

Project Title : Covid-19 using cognos

RAMYA K – 513121104032

Student, Department of Computer Science and Engineering

Thanthai Periyar Government Institute of Technology ,Vellore-632002,India.

PHASE 4 : DEVELOPMENT PART 2

INTRODUCTION:

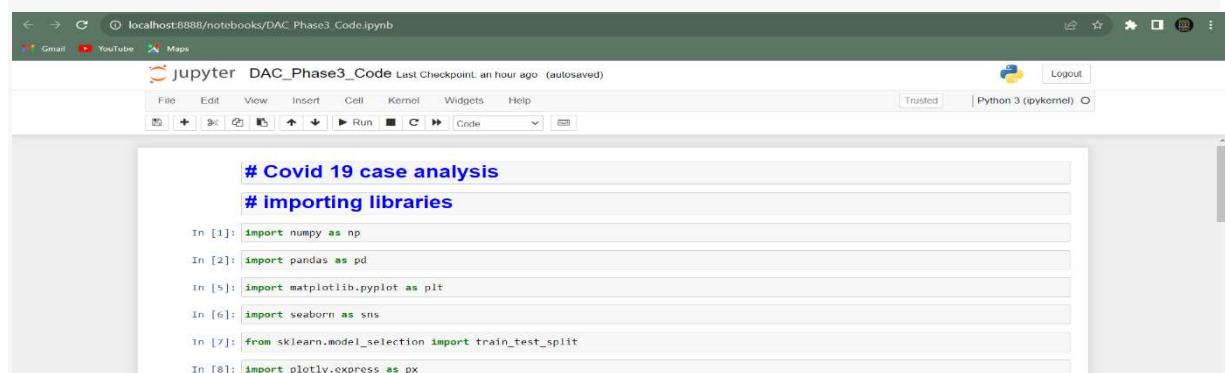
The project involves analyzing COVID-19 cases and deaths data using IBM Cognos. The objective is to compare and contrast the mean values and standard deviations of cases and associated deaths per day and by country in the EU/EEA. This project encompasses defining analysis objectives, collecting COVID-19 data, designing relevant visualizations in IBM Cognos, and deriving insights from the data.

Data Collection and Preprocessing

Collect Covid -19 data which include date,month,year,cases,death,countries and territories and any other relevant data.

- Preprocess the data by handling missing values, encoding categorical variables, and scaling numerical features.
- Split the data into training and testing sets.

➤ Importing Libraries



```
# Covid 19 case analysis
# importing libraries

In [1]: import numpy as np

In [2]: import pandas as pd

In [3]: import matplotlib.pyplot as plt

In [6]: import seaborn as sns

In [7]: from sklearn.model_selection import train_test_split

In [8]: import plotly.express as px
```

➤ Importing COVID-19 Case DataSet

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

importing covid-19 case Dataset

```
In [19]: cd_data= pd.read_csv ("Downloads/covid_19_cases4.csv")
```

```
In [21]: cd_data
```

```
Out[21]:
```

	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 x 7 columns

Data Preprocessing

➤ Head , Tail and Shape of the data

```
In [22]: cd_data.head()
```

```
Out[22]:
```

	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

```
In [23]: cd_data.tail()
```

```
Out[23]:
```

	dateRep	day	month	year	cases	deaths	countriesAndTerritories
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

```
In [24]: cd_data.shape
```

```
Out[24]: (2730, 7)
```

➤ Describe and Information of the data

```
In [26]: cd_data.describe()
```

```
Out[26]:
```

