**Title: COVID-19 Data Analysis and Prediction**

**Abstract:**

The COVID-19 pandemic has significantly impacted global health and economies, and the analysis of COVID-19 data is crucial for understanding trends, making predictions, and shaping public health responses. This project focuses on the analysis and prediction of COVID-19 infection rates, vaccination progress, and public health responses using real-world datasets. The primary objectives of the project are to explore infection trends, predict future cases, perform correlation analysis between various factors such as infections, deaths, and vaccinations, and use clustering techniques to group countries based on their COVID-19 response metrics.

The project is built using Python and several libraries such as Pandas for data manipulation, Matplotlib and Seaborn for data visualization, and Scikit-learn for machine learning tasks. The analysis involves:

1. **Data Preprocessing**: Cleaning the dataset to handle missing values, convert date columns to the appropriate format, and handle any inconsistencies in the data.
2. **Correlation Analysis**: Evaluating the relationships between key factors like total cases, total deaths, total vaccinations, GDP, and population across countries using a correlation matrix.
3. **Trend Analysis**: Visualizing trends over time using moving averages to understand the progression of cases, deaths, and vaccinations.
4. **Prediction Modeling**: Employing Linear Regression to predict future COVID-19 cases based on historical data, evaluating the model's performance using Mean Absolute Error (MAE).
5. **Clustering Countries**: Using K-Means clustering to group countries based on their COVID-19 infection rates, vaccination coverage, and death rates to identify patterns and groupings in response strategies.

The project aims to provide insights into the effectiveness of vaccination campaigns, the correlation between GDP and infection rates, and to forecast the potential future trajectory of the pandemic. The findings from this analysis can assist public health authorities and decision-makers in managing ongoing and future public health crises more effectively.

The project leverages synthetic data for demonstration purposes and is adaptable to real-world datasets, offering a flexible and scalable approach to pandemic data analysis. Through this work, we aim to showcase the potential of data science and machine learning in solving complex public health challenges.

**Code:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from sklearn.linear\_model import LinearRegression

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

from sklearn.metrics import mean\_absolute\_error

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

# Load COVID-19 datasets

def load\_data():

    url = 'synthetic\_covid19\_data.csv'  # Replace this with your local CSV file if using locally

    data = pd.read\_csv(url)

    return data

# Data Preprocessing and cleaning

def preprocess\_data(df):

    df['date'] = pd.to\_datetime(df['date'], errors='coerce')

    df.dropna(subset=['total\_cases', 'total\_deaths', 'total\_vaccinations'], inplace=True)

    return df

# Correlation analysis

def correlation\_analysis(df):

    df\_corr = df[['total\_cases', 'total\_deaths', 'total\_vaccinations', 'total\_population', 'gdp']].dropna()

    correlation\_matrix = df\_corr.corr()

    plt.figure(figsize=(10, 8))

    sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

    plt.title("Correlation Matrix of COVID-19 Data")

    plt.show()

# Trend Analysis using Moving Averages

def plot\_moving\_average\_trends(df, country='World', window=7):

    country\_data = df[df['location'] == country]

    country\_data['moving\_avg\_cases'] = country\_data['total\_cases'].rolling(window=window).mean()

    country\_data['moving\_avg\_deaths'] = country\_data['total\_deaths'].rolling(window=window).mean()

    country\_data['moving\_avg\_vaccinations'] = country\_data['total\_vaccinations'].rolling(window=window).mean()

    plt.figure(figsize=(12,6))

    plt.plot(country\_data['date'], country\_data['moving\_avg\_cases'], label='7-Day Moving Average (Cases)', color='blue')

    plt.plot(country\_data['date'], country\_data['moving\_avg\_deaths'], label='7-Day Moving Average (Deaths)', color='red')

    plt.plot(country\_data['date'], country\_data['moving\_avg\_vaccinations'], label='7-Day Moving Average (Vaccinations)', color='green')

    plt.title(f"COVID-19 Moving Average Trends Over Time for {country}")

    plt.xlabel('Date')

    plt.ylabel('Count')

    plt.legend()

    plt.xticks(rotation=45)

    plt.show()

