

COVID-19 WORLD VACCINATION **PROGRESS USING PYTHON**

Phase 3 submission Documents

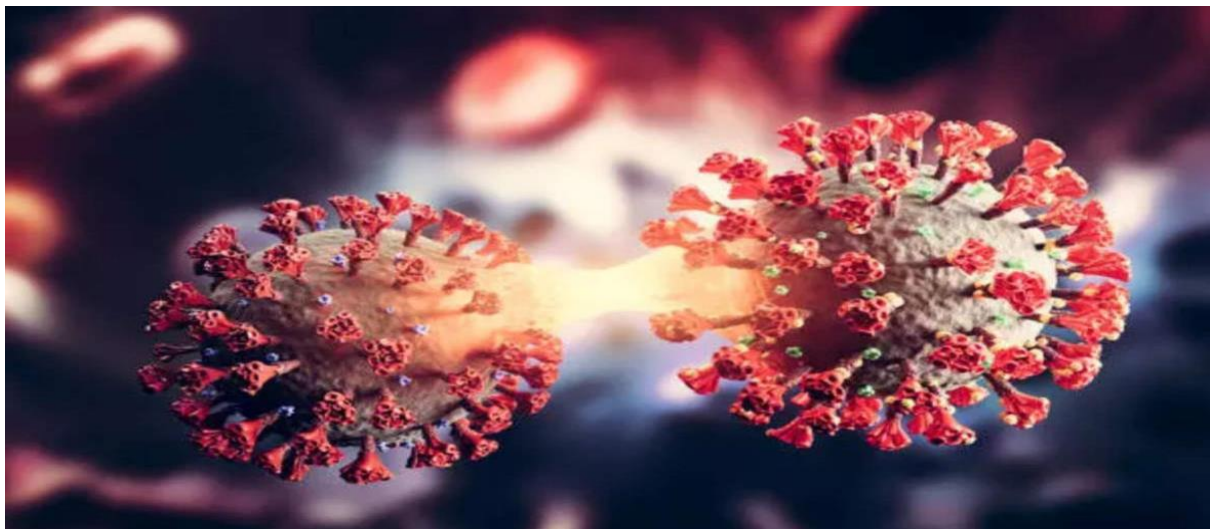
Project Title : Covid-19 Vaccine

Analysis

Name: Santhiya.B

Reg No:712221104018

**College:park college of engineering
and technology**



Covid-19 vaccine Analysis

Introduction:

- ❖ The analysis of COVID-19 vaccines is a critical area of study, especially in the context of the global response to the pandemic. This analysis involves assessing various aspects of COVID-19 vaccines, including their development, efficacy, safety, distribution, and impact on public health.
- ❖ Researchers and experts have been evaluating data from clinical trials, real-world studies, and post-market surveillance to understand how these vaccines perform in different populations.
- ❖ The goal of such analysis is to provide evidence-based information to guide vaccination strategies, monitor vaccine safety, and contribute to the ongoing effort to control the spread of the virus. In this analysis, factors such as vaccine technology, effectiveness against different variants, vaccine hesitancy, and equity in distribution have also been subjects of significant interest and investigation.

Given data set:

	location	date	vaccine	total_vaccinations
0	Argentina	2020-12-29	Moderna	2
1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	20481
4	Argentina	2020-12-30	Moderna	2
5	Argentina	2020-12-30	Oxford/AstraZeneca	3
6	Argentina	2020-12-30	Sinopharm/Beijing	1
7	Argentina	2020-12-30	Sputnik V	40583
8	Argentina	2020-12-31	Moderna	2
9	Argentina	2020-12-31	Oxford/AstraZeneca	3

Necessary step to follow:

1. Import Libraries:

Start by importing the necessary libraries:

Program:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from folium.features import Choropleth
import folium
from folium.features import Tooltip
import seaborn as sns
```

2.Load the Dataset:

Load your dataset into a pandas Dataframe.you can typically find covid-19 vaccine analysis dataset in CSV format ,but you can Adapt code to other format as needed.

Program:

```
df = pd.read_csv(covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv)
```

```
df['date'] = pd.to_datetime(df['date'])
```

In our dataset, the Total Vaccinations represent the cumulative sum of vaccinations up to that date. To express the usage of different vaccines by countries, we need to clean the dataset and transform it.

