**NAAN MUDHALVAN-IBM**

COVID-19 VACCINE ANALYSIS



|  |  |
| --- | --- |
| **NAME** | **SALADAGU KALYANI** |
| **ROLE** | **TEAM MEMBER** |
| **NAAN MUDHALVAN ID** | **au732121104026** |
| **GROUP** | **3** |
| **COLLEGE** | **NANDHA COLLEGE OF TECHNOLOGY** |
| **DATE** | **01/11/2023** |

**COVID-19 VACCINE ANALYSIS**

**DESCRIPTION:**

COVID-19 vaccines are special shots that help protect people from getting sick with the coronavirus, which causes COVID-19. These vaccines were created to prevent the spread of the virus and reduce the severity of the illness.

**1. Problem Understanding:**

* In the ongoing battle against the COVID-19 pandemic, it is imperative to closely monitor and assess the progress of vaccination campaigns. One of the key challenges is to determine how many individuals have been vaccinated within a specific population or geographic region. Without a comprehensive understanding of vaccination coverage, it is difficult to evaluate the effectiveness of vaccination efforts, identify underserved communities, and make informed decisions regarding resource allocation and future vaccination strategies.

**2. Solution for solving the problem:**

* The solution to this problem is to develop a robust and real-time COVID-19 Vaccine Coverage Tracking System that accurately calculates and reports the number of individuals who have received COVID-19 vaccinations.
* By monitoring how many individuals have been vaccinated within a specific population or region, this system provides valuable data for analysis, decision-making, and resource allocation. To do this we have proposed.

1. **Data Collection and Integration**:
   * Collect vaccination data from multiple sources.
   * Integrate the data into a centralized repository.
2. **Data Preprocessing**:
   * Clean and preprocess the data to address missing values and inconsistencies.
3. **Calculation of Vaccination Coverage**:
   * Develop algorithms to calculate vaccination coverage rates for different demographic groups and geographic regions.
4. **Data Visualization**:
   * Create basic visualizations (e.g., charts, graphs) to illustrate vaccination coverage trends.
5. **Real-Time Data Updates**:
   * Implement mechanisms for receiving and processing real-time data updates.
6. **Reporting**:
   * Generate regular reports summarizing vaccination coverage findings.
7. **Ethical Considerations**:
   * Ensure compliance with ethical and privacy standards in data handling.

**DESIGN THINKING APPROACH FOR SOLVING THE PROBLEM OF TO DETERMINE HOW MANY INDIVIDUALS HAVE BEEN VACCINATED WITHIN A SPECIFIC POPULATION OR GEOGRAPHIC REGION USING COVID-19 VACCINE ANALYSIS**.

**1. Empathize**

* Collect vaccination data while respecting individuals' privacy and emotional well-being.
* Analyse data considering cultural nuances and demographic disparities with empathy, ensuring inclusive communication and equitable resource allocation.
* Communicate vaccination coverage findings with compassion and clear, accessible information**.**

**2. Define**

* Covid-19 vaccine analysis involves the examination of data related to vaccine distribution, coverage, and effectiveness to inform public health decisions and optimize vaccination campaigns, contributing to the global effort to control the pandemic.
* It entails the comprehensive assessment of individuals' vaccination status within a specified population or region.
* And providing insights into both the vaccinated and unvaccinated segments, facilitating equitable vaccine distribution and pandemic control strategies."

**3.Ideate:**

* Gather vaccination data from reliable sources.
* Integrate data into a centralized repository.
* Clean and preprocess data to address issues.
* Standardize data formats and ensure accuracy.

**4.Prototype**

* Simulate a small dataset with sample vaccination records.
* Calculate basic coverage rates.
* Create a simple user interface for data input and visualization.
* Develop a demo dashboard for displaying coverage data.
* Provide basic documentation.

**5.Test**

* Start with a small-scale pilot test.
* Test your prototype with a limited dataset and a small user group.
* Identify any technical issues, usability problems, or data quality issues.
* Collect feedback from users regarding their experience.

**6. Implement**

* Move forward with full implementation.
* Address issues identified during testing and refine the project accordingly.
* Deploy the full-scale COVID-19 vaccine coverage analysis system.
* Ensure data sources are integrated and updated regularly.

**7. Feedback and Iterate:**

* Continuously improve based on feedback.
* Solicit feedback from users and stakeholders regularly.
* Use feedback to make iterative improvements to the system.
* Consider additional features or enhancements suggested by users.

**8. Scale and Optimize:**

* Scale up the project as needed.
* Assess scalability requirements and optimize system performance.
* Expand data sources and coverage to larger populations or regions.
* Implement efficient data processing and storage solutions.

**9. Educate and Train:**

* Ensure effective utilization.
* Provide training and educational materials to users and administrators.
* Conduct workshops or webinars to familiarize stakeholders with the system.
* Promote best practices for data interpretation and decision-making.