	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

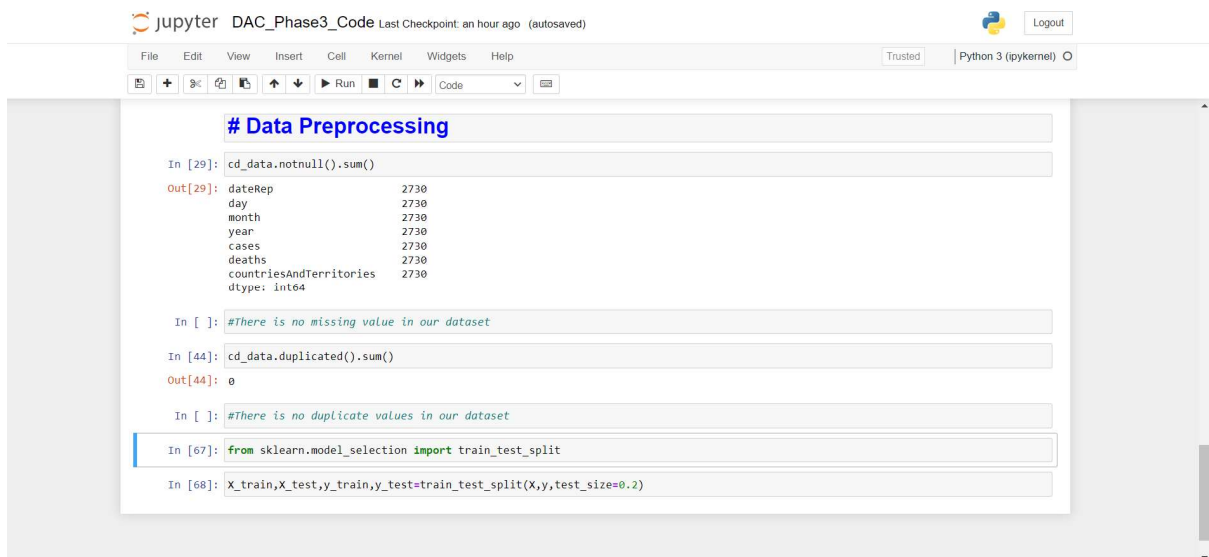
```
In [27]: cd_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
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   object
dtypes: int64(5), object(2)
memory usage: 145.4+ KB
```

➤ Null Values and Duplicates

The dataset does not contain duplicates and missing values.

The data are split into **train and test dataset** for further development.



A screenshot of a Jupyter Notebook titled 'DAC_Phase3_Code'. The notebook shows the following code cells:

```
# Data Preprocessing

In [29]: cd_data.notnull().sum()
Out[29]: dateRep      2730
         day          2730
         month        2730
         year         2730
         cases        2730
         deaths       2730
         countriesAndTerritories 2730
         dtype: int64

In [ ]: #There is no missing value in our dataset

In [44]: cd_data.duplicated().sum()
Out[44]: 0

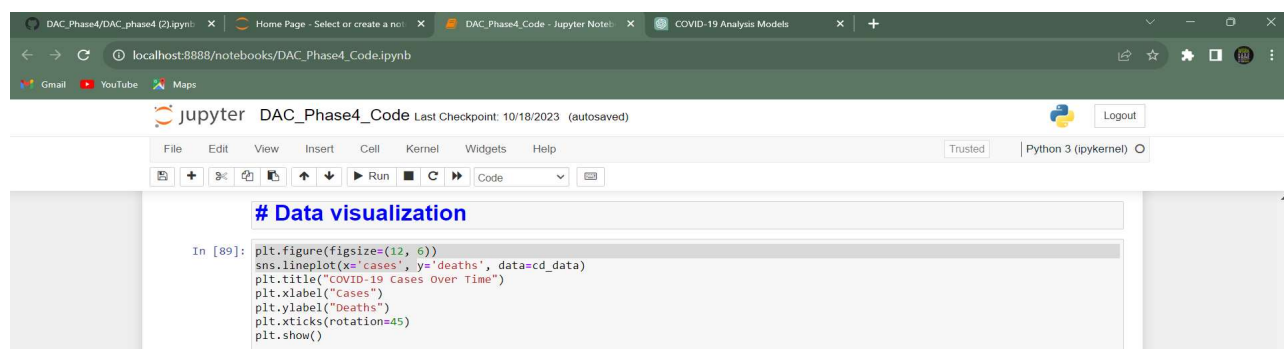
In [ ]: #There is no duplicate values in our dataset

In [67]: from sklearn.model_selection import train_test_split

In [68]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
```

