IBM NAAN MUDHALVAN

APPLIED DATA SCIENCE

COVID-19 VACCINE ANALYSIS

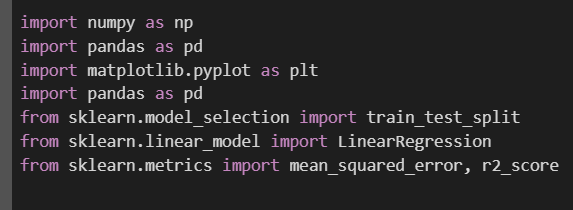
PHASE – 4 PROJECT SUBMISSION

|  |  |
| --- | --- |
| DOMAIN | APPLIED DATA SCIENCE |
| PROJECT TOPIC | COVID-19 VACCINE ANALYSES |
| TEAM MEMBER  AND  REGISTER NUMBER | BALAJI S (420421104009)  VIGNESH K (420421104085)  DEEPAK S (420421104013)  AJITH K (420421104004) |

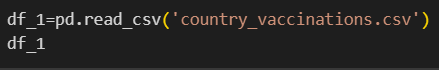
Dataset link: <https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress>

Program:

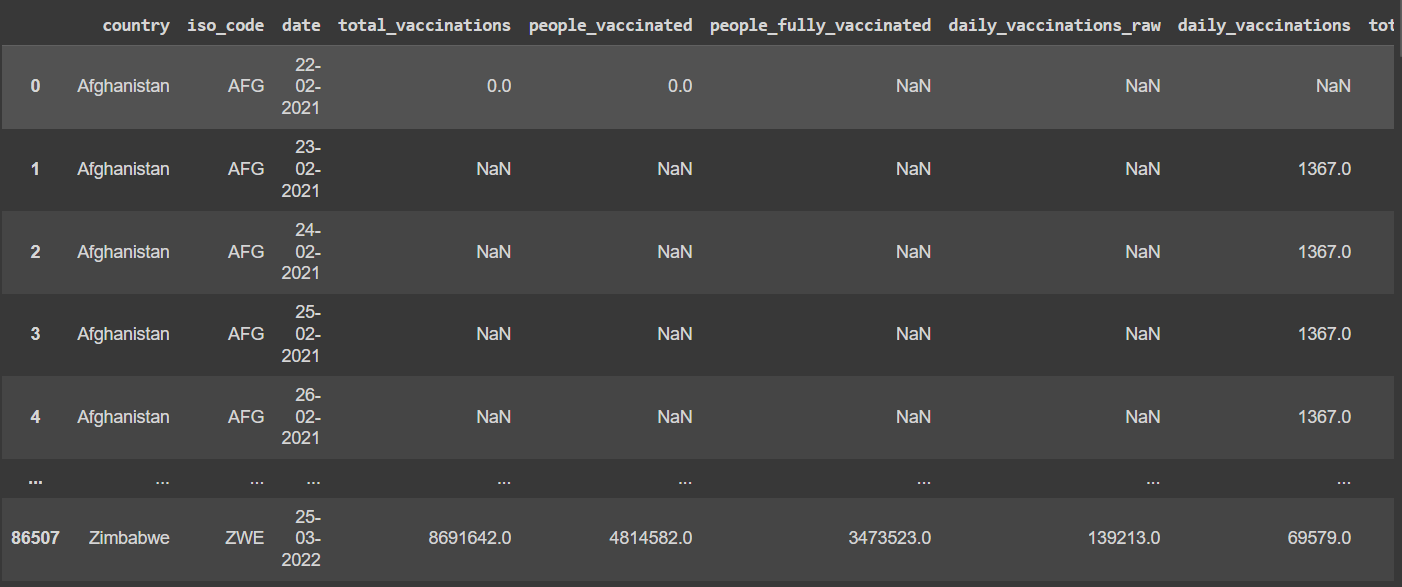
STEP 1: Import the libraries.



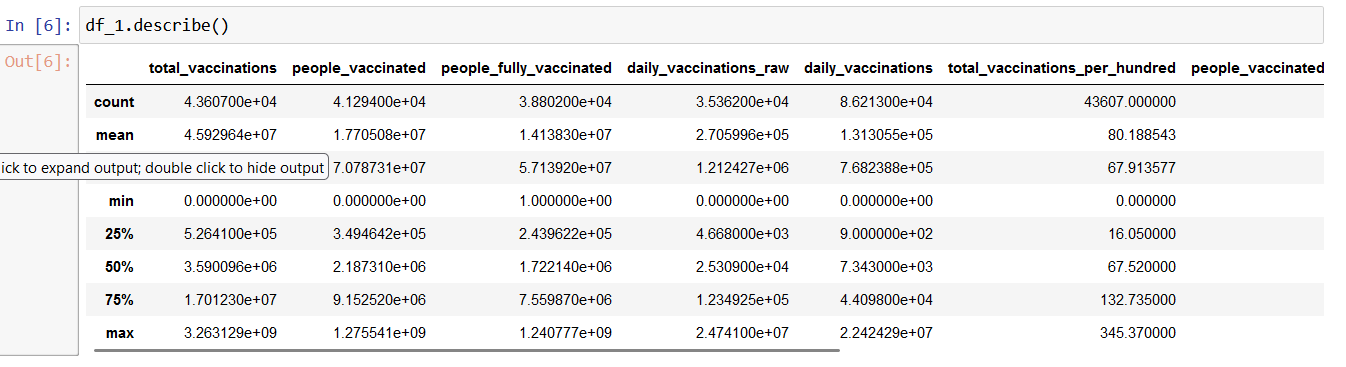
STEP 2: Load the datasets:



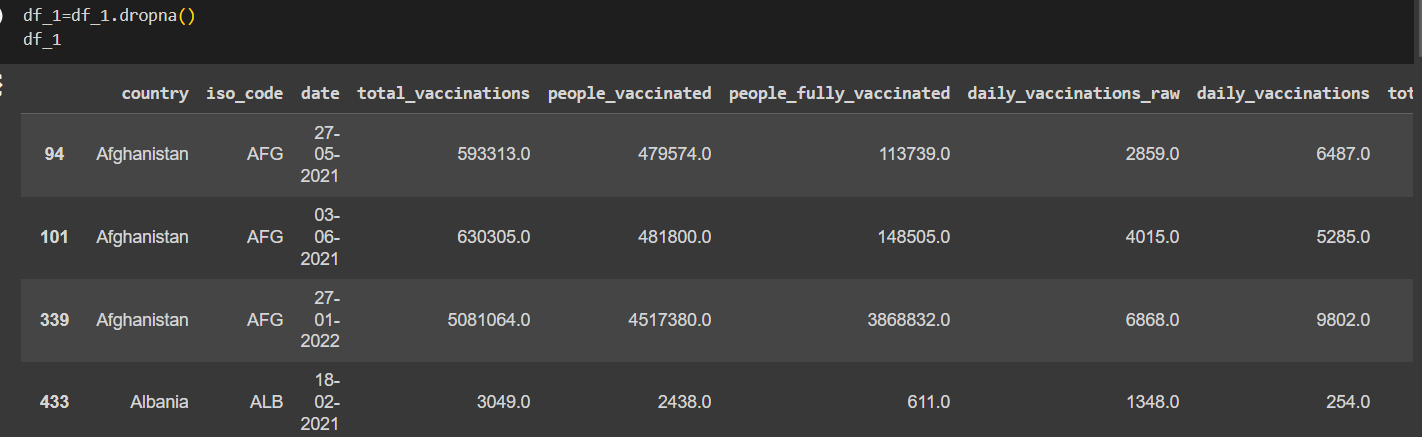
Output:



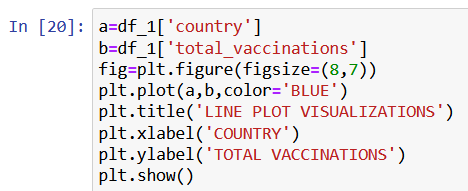
STEP 3: Describe the datasets.



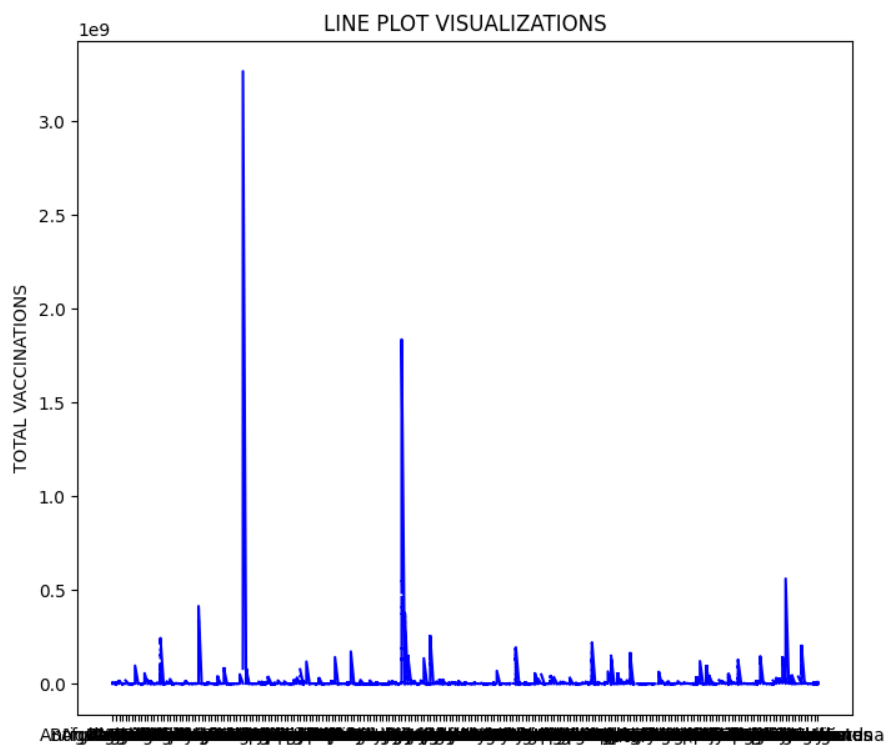
STEP 4: Data cleaning.

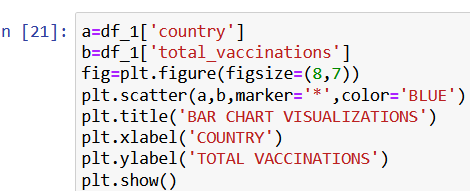


STEP 5: Data visualization.

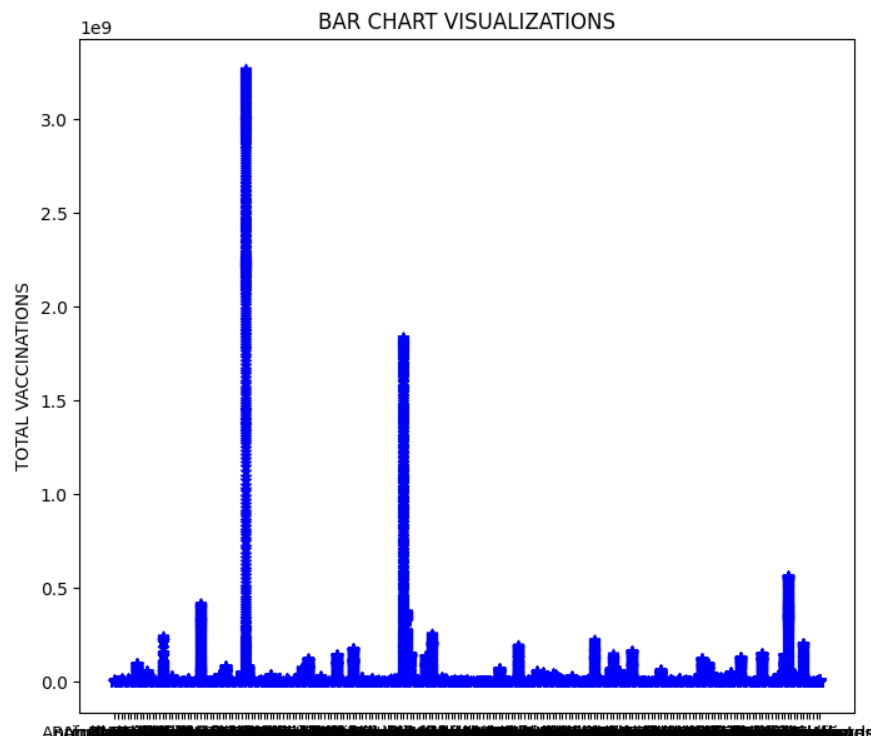


Output:





Output:



STEP 6: Training and evaluation of the datasets by linear regression models.

# Define features (X) and target (y)

X = df\_1[['total\_vaccinations', 'daily\_vaccinations']]  # Specify the relevant features

y = df\_1['daily\_vaccinations\_per\_million']  # The target variable you want to predict

# 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

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

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

print("Mean Squared Error:", mse)

print("R-squared (R2) Score:", r2)

# You can save the trained model for later use

# Example: save the model to a file

from joblib import dump

dump(model, 'covid\_model.joblib')

output:

