**COVID Vaccines Analysis**

**DAC\_Phase3 (Development Part 1** **)**

**Title:**

The Covid-19 vaccine analysis by collecting and preprocessing the data. Collect and preprocess the COVID-19 vaccine data for analysis.

**Dataset link:**

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

**Abstract:**

The rapid distribution and administration of vaccines are of paramount importance in the global effort to combat infectious diseases, and the comprehensive analysis of vaccine distribution data can provide valuable insights for optimizing public health strategies. With the ongoing challenges posed by the COVID-19 pandemic and the necessity to manage vaccine supply chains effectively, the utilization of advanced machine learning techniques has emerged as a key innovation.

This document focuses on the systematic collection and preprocessing of data related to COVID-19 vaccines. The study leverages diverse datasets, including clinical trial results, real-world vaccination data, adverse event reports, and epidemiological statistics, to provide a comprehensive analysis of the vaccines' performance. The goal is to gain insights into vaccine efficacy, safety, and the impact of different vaccination strategies.

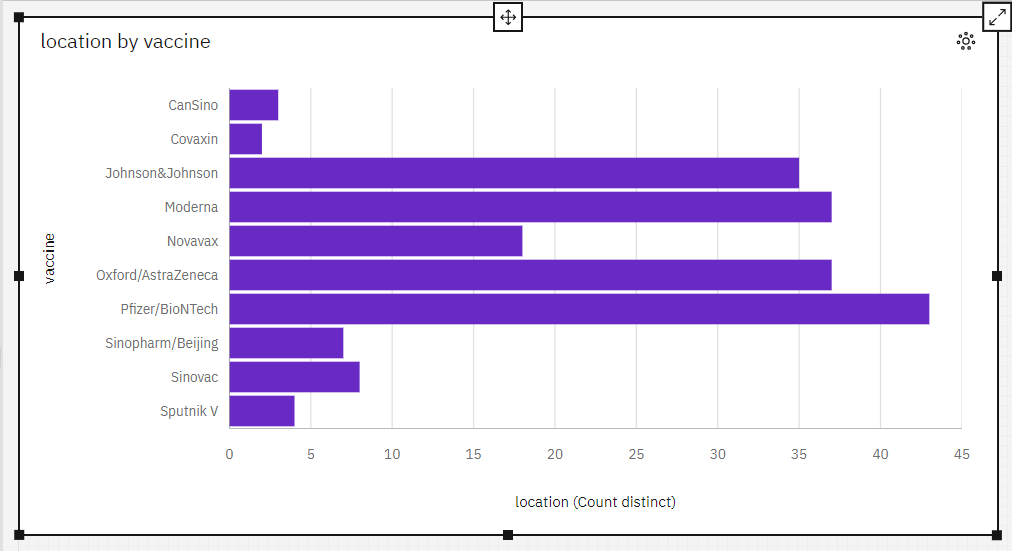
**Benefits:**

**.** **Improved resource allocation:** By understanding distribution patterns, authorities can allocate vaccines where they are needed most.

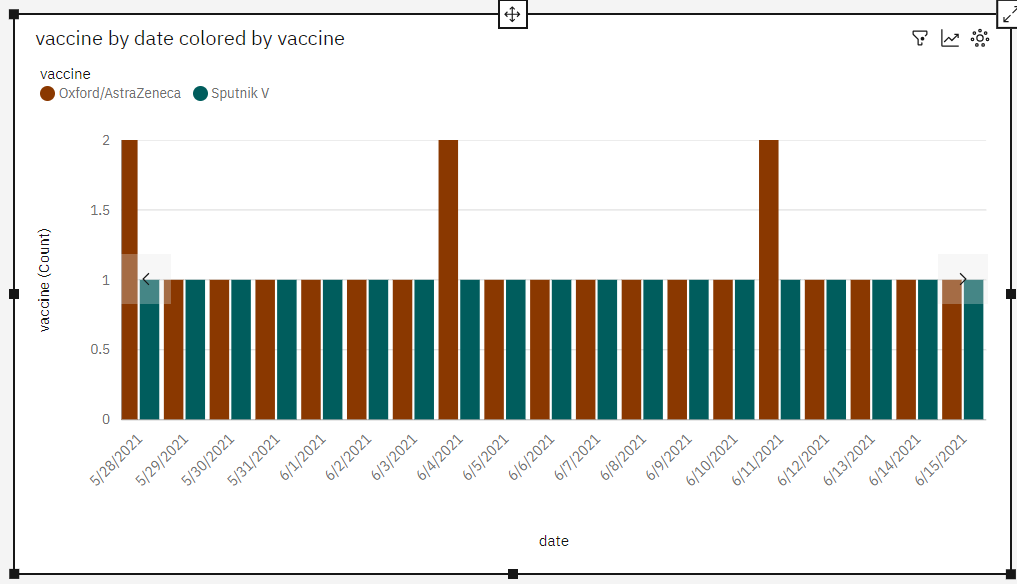
**.** **Enhanced planning:** Time series forecasting can help healthcare systems prepare for future demand and optimize distribution logistics.