**10. Celebrate Success:**

* Organize a recognition event or ceremony to acknowledge the contributions of the project team, stakeholders, and community partners who supported the COVID-19 vaccine analysis initiative.
* Highlight key achievements, such as improvements in vaccination coverage, data-driven decision-making, or equitable vaccine distribution**.**

**DATASET AND ITS DETAIL (WWW.KAGGLE.COM/DATA):**

The dataset was obtained from Kaggle, a well-known data science and machine learning platform. It provides a comprehensive COVID-19 dataset with information on cases, vaccinations, demographics, and other related variables.

LINK:https://www.kaggle.com/datasets/swatikhedekar/state-wise-india-covid19vaccination

**DETAILS ABOUT COLUMNS**

The columns are;

* State/UTs
* Total Vaccination Doses
* Dose 1
* Dose 2
* Population

**State/UTs:**

* The "State/UTs" column in the dataset refers to the Indian states and union territories (UTs) where COVID-19 vaccination data is recorded.
* Each entry in this column specifies the specific region or administrative division in India where the vaccination data is associated.

**Total Vaccination Doses:**

* The "Total Vaccination Doses Administered" column in the dataset represents the cumulative count of COVID-19 vaccine doses administered within each Indian state or union territory.
* This count includes all doses given, including both the first dose (initial vaccination) and the second dose (booster or follow-up vaccination), if applicable.

**Dose 1:**

* The "Dose 1" column in the dataset represents the number of individuals who have received the first dose of a COVID-19 vaccine within each Indian state or union territory.

**Dose 2:**

* The "Dose 2" column in the dataset represents the number of individuals who have received the second dose (booster or follow-up dose) of a COVID-19 vaccine within each Indian state or union territory.

**Population:**

* The "Population" column in the dataset provides the estimated population of each Indian state or union territory (UT).
* This figure represents the total number of residents in each region and serves as a fundamental demographic statistic.
* It's a crucial reference point for assessing COVID-19 vaccination coverage in relation to the population size of each state or UT.
* Comparing vaccination data to population data helps in understanding the proportion of people who have been vaccinated relative to the total population, which is vital for evaluating the effectiveness and reach of vaccination efforts in different regions of India.

**Tools and Libraries**

* **Python:** Utilize Python as the primary programming language.
* **Data Analysis:** Pandas for data manipulation, NumPy for numerical operations.
* **Data Visualization:** Matplotlib, Seaborn, Plotly for creating visualizations.
* **Geospatial Analysis:** Geographic Information System (GIS) tools like GeoPandas.
* **Machine Learning:** Scikit-learn for predictive modeling.
* **Web Development:** Flask or Django for building interactive dashboards.

**Library Installation:**

* To install the necessary libraries, you can use Python's package manager, pip. Open your command line or terminal and run the following commands
* pip install pandas requests matplotlib seaborn numpy geopandas folium scikit-learn plotly dash streamlit

**TRAIN AND TEST:**

**Data Loading:**

* Start by importing the dataset from Kaggle using Python and Pandas**.**

**Data Preprocessing:**

* Clean and prepare the dataset, addressing missing values and performing feature engineering if necessary.

**Data Splitting:**

* Divide the data into training and testing sets, with the training set usually being larger (e.g., 80% for training and 20% for testing).

**Model Selection:**

* Choose a machine learning model suitable for your analysis, such as regression or decision trees.

**Model Training:**

* Train the selected model using the training data.

**Model Evaluation:**

* Assess the model's performance on the testing data using metrics like Mean Absolute Error (MAE) or R-squared (R^2).

**METRICS:**

1. **Vaccination Coverage Rate (VCR):** This metric calculates the percentage of the population that has received the COVID-19 vaccine. A higher VCR indicates a more successful vaccination campaign.
2. **False Positive Rate (FPR):** This metric assesses the accuracy of your system in identifying vaccinated individuals. It measures the percentage of individuals incorrectly classified as unvaccinated (false negatives). A lower FPR is desired to minimize false alarms.
3. **False Negative Rate (FNR):** FNR measures the percentage of vaccinated individuals who are incorrectly classified as unvaccinated (false negatives). A lower FNR is essential to ensure that vaccinated individuals are correctly identified.

**INNOVATIVE PHASE:**

The objective of our project is to enhance public health protection by analysing data on vaccinated individuals. In the event of a future COVID-19 outbreak, our project aims to leverage our dataset on vaccination records to accurately identify and isolate individuals who have not been vaccinated. This proactive approach is designed to help mitigate the spread of the virus and safeguard the health of the community. To assess the effectiveness of our project, we will utilize a range of metrics, including vaccination coverage rate, false positive rate, false negative rate, precision, recall, F1 score, accuracy, specificity, and relevant area-under-the-curve measurements to ensure the accurate identification of vaccinated and unvaccinated individuals.

