Data analytics with Cognos

Covid vaccine analysis

Phase 3: Development Part 1

**topic:** Begin conducting the covid-19 vaccine analysis by collecting and preprocessing the data.



Covid vaccine analysis

Introduction:

* The rapid development and deployment of COVID-19 vaccines have marked a significant milestone in the global fight against the pandemic.
* Vaccination efforts worldwide have generated a wealth of data, offering invaluable insights into the effectiveness of these vaccines, distribution patterns, and public response.
* This explores the vital process of COVID-19 vaccine analysis, shedding light on its methodologies, challenges, and the transformative impact it has on public health strategies.

Importance of loading and processing dataset:

* Loading and preprocessing the dataset is an important first step in

building any machine learning model.

* However, it is especially important for total\_vaccinations, people\_vaccinated, people\_fully\_vaccinated etc..
* By loading and preprocessing the dataset, we can ensure that the

machine learning algorithm is able to learn from the data effectively and accurately.

Challenges involved in loading and preprocessing a covid vaccine analysis dataset;

There are a number of challenges involved in loading and preprocessing a covid vaccine analysis dataset, including:

* Handling missing values:
* covid vaccine analysis datasets often contain missing values, which can Be due to a variety of factors, such as human error or incomplete data collection.
* Common methods for handling missing values include dropping the rows with missing values, imputing the missing values with the mean or median of the feature, or using a more sophisticated method such as multiple imputation.
* Encoding categorical variables:
* covid vaccine analysis datasets often contain categorical features, such as the type of house, the neighborhood, and the school district.
* These features need to be encoded before they can be used by machine learning models.
* One common way to encode categorical variables is to use one-hot encoding.
* Scaling the features:
* It is often helpful to scale the features before training a
* machine learning model.
* This can help to improve the performance of the model and make it more robust to outliers.
* There are a variety of ways to scale the features, such as min-max scaling and standard scaling.
* Splitting the dataset into training and testing sets:
* Once the data has been pre-processed, we need to split the

dataset into training and testing sets.

* The training set will be used to train the model, and the testing set will be used to evaluate the performance of the model on unseen data.
* It is important to split the dataset in a way that is representative of the real world distribution of the data.

How to overcome the challenges of loading and preprocessing a covid vaccine analysis dataset:

There are a number of things that can be done to overcome the

challenges of loading and preprocessing a covid vaccine analysis dataset, including:

* Use a data preprocessing library:
* There are a number of libraries available that can help with data preprocessing tasks, such as handling missing values, encoding categorical variables, and scaling the features.
* Carefully consider the specific needs of your model:
* The best way to preprocess the data will depend on the specific

machine learning algorithm that you are using.

* It is important to carefully consider the requirements of the algorithm and to preprocess the data in a way that is compatible with the algorithm.
* Validate the preprocessed data:
* It is important to validate the preprocessed data to ensure that it is in a format that can be used by the machine learning algorithm and that it is of high quality.
* This can be done by inspecting the data visually or by using statistical methods.

1.Loading the dataset:

* Loading the dataset using machine learning is the process of bringing the data into the machine learning environment so that it can be used to train and evaluate a model.
* The specific steps involved in loading the dataset will vary depending on the machine learning library or framework that is being used.
* However, there are some general steps that are common to most machine learning frameworks:

1. Identify the dataset:

* The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

1. Load the dataset:

* Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

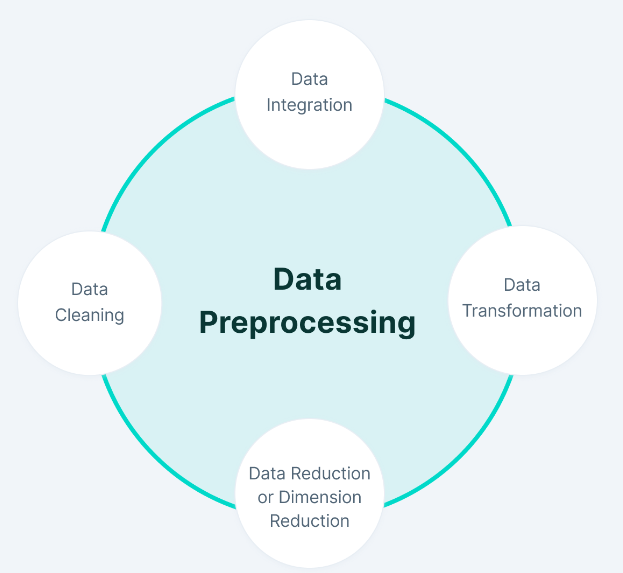
1. Preprocess the dataset:

* Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format, and splitting the data into training and test sets.

