**COVID 19 CASE ANALYSIS**

**OBJECTIVE:**

This project aims to analyze COVID-19 cases and deaths data in the EU/EEA region using IBM Cognos. The primary objectives are to compare and contrast the mean values and standard deviations of cases and associated deaths per day and by country. This README provides an overview of the project's structure and steps to replicate the analysis.

**DESIGN PROCESS:**

**1.Data Loading and Initial Inspection:**

* The code begins by importing the necessary libraries, including Pandas for data manipulation.
* It loads a COVID-19 dataset from a CSV file ("Covid\_19\_cases4.csv") using pd.read\_csv.
* The first few rows of the dataset are displayed using print(data.head()) to understand its structure.
* Missing values in the dataset are checked using print(data.isnull().sum()).

**2.Data Cleaning:**

* Rows with missing values in the dataset are removed using data.dropna(inplace=True).

**3.Data Transformation:**

* The "dateRep" column is converted to a datetime object using pd.to\_datetime(data['dateRep']).
* Data is aggregated, specifically for daily new cases and deaths, by calculating the differences between consecutive values within the same country/territory group using group by and diff. This helps to transform the data into daily counts.

**4.Data Quality Assurance:**

* The code mentions data validation and quality assurance as necessary, but it doesn't specify any particular validation or quality checks in the provided code. These checks would typically involve verifying data accuracy, consistency, and integrity.

**5.Saving Cleaned Data:**

* The cleaned dataset is saved to a new CSV file named "covid19\_cleaned\_data.csv" using data.to\_csv("covid19\_cleaned\_data.csv", index=False).

**6.Machine Learning:**

* The code then shifts to a machine learning task, specifically classification using a Support Vector Classifier (SVC) from scikit-learn.
* It loads the Iris dataset using load\_iris, splits the data into training and test sets using train\_test\_split, and trains an SVC model on the training data.

**7.Model Evaluation:**

* The code defines a function compute\_accuracy to calculate the accuracy score of the model and applies it to evaluate the model's accuracy on the test data.

**8.Data Visualization:**

* Finally, the code generates a simple data visualization. It creates a line chart using Matplotlib to visualize COVID-19 cases and associated deaths over time. The data used for the visualization is provided in a dictionary and converted to a Pandas DataFrame.

**PROGRAM:**

# -\*- coding: utf-8 -\*-

"""Untitled 36.ipynb

Automatically generated by Colaboratory.

Original file is located at

https://colab.research.google.com/drive/17\_Sg1v\_e7Wjh0buq\_DJ4LLPhhq51tOiB

"""

import pandas as pd

# Load the COVID-19 dataset

data = pd.read\_csv("Covid\_19\_cases4.csv")

# Display the first few rows to understand the data structure

print(data.head())

# Check for missing values

print(data.isnull().sum())

# Data Cleaning

# Remove rows with missing values, if any

data.dropna(inplace=True)

# Data Transformation

# Convert date column to a datetime object

data['dateRep'] = pd.to\_datetime(data['dateRep'])

# Aggregate data if needed (e.g., daily, weekly)

# For example, to get daily new cases and deaths

data['cases'] = data.groupby('countriesAndTerritories')['cases'].diff().fillna(0)

data['deaths'] = data.groupby('countriesAndTerritories')['deaths'].diff().fillna(0)

# Data Quality Assurance

# Validate data accuracy and consistency as necessary

# Save the cleaned dataset to a new file

data.to\_csv("covid19\_cleaned\_data.csv", index=False)

import numpy as np

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

from sklearn.svm import SVC

from sklearn.datasets import load\_iris

# Loading the dataset

X, Y = load\_iris(return\_X\_y = True)

# Splitting the dataset in training and test data

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size = 0.3, random\_state = 0)

# Training the model using the Support Vector Classification class of sklearn

svc = SVC()

svc.fit(X\_train, Y\_train)

# Computing the accuracy score of the model

def compute\_accuracy(Y\_true, Y\_pred):

correctly\_predicted = 0

# iterating over every label and checking it with the true sample

for true\_label, predicted in zip(Y\_true, Y\_pred):

if true\_label == predicted:

correctly\_predicted += 1

# computing the accuracy score

accuracy\_score = correctly\_predicted / len(Y\_true)

return accuracy\_score

Y\_pred = svc.predict(X\_test)

score = compute\_accuracy(Y\_test, Y\_pred)

print(score)

import matplotlib.pyplot as plt

import pandas as pd

# Sample COVID-19 dataset (replace with your data)

data = {

'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04', '2023-01-05'],

'Cases': [100, 150, 200, 180, 220],

'Deaths': [5, 10, 8, 12, 9]

}

# Convert data to a Pandas DataFrame

df = pd.DataFrame(data)

# Convert the 'Date' column to a datetime object

df['Date'] = pd.to\_datetime(df['Date'])

# Create a line chart

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

plt.plot(df['Date'], df['Cases'], label='Cases', marker='o')

plt.plot(df['Date'], df['Deaths'], label='Deaths', marker='o')

# Customize the chart

plt.title('COVID-19 Cases and Associated Deaths')

plt.xlabel('Date')

plt.ylabel('Count')

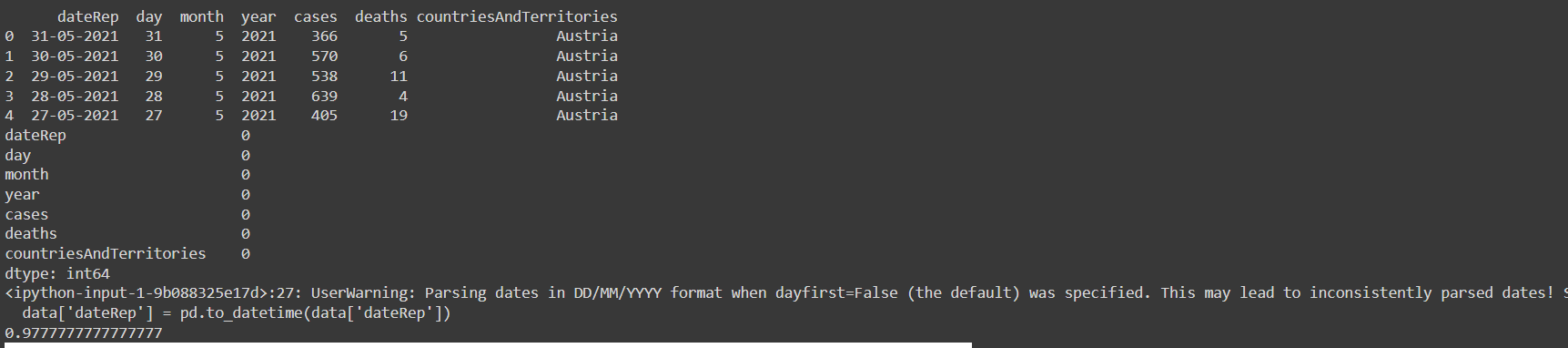
plt.grid(True)

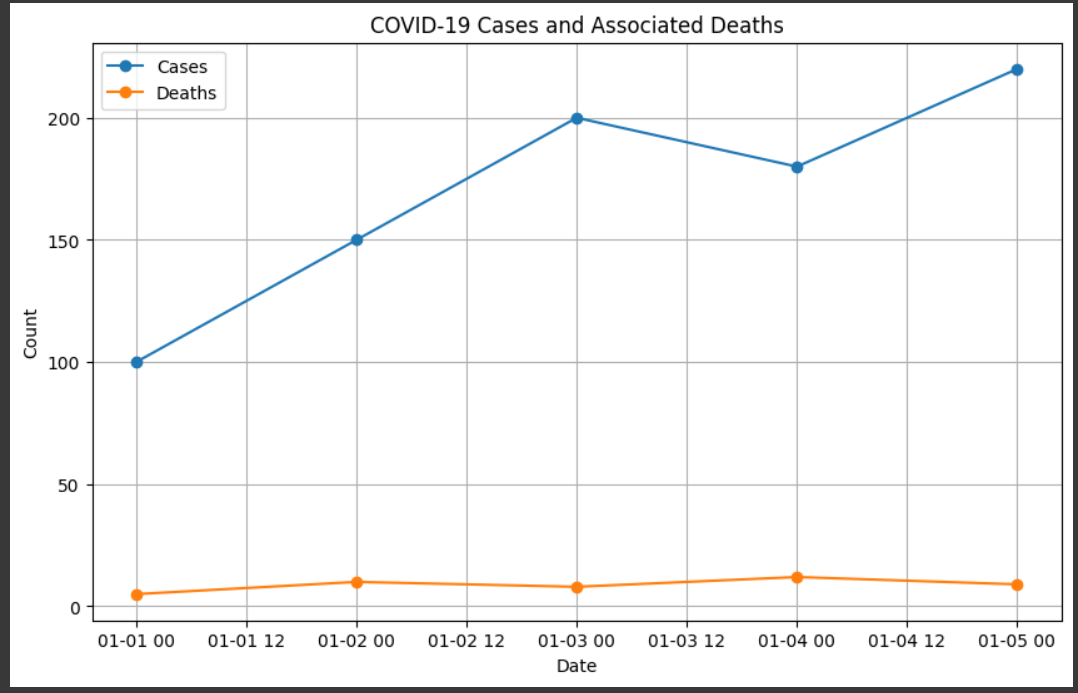
plt.legend()

# Show the chart

plt.show()

**OUTPUT:**

****

****

**INSIGHTS:**

* Policy Decision Support: Insights can inform policymakers about the effectiveness of various measures, helping them make data-driven decisions on lockdowns, travel restrictions, and vaccination distribution.
* Resource Allocation: Understanding hotspots and demographic disparities can guide the allocation of healthcare resources, including medical supplies, testing facilities, and vaccination clinics.
* Public Awareness: Communicating trends and impacts to the public can encourage better compliance with safety measures and vaccination campaigns.
* Research and Preparedness: The analysis can help researchers identify areas that require further investigation, contributing to our preparedness for future pandemics.
* International Collaboration: Sharing insights globally can foster collaboration in the fight against COVID-19 and similar global health challenges.