**Development:**

The project involves analyzing COVID-19 data to gain insights into the impact of vaccination campaigns on public health. We aim to understand vaccination rates, vaccine efficacy, and the relationship between vaccination and the spread of COVID-19

**GIVEN DATA SET:**

The dataset "State-wise India COVID-19 Vaccination" on Kaggle provides detailed information about the COVID-19 vaccination efforts in different states of India. It includes data related to the administration of COVID-19 vaccines, the type of vaccines used, and the number of individuals vaccinated.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **State/UTs** | **Total Vaccination Doses** | **Dose 1** | **Dose 2** | **Population** |
| Andaman and Nicobar | 629054 | 311893 | 317161 | 399001 |
| Andhra Pradesh | 84147957 | 40624263 | 43523694 | 91702478 |
| Arunachal Pradesh | 1596166 | 856732 | 739434 | 1711947 |
| Assam | 42998698 | 22535419 | 20463279 | 35998752 |
| Bihar | 119963226 | 62590002 | 57373224 | 128500364 |
| Chandigarh | 2001114 | 1088086 | 913028 | 1158040 |
| Chhattisgarh | 36927545 | 18855121 | 18072424 | 32199722 |
| Dadra and Nagar Haveli and Daman and Diu | 1319914 | 729023 | 590891 | 773997 |
| Delhi | 30710281 | 16555043 | 14155238 | 19301096 |
| Goa | 2572559 | 1353009 | 1219550 | 1521992 |
| Gujarat | 98534412 | 49269034 | 49265378 | 70400153 |
| Haryana | 40581317 | 21938012 | 18643305 | 28900667 |
| Himachal Pradesh | 11770370 | 6028184 | 5742186 | 7503010 |
| Jammu and Kashmir | 20272520 | 9936338 | 10336182 | 14999397 |
| Jharkhand | 36992685 | 21243308 | 15749377 | 40100376 |
| Karnataka | 100209012 | 49971474 | 50237538 | 69599762 |
| Kerala | 50729256 | 26999013 | 23730243 | 34698876 |
| Ladakh | 404656 | 219238 | 185418 | 290492 |
| Lakshadweep | 112378 | 56831 | 55547 | 66001 |
| Madhya Pradesh | 107931053 | 54060775 | 53870278 | 85002417 |
| Maharashtra | 155773246 | 84570317 | 71202929 | 124904071 |
| Manipur | 2666749 | 1457120 | 1209629 | 3436948 |
| Meghalaya | 2348527 | 1329229 | 1019298 | 3772103 |
| Mizoram | 1440084 | 783477 | 656607 | 1308967 |
| Nagaland | 1515042 | 835771 | 679271 | 2073074 |
| Odisha | 60803739 | 31331147 | 29472592 | 47099270 |
| Puducherry | 1620765 | 903986 | 716779 | 1646050 |
| Punjab | 41717794 | 22309609 | 19408185 | 30501026 |
| Rajasthan | 97164120 | 51029686 | 46134434 | 79502477 |
| Sikkim | 1045753 | 539208 | 506545 | 658019 |
| Tamil Nadu | 107856629 | 56110543 | 51746086 | 83697770 |
| Telangana | 58332610 | 29547155 | 28785455 | 38157311 |
| Tripura | 4962881 | 2653391 | 2309490 | 4184959 |
| Uttar Pradesh | 299773777 | 153669397 | 146104380 | 231502578 |
| Uttarakhand | 16068172 | 8164652 | 7903520 | 11700099 |
| West Bengal | 128418265 | 67232447 | 61185818 | 100896618 |

**Tasks Completed:**

With this dataset, I have developed code using the Python programming language. In this phase, I have performed the following actions:

* List Files in Current Directory
* Viewing the First Rows of a DataFrame
* Calculation of Mean
* Minimum Members Vaccinated
* Maximum Members Vaccinated

1. **List Files in Current Directory**: List Files in Current Directory is a concise and appropriate title for this code snippet. It accurately describes the main action being performed in the code, which is using the **ls** shell command to list the files in the current directory.
2. **Viewing the First Rows of a DataFrame:** It serves the purpose of displaying the first few rows of a DataFrame for quick inspection and understanding of the dataset.
3. **Calculation of Mean:** I calculated the mean (average) value for a specific variable within the dataset to understand the central tendency of that variable.
4. **Minimum Members Vaccinated:** I identified the state or union territory with the lowest number of members vaccinated and reported the corresponding vaccination figure.
5. **Maximum Members Vaccinated:** Similarly, I identified the state or union territory with the highest number of members vaccinated and reported the associated vaccination count.

These actions provide a foundational understanding of the dataset and facilitate the exploration and interpretation of key statistics related to COVID-19 vaccination across different regions.

**List Files in Current Directory:**

import pandas as pd

!ls

COVID-india-statewise.csv sample\_data

**Explanation:**

1. **import pandas as pd**: This line imports the Pandas library and assigns it the alias 'pd.' Pandas is a powerful data manipulation library used for working with structured data, particularly in tabular form (like spreadsheets). By aliasing it as 'pd,' you can use 'pd' as a shorthand to access Pandas functions in your code.
2. **!ls**: This is not a Python command but a shell command (commonly used in Unix/Linux-based systems). It's used to list the files and directories in the current directory. In a Jupyter Notebook or IPython environment, you can use **!** to run shell commands from within your Python environment.