Some common data preprocessing tasks include:

* Data cleaning:
* This involves identifying and correcting errors and inconsistencies in the data. For example, this may involve removing duplicate records, correcting typos, and filling in missing values.
* Data transformation:
* This involves converting the data into a format that is suitable for the analysis task. For example, this may involve converting categorical data to numerical data, or scaling the data to a suitable range.
* Feature engineering:
* This involves creating new features from the existing data. For example, this may involve creating features that represent interactions between variables, or features that represent summary statistics of the data.
* Data integration:
* This involves combining data from multiple sources into a single dataset. This may involve resolving inconsistencies in the data, such as different data formats or different variable names.

Data preprocessing is an essential step in many data science projects. By carefully preprocessing the data, data scientists can improve the accuracy and reliability of their results.



CODE:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph\_objects as go

import matplotlib.patches as mpatches

from plotly.subplots import make\_subplots

from wordcloud import WordCloud

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import seaborn as sns

sns.set(color\_codes = True)

sns.set(style="whitegrid")

import plotly.figure\_factory as ff

from plotly.colors import n\_colors

df=pd.read\_csv("C:\\Users\\91866\\Downloads\\archive\\country\_vaccinations.csv")

print(df)

df.info()

df.isnull().sum()

df.fillna(value = 0, inplace = True)

df.total\_vaccinations = df.total\_vaccinations.astype(int)

df.people\_vaccinated = df.people\_vaccinated.astype(int)

df.people\_fully\_vaccinated = df.people\_fully\_vaccinated.astype(int)

df.daily\_vaccinations\_raw = df.daily\_vaccinations\_raw.astype(int)

df.daily\_vaccinations = df.daily\_vaccinations.astype(int)

df.total\_vaccinations\_per\_hundred = df.total\_vaccinations\_per\_hundred.astype(int)

df.people\_fully\_vaccinated\_per\_hundred = df.people\_fully\_vaccinated\_per\_hundred.astype(int)

df.daily\_vaccinations\_per\_million = df.daily\_vaccinations\_per\_million.astype(int)

df.people\_vaccinated\_per\_hundred = df.people\_vaccinated\_per\_hundred.astype(int)

date = df.date.str.split('-', expand =True)

date

df['year'] = date[0]

df['month'] = date[1]

df['day'] = date[2]

df.year = pd.to\_numeric(df.year)

df.month = pd.to\_numeric(df.month)

df.day = pd.to\_numeric(df.day)

df.date = pd.to\_datetime(df.date)

df.head()

print('Data point starts from ',df.date.min(),'n')

print('Data point ends at ',df.date.max(),'n')

print('Total no of countries in the data set ',len(df.country.unique()),'n')

print('Total no of unique vaccines in the data set ',len(df.vaccines.unique()),'n')

df.describe()

df.country.unique()

def size(m,n):

fig = plt.gcf();

fig.set\_size\_inches(m,n);

wordCloud = WordCloud(

background\_color='blue',

max\_font\_size = 50).generate(' '.join(df.country))

plt.figure(figsize=(12,4))

plt.axis('off')

plt.imshow(wordCloud)

plt.show()

fig = px.line(df, x = 'date', y ='daily\_vaccinations', color = 'country')

fig.update\_layout(

title={

'text' : "Daily vaccination trend",

'y':0.95,

'x':0.5

},

xaxis\_title="Date",

yaxis\_title="Daily Vaccinations"

)

fig.show()

OUTPUT

READING THE DATASET:

country iso\_code date total\_vaccinations \

0 Afghanistan AFG 2021-02-22 0.0

1 Afghanistan AFG 2021-02-23 NaN

2 Afghanistan AFG 2021-02-24 NaN

3 Afghanistan AFG 2021-02-25 NaN

4 Afghanistan AFG 2