Covid-19 vaccines analysis

Phase 2 : INNOVATION

TEAM MEMBERS:

TEAM LEADER : PARVATHALA HARINI-211521104102

TATIPARTHI SRAVANI-211521104169

CHICHILI TEJASWINI REDDY-211521104027

SANDRA CHAITHANYA-211521104134

MOUNIKA GAJENDHRAN-21152104091

COVID-19 VACCINES ANALYSIS:

Innovation in the development and analysis of COVID-19 vaccines has been crucial in the global effort to combat the pandemic. Here are some key innovations and methods used for the analysis of COVID-19 vaccines:

Messenger RNA (mRNA) Technology: Both the Pfizer-BioNTech and Moderna COVID-19 vaccines are based on mRNA technology. This approach allows for the rapid development of vaccines by encoding a portion of the virus's genetic material, enabling the immune system to recognize and respond to it. This innovation has been groundbreaking in vaccine development.

Virus-Like Particle (VLP) Vaccines: Some COVID-19 vaccines, like the one developed by Novavax, use VLPs that mimic the virus's structure without containing genetic material. These VLPs stimulate an immune response without causing the disease. Analyzing the safety and efficacy of VLP vaccines has been a significant focus of research.

Viral Vector Vaccines: Vaccines like the AstraZeneca and Johnson & Johnson vaccines use adenovirus vectors to deliver a piece of the COVID-19 virus's genetic material. Analyzing the effectiveness of this approach and its potential side effects has been a critical part of vaccine analysis.

Clinical Trials: The analysis of vaccine candidates involves extensive clinical trials to assess safety and efficacy. Innovative trial designs, such as adaptive trials, have been used to accelerate the evaluation process.

Genomic Sequencing: Sequencing the virus's genome allowed for the rapid identification of variants and assessment of their impact on vaccine efficacy. Continuous genomic surveillance is essential to understand how vaccines perform against evolving strains.

Immunogenicity Assessment: Innovative assays and techniques have been developed to evaluate the immunogenicity of vaccines, measuring antibody and T-cell responses to determine their effectiveness.

DATASET LINK :
https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress

DETAIL ABOUT THE LIBRARIES AND THE WAY DOWNLOAD:
PANDAS : Pandas is a popular open-source data manipulation and analysis library for Python. It provides easy-to-use data structures and data analysis tools for working with structured data. Pandas is an essential tool for data scientists, analysts, and anyone working with data in Python.
Data Structures:
Pandas offers two primary data structures: Series and DataFrame.

Series: A one-dimensional array-like object that can hold any data type. It is essentially a labeled NumPy array.
DataFrame: A two-dimensional table of data with rows and columns. It is the most commonly used structure for data analysis and is similar to a spreadsheet

```
C:\Users\chait>pip install pandas
Collecting pandas
  Obtaining dependency information for pandas from https://files.pythonhosted.org/packages/89/c8/466196b756d74326820f
068540f0f/pandas-2.1.1-cp312-cp312-win_amd64.whl.metadata
  Downloading pandas-2.1.1-cp312-cp312-win_amd64.whl.metadata (18 kB)
Collecting numpy>=1.26.0 (from pandas)
  Obtaining dependency information for numpy>=1.26.0 from https://files.pythonhosted.org/packages/98/d7/1cc7a1111840
808ebaa90ee95902/numpy-1.26.0-cp312-cp312-win_amd64.whl.metadata
  Downloading numpy-1.26.0-cp312-cp312-win_amd64.whl.metadata (61 kB)
     ---------------------------------------- 61.1/61.1 kB 180.9 kB/s eta 0:00:00
Collecting python-dateutil>=2.8.2 (from pandas)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ---------------------------------------- 247.7/247.7 kB 379.9 kB/s eta 0:00:00
Collecting pytz>=2020.1 (from pandas)
  Obtaining dependency information for pytz>=2020.1 from https://files.pythonhosted.org/packages/32/4d/aaf7eff5deb402
9f8ef9994261fc2/pytz-2023.3.post1-py2.py3-none-any.whl.metadata
  Downloading pytz-2023.3.post1-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.1 (from pandas)
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
     ---------------------------------------- 341.8/341.8 kB 848.1 kB/s eta 0:00:00
Collecting six>=1.5 (from python-dateutil>=2.8.2->pandas)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Downloading pandas-2.1.1-cp312-cp312-win_amd64.whl (10.5 MB)
     ---------------------------------------- 10.5/10.5 MB 695.0 kB/s eta 0:00:00
Downloading numpy-1.26.0-cp312-cp312-win_amd64.whl (15.5 MB)
     ---------------------------------------- 15.5/15.5 MB 611.9 kB/s eta 0:00:00
Downloading pytz-2023.3.post1-py2.py3-none-any.whl (502 kB)
     ---------------------------------------- 502.5/502.5 kB 927.0 kB/s eta 0:00:00
Installing collected packages: pytz, tzdata, six, numpy, python-dateutil, pandas
Successfully installed numpy-1.26.0 pandas-2.1.1 python-dateutil-2.8.2 pytz-2023.3.post1 six-1.16.0 tzdata-2023.3
```