**Viewing the First Rows of a DataFrame**:

A screenshot of a computer

Description automatically generateddf.head

**Calculation of Mean:**

A screenshot of a computer

Description automatically generateddf.describe()

**Graphs:**

**Distributions**

A graph of a vaccination

Description automatically generated

A graph with blue squares

Description automatically generated

A graph with a blue rectangle

Description automatically generated

A graph with blue squares

Description automatically generated

**2-d distributions:**

A graph of a vaccination

Description automatically generated

**2-d distributions:**

A graph with blue dots

Description automatically generated

A graph with blue dots

Description automatically generated

**A graph of a vaccination

Description automatically generatedValues :**

**A graph with a line and numbers

Description automatically generated**

A graph with a line and numbers

Description automatically generated

A graph with a line and text

Description automatically generated

A graph of a vaccination

Description automatically generated

**Minimum Members Vaccinated:**

A screen shot of a computer

Description automatically generateddf.min()

**Maximum Members Vaccinated:**

df.max()

A screenshot of a computer

Description automatically generated

**Analysis:**

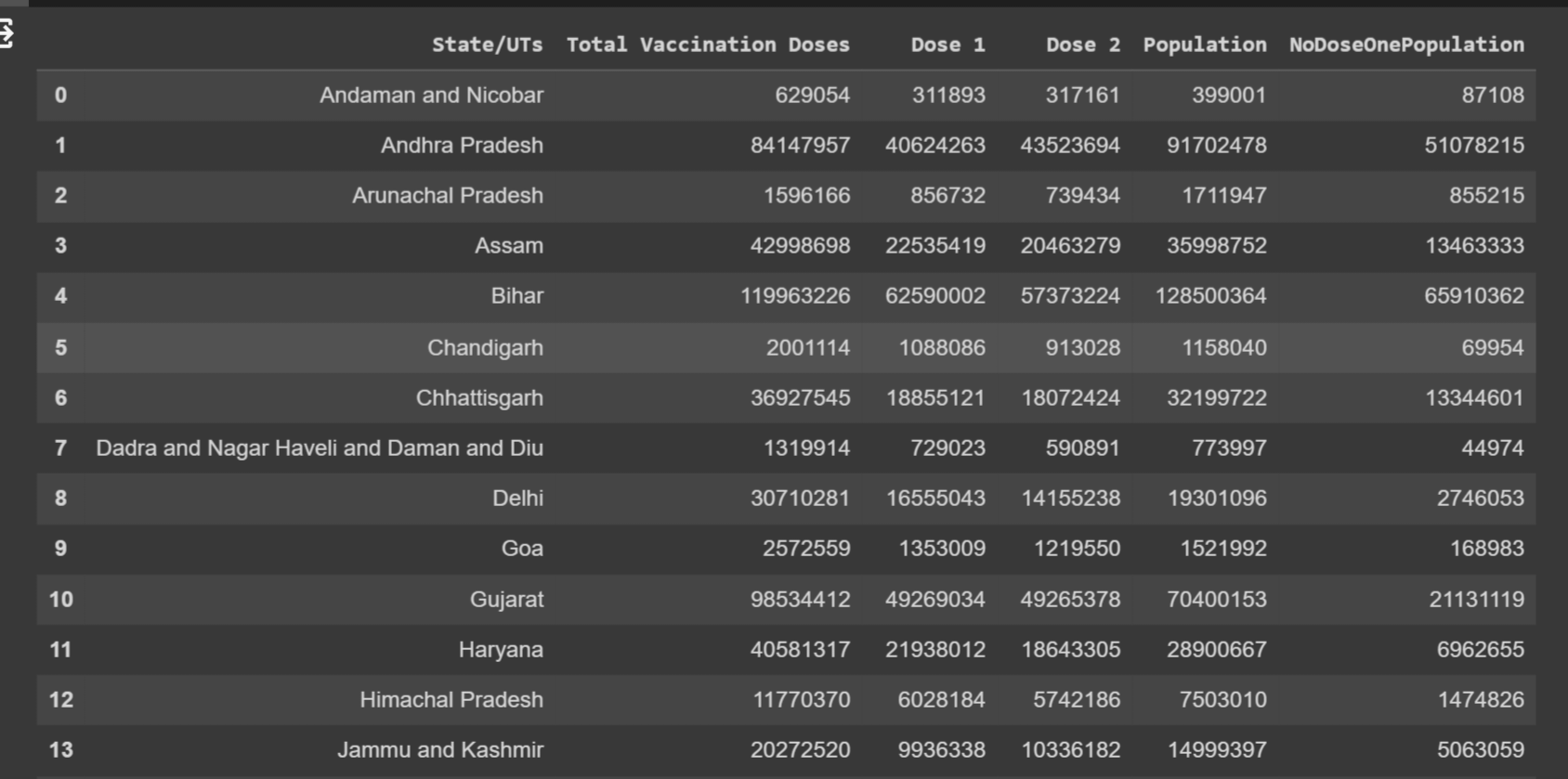
# dose 1 - Not vaccinated

NoDoseOnepopulation = df["Population"] - df["Dose 1"]

