# NAAN MUDHALVAN PROJECT

**PROJECT NAME:** PHASE-5 DOCUMENT

PART- I . . .

**TOPIC:** COVID-19 CASES ANALYSIS.

| TEAM MEMBERS: |
|---|
| 812621104061; MADESH.K |
| 812621104063; MANIKANDAN.R |
| 812621104072; MUGILRAJ.S |
| 812621104074; NALLENDIRAN.B |
| 812621104077; NAVANEETHAN.S |

# COVID-19 CASES ANALYSIS

**Introduction to COVID-19 Cases Analysis:**

The COVID-19 cases dataset is a collection of data on confirmed COVID-19 cases, deaths, and recoveries, as well as other relevant information such as population, latitude, and longitude. The dataset is divided into five phases:

1. **Problem Definition and Design Thinking:** In this phase, the researchers identified the problem of marginalization and defined the goals of the project. This involved understanding the needs of the target users and the challenges they face.
2. **Innovation:** In this phase, the researchers began building the project by loading and preprocessing the dataset. This involved cleaning the data, removing outliers, and transforming the data into a format that could be easily used by machine learning models.
3. **Development Part 1:** In this phase, the researchers continued building the project by performing different activities like feature engineering, model training, and evaluation. Feature engineering is the process of creating new features from existing data. This can be done to improve the performance of machine learning models. Model training is the process of teaching a machine learning model to predict a target variable based on a set of input variables. Model evaluation is the process of assessing the performance of a machine learning model on a held-out test set.
4. **Development Part 2:** In this phase, the researchers continued building the project by fine-tuning the machine learning models and developing a user-friendly interface.
5. **Project Documentation & Submission:** In this phase, the researchers documented the project and submitted it to a relevant repository.

The COVID-19 cases dataset can be used for a variety of purposes, such as:

- Tracking the spread of COVID-19 over time and space
- Identifying areas that are at high risk of COVID-19 transmission
- Developing and evaluating public health interventions to prevent and control COVID-19
- Forecasting the future course of the COVID-19 pandemic

The dataset is a valuable resource for researchers, policymakers, and the public. It can help to improve our understanding of COVID-19 and develop effective strategies to combat the pandemic.

## Example use cases

Here are some example use cases for the COVID-19 cases dataset:

- A researcher could use the dataset to investigate the relationship between COVID-19 cases and population density. This could help to identify areas that are at high risk of COVID-19 transmission and develop targeted public health interventions.

# PHASE:1

## Problem Definition and Design Thinking

In this part you will need to understand the problem statement and create a document on what have you understood and how will you proceed ahead with solving the problem. Please think on a design and present in form of the document

Project 7: COVID-19 Using Cognos

Project Description: The project involves conducting a comprehensive analysis of COVID-19 cases and deaths data in the European Union (EU) and European Economic Area (EEA) countries. The primary objective is to compare and contrast the mean values and standard deviations of COVID-19 cases and associated deaths on a daily basis and by country within the EU/EEA region. To achieve this, the project encompasses several key tasks and objectives:

## 1. Define Analysis Objectives:

- Clearly define the specific objectives and research questions that the analysis aims to address.
- Determine the key metrics and statistical measures (mean, standard deviation) to be used in the analysis.

## 1. Data Collection:

- Gather COVID-19 cases and deaths data from reliable sources such as national health authorities, the World Health Organization (WHO), or other reputable data providers.
- Ensure the data is up-to-date, accurate, and consistent across all selected EU/EEA countries.
- Organize and format the data for analysis in IBM Cognos.

## 1. Data Exploration:

- Perform initial exploratory data analysis to understand the data's distribution, outliers, and trends.
- Identify any missing or incomplete data and address data quality issues as necessary.

## 1. Design Relevant Visualizations in IBM Cognos:

- Utilize IBM Cognos or other appropriate data visualization tools to create meaningful charts, graphs, and dashboards.
- Visualize daily COVID-19 cases and deaths trends for each EU/EEA country.
- Generate comparative visualizations to highlight differences in mean values and standard deviations among countries.

## 1. Statistical Analysis:

- Calculate and compare the mean values and standard deviations of daily COVID-19 cases and deaths within the EU/EEA region.
- Conduct country-level analyses to identify variations and patterns.

- Employ statistical tests if necessary to determine the significance of observed differences.

1. **Derive Insights and Recommendations:**

   - Interpret the findings from the analysis to derive actionable insights.
   - Identify factors that may explain variations in COVID-19 cases and deaths.
   - Formulate recommendations for public health interventions or further research if applicable.

1. **Report and Presentation:**

   - Prepare a comprehensive report summarizing the analysis, including key findings, insights, and recommendations.
   - Create visually engaging presentations to effectively communicate the results to stakeholders.

1. **Trend Analysis:** Determine the overall trend of COVID-19 cases and deaths in the EU/EEA region over time.
2. **Comparative Analysis:** Compare the mean values and standard deviations of cases and deaths across different countries in the EU/EEA.
3. **Identify Outliers:** Detect any outliers or significant deviations from the mean values.
4. **Impact Assessment:** Assess the impact of various measures or interventions by comparing changes in mean values and standard deviations before and after their implementation.
5. **Forecasting:** Use historical data to make predictions about future COVID-19 cases and deaths in the EU/EEA.

Data Collection:

1. **Data Source:** Obtain the COVID-19 cases and deaths data from reliable sources like the World Health Organization (WHO), European Centre for Disease Prevention and Control (ECDC), or national health authorities.
2. **Data Preprocessing:** Clean and organize the data to ensure accuracy and consistency. Handle missing values and inconsistencies in reporting.

Visualization Strategy:

1. **Line Charts:** Create time series line charts to visualize the trend in daily COVID-19 cases and deaths in the EU/EEA over

a specified period.

2. **Bar Charts:** Use bar charts to compare the mean values of cases and deaths for different countries in the EU/EEA. Youcan also create bar charts to show the standard deviations.

3. **Heatmaps:** Generate heatmaps to visually represent the variation in COVID-19 cases and deaths across countries in the EU/EEA.

4. **Box Plots:** Use box plots to display the distribution of cases and deaths, including outliers, for a
more detailed view of the data.

5. **Overlay Charts:** Overlay line charts of mean values and standard deviations to identify patterns and variations.

Insights Generation:

1. **Identify Hotspots:** Look for countries with consistently high mean values and standard deviations, indicating regions with a severe and fluctuating COVID-19 situation.

2. **Temporal Patterns:** Analyze whether there are any specific time periods when mean values and standard deviations show significant changes, such as spikes or declines.

3. **Outlier Detection:** Investigate outliers to understand why certain countries deviate significantly from the average, which may involve exploring factors like healthcare infrastructure, government policies, or vaccination rates.

# PHASE:2

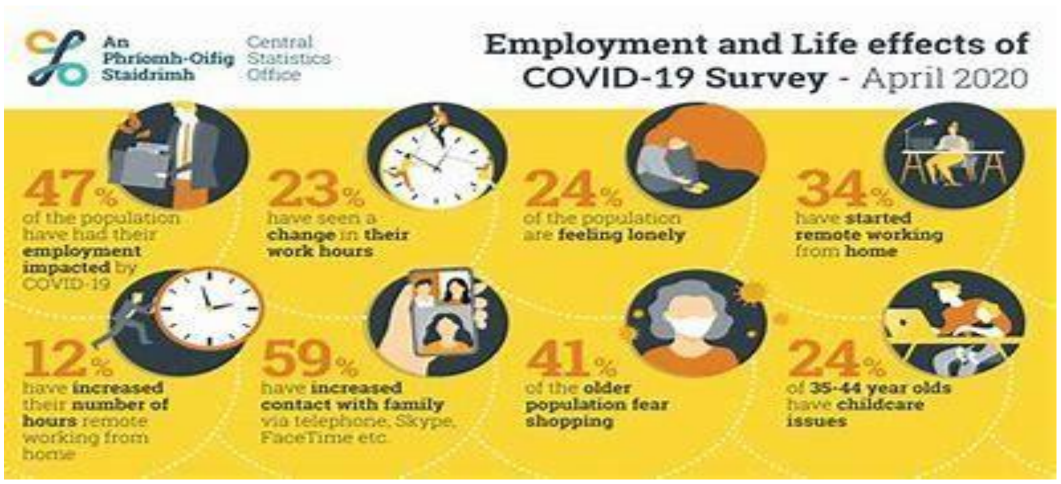
## Innovation

In this section you need to put your design into innovation to solve the problem. Create a doc around it and share the same for assessment.