NumPy (Numerical Python) is a powerful library in Python for numerical and mathematical operations. It provides support for working with large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. Here are some common uses of the NumPy library.

Array Creation: You can create arrays easily using NumPy. For example, you can create arrays from lists or even generate arrays with specific values and patterns using functions like np.array(), np.zeros(), np.ones(), np.arange(), and np.linspace().

Array Manipulation: NumPy offers various functions for manipulating arrays. You can reshape, concatenate, split, and transpose arrays as needed.

Mathematical Operations: You can perform element-wise mathematical operations on arrays. NumPy provides functions for addition, subtraction, multiplication, division, exponentiation, and more.

Linear Algebra: NumPy is well-suited for linear algebra operations. You can compute matrix multiplication, determinant, inverse, eigenvalues, and eigenvectors using functions like np.dot(), np.linalg.det(), np.linalg.inv(), and np.linalg.eig().

Statistical Analysis: NumPy includes functions for statistical analysis, such as mean, median, variance, standard deviation, and correlation. You can use functions like np.mean(), np.median(), np.var(), and np.corrcoef().

Random Number Generation: NumPy provides tools for generating random numbers and random arrays. This is useful for simulations and experiments. You can use functions like np.random.rand(), np.random.randn(), and np.random.randint().

Indexing and Slicing: You can access and manipulate elements within arrays using indexing and slicing, similar to standard Python lists.

Broadcasting: NumPy supports broadcasting, which allows you to perform operations on arrays with different shapes. This can simplify many operations, making your code more concise.

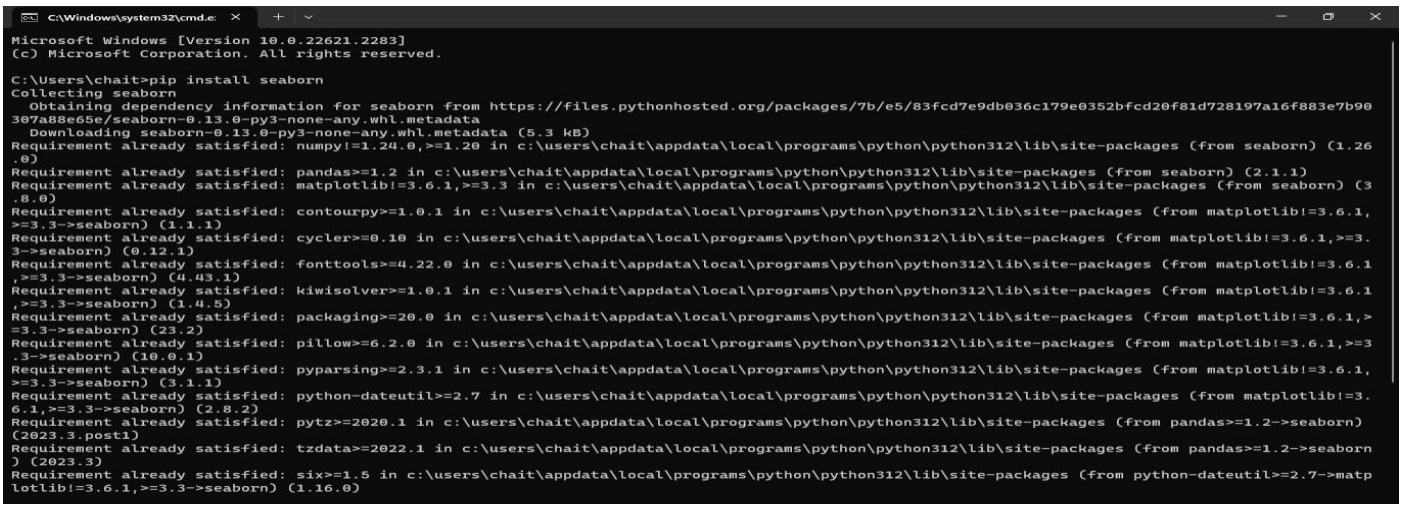
File I/O: NumPy can read and write data to/from files. You can save and load NumPy arrays from various file formats, including binary and text formats.