# Predict COVID-19 cases using Linear Regression

def predict\_covid\_cases(df, country='United States'):

    country\_data = df[df['location'] == country]

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

    country\_data['date\_ordinal'] = country\_data['date'].apply(lambda x: x.toordinal())  # Convert date to ordinal

    X = country\_data[['date\_ordinal']]

    y = country\_data['total\_cases']

    X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, shuffle=False)

    model = LinearRegression()

    model.fit(X\_train, y\_train)

    y\_pred = model.predict(X\_test)

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

    print(f"Mean Absolute Error: {mae}")

    plt.figure(figsize=(12,6))

    plt.plot(country\_data['date'], country\_data['total\_cases'], label='Actual Cases', color='blue')

    plt.plot(country\_data['date'].iloc[len(X\_train):], y\_pred, label='Predicted Cases', color='red')

    plt.title(f"COVID-19 Cases Prediction for {country}")

    plt.xlabel('Date')

    plt.ylabel('Total Cases')

    plt.legend()

    plt.xticks(rotation=45)

    plt.show()

# K-Means Clustering of countries based on infection/vaccination rates

def cluster\_countries(df):

    country\_data = df[['location', 'total\_cases', 'total\_vaccinations', 'total\_deaths', 'total\_population']].dropna()

    scaler = StandardScaler()

    scaled\_data = scaler.fit\_transform(country\_data[['total\_cases', 'total\_vaccinations', 'total\_deaths']])

    kmeans = KMeans(n\_clusters=3, random\_state=42)

    country\_data['Cluster'] = kmeans.fit\_predict(scaled\_data)

    plt.figure(figsize=(10,6))

    sns.scatterplot(x='total\_cases', y='total\_vaccinations', hue='Cluster', data=country\_data, palette='Set1')

    plt.title("Clustering of Countries Based on COVID-19 Data")

    plt.xlabel('Total Cases')

    plt.ylabel('Total Vaccinations')

    plt.show()

    return country\_data[['location', 'Cluster']]

# Main function to run the analysis

def main():

    df = load\_data()  # Load dataset

    df = preprocess\_data(df)  # Preprocess the data

    # Perform Analysis

    correlation\_analysis(df)  # Perform correlation analysis

    plot\_moving\_average\_trends(df, country='United States')  # Plot trends for a specific country

    predict\_covid\_cases(df, country='United States')  # Predict COVID-19 cases for a country