**.** **Informed decision-making:** Data-driven insights enable public health officials to make well-informed decisions regarding vaccine distribution and management.

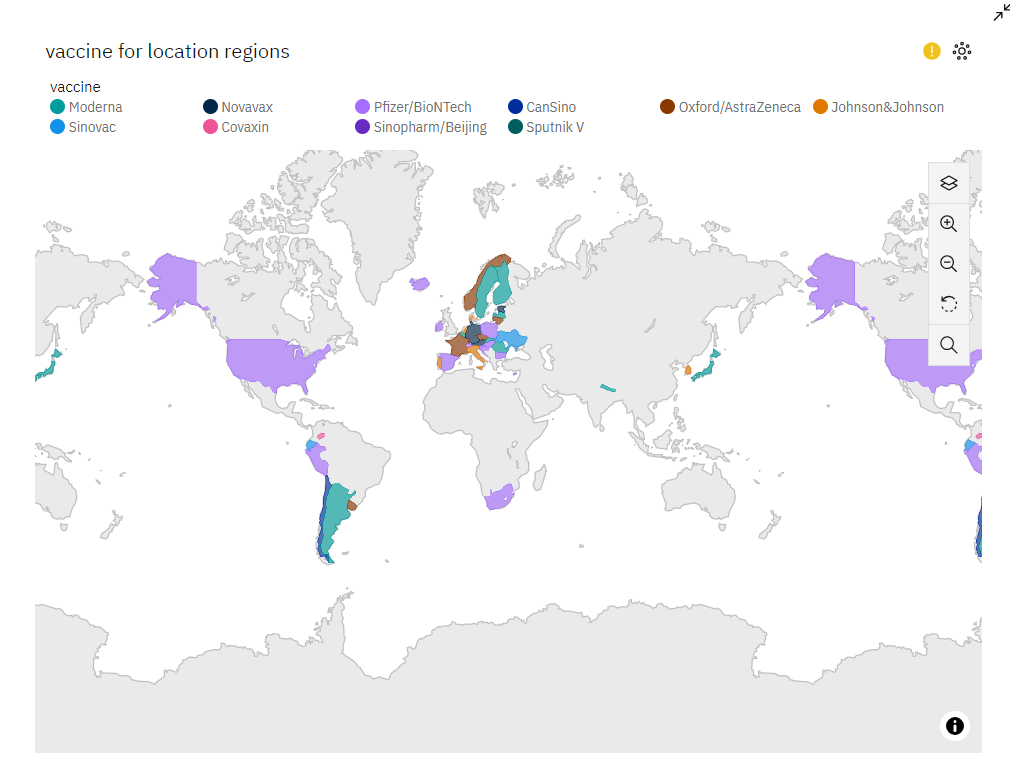
**Analysis using Cognos tool:**

****

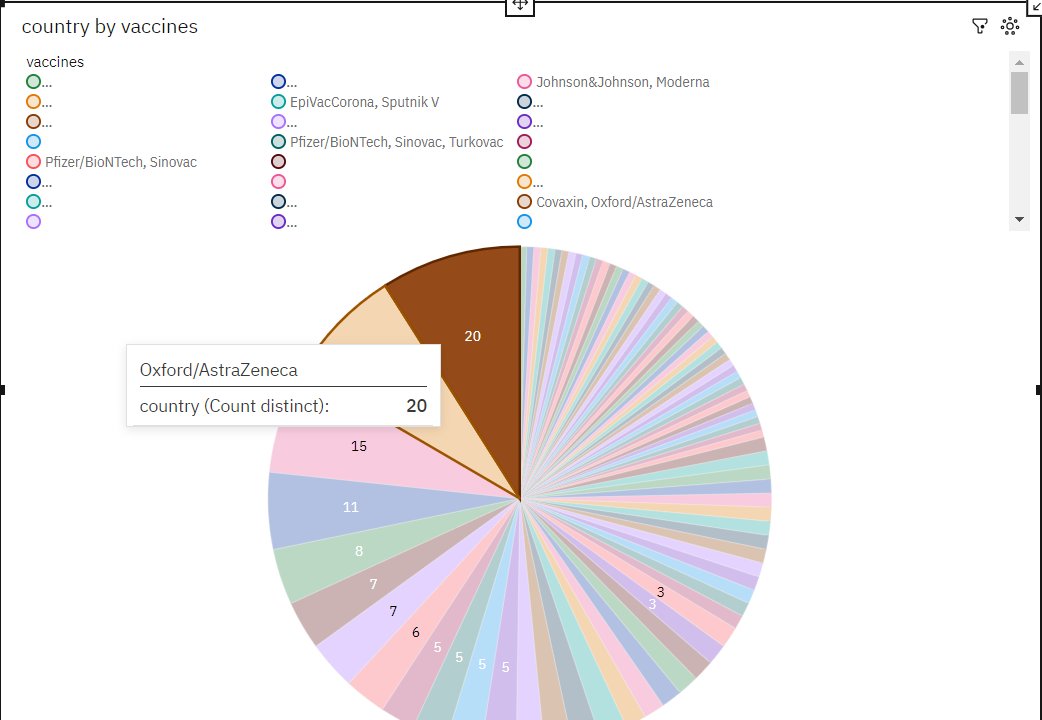
* Pfizer/BioNTech is the most frequently occurring category of vaccine with a count of 8888 items with location values (25 % of the total).
* The total number of results for location, across all vaccines, is almost 36 thousand.



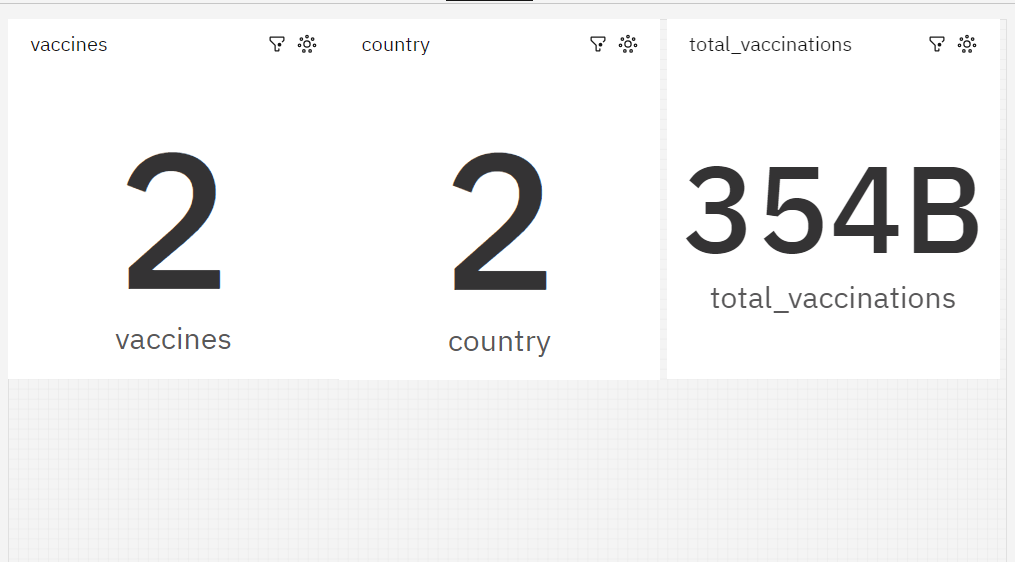
* It is projected that by 2022-06-29, Oxford/AstraZeneca will exceed Sputnik V in vaccine by 0.12.
* From 2021-02-04 to 2021-02-05, Oxford/AstraZeneca's vaccine increased by 100%.
* 2021-02-12 (0.3 %), 2021-12-17 (0.3 %), 2021-12-24 (0.3 %), 2021-02-05 (0.3 %), and 2021-12-31 (0.3 %) are the most frequently occurring categories of date with a combined count of 15 items with vaccine values (1.5 % of the total) .
* Oxford/AstraZeneca is the most frequently occurring category of vaccine with a count of 515 items with vaccine values (53 % of the total).
* The total number of results for vaccine, across all dates, is 971.



* European Union location accounted for 41% of Pfizer/BioNTech total\_vaccinations compared to 24% for Moderna.
* location European Union has the highest total\_vaccinations at approximately 193 billion, out of which vaccine Pfizer/BioNTech contributed the most at approximately 141 billion.
* vaccine Pfizer/BioNTech has the highest total total\_vaccinations due to location European Union.
* The total number of results for vaccine, across all locations, is almost 36 thousand.



* Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech (23.3 %) and Moderna, Oxford/AstraZeneca, Pfizer/BioNTech (21.6 %) are the most frequently occurring categories of vaccines with a combined count of 1756 items with country values (44.9 % of the total).
* The total number of results for country, across all vaccines, is nearly four thousand.



* total\_vaccinations has a weak weekly trend. The smallest values typically occur on Saturday.
* total\_vaccinations has a strong upward trend.
* date 2021-01-15 has the lowest total total\_vaccinations at 0.0, followed by 2021-01-16 at 191181.0.
* date 2021-02-14 has the highest total total\_vaccinations at 0.0, followed by 2022-03-29 at 2.088846244E9.
* Based on the current forecasting, total\_vaccinations may reach over 2.3 billion by date 2022-06-25.
* total\_vaccinations has unusually low values for 9 time points, the most notable of which are 2021-08-12, 2021-07-18, 2021-07-31,2021-08-15, and 2021-08-02.
* From 2021-08-15 to 2021-08-16, total\_vaccinations increased by 502%.
* The overall number of results for total\_vaccinations is 721.

**CLEANING THE DATASET :**

import pandas as pd

import numpy as np

# Load your dataset into a Pandas DataFrame

data = pd.read\_csv("country\_vaccinations.csv")

# Handling Missing Data

# Check for missing values in the dataset

data.isnull().sum()

# Depending on your analysis, you can either drop rows with missing values or fill them with appropriate values (e.g., zeros or the mean of the column).

# Drop rows with missing values

data.dropna(inplace=True)

#  Data Type Conversion

# Ensure the data types of columns are appropriate for analysis

data['date'] = pd.to\_datetime(data['date'])

#  Feature Engineering

# Create new columns or features if needed

data['Vaccination Rate'] = data['daily\_vaccinations'] / data['total\_vaccinations\_per\_hundred']

#  Data Scaling and Normalization

# Scale and normalize numeric columns if necessary

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

data[['total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred','people\_fully\_vaccinated\_per\_hundred']] = scaler.fit\_transform(data[['total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred','people\_fully\_vaccinated\_per\_hundred']])

#  Encoding Categorical Data

# If you have categorical data like 'Country', you can encode it into numerical values.

from sklearn.preprocessing import LabelEncoder

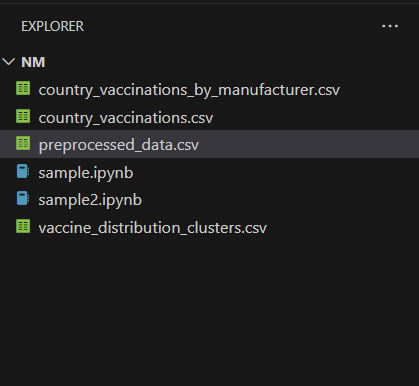
encoder = LabelEncoder()

data['country'] = encoder.fit\_transform(data['country'])

#  Save Preprocessed Data

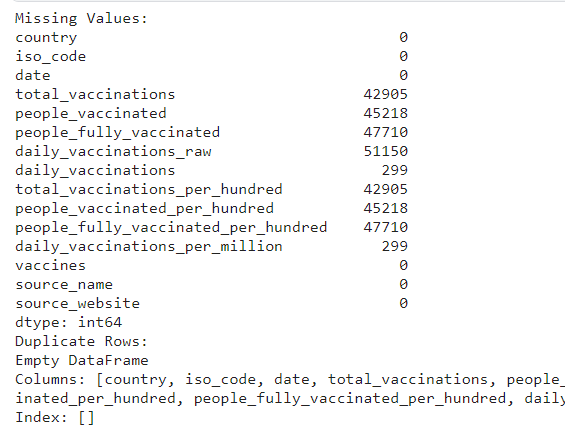
# Save the preprocessed data to a new CSV file for further analysis

data.to\_csv("preprocessed\_data.csv", index=False)

****

**CLEANING THE DATASET**

**BEFORE :**

****

**CODE:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

df=pd.read\_csv('preprocessed\_data.csv')

# Check for missing values

missing\_values = df.isnull().sum()

# Check for duplicates

duplicate\_rows = df[df.duplicated(keep='first')]

print("Missing Values:")

print(missing\_values)

print("Duplicate Rows:")

print(duplicate\_rows)

**AFTER:**

****

**Conclusion:**

In the face of global health crises like the COVID-19 pandemic, advanced machine learning techniques offer innovative solutions to the challenges of vaccine distribution. This document explores the application of clustering and time series forecasting to uncover hidden patterns in vaccine distribution and adverse effects data. By doing so, we aim to contribute to more effective vaccine allocation, better planning, and informed decision-making in the realm of public health.