df = df.assign(NoDoseOnePopulation=NoDoseOnepopulation)

df

**OUTPUT:**

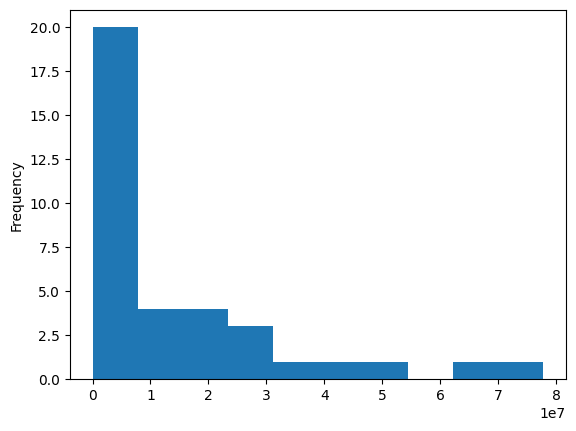
****

**A screenshot of a computer

Description automatically generated**A screenshot of a computer screen

Description automatically generated

<Axes: ylabel='Frequency'>



**Dose 2 - Not vaccinated:**

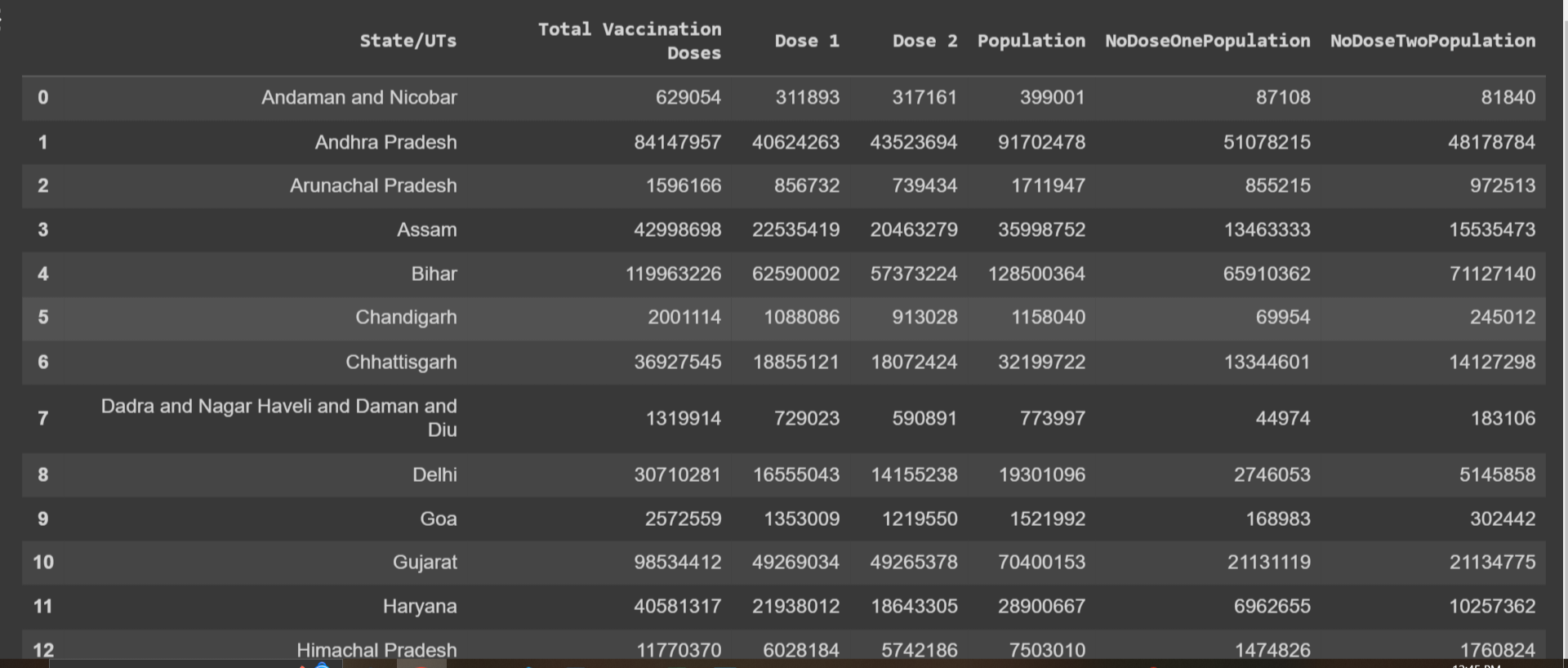
#dose 2 - Not vaccinated

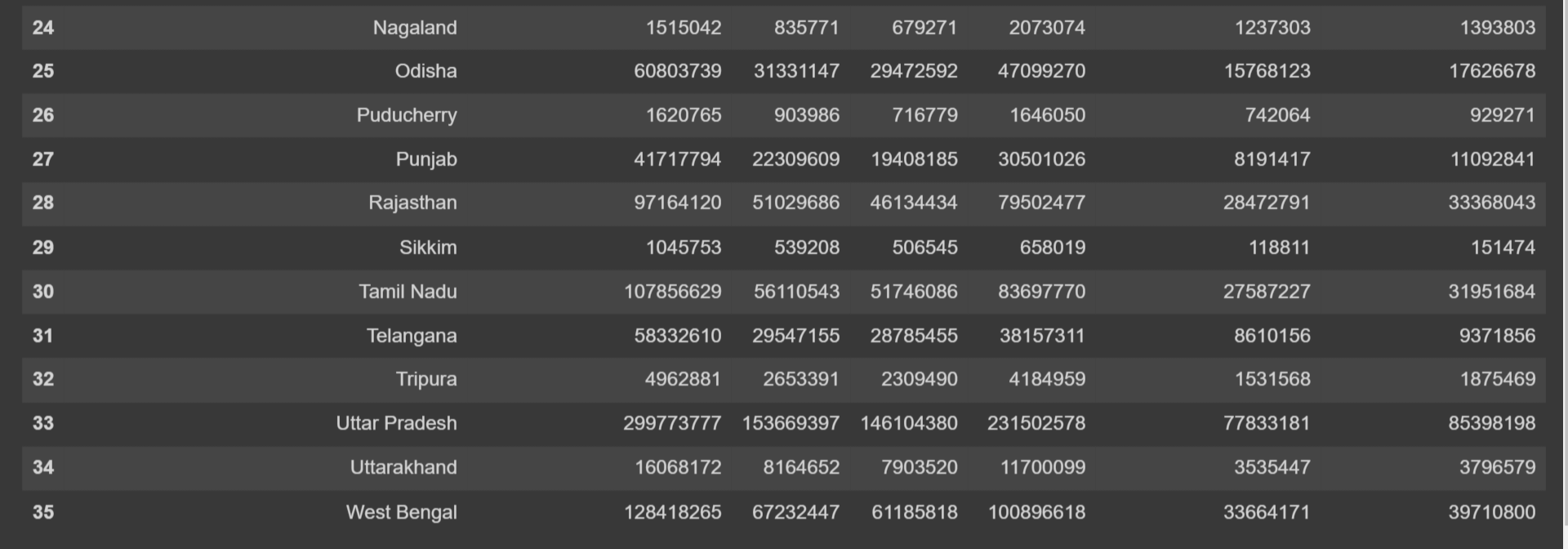
NoDoseTwoPopulation = df["Population"] - df["Dose 2"]

df = df.assign(NoDoseTwoPopulation=NoDoseTwoPopulation)

df

A screenshot of a computer screen

Description automatically generated**OUTPUT**

****

**VISUALIZATION:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

data = pd.read\_csv(r"/content/COVID-19 India Statewise Vaccine Data.csv")

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

sns.barplot(data=data, x='State/UTs', y='Total Vaccination Doses')

plt.xticks(rotation=90)

plt.title('Total Vaccination Doses by State/UTs')

plt.show()

correlation\_matrix = data.corr()

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

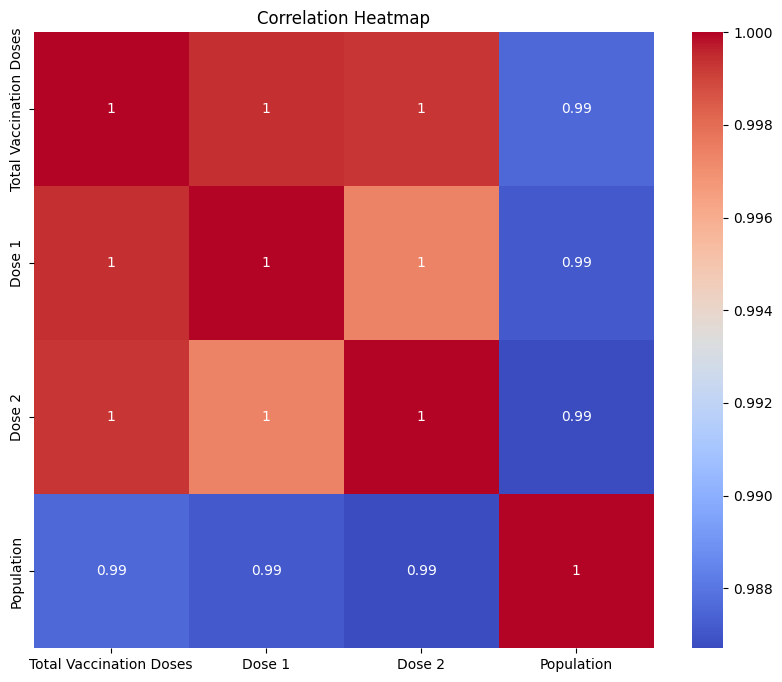
plt.title('Correlation Heatmap')

plt.show()