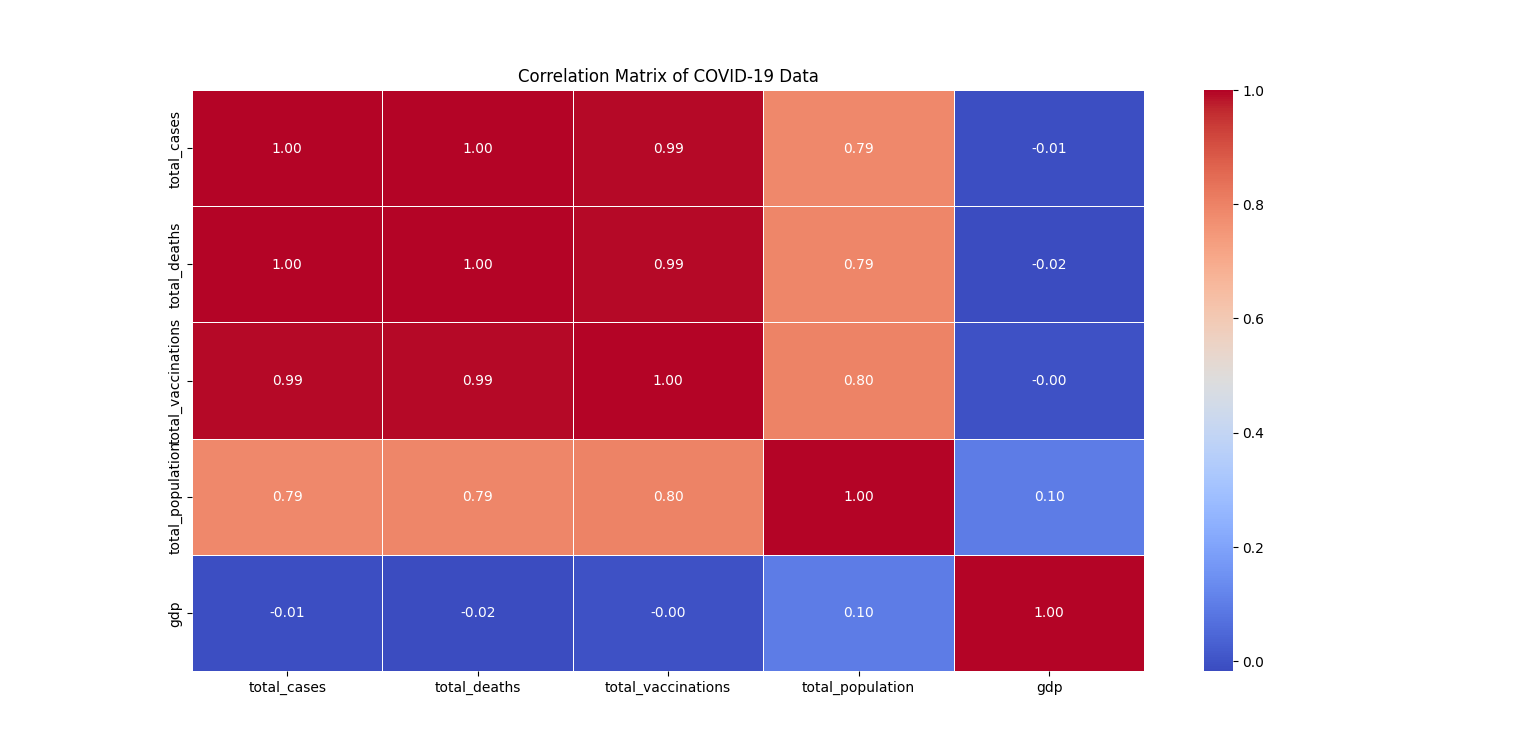
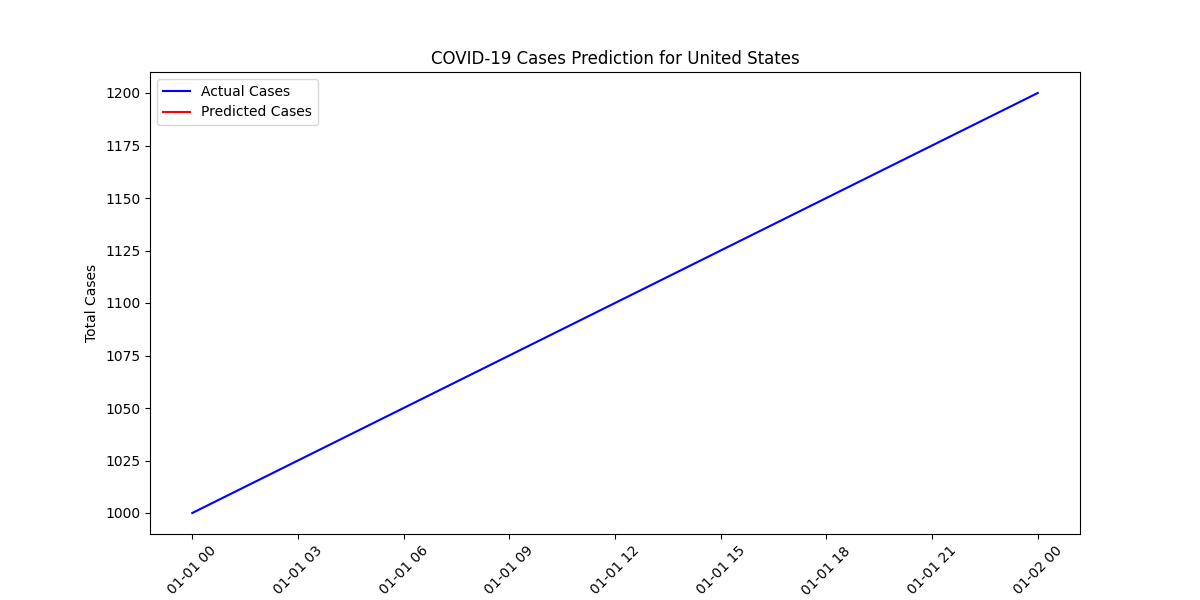
    clustered\_data = cluster\_countries(df)  # Perform K-Means clustering