```
Data=pd.DataFrame(columns=['Country', 'Vaccine', 'Total_vaccine'])
```

```
For country in df["location"].unique():
```

```
    For vaccine in df["vaccine"].unique():
```

```
        Filtered_data = df[(df['location'] == country) &
        (df['vaccine'] == vaccine)]
```

```
        Total_count = filtered_data['total_vaccinations'].max()
```

```
        Data = pd.concat([data, pd.DataFrame({'Country': [country],
        'Vaccine': [vaccine], 'Total_vaccine': [total_count]})],
        ignore_index=True)
```

Challenge involved in loading and preprocessing a covid-19 vaccine analysis dataset:

- **Data Source and Format:** The dataset may be scattered across multiple sources, in various formats, such as CSV, JSON, or APIs. Gathering and harmonizing these sources can be complex.
- **Data Transformation:** Preprocessing might involve scaling, normalization, or encoding categorical variables to make the data suitable for machine learning algorithms.
- **Data Integration:** Combining COVID-19 vaccine data with other relevant datasets, such as demographic or geographic information, can be complex but can provide valuable insights.

How to overcome the challenge involved in loading and preprocessing a covid-19 vaccine analysis data set:

- **Data Collection and Integration:** Collect data from reliable sources and ensure it's well-documented. Combine multiple data sources into a single, cohesive dataset.

➤ **Data Cleaning:**

- Handle missing data by imputation or removal, depending on the extent of missing values.
- Detect and address errors or inconsistencies.
- Handle duplicates and outliers appropriately.

Loading the dataset:

- ❖ **Data Source:** Identify the source from which you want to obtain the dataset. Common sources include government health agencies, research institutions, or publicly available datasets on platforms like Kaggle or GitHub.
- ❖ **Load the Dataset:** Use code to load the dataset into your chosen programming environment. For example, in Python with Pandas, you can use `pd.read_csv()` to load a CSV file.

Program:

```
data_2=pd.DataFrame(columns=['Country', 'Vaccine'])
data["Total_vaccine"] = pd.to_numeric(data["Total_vaccine"], errors="coerce")
for country in data["Country"].unique():
    new_data = data[data["Country"] == country]
    max_vaccine = new_data.loc[new_data["Total_vaccine"].idxmax(), "Vaccine"]
    data_2 = pd.concat([data_2, pd.DataFrame({'Country': [country], 'Vaccine': [max_vaccine]})], ignore_index=True)
```