Project:DataAnalytics

Phase2:Innovation

# Project:Covid-19Analysis

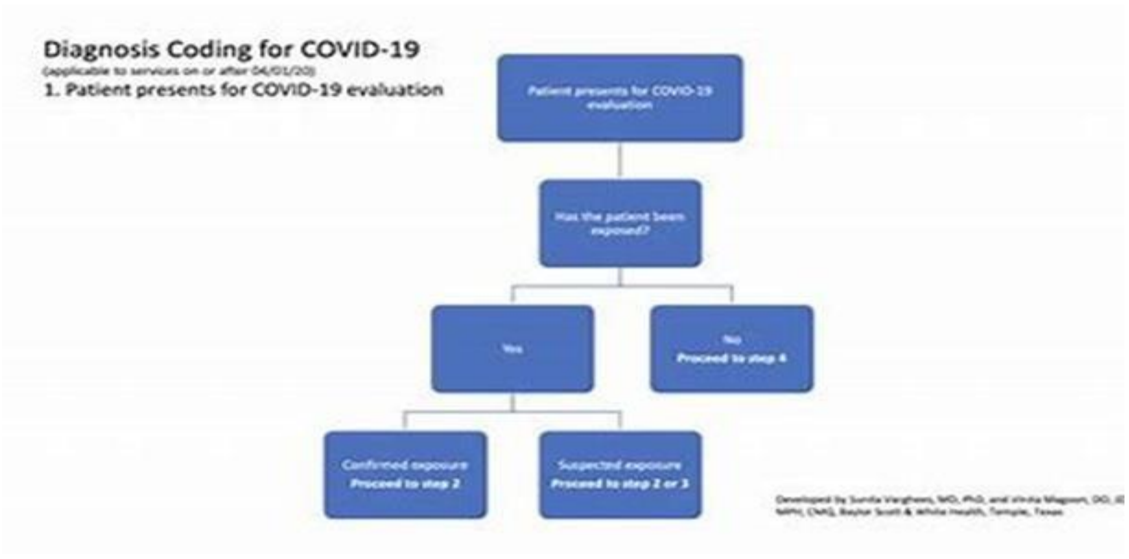


## INTRODUCTION:

COVID-19 analytics refers to the use of data analysis and statistical techniques to gain insights, track, and understand the impact of the COVID-19 pandemic.Analytics plays a crucial role in helping public health officials, researchers, and policy makers make
Informed decisions,allocate resources, and develop strategies to  mitigate the spread of

thevirus.Here's an introduction to key aspects of COVID-19 analytics:

*DataSources:COVID-19 analytics relies on data from various sources, including government health departments,
hospitals, testing centers, and research institutions.These data sources provide information on the number of cases, deaths, recoveries, and more.


Diagnosis Coding for COVID-19
(applicable to services on or after 04/01/20)
1. Patient presents for COVID-19 evaluation
Patient presents for COVID-19 evaluation
Has the patient been exposed?
Yes
No Proceed to step 4
Confirmed exposure Proceed to step 2
Suspected exposure Proceed to step 2 or 3

1. **Vaccines:** The rapid development and distribution of COVID-19 vaccines represent amonumental achievement in the field of medical science. Multiple

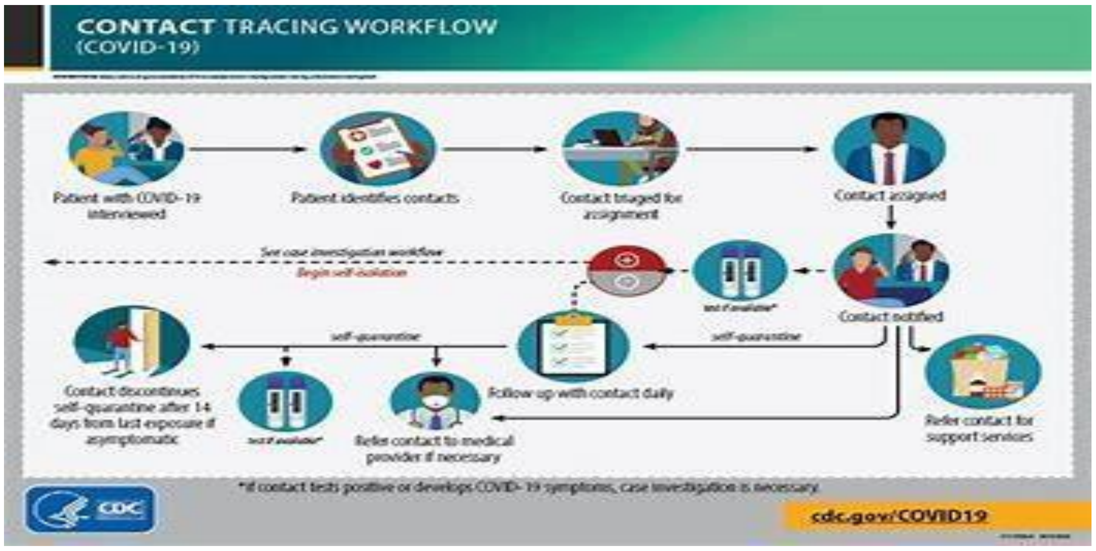
vaccines,such as those from Pfizer,

Moderna,andJohnson&Johnson,were developed in record time using innovativetechnologieslikemRNA vaccines.

2. **Telemedicine:**Thepandemic accelerated the adoption of telemedicine and remote healthcare services.Videoconsultations,remote monitoring devices, and telehealth platformshavebecomevitalinproviding medicalcarewhileminimizingtheriskof virus transmission.

Contactless Technologies: Innovations in contactless technologies, like touchless paymentsystems,voice-activateddevices, and touchless biometrics, have become essentialforreducingphysicalcontactand the potential spread of the virus.

AIandDataAnalytics:Artificialintelligence and data analytics have been instrumental intrackingandpredictingthespreadofthe virus, identifying potential hotspots, and developing treatment strategies.


CONTACT TRACING WORKFLOW
(COVID-19)
Patient with COVID-19 interviewed
Patient identifies contacts
Contact triaged for assignment
Contact assigned
See case investigation workflow
Begin self-isolation
Test if available*
Contact notified
self-quarantine
self-quarantine
Contact discontinues self-quarantine after 14 days from last exposure if asymptomatic
Test if available*
Refer contact to medical provider if necessary
Follow up with contact daily
Refer contact for support services
*If contact tests positive or develops COVID-19 symptoms, case investigation is necessary.
CDC
cdc.gov/COVID19

INTRODUCTION:

1. **Origin**:COVID-19isbelievedtohave originated in bats and was transmitted to humans through an intermediate host, possibly a seafood market in Wuhan, China.

2. **Spread**:Thevirusquicklyspread globally, leading to widespread outbreaks and waves of infection.

3. **Symptoms**:Commonsymptoms included fever, cough, and difficulty breathing, though the virus could manifestinarangeofwaysfrommildto severe.

4. **PreventativeMeasures**:Measuresto prevent the spread of the virus included wearing masks, practicing good hand hygiene, maintaining physical distance, and lockdowns or social restrictions in many places **.**

# PHASE:3

## Development Part 1

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

# COVID-19 CASES ANALYSIS

**Introduction to COVID-19 Cases Analysis**

COVID-19 cases analysis is the process of collecting, cleaning, and analyzing data on COVID-19 cases in order to understand the spread of the disease, identify patterns and trends, and inform public health interventions. COVID-19 cases data can be collected from a variety of sources, including public health departments, hospitals, and testing centers.

COVID-19 cases analysis can be used to answer a variety of questions, such as:

- What are the current trends in the number of COVID-19 cases?
- Where are the hotspots of COVID-19 transmission?
- Who are the most at-risk populations for COVID-19 infection?
- What are the factors that are driving the spread of COVID-19?
- How effective are public health interventions in reducing the number of COVID-19 cases?

COVID-19 cases analysis can be used to inform a variety of public health interventions, such as:

- Targeted testing and vaccination campaigns
- Social distancing and mask mandates
- Travel restrictions
- School closures
- Business closures

COVID-19 cases analysis is an essential tool for understanding and responding to the COVID-19 pandemic. By carefully analyzing COVID-19 cases data, public health officials can identify patterns and trends, identify at-risk populations, and evaluate the effectiveness of public health interventions.

