Imputer(missing\_values='NaN', strategy='mean', axis=0)

## imputer = imputer.fit(X[:, columns\_with\_missing\_data])

## X[:, columns\_with\_missing\_data] = imputer.transform(X[:, columns\_with\_missing\_data])

**4.Encoding Categorical Data.(one-hot encoding):**

One-hot encoding is a technique used to convert categorical data into a numerical format. Each category becomes a binary feature (0 or 1) in a new column, making it suitable for machine learning algorithms.

## Encode=OneHotEncode(categoricalfeatures=categoricalcolumn)

## X = encode.fit\_transform(X).toarray()

**5.Splitting the data set into test set and training set.( import train train\_test\_split)(X\_train,X\_test, Y\_train,Y\_test):**

Before building a machine learning model, it's essential to divide our dataset into two sets: a training set and a test set. The training set is used to train the model, while the test set is used to evaluate its performance.

## X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.2, random\_state=0)

**6.Feature Scaling.(import StandardScaler):**

Feature scaling ensures that all features have the same scale, typically with a mean of 0 and a standard deviation of 1.

## scaler = StandardScaler()

## X\_train = scaler.fit\_transform(X\_train)

## X\_test = scaler.transform(X\_test)

**PREPROCESSING THE DATASET:**

Preprocessing of data in a dataset refers to the various techniques and operations applied to the data before using it for analysis, modeling, Here's a more detailed explanation of data preprocessing within the context of a dataset:

**1. Data Cleaning:**

Handling Missing Values: Identify and deal with missing data, which may involve filling in missing values, removing rows with missing data, or using imputation techniques.

Dealing with Duplicates: Detect and remove duplicate records to ensure data integrity.

**2. Data Transformation:**

Feature Scaling: Normalize or standardize numerical features to bring them to a similar scale. This is important for algorithms sensitive to feature scales.

Feature Encoding: Convert categorical variables into a numerical format using techniques like one-hot encoding or label encoding.

Feature Engineering: Create new features or modify existing ones to capture relevant information and patterns in the data.

Binning: Group continuous data into bins or categories to simplify analysis.

Log Transformation: Apply logarithmic transformations to features when necessary to make their distribution more normal.

**3. Data Reduction:**

Dimensionality Reduction: Reduce the number of features, often using techniques like Principal Component Analysis (PCA) or feature selection to select the most relevant variables.

Outlier Detection and Handling: Identify and deal with outliers, which can distort analysis and modeling results.

**PERFORMING DIFFERENT ANALYSIS**:

Performing different types of analysis on a dataset depends on the goals of your analysis and the nature of the data. Here are some common types of analysis that you might perform on a dataset:

**Descriptive Analysis:**

Summarize and describe the main characteristics of the dataset, including measures of central tendency, dispersion, and visualizations such as histograms, box plots, and bar charts.

**Exploratory Data Analysis (EDA):**

Explore the dataset to uncover patterns, relationships, and anomalies.

Visualize data using scatter plots, heatmaps, and correlation matrices.

Identify potential outliers and trends.

**Statistical Analysis:**

Conduct hypothesis testing and statistical inference to make inferences about the data.

Perform t-tests, ANOVA, chi-squared tests, and other statistical tests as appropriate.

**Code:**

Analyzing a COVID-19 dataset involves various tasks such as loading data, cleaning it, visualizing trends, and performing statistical analysis. Here's a Python code example that demonstrates how to perform basic COVID-19 data analysis using a sample dataset. You can adjust this code to work with your specific COVID-19 dataset**:**

## import pandas as pd

## import matplotlib.pyplot as plt

## url = "<https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress>"

## covid\_data = pd.read\_csv(url)

## covid\_data = covid\_data.transpose()

## covid\_data.columns = covid\_data.iloc[0]

## covid\_data = covid\_data[1:]

## covid\_data.index = pd.to\_datetime(covid\_data.index)

## countries\_to\_analyze = ['US', 'India', 'China']

## covid\_data = covid\_data[countries\_to\_analyze]

## plt.figure(figsize=(12, 6))

## for country in countries\_to\_analyze:

## plt.plot(covid\_data.index, covid\_data[country], label=country)

## plt.xlabel('Date')

## plt.ylabel('Confirmed Cases')

## plt.title('COVID-19 Daily Confirmed Cases')

## plt.legend()

## plt.grid(True)

## plt.show()

**CONCLUSION:**

In our analysis of the COVID-19 vaccine dataset, we have examined various aspects of vaccine distribution, effectiveness, and public response.

Our analysis revealed that vaccine distribution efforts have been substantial, with a significant number of vaccine doses administered worldwide. This has contributed to the global effort to control the spread of COVID-19.