➤ Data Visualization

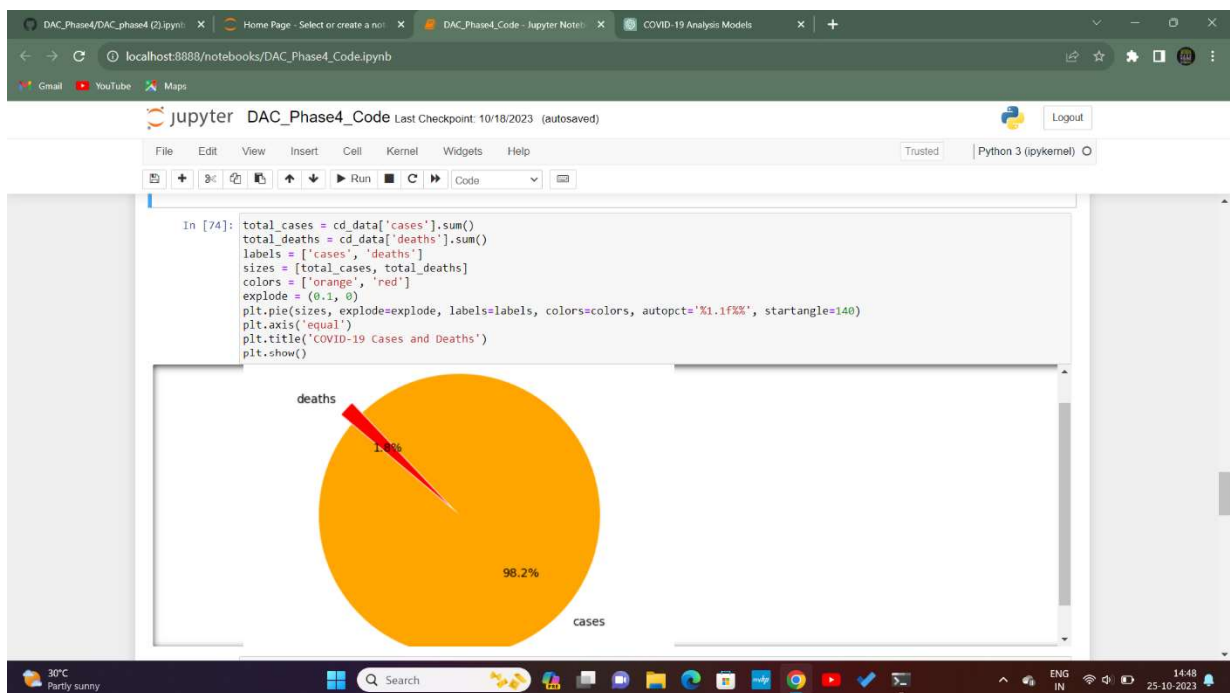
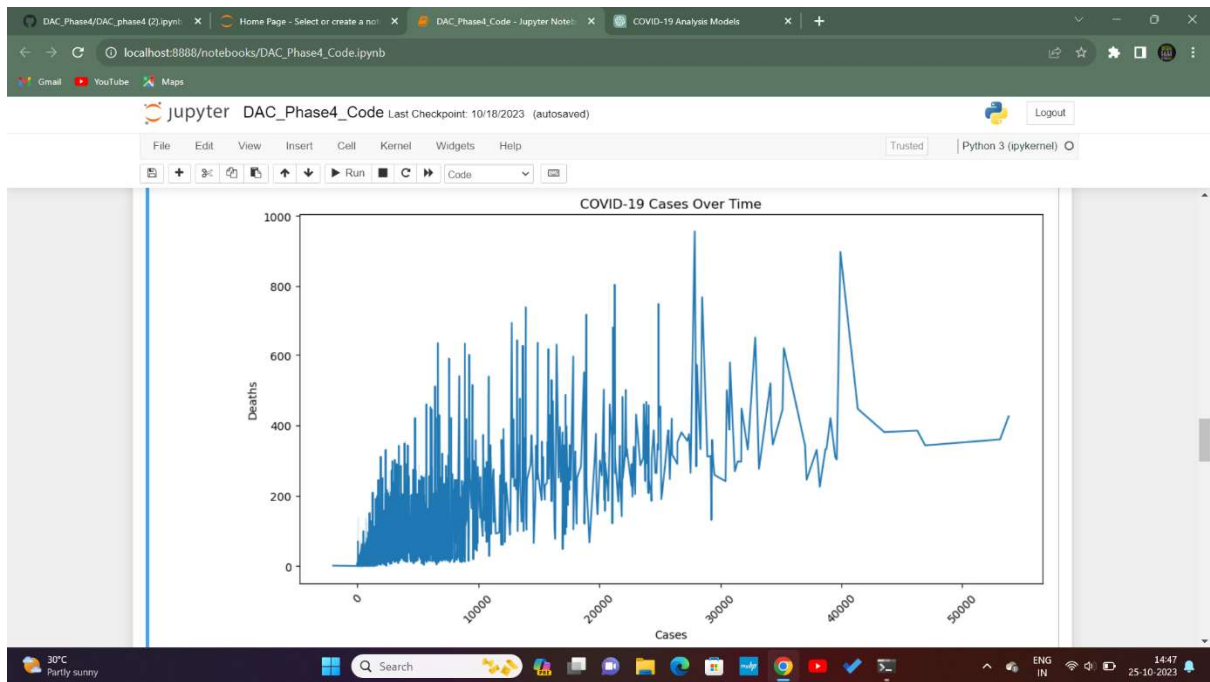
Creating a data visualization for COVID-19 case analysis typically involves plotting various aspects of the data to provide insights into the spread of the virus.