**Examples of COVID-19 Cases Analysis**

Here are some examples of COVID-19 cases analysis:

- A public health department might analyze COVID-19 cases data to identify the neighborhoods in a city with the highest rates of transmission. This information could be used to target testing and vaccination efforts to those neighborhoods.
- A hospital might analyze COVID-19 cases data to identify the characteristics of patients who are most likely to be hospitalized with COVID-19. This information could be used to develop early warning systems for identifying patients who are at risk of severe illness.

- A research team might analyze COVID-19 cases data from multiple countries to identify the factors that are driving the spread of the disease. This information could be used to develop more effective public health interventions.

COVID-19 cases analysis is a complex and rapidly evolving field. However, by carefully analyzing COVID-19 cases data, public health officials and researchers can gain valuable insights into the spread of the disease and inform effective public health interventions.

## CONTENT:

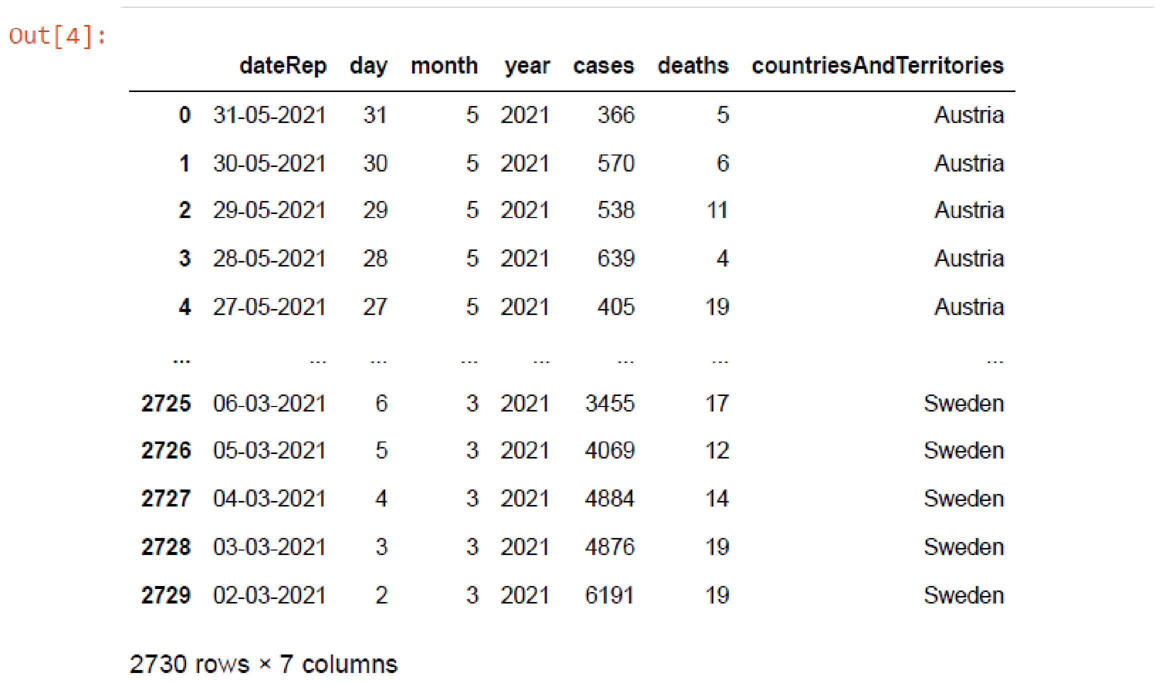
In this technology projects you will begin building your project by loading and preprocessing thedataset. Perform different analysis and visualization using IBM Cognos. After performing therelevant activities create a document around it and share the same for assessment.

## GIVEN DATASET:

https://www.kaggle.com/datasets/chakradharmattapalli/covid-19-cases

## LOAD THE GIVEN DATASET USING PYTHON PROGRAM:

```
import  pandas as pd
dataframe=pd.read_csv("Covid_19_cases4.csv")
dataframe
```



Out[4]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories |
|---|---|---|---|---|---|---|---|
| **0** | 31-05-2021 | 31 | 5 | 2021 | 366 | 5 | Austria |
| **1** | 30-05-2021 | 30 | 5 | 2021 | 570 | 6 | Austria |
| **2** | 29-05-2021 | 29 | 5 | 2021 | 538 | 11 | Austria |
| **3** | 28-05-2021 | 28 | 5 | 2021 | 639 | 4 | Austria |
| **4** | 27-05-2021 | 27 | 5 | 2021 | 405 | 19 | Austria |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **2725** | 06-03-2021 | 6 | 3 | 2021 | 3455 | 17 | Sweden |
| **2726** | 05-03-2021 | 5 | 3 | 2021 | 4069 | 12 | Sweden |
| **2727** | 04-03-2021 | 4 | 3 | 2021 | 4884 | 14 | Sweden |
| **2728** | 03-03-2021 | 3 | 3 | 2021 | 4876 | 19 | Sweden |
| **2729** | 02-03-2021 | 2 | 3 | 2021 | 6191 | 19 | Sweden |

2730 rows × 7 columns

## DATA PREPROCESSING:

Data preprocessing is the process of cleaning, transforming, and organizing raw data to make it suitable for machine learning algorithms. It is an essential step in any machine learning project,as the quality of the preprocessed data directly impacts the performance of the trained model.

### Here are some common data preprocessing steps:

1. **DATA CLEANING:** This involves identifying and correcting errors and inconsistencies in thedata, such as missing values, duplicate records, and typos.
2. **DATA TRANSFORMATION:** This involves converting the data into a format that is compatible with the chosen machine learning algorithm. For example, categorical data may need tobe encoded as numerical data, and features may need to be scaled to a common range.
3. **FEATURE ENGINEERING:** This involves creating new features from the existing data or transforming existing features in a way that makes them more informative for the machine learning algorithm. For example, you might create a new feature that is the ratio of two other features.
4. **DATA SPLITTING:** This involves dividing the preprocessed data into two sets: a training setand a test set. The training set is used to train the machine learning model, and the testset is used to evaluate the performance of the trained model on unseen data.The specific data

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

preprocessing steps that you need to perform will vary depending on the specific machine learning project that you are working on. However, the steps outlined above are a good starting point.

## Here are some additional tips for data preprocessing:

● Understand your data: Before you start preprocessing your data, it is important to understand the nature of the data and the specific machine learning algorithm that you will be using.This will help you to identify the most important data preprocessing steps to perform.

● Use a consistent approach: When preprocessing your data, it is important to use aconsistent approach across all of your data. This will help to ensure that your data is consistent and that your machine learning model is trained on a fair representation of the data.

```python
#Step 1: Import the
necessary libraries#
importing libraries
import pandas as pd
import scipy
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import seaborn as sns
```

```python
# Load the dataset
df = pd.read_csv('Covid_19_cases4.csv')
print(df.head())
```

```
                  dateRep  day  month  year  cases  deaths
            countriesAndTerritories0  31-05-2021         31                 5  2021      3 6
            1  30-05-2021   30      5  2021    570       6
                            Austria
            2  29-05-2021   29      5  2021    538      11
                            Austria
            3  28-05-2021   28      5  2021    639       4
                            Austria
            4  27-05-2021   27      5  2021    405      19
                            Austria
```

```python
#Check the data info
df.info()
```

```
            <class
            'pandas.core.frame.D
            ataFrame'>
            RangeIndex: 2730
            entries, 0 to 2729
            Data columns (total
            7 columns):
             #   Column                   Non-Null Count  Dtype
    ---  -------                  --------------  ------
             0   dateRep                  2730 non-null   object
             1   day                      2730 non-null   int64
             2   month                    2730 non-null   int64
             3   year                     2730 non-null   int64
             4   cases                    2730 non-null   int64
             5   deaths                   2730 non-null   int64
             6   countriesAndTerritories
            2730 non-null                             objectdtypes: int64(5), object 2)
            memory usage: 149.4+ KB
```

```python
#As we can see from the above info that the our dataset has 100 rows and each columns ha
#We can also check the null values using df.isnull()
df.isnull().sum()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

```
Out[4]: dateRep                     0
        day                         0
        month                       0
        year                        0
        cases                       0
        deaths                      0
        countriesAndTerritories     0
        dtype: int64
```