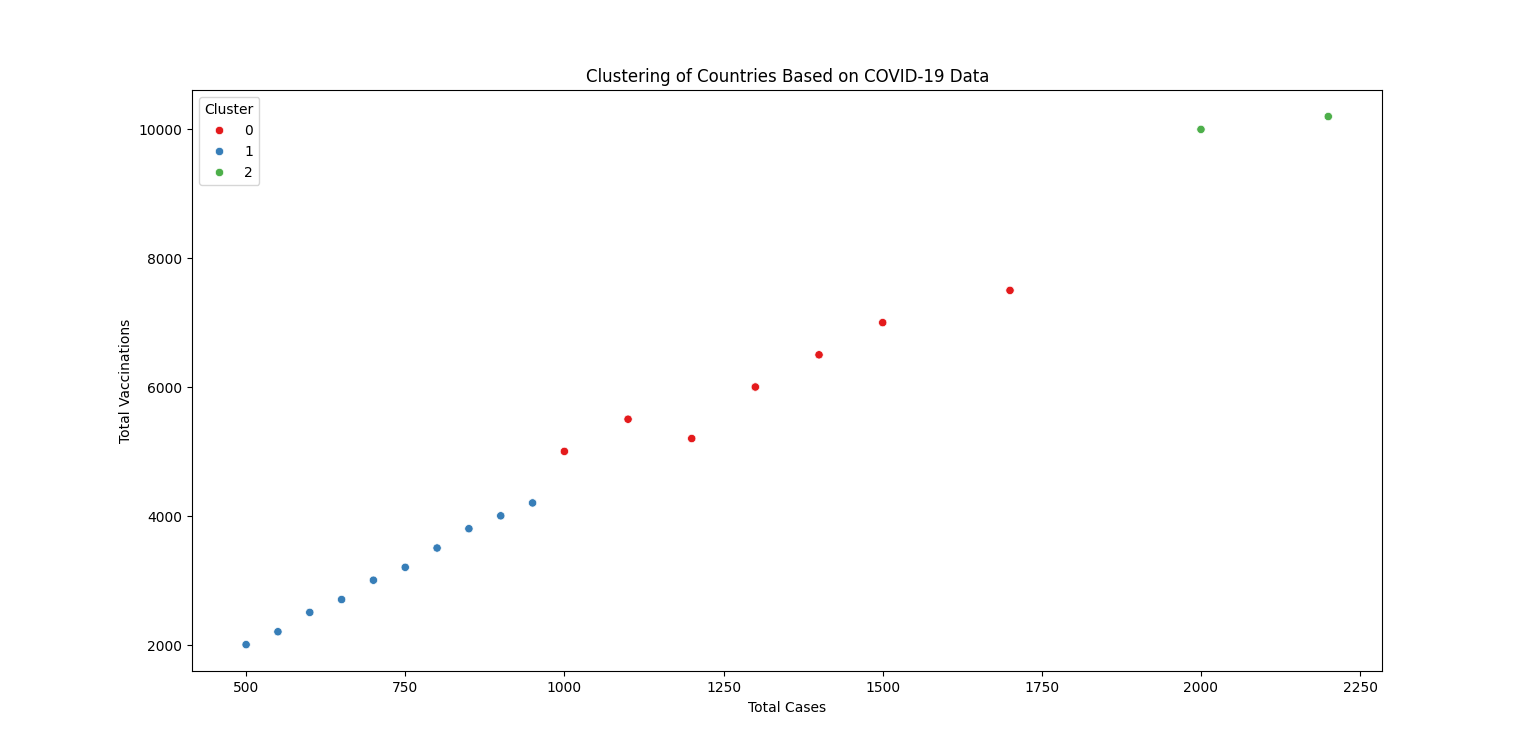
    print(clustered\_data.head())  # Display clustering results

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Output:**

****

****