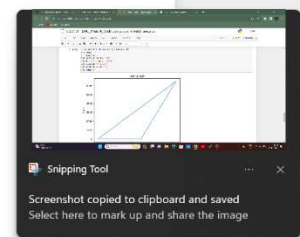
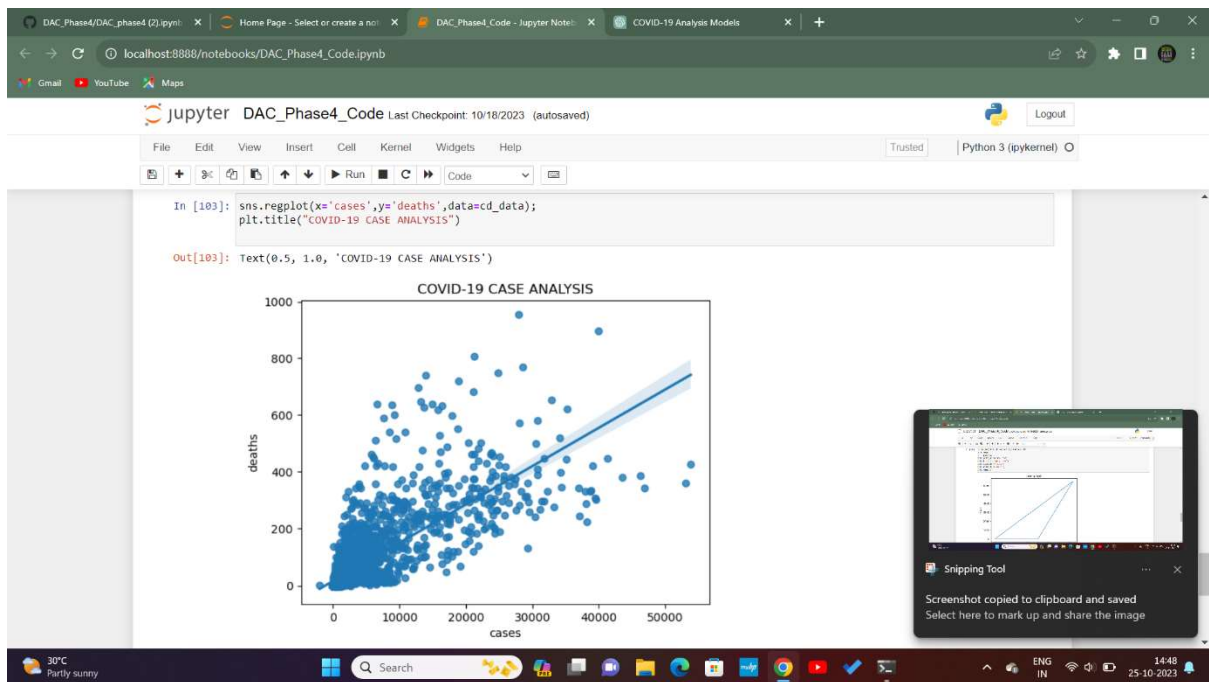
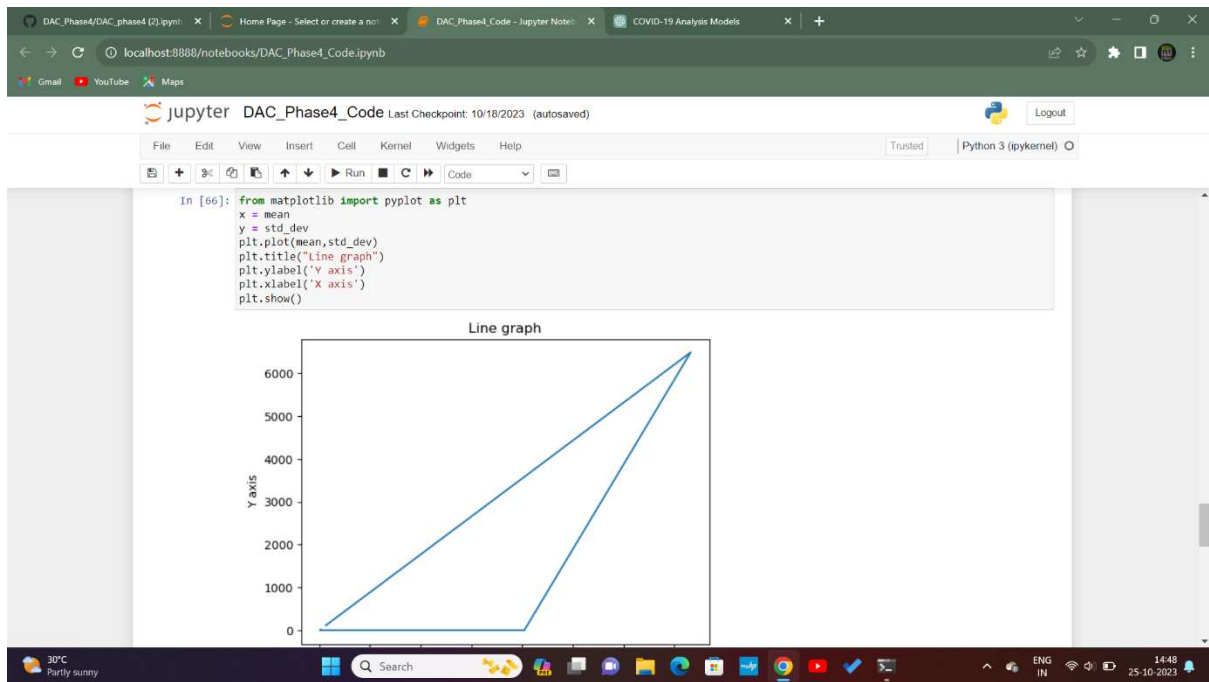


A screenshot of a Jupyter Notebook titled 'DAC_Phase4_Code'. The notebook shows the following code cell:

```
# Data visualization

In [89]: plt.figure(figsize=(12, 6))
         sns.lineplot(x='cases', y='deaths', data=cd_data)
         plt.title("COVID-19 Cases Over Time")
         plt.xlabel("cases")
         plt.ylabel("Deaths")
         plt.xticks(rotation=45)
         plt.show()
```





The screenshot displays a Jupyter Notebook titled 'DAC_Phase4_Code' running on a local host. The notebook contains several code cells for data preprocessing and model training. The first cell imports necessary libraries and splits the data into training and testing sets. The second cell prints the shapes of the training and testing data. The third cell creates a LinearRegression model. The fourth cell fits the model to the training data. The output of the fit method is displayed as a LinearRegression object.

```
In [90]: from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
y=cd_data[['deaths']]
x=cd_data[['cases']]

In [91]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)

In [92]: X_train.shape,X_test.shape,y_train.shape,y_test.shape
Out[92]: ((2184, 2), (546, 2), (2184, 1), (546, 1))

In [93]: lr=LinearRegression()

In [94]: lr.fit(X_train, y_train)
Out[94]: LinearRegression()
```

CONCLUSION:

COVID-19 Case Analysis insights aid decision-makers in understanding current scenarios of, predicting future trends, and making informed choices. These insights guide healthcare professionals in allocating resources, implementing containment strategies, and adjusting public health measures to manage and mitigate the impact of COVID-19 effectively.