Signal Processing: NumPy includes functions for signal processing, like Fourier transforms, convolution, and filtering, making it useful for audio and image processing.

Data Analysis and Data Manipulation: NumPy is often used in conjunction with libraries like Pandas for data analysis and manipulation. NumPy arrays can be easily converted to Pandas DataFrames.

Machine Learning: Many machine learning libraries, such as scikit-learn and TensorFlow, use NumPy arrays as their data structures. NumPy provides a common interface for data handling in these libraries.

Plotting: While NumPy itself is not a plotting library, it can be used in combination with libraries like Matplotlib to create data visualizations and plots.



Seaborn is a Python data visualization library built on top of Matplotlib. It is designed for creating informative and attractive statistical graphics. Seaborn provides a high-level interface for creating aesthetically pleasing and informative statistical plots, making it a popular choice for data analysts and data scientists. Here are some of the key features and common uses of Seaborn:

Improved Aesthetics: Seaborn comes with built-in themes and color palettes that make your plots more visually appealing. It offers a more polished and attractive look compared to the default Matplotlib plots.

Statistical Plotting: Seaborn specializes in creating various statistical plots, such as scatter plots, bar plots, box plots, violin plots, pair plots, and joint plots. These plots are designed to help you understand data distributions and relationships.

Categorical Data Visualization: Seaborn makes it easy to work with categorical data by providing functions to create categorical plots like bar plots, count plots, and point plots. These are useful for visualizing data with discrete categories.

Regression Analysis: Seaborn simplifies the process of visualizing regression models with functions like lmplot() and regplot(). These can help you explore relationships between variables and fit regression lines to your data.

Time Series Plotting: Seaborn offers support for time series data visualization. You can create time series plots with different time units and styles, including line plots and heatmaps.

Data Exploration: Seaborn can be used to explore your data by creating distribution plots, histograms, and kernel density estimation (KDE) plots, helping you understand data characteristics and identify patterns.

Multi-Plot Grids: Seaborn provides functions to create multi-plot grids, such as pair plots and facet grids. These are useful for visualizing relationships between multiple variables in a structured way.

Color Palettes: Seaborn includes various color palettes that can be applied to your plots. You can customize the colors of your visualizations or choose from predefined palettes to match the context of your data.

Automatic Data Aggregation: Some Seaborn functions can automatically perform data aggregation and summarization, simplifying the creation of complex plots like bar plots and box plots.

Integration with Pandas: Seaborn works well with Pandas DataFrames, making it easy to create visualizations directly from your data without extensive data manipulation.

Customization: While Seaborn provides attractive defaults, it also allows for extensive customization. You can fine-tune the appearance of your plots using various parameters and styles.

TEST AND TRIAL:

Creating a data science project to analyze COVID-19 vaccine data can be a valuable and relevant endeavor. Here's a step-by-step guide to get you started with a test and trial project for COVID-19 vaccine analysis:

1. Define the Problem and Objectives:

Clearly state the problem you want to address. For example, you can analyze the effectiveness of different COVID-19 vaccines, vaccination rates, side effects, or any other specific aspect.
Define your project's objectives, such as gaining insights, making predictions, or providing recommendations.

2. Data Collection:

Gather relevant data from trusted sources. You can consider data sources like government health departments, the World Health Organization (WHO), or academic institutions.
Make sure the data is in a structured format (CSV, Excel, etc.) and contains relevant information.