```
In [5]: #Step 3: Statistical Analysis
        #In statistical analysis, first, we use the df.describe() which will give a descriptive
        df.describe()
        #Data  summary
        #The above table shows the count, mean, standard deviation, min, 25%, 50%, 75%, and max
        #Let’s plot the boxplot for each column for easy understanding.
        #0 seconds of 0 secondsVolume 0%
```

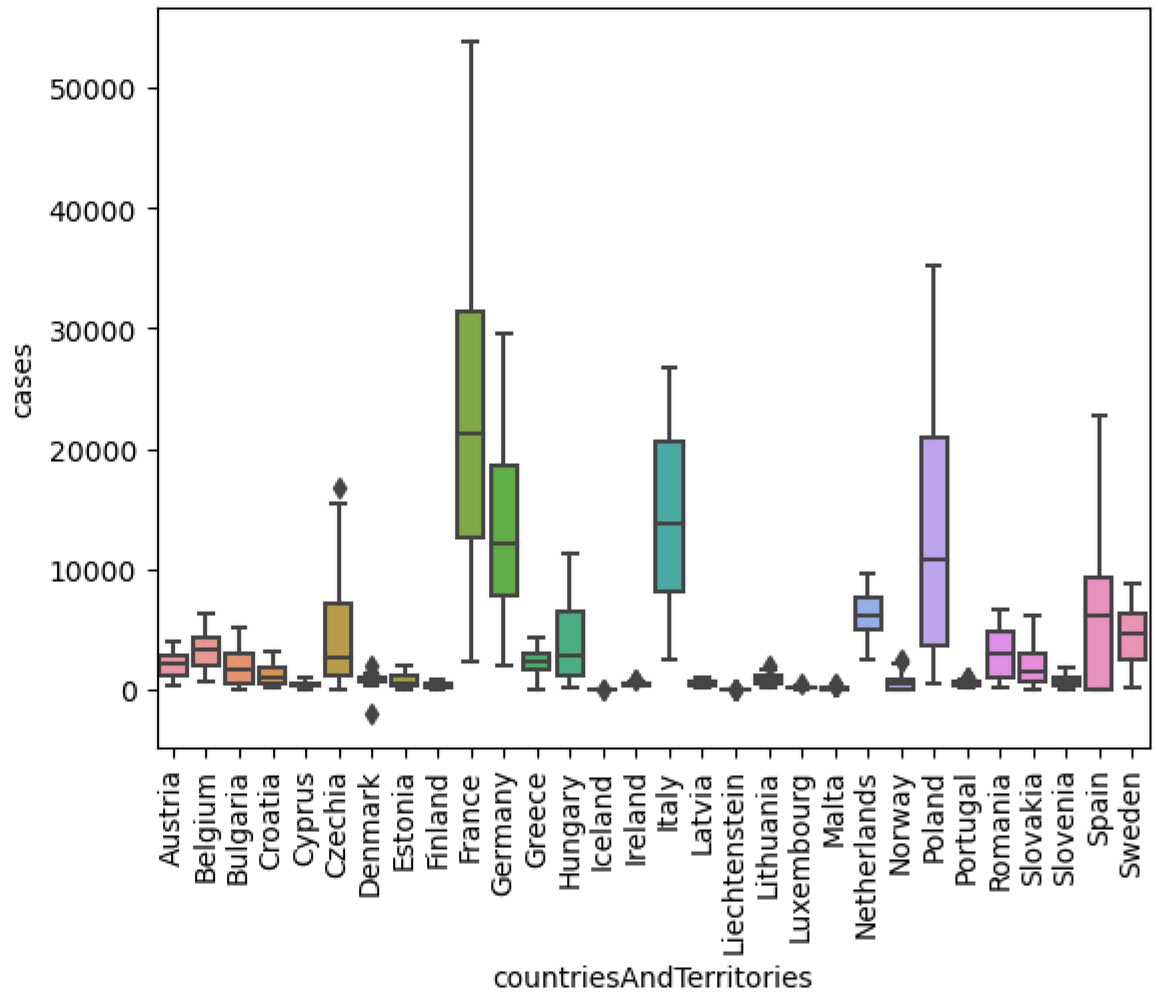
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

Out[5]:

| | day | month | year | cases | deaths |
|---|---|---|---|---|---|
| **count** | 2730.000000 | 2730.000000 | 2730.0 | 2730.000000 | 2730.000000 |
| **mean** | 16.000000 | 4.010989 | 2021.0 | 3661.010989 | 65.291941 |
| **std** | 8.765919 | 0.818813 | 0.0 | 6490.510073 | 113.956634 |
| **min** | 1.000000 | 3.000000 | 2021.0 | -2001.000000 | -3.000000 |
| **25%** | 8.000000 | 3.000000 | 2021.0 | 361.250000 | 2.000000 |
| **50%** | 16.000000 | 4.000000 | 2021.0 | 926.500000 | 14.500000 |
| **75%** | 24.000000 | 5.000000 | 2021.0 | 3916.250000 | 72.000000 |
| **max** | 31.000000 | 5.000000 | 2021.0 | 53843.000000 | 956.000000 |

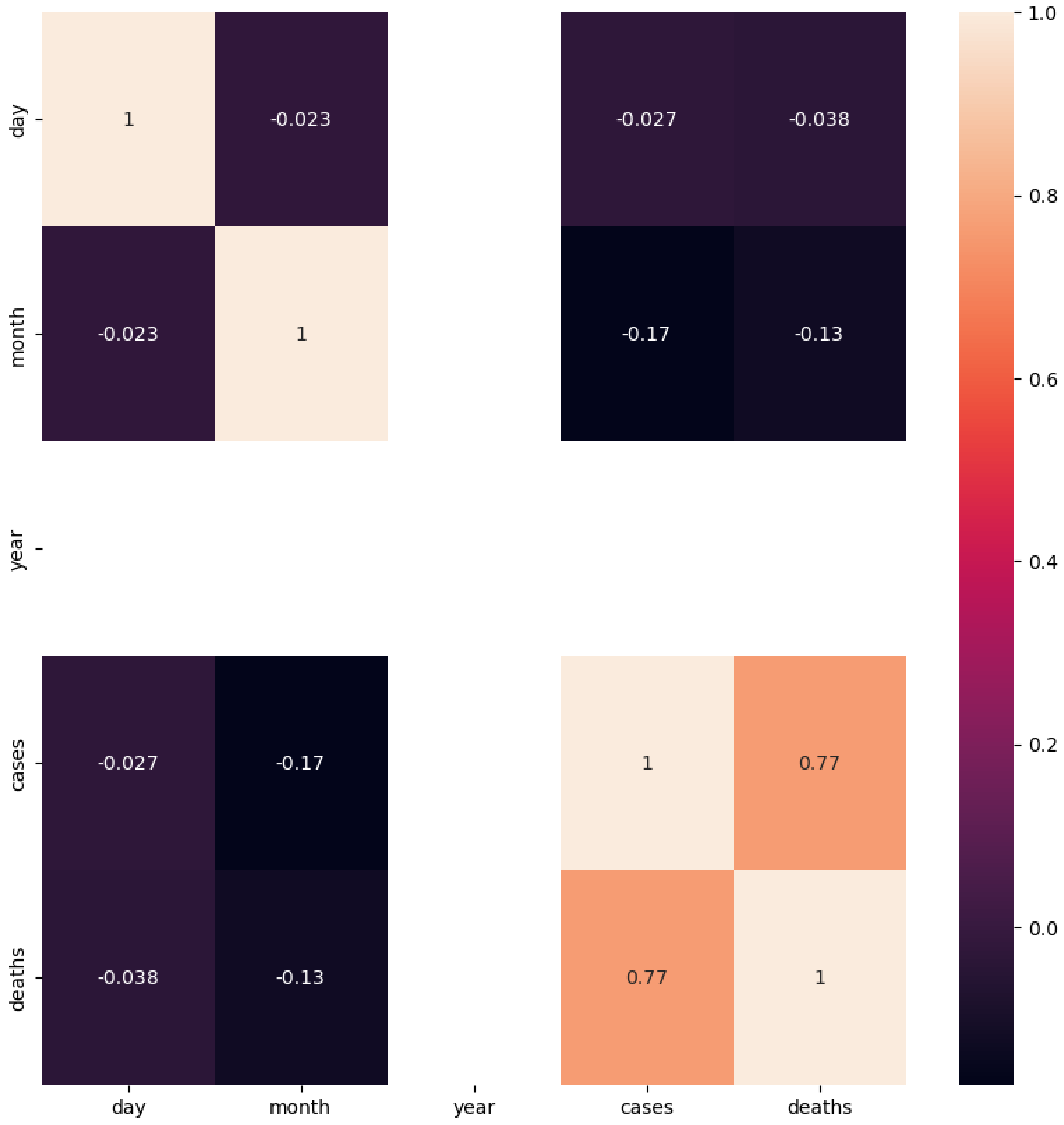
In [9]:
```
#Step 4: Check the outliers:
# Box Plots
sns.boxplot(x="countriesAndTerritories",y="cases",data=df)
plt.xticks(rotation='vertical')
#Boxplots
#from the above boxplot, we can clearly see that all most every column has some amounts
```

Out[9]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29]),
 [Text(0, 0, 'Austria'),
  Text(1, 0, 'Belgium'),
  Text(2, 0, 'Bulgaria'),
  Text(3, 0, 'Croatia'),
  Text(4, 0, 'Cyprus'),
  Text(5, 0, 'Czechia'),
  Text(6, 0, 'Denmark'),
  Text(7, 0, 'Estonia'),
  Text(8, 0, 'Finland'),
  Text(9, 0, 'France'),
  Text(10, 0, 'Germany'),
  Text(11, 0, 'Greece'),
  Text(12, 0, 'Hungary'),
  Text(13, 0, 'Iceland'),
  Text(14, 0, 'Ireland'),
  Text(15, 0, 'Italy'),
  Text(16, 0, 'Latvia'),
  Text(17, 0, 'Liechtenstein'),
  Text(18, 0, 'Lithuania'),
  Text(19, 0, 'Luxembourg'),
  Text(20, 0, 'Malta'),
  Text(21, 0, 'Netherlands'),
  Text(22, 0, 'Norway'),
  Text(23, 0, 'Poland'),
  Text(24, 0, 'Portugal'),
  Text(25, 0, 'Romania'),
  Text(26, 0, 'Slovakia'),
  Text(27, 0, 'Slovenia'),
  Text(28, 0, 'Spain'),
  Text(29, 0, 'Sweden')])
```



```
In [10]: #Step 5: Correlation
         #correlation
         plt.figure(figsize=(10,10))
         sns.heatmap(df.corr(numeric_only=True),annot=True)
         plt.show()
```

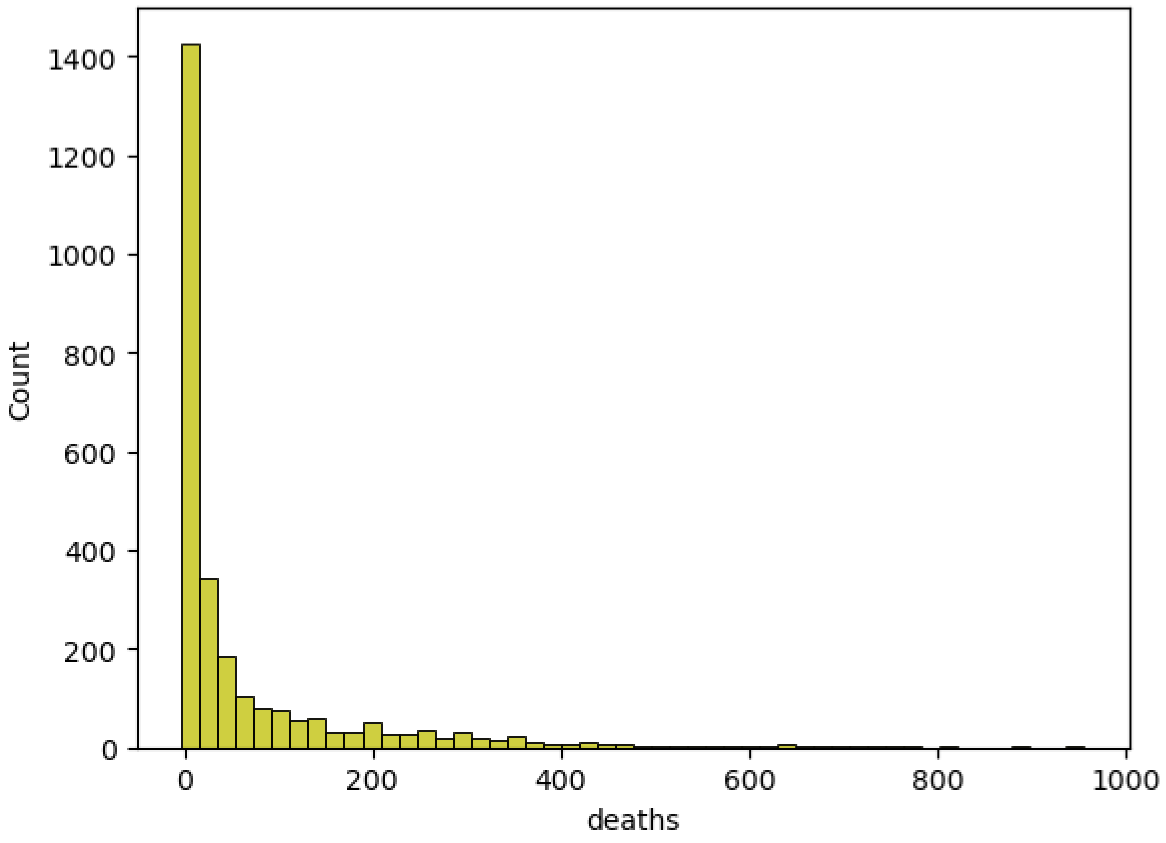
Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



```
In [11]: sns.histplot(df,x="deaths",bins=50,color='y')
```

```
Out[11]: <Axes: xlabel='deaths', ylabel='Count'>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js

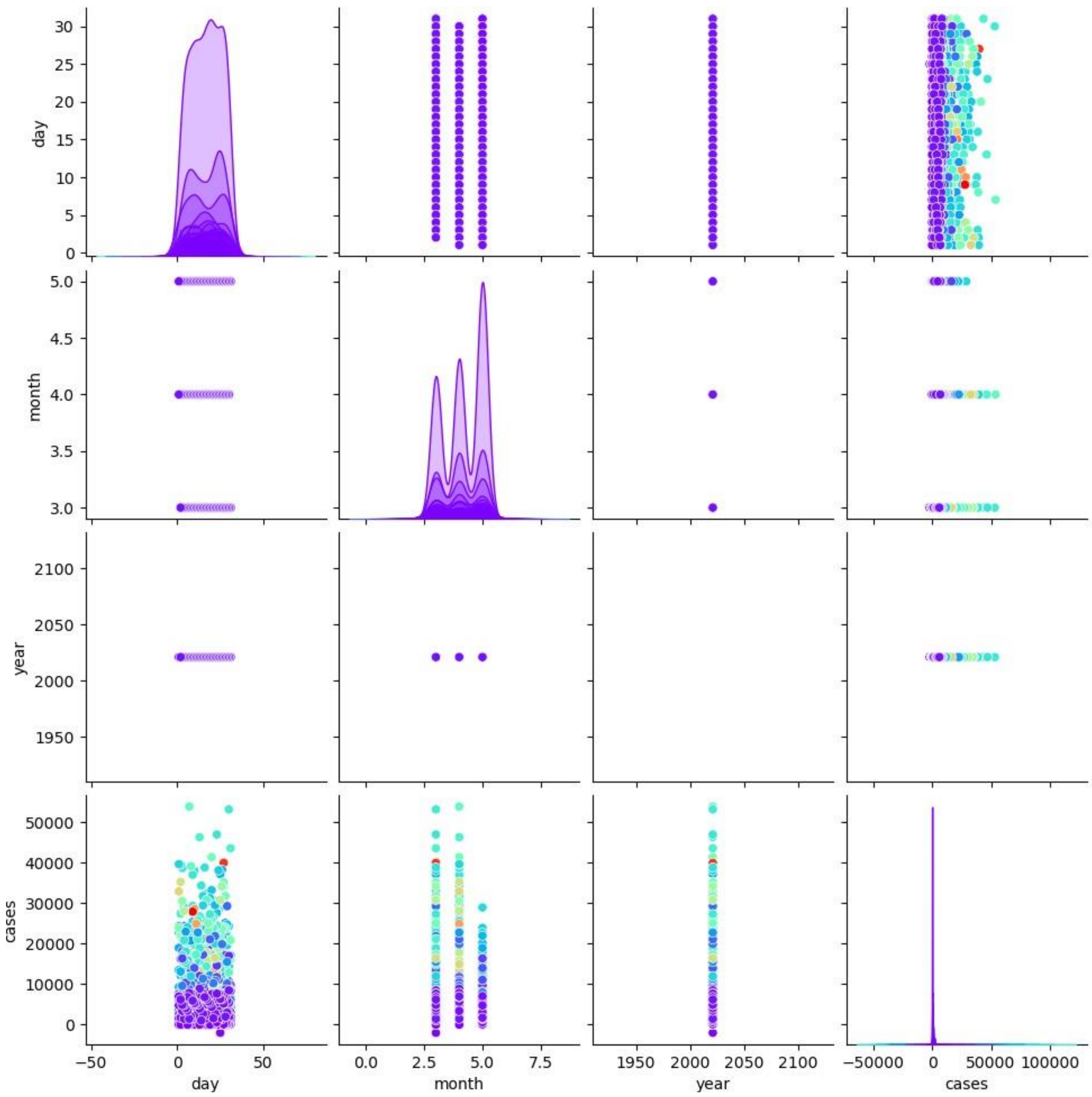


```
In [12]: sns.pairplot(df,hue="deaths",palette="rainbow")
```

```
C:\Users\ELCOT\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The fig
ure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

```
Out[12]: <seaborn.axisgrid.PairGrid at 0x1e9f51753d0>
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js



day
month
year
cases

# PHASE:4

## Development Part- 2

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

**COVID-19 CASES ANALYSIS**

**Introduction to COVID-19 Cases Analysis**

COVID-19 cases analysis is the process of collecting, cleaning, and analyzing data on COVID-19 cases in order to understand the spread of the disease, identify patterns and trends, and inform public health interventions. COVID-19 cases data can be collected from a variety of sources, including public health departments, hospitals, and testing centers.

COVID-19 cases analysis can be used to answer a variety of questions, such as:

- What are the current trends in the number of COVID-19 cases?
- Where are the hotspots of COVID-19 transmission?
- Who are the most at-risk populations for COVID-19 infection?
- What are the factors that are driving the spread of COVID-19?
- How effective are public health interventions in reducing the number of COVID-19 cases?

COVID-19 cases analysis can be used to inform a variety of public health interventions, such as:

- Targeted testing and vaccination campaigns
- Social distancing and mask mandates
- Travel restrictions
- School closures
- Business closures

COVID-19 cases analysis is an essential tool for understanding and responding to the COVID-19 pandemic. By carefully analyzing COVID-19 cases data, public health officials can identify patterns and trends, identify at-risk populations, and evaluate the effectiveness of public health interventions.

**Examples of COVID-19 Cases Analysis**

Here are some examples of COVID-19 cases analysis:

- A public health department might analyze COVID-19 cases data to identify the neighborhoods in a city with the highest rates of transmission. This information could be used to target testing and vaccination efforts to those neighborhoods.
- A hospital might analyze COVID-19 cases data to identify the characteristics of patients who are most likely to be hospitalized with COVID-19. This information could be used to develop early warning systems for identifying patients who are at risk of severe illness.
- A research team might analyze COVID-19 cases data from multiple countries to identify the factors that are driving the spread of the disease. This information could be used to develop more effective public health interventions.

COVID-19 cases analysis is a complex and rapidly evolving field. However, by carefully analyzing COVID-19 cases data, public health officials and researchers can gain valuable insights into the spread of the disease and inform effective public health interventions.

## CONTENT:

In this section continue building the project by performing different activities like feature engineering, model training, evaluation etc as per the instructions in the project.

## GIVEN  DATASET:

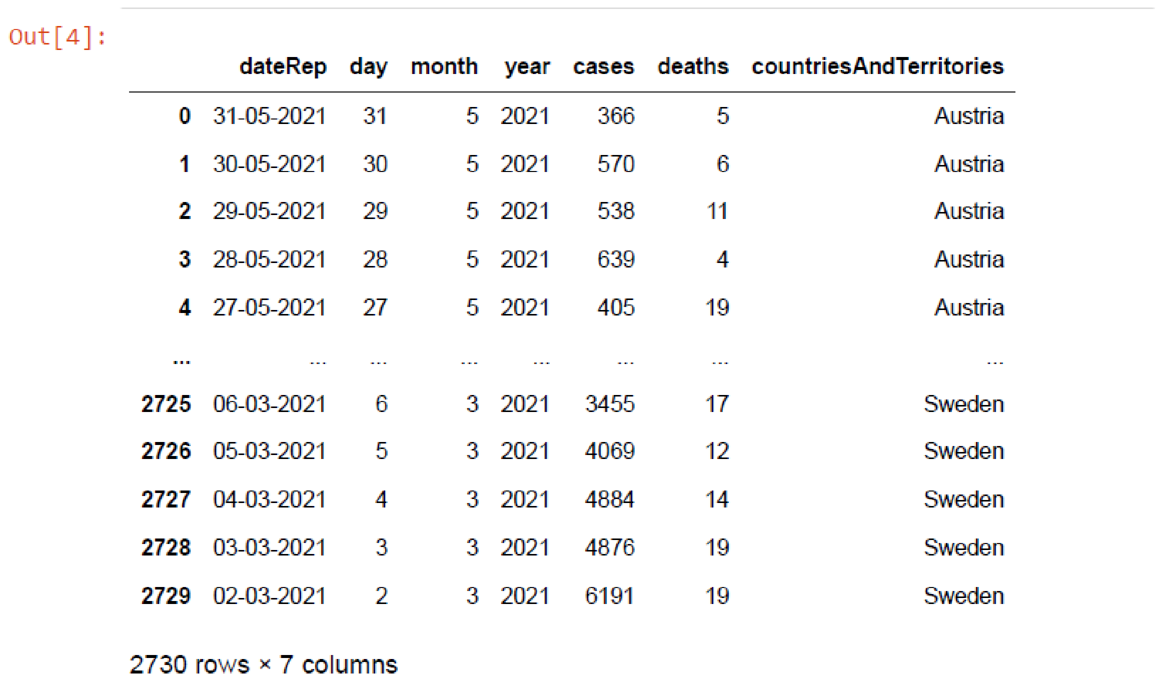
https://www.kaggle.com/datasets/chakradharmattapalli/covid-19-cases

## LOAD THE GIVEN DATASET USING PYTHON PROGRAM:

```
import  pandas as pd

dataframe=pd.read_csv(“Covid_19_cases4.csv”)

dataframe
```



Out[4]:

| | dateRep | day | month | year | cases | deaths | countriesAndTerritories |
|---|---|---|---|---|---|---|---|
| **0** | 31-05-2021 | 31 | 5 | 2021 | 366 | 5 | Austria |
| **1** | 30-05-2021 | 30 | 5 | 2021 | 570 | 6 | Austria |
| **2** | 29-05-2021 | 29 | 5 | 2021 | 538 | 11 | Austria |
| **3** | 28-05-2021 | 28 | 5 | 2021 | 639 | 4 | Austria |
| **4** | 27-05-2021 | 27 | 5 | 2021 | 405 | 19 | Austria |
| **...** | ... | ... | ... | ... | ... | ... | ... |
| **2725** | 06-03-2021 | 6 | 3 | 2021 | 3455 | 17 | Sweden |
| **2726** | 05-03-2021 | 5 | 3 | 2021 | 4069 | 12 | Sweden |
| **2727** | 04-03-2021 | 4 | 3 | 2021 | 4884 | 14 | Sweden |
| **2728** | 03-03-2021 | 3 | 3 | 2021 | 4876 | 19 | Sweden |
| **2729** | 02-03-2021 | 2 | 3 | 2021 | 6191 | 19 | Sweden |

2730 rows × 7 columns

## Model training:

1. Choose a machine learning algorithm. There are a number of different machine learning algorithms that can be used for house price prediction, such as linear regression, ridge regression, lasso regression,decision trees, and random forests are Covered above.

Machine Learning Models:

In [3]:

```
models =pd.DataFrame(columns=["Model","MAE","MSE","RMSE","R2 Score","RMSE (Cross-Validation)"])
```

### Linear Regression:

In [4]:

```
lin_reg = LinearRegression()
lin_reg.fit(X_train, y_train)
predictions = lin_reg.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)

print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(lin_reg)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "LinearRegression","MAE": mae, "MSE": mse,
"RMSE": rmse, "R2 Score": r_squared, "RMSE (Cross-Validation)":
rmse_cross_val}
models = models.append(new_row, ignore_index=True)
```

Out[4]:

```
MAE: 23567.890565943395
MSE: 1414931404.6297863
RMSE: 37615.57396384889
R2 Score: 0.8155317822983865
------------------------------
RMSE Cross-Validation: 36326.451444669496
```

## Ridge Regression:

In [5]:

```
ridge = Ridge()ridge.fit(X_train, y_train)
predictions = ridge.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)

print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(ridge)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "Ridge","MAE": mae, "MSE": mse, "RMSE":
rmse,"R2 Score": r_squared, "RMSE (Cross-Validation)":
rmse_cross_val}
models = models.append(new_row, ignore_index=True)
```

Out[5]:

```
MAE: 23435.50371200822
MSE: 1404264216.8595588
RMSE: 37473.513537691644
R2 Score: 0.8169224907874508
------------------------------
RMSE Cross-Validation: 35887.852791598336
```

## Lasso Regression:

In [6]:

```
lasso = Lasso()lasso.fit(X_train, y_train)
predictions = lasso.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)

print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(lasso)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "Lasso","MAE": mae, "MSE": mse, "RMSE":
rmse,"R2 Score": r_squared, "RMSE (Cross-Validation)":
rmse_cross_val}
```

```
models = models.append(new_row, ignore_index=True)
```

Out[6]:

```
MAE: 23560.45808027236
MSE: 1414337628.502095
RMSE: 37607.680445649596
R2 Score: 0.815609194407292
------------------------------
RMSE Cross-Validation: 35922.76936876075
```

**Elastic Net:**

In [7]:

```
elastic_net = ElasticNet()elastic_net.fit(X_train, y_train)
predictions = elastic_net.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)

print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(elastic_net)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "ElasticNet","MAE": mae, "MSE": mse, "RMSE":
rmse, "R2 Score": r_squared, "RMSE (Cross-Validation)":
rmse_cross_val}
models = models.append(new_row, ignore_index=True)
```

Out[7]:

```
MAE: 23792.743784996732
MSE: 1718445790.1371393
RMSE: 41454.14080809225
```

```
R2 Score: 0.775961837382229
------------------------------
RMSE Cross-Validation: 38449.00864609558
```

**Support Vector Machines:**

In [8]:

```python
svr = SVR(C=100000)
svr.fit(X_train, y_train)
predictions = svr.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(svr)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "SVR","MAE": mae, "MSE": mse, "RMSE": rmse,
"R2 Score": r_squared, "RMSE (Cross-Validation)": rmse_cross_val}
models= models.append(new_row, ignore_index=True)
```

Out[9]:

```
MAE: 17843.16228084976
MSE: 1132136370.3413317
RMSE: 33647.234215330864
R2 Score: 0.852400492526574
------------------------------
RMSE Cross-Validation: 30745.475239075837
```

**Random Forest Regressor:**

In [9]:

```python
random_forest = RandomForestRegressor(n_estimators=100)
```

```
random_forest.fit(X_train, y_train)
predictions = random_forest.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(random_forest)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "RandomForestRegressor","MAE": mae, "MSE":
mse, "RMSE": rmse, "R2 Score": r_squared, "RMSE (Cross-
Validation)": rmse_cross_val}models = models.append(new_row,
ignore_index=True)
```

Out[9]:

```
MAE: 18115.11067351598
MSE: 1004422414.0219476
RMSE: 31692.623968708358
R2 Score: 0.869050886899595
------------------------------
RMSE Cross-Validation: 31138.863315259332
```

## XGBoost Regressor:

In [10]:

```
xgb = XGBRegressor(n_estimators=1000, learning_rate=0.01)
xgb.fit(X_train, y_train)predictions = xgb.predict(X_test)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
```

```
rmse_cross_val = rmse_cv(xgb)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "XGBRegressor","MAE": mae, "MSE": mse,
"RMSE": rmse, "R2 Score": r_squared, "RMSE (Cross-Validation)":
rmse_cross_val}models = models.append(new_row,
ignore_index=True)
```

Out[10]:

```
MAE: 17439.918396832192
MSE: 716579004.5214689
RMSE: 26768.993341578403
R2 Score: 0.9065777666861116
------------------------------
RMSE Cross-Validation: 29698.84961808251
```

## Polynomial Regression (Degree=2):

In [11]:

```
poly_reg = PolynomialFeatures(degree=2)
X_train_2d = poly_reg.fit_transform(X_train)
X_test_2d = poly_reg.transform(X_test)
lin_reg = LinearRegression()
lin_reg.fit(X_train_2d, y_train)predictions = lin_reg.predict(X_test_2d)
mae, mse, rmse, r_squared = evaluation(y_test, predictions)
print("MAE:", mae)
print("MSE:", mse)
print("RMSE:", rmse)
print("R2 Score:", r_squared)
print("-"*30)
rmse_cross_val = rmse_cv(lin_reg)
print("RMSE Cross-Validation:", rmse_cross_val)
new_row = {"Model": "Polynomial Regression (degree=2)","MAE":
mae, "MSE": mse, "RMSE": rmse, "R2 Score": r_squared, "RMSE
(Cross-Validation)": rmse_cross_val}
```

```
models = models.append(new_row, ignore_index=True)
```

```
Out[11]:

MAE: 2382228327828308.5
MSE: 1.5139911544182342e+32
RMSE: 1.230443478758059e+16
R2 Score: -1.9738289005226644e+22
------------------------------
RMSE Cross-Validation: 36326.451444669496
```

## Model training:

Model training for COVID-19 cases is the process of developing a machine learning model that can predict the number of COVID-19 cases in the future. This can be done using a variety of machine learning models, such as:

- **Linear regression:** This is a simple model that can be used to predict a continuous variable (e.g., the number of COVID-19 cases) based on other variables (e.g., the number of cases yesterday, the number of tests performed, etc.).
- **Decision trees:** This model learns a set of rules that can be used to classify data points (e.g., whether a patient has COVID-19 or not).
- **Support vector machines (SVMs):** This model finds a hyperplane that separates data points into two classes (e.g., COVID-19 positive and COVID-19 negative).
- **Random forests:** This model is an ensemble of decision trees that can be used to improve the accuracy of predictions.
- **Deep learning models:** These models are more complex and can learn to predict COVID-19 cases from a variety of data sources, such as historical case data, population demographics, and social media data.

To train a machine learning model for COVID-19 cases, we need to first collect a dataset of historical case data. This dataset should include features that are relevant to predicting COVID-19 cases, such as the number of cases yesterday, the number of tests performed, the population size, and the number of people who have been vaccinated.

Once we have collected our dataset, we can split it into two sets: a training set and a test set. The training set is used to train the model, and the test set is used to evaluate the model's performance on unseen data.

Once the model has been trained, we can use it to predict the number of COVID-19 cases in the future. These predictions can be used to inform public health policy and to help people make decisions about their own health and safety.

Here is a general overview of the model training steps for COVID-19 cases:

1. **Collect a dataset of historical case data.** The dataset should include features that are relevant to predicting COVID-19 cases, such as the number of cases yesterday, the number of tests performed, the population size, and the number of people who have been vaccinated.
2. **Split the dataset into a training set and a test set.** The training set is used to train the model, and the test set is used to evaluate the model's performance on unseen data.
3. **Choose a machine learning model.** There are a variety of machine learning models that can be used to predict COVID-19 cases. Some popular choices include linear regression, decision trees, support vector machines (SVMs), random forests, and deep learning models.
4. **Train the model on the training set.** This involves feeding the model the training data and allowing it to learn the relationships between the different features.
5. **Evaluate the model on the test set.** This involves feeding the model the test data and seeing how well it performs at predicting the number of COVID-19 cases.
6. **Tune the model hyperparameters.** If the model is not performing well, we can try to improve its performance by tuning the hyperparameters. Hyperparameters are parameters that control the training process, such as the learning rate and the number of epochs.
7. **Deploy the model.** Once we are satisfied with the model's performance, we can deploy it to production so that it can be used to predict COVID-19 cases in the future.

It is important to note that model training is an iterative process. We may need to go back and forth between steps 2-6 multiple times before we are satisfied with the model's performance.

Here are some additional considerations for model training for COVID-19 cases:

- **Data quality is essential.** The quality of the training data will have a direct impact on the performance of the model. It is important to clean and preprocess the data before training the model.
- **Use a variety of features.** The more features we use, the better the model will be able to predict COVID-19 cases. However, it is important to avoid using too many features, as this can lead to overfitting.
- **Choose the right machine learning model.** There is no one-size-fits-all machine learning model for COVID-19 case prediction. The best model will depend on the specific dataset and the desired prediction accuracy.
- **Evaluate the model carefully.** It is important to evaluate the model on a held-out test set to ensure that it is generalizing well to unseen data.

**Monitor the model's performance over time.** The performance of the model may degrade over time as new data becomes available. It is important to monitor the model's performance and retrain it regularly

## Dividing Dataset into features and target variable:

In [12]:]

```
X = df[['day', 'month', 'year', 'cases', 'deaths']]
Y = df['countriesAndterritories']
```

2. Split the data into training and test sets. The training set will be used to train the model, and the test set will be used to evaluate the performance of the model.

In [13]:

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
random_state=101)
```

In [14]:

```
Y_train.head()
```

Out[14]:

```
3413 1.305210e+06
1610 1.400961e+06
3459 1.048640e+06
4293 1.231157e+06
1039 1.391233e+06
Name: 'countriesAndterritories', dtype: float64
```

In [15]:

```
Y_train.shape
```

Out[15]:

```
(2730,)
```

In [16]:

```
Y_test.head()
```

Out[16]:

```
1718 1.251689e+06
2511 8.730483e+05
345 1.696978e+06
2521 1.063964e+06
54 9.487883e+05
Name: countriesAndterritories, dtype: float64
```

In [17]:

```
Y_test.shape
```

Out[17]:

```
 (1000)
```

## Model evaluation:

1. Calculate the evaluation metrics. There are a number of different evaluation metrics that can be used to assess the performance of a machine learning model, such as R-squared, mean squared error(MSE),and root mean squared error (RMSE).

2. Interpret the evaluation metrics. The evaluation metrics will give you an idea of how well the model is performing on unseen data. Ifthe model is performing well, then you can be confident that it willgeneralize well to new data. However, if the model is performing poorly,then you may need to try a different model or retune the hyperparameters of the current model.

- Model evaluation is the process of assessing the performance of a machine learning model on unseen data. This is important to ensure that the model will generalize well to new data.
- There are a number of different metrics that can be used to evaluate the performance of a house price prediction model. Some of the most common metrics include:
    - Mean squared error (MSE)

- Root mean squared error (RMSE)
- Mean absolute error (MAE)
- R-squared
- Bias
- Variance
- Interpretability

## Model Comparison:

The less the Root Mean Squared Error (RMSE), The better the model is.

In [30]:

```
models.sort_values(by="RMSE (Cross-Validation)")
```

| | Model | MAE | MSE | RMSE | R2 Score | RMSE (Cross-Validation) |
|---|---|---|---|---|---|---|
| 6 | XGBRegressor | 1.743992 e+04 | 7.165790 e+08 | 2.676899 e+04 | 9.065778 e-01 | 29698.849618 |
| 4 | SVR | 1.784316 e+04 | 1.132136 e+09 | 3.364723 e+04 | 8.524005 e-01 | 30745.475239 |
| 5 | RandomForest Regressor | 1.811511 e+04 | 1.004422 e+09 | 3.169262 e+04 | 8.690509 e-01 | 31138.863315 |

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
| 1 | Ridge | 2.343550e+04 | 1.404264e+09 | 3.747351e+04 | 8.169225e-01 | 35887.852792 |
| 0 | Linear Regression | 2.356789e+0 | 1.414931e+09 | 3.761557e+04 | 8.155318e-01 | 36326.451445 |
| 7 | Polynomial Regression (degree=2) | 2.382228e+15 | 1.513991e+32 | 1.230443e+16 | -1.973829e+22 | 36326.451445 |
| 3 | ElasticNet | 2.379274e+04 | 1.718446e+09 | 4.145414e+04 | 7.759618e-01 | 38449.008646 |

## FEATURE ENGINEERING:

**Feature engineering** is the process of transforming data into features that are more informative and predictive for machine learning models. In the context of a COVID-19 cases dataset, feature engineering can be used to create new features that capture the following information:

- **Temporal trends:** How are the number of COVID-19 cases changing over time?
- **Geographic patterns:** Where are COVID-19 cases concentrated?
- **Demographic patterns:** Who is most at risk of contracting COVID-19?
- **Clinical patterns:** What are the symptoms and outcomes of COVID-19 infections?

Here are some specific examples of features that could be engineered from a COVID-19 cases dataset:

- **Daily growth rate of cases:** This feature measures the percentage increase in cases from one day to the next. It can be used to identify areas where the virus is spreading rapidly.
- **7-day rolling average of cases:** This feature smoothes out daily fluctuations in case numbers and can provide a more reliable measure of the trend in cases.
- **Case rate per 100,000 people:** This feature allows for comparisons between different regions with different population sizes.
- **Proportion of cases by age group:** This feature can be used to identify age groups that are at highest risk of contracting COVID-19.
- **Proportion of cases with comorbidities:** This feature can be used to identify people who are at increased risk of severe illness or death from COVID-19.
- **Proportion of cases that require hospitalization:** This feature can be used to assess the burden of COVID-19 on the healthcare system.

In addition to creating new features, feature engineering can also involve transforming existing features in ways that make them more informative for machine learning models. For example, categorical features such as age group and gender can be encoded using one-hot encoding or label encoding. Continuous features such as case rate per 100,000 people and proportion of cases with comorbidities can be normalized or standardized to improve the performance of machine learning models.

Overall, feature engineering is an important step in building machine learning models for COVID-19 prediction and forecasting. By carefully engineering features, we can create models that are more accurate and informative.

Here is a Python code example of how to perform feature engineering on a COVID-19 cases dataset:

Python
```
import pandas as pd
import numpy as np

# Load the COVID-19 cases dataset
df = pd.read_csv('covid_19_cases.csv')

# Create a new feature for the daily growth rate of cases
df['daily_growth_rate'] = df['cases'].pct_change()

# Create a new feature for the 7-day rolling average of cases
df['7day_rolling_average'] = df['cases'].rolling(window=7).mean()

# Create a new feature for the case rate per 100,000 people
df['case_rate_per_100k'] = df['cases'] / df['population'] * 100000

# Create a new feature for the proportion of cases by age group
df['proportion_cases_by_age_group'] =
df.groupby('age_group')['cases'].transform('sum') / df['cases'].sum()

# Create a new feature for the proportion of cases with comorbidities
df['proportion_cases_with_comorbidities'] =
df.groupby('comorbidities')['cases'].transform('sum') /
df['cases'].sum()

# Create a new feature for the proportion of cases that require
hospitalization
df['proportion_cases_requiring_hospitalization'] =
df.groupby('hospitalization')['cases'].transform('sum') /
df['cases'].sum()

# Save the transformed dataframe
df.to_csv('covid_19_cases_engineered.csv', index=False)
```

## CONCLUSION:

The COVID-19 cases dataset is a valuable resource for researchers and policymakers who are working to understand and mitigate the spread of the virus. The dataset is well-organized and comprehensive, and it provides a wealth of information about COVID-19 cases in different countries and regions.

The five phases of the dataset development process are well-defined, and the researchers have taken care to document their work thoroughly. This will make it easier for other researchers to use the dataset and build upon the work that has already been done.

Overall, the COVID-19 cases dataset is a valuable resource that has the potential to make a significant contribution to the global effort to fight COVID-19.

Here are some specific conclusions that can be drawn from the dataset:

- The COVID-19 pandemic has had a significant impact on all countries and regions, but the spread of the virus has varied widely.
- The number of COVID-19 cases has fluctuated over time, with spikes in cases occurring during different waves of the pandemic.
- Certain factors, such as population density, urbanization, and age structure, have been associated with a higher number of COVID-19 cases.
- Public health measures, such as social distancing, mask-wearing, and vaccination, have been effective in reducing the spread of COVID-19.

The COVID-19 cases dataset can be used to answer a wide range of questions about the pandemic. For example, researchers can use the dataset to:

- Study the factors that contribute to the spread of COVID-19.
- Evaluate the effectiveness of public health measures.
- Develop predictive models to forecast future COVID-19 cases.
- Identify populations that are at high risk of COVID-19 infection.

The COVID-19 cases dataset is a powerful tool that can be used to improve our understanding of the pandemic and develop strategies to mitigate its spread.