```
data_2.head()
```

```
data.dropna(axis=0,inplace=True)
```

```
data.head(20)
```

Loading dataset:

Output:

	Country	Vaccine	Total_vaccine
0	Argentina	Moderna	6507561
1	Argentina	Oxford/AstraZeneca	25977231
2	Argentina	Sinopharm/Beijing	28322602
3	Argentina	Sputnik V	20405678
4	Argentina	CanSino	610540
5	Argentina	Pfizer/BioNTech	14681054
10	Austria	Moderna	1585063
11	Austria	Oxford/AstraZeneca	1588222
15	Austria	Pfizer/BioNTech	14584985
16	Austria	Johnson&Johnson	363548
17	Austria	Novavax	3682
20	Belgium	Moderna	4267394
21	Belgium	Oxford/AstraZeneca	2846716
25	Belgium	Pfizer/BioNTech	17451842
26	Belgium	Johnson&Johnson	425639
27	Belgium	Novavax	36
30	Bulgaria	Moderna	491663
31	Bulgaria	Oxford/AstraZeneca	478541
35	Bulgaria	Pfizer/BioNTech	2852218
36	Bulgaria	Johnson&Johnson	511702

Preprocessing the Dataset:

- ❖ **Data Cleaning:** Remove or handle missing values, duplicates, and outliers. Ensure data consistency and accuracy.
- ❖ **Data Transformation:** Perform necessary transformations, like normalization or scaling, to ensure all data is on the same scale.

Visualization and preprocessing of data:

In[1]:

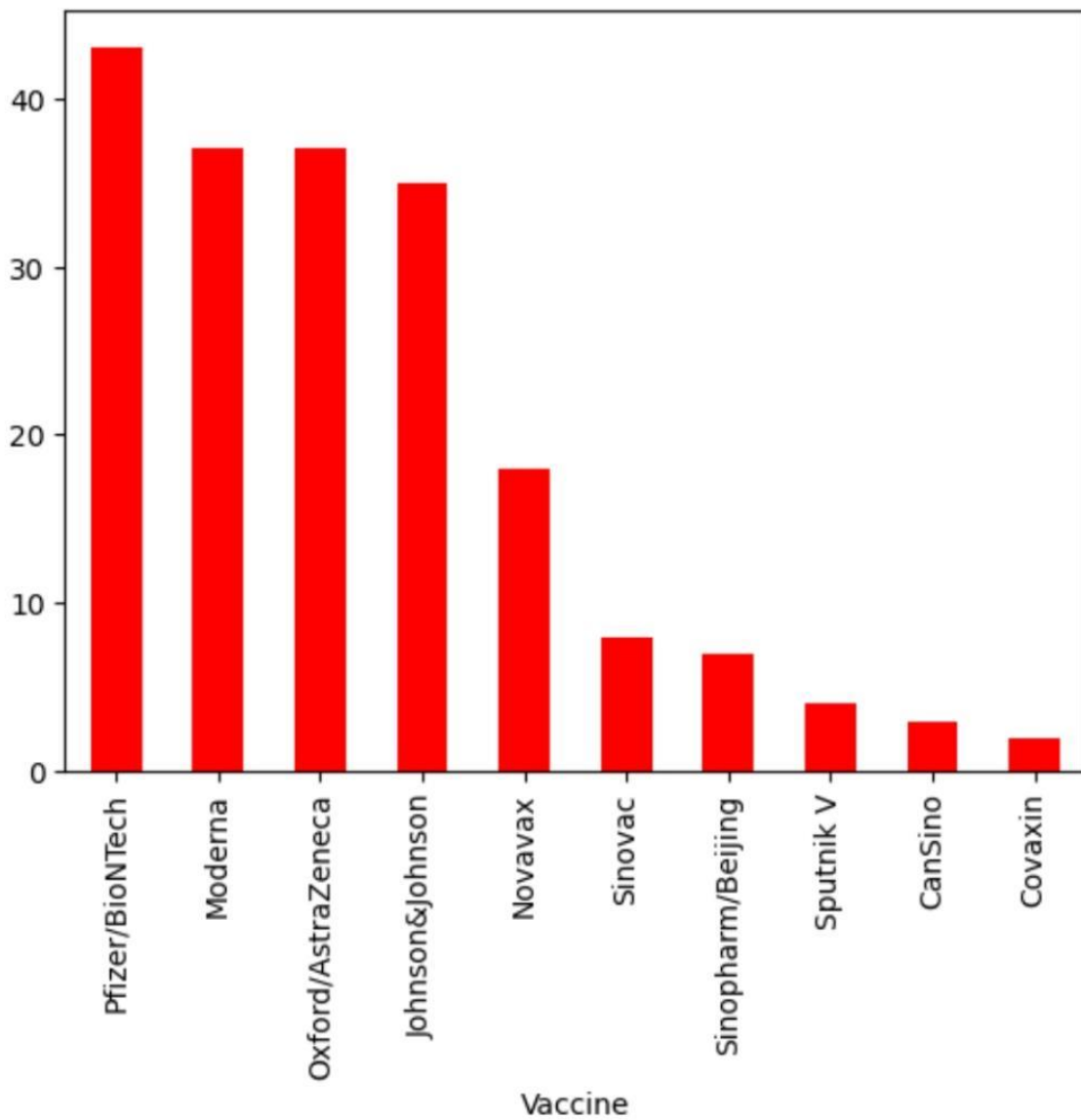
```
Number_of_vaccines = data.groupby('Vaccine')[ 'Country' ].nunique()
```

In[1]:

```
number_of_vaccines.sort_values(ascending=False).plot(kind="bar", color="r")
```


Out[1]:

<Axes: xlabel='Vaccine'>



In[2]:

```
Dtfrm.set_index("Country", inplace=True)
```

In[2]:

```
color=["Lightblue","Purple","Green","Orange",  
,"darkgoldenrod","tan","Gray","Blue","Pink","Li  
ghtgreen"]  
dtfrm["average_vaccination_count"].sort_values(  
ascending=False).head(10).plot(kind="bar",color  
=color)
```

Out[2]:

<Axes: xlabel='Country'>