3. Data Preprocessing:

Clean and prepare the data. This may involve handling missing values, data transformation, and feature engineering.
Ensure that the data is ready for analysis by converting it into a format that can be easily loaded into Python or another programming environment.

4. Data Exploration and Analysis:

Use Python libraries like NumPy, Pandas, and Seaborn to explore and visualize the data.
Generate summary statistics, histograms, scatter plots, and other visualizations to understand the data better.
Perform statistical analyses to identify trends and correlations.

5. Hypothesis Testing (Optional):

If relevant, you can perform hypothesis testing to validate or invalidate certain hypotheses regarding vaccine effectiveness or other factors.

6. Machine Learning (Optional):

```
In [2]: df.shape

        df.isnull()
        df.isnull().sum()
        df.isnull().sum().sum()
Out[2]: 323414
```

If you want to make predictions or build models, consider using machine learning techniques. For example, you can create models to predict vaccine distribution or vaccine uptake rates.

7. Data Visualization:

Use Seaborn and Matplotlib to create informative and visually appealing charts, graphs, and dashboards to present your findings.

8. Interpretation and Insights:

Summarize your findings, draw conclusions, and provide actionable insights.
Explain the implications of your analysis and how it can be used to address the problem or achieve your objectives.

9. Recommendations:

Based on your analysis, offer recommendations or suggestions. For example, you might recommend targeted vaccination campaigns, policy changes, or further research.

10. Documentation and Reporting:

Document your project thoroughly, including the data sources, methods, and code.
Create a report or presentation to communicate your findings and insights effectively.

11. Testing and Validation:

Ensure that your code is well-tested and the results are validated.
Share your project with colleagues or experts in the field to get feedback and validation.

12. Deployment (Optional):

If your analysis results in a useful tool or application, consider deploying it for wider use.

13. Share and Communicate:

Share your findings and insights with the community through articles, blog posts, or presentations.
Remember that the COVID-19 situation is dynamic, so it's important to keep your data and analysis up to date. Additionally, make sure to follow ethical guidelines and respect data privacy when working with sensitive health data

```
In [1]: import pandas as pd
        df=pd.read_csv('country_vaccinations.csv')
        df.head(10)
```

Out[1]:

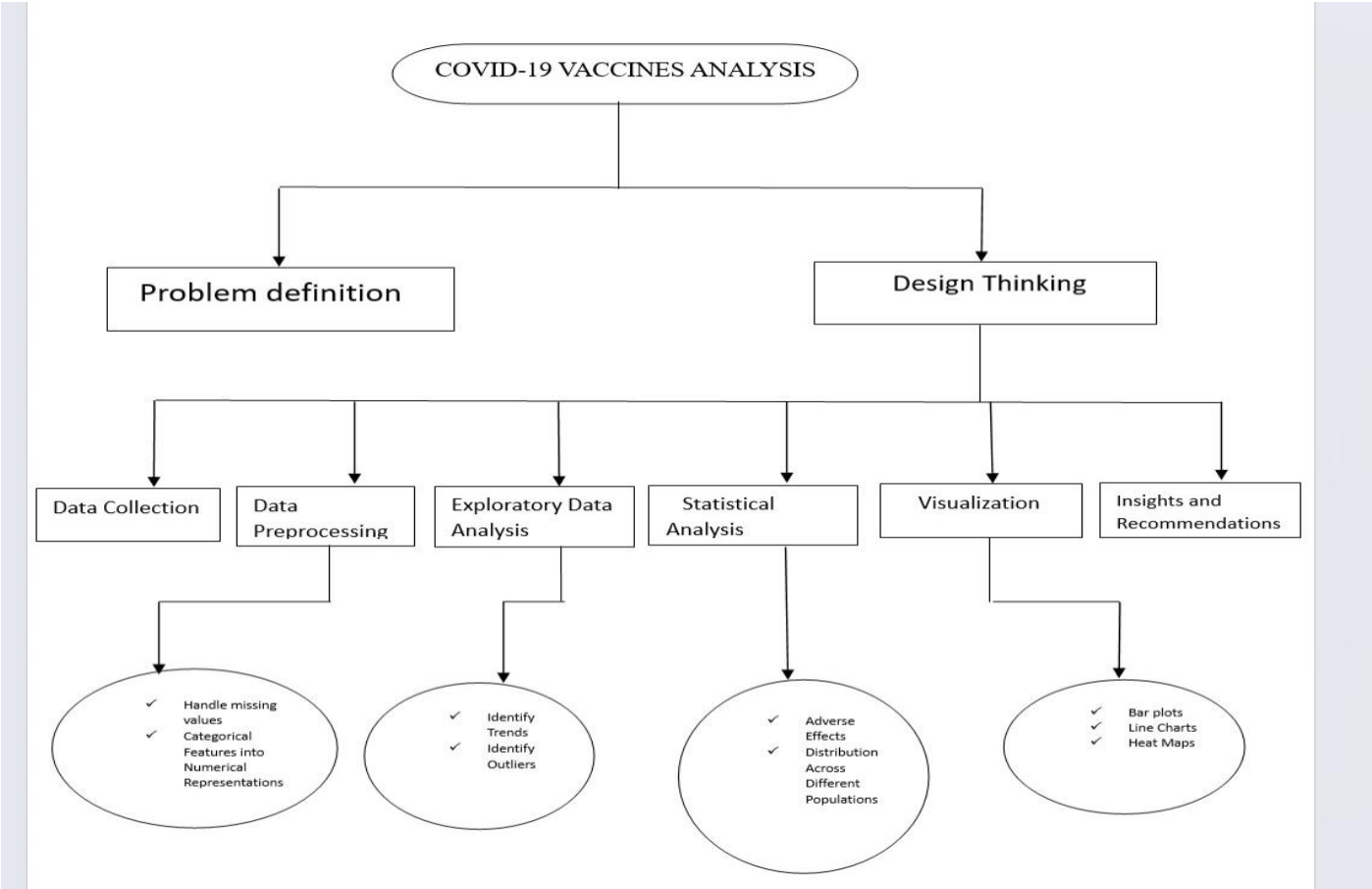
	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
5	Afghanistan	AFG	2021-02-27	NaN	NaN	NaN	NaN	1367.0	
6	Afghanistan	AFG	2021-02-28	8200.0	8200.0	NaN	NaN	1367.0	
7	Afghanistan	AFG	2021-03-01	NaN	NaN	NaN	NaN	1580.0	

In [5]:
```
#filling null values


df2=df.fillna(value=0)
df2

df2.isnull().sum().sum()

df3=df.fillna(value=df['total_vaccinations_per_hundred'].median())
df3
```

Out[5]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccina
0	Afghanistan	AFG	2021-02-22	0.00	0.00	67.52	67.52	67.52	
1	Afghanistan	AFG	2021-02-23	67.52	67.52	67.52	67.52	1367.00	
2	Afghanistan	AFG	2021-02-24	67.52	67.52	67.52	67.52	1367.00	
3	Afghanistan	AFG	2021-02-25	67.52	67.52	67.52	67.52	1367.00	
4	Afghanistan	AFG	2021-02-26	67.52	67.52	67.52	67.52	1367.00	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.00	4814582.00	3473523.00	139213.00	69579.00	

REST OF EXPLANATION AND METRICS OF ACCURACY :



As we conclude this analysis, we extend our gratitude to the dedicated healthcare professionals administering vaccines, the scientists driving vaccine development and safety monitoring, and the policymakers shaping vaccination strategies. The path forward is illuminated by the collaborative spirit of our global community, as well as the lessons learned from data-driven analysis. In the journey ahead, we offer the following overarching recommendations:

1: **. Vaccination Prioritization and Outreach** Prioritize vaccinations for underserved communities and engage community leaders to build trust and facilitate vaccine access.

2**.Safety and Surveillance**: Maintain rigorous safety monitoring systems and transparent reporting mechanisms to promptly address adverse events and maintain public confidence.

3.**Communication and Education**: Continuously refine and disseminate evidence-based communication strategies to counter misinformation and hesitancy.

4. **Research and Adaptation**: Invest in ongoing research to assess long-term vaccine efficacy, address emerging variants, and inform booster shot strategies.