**OUTPUT:**

A graph of different colored lines

Description automatically generated



**FEATURE ENGINEERING - EXPLORATARY DATA ANALYSIS(EDA):**

import pandas as pd

import matplotlib.pyplot as plt

# Create a Pandas DataFrame with the provided data

data = pd.DataFrame({

    'State/UTs': ['Andaman and Nicobar', 'Andhra Pradesh', 'Arunachal Pradesh', 'Assam', 'Bihar', 'Chandigarh', 'Chhattisgarh', 'Dadra and Nagar Haveli and Daman and Diu', 'Delhi', 'Goa'],

    'Total Vaccination Doses': [629054, 84147957, 1596166, 42998698, 119963226, 2001114, 36927545, 1319914, 30710281, 2572559],

    'Dose 1': [311893, 40624263, 856732, 22535419, 62590002, 1088086, 18855121, 729023, 16555043, 1353009],

    'Dose 2': [317161, 43523694, 739434, 20463279, 57373224, 913028, 18072424, 590891, 14155238, 1219550],

    'Population': [399001, 91702478, 1711947, 35998752, 128500364, 1158040, 32199722, 773997, 19301096, 1521992]

})

# Display the first few rows of the dataset

print(data.head())

# Calculate basic statistics

summary\_statistics = data.describe()

print(summary\_statistics)

# Create a bar plot to visualize Total Vaccination Doses by State/UTs

plt.figure(figsize=(10, 6))

plt.bar(data['State/UTs'], data['Total Vaccination Doses'])

plt.title('Total Vaccination Doses by State/UTs')

plt.xlabel('State/UTs')

plt.ylabel('Total Vaccination Doses')

plt.xticks(rotation=45)

plt.show()

# Create a scatter plot to visualize the relationship between Dose 1 and Dose 2

plt.figure(figsize=(10, 6))

plt.scatter(data['Dose 1'], data['Dose 2'], alpha=0.5)

plt.title('Dose 1 vs. Dose 2')

plt.xlabel('Dose 1')

plt.ylabel('Dose 2')

plt.grid(True)

plt.show()

**OUTPUT:**

State/UTs Total Vaccination Doses Dose 1 Dose 2 \

0 Andaman and Nicobar 629054 311893 317161

1 Andhra Pradesh 84147957 40624263 43523694

2 Arunachal Pradesh 1596166 856732 739434

3 Assam 42998698 22535419 20463279

4 Bihar 119963226 62590002 57373224

Population

0 399001

1 91702478

2 1711947

3 35998752

4 128500364

Total Vaccination Doses Dose 1 Dose 2 Population

count 1.000000e+01 1.000000e+01 1.000000e+01 1.000000e+01

mean 3.228665e+07 1.654986e+07 1.573679e+07 3.132674e+07

std 4.112141e+07 2.100441e+07 2.016206e+07 4.449801e+07

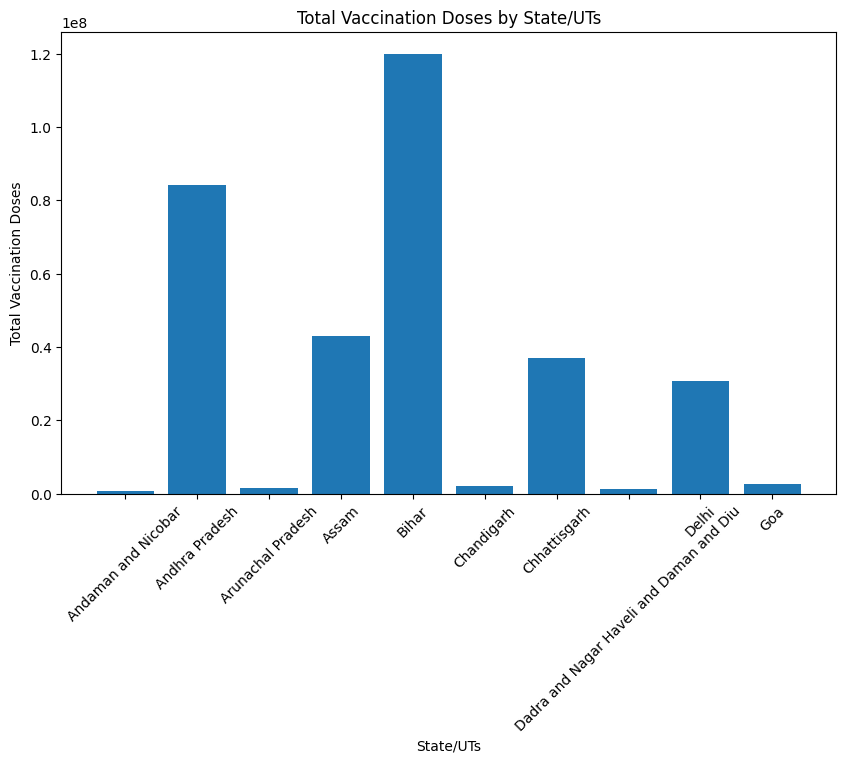
min 6.290540e+05 3.118930e+05 3.171610e+05 3.990010e+05

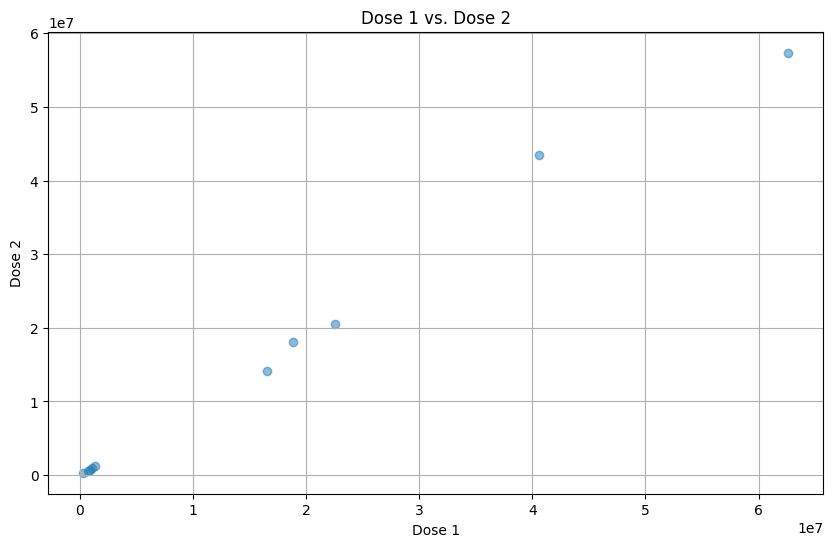
25% 1.697403e+06 9.145705e+05 7.828325e+05 1.249028e+06

50% 1.664142e+07 8.954026e+06 7.687394e+06 1.050652e+07

75% 4.148091e+07 2.161534e+07 1.986557e+07 3.504899e+07

max 1.199632e+08 6.259000e+07 5.737322e+07 1.285004e+08





**PREDICTIVE ANALYSIS:**

* Predictive analysis in data science is a process of using historical data to make informed predictions or forecasts about future events or outcomes.

**PROGRAM:**

import pandas as pd

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

import matplotlib.pyplot as plt

# Load the dataset

df = pd.read\_csv(r"/content/COVID-19 India Statewise Vaccine Data.csv")

# Select relevant features

X = df[['Dose 1', 'Dose 2']]

y = df['Total Vaccination Doses']

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create a linear regression model

model = LinearRegression()

# Train the model on the training data

model.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = model.predict(X\_test)

# Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

# Visualize the results

plt.scatter(y\_test, y\_pred)

plt.xlabel("Actual Total Vaccination Doses")

plt.ylabel("Predicted Total Vaccination Doses")

plt.title("Actual vs. Predicted Total Vaccination Doses")

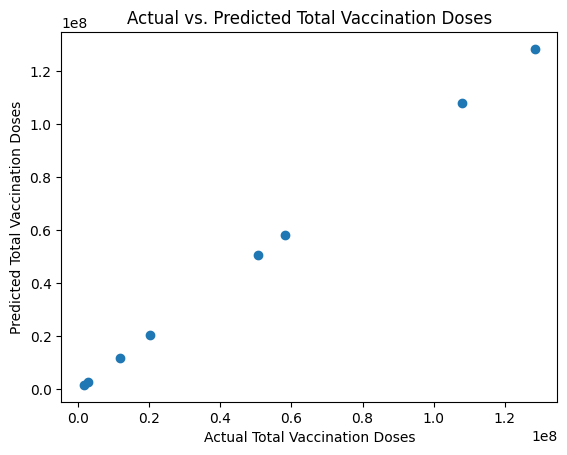
plt.show()

print("Mean Absolute Error:", mae)

print("Mean Squared Error:", mse)

print("R-squared (R^2):", r2)

**OUTPUT:**



Mean Absolute Error: 1.3300450518727303e-08

Mean Squared Error: 2.248160967284474e-16

R-squared (R^2): 1.0

**Conclusion:**

In the fight against the COVID-19 pandemic, a comprehensive and data-driven approach is indispensable. The project of COVID-19 Vaccine Analysis presented here plays a vital role in monitoring and assessing vaccination campaigns, ultimately contributing to the protection of public health. the analysis of the 'State-wise India COVID-19 Vaccination' dataset provided valuable insights into the vaccination efforts across different states and union territories in India. Key statistics, including the mean number of vaccination doses administered, the total count of records, and the states with the minimum and maximum members vaccinated, were calculated. These statistics shed light on the variation in vaccination coverage across regions, offering essential information for assessing and planning public health strategies. This initial analysis serves as a foundational step for further in-depth exploration and data-driven decision-making in the fight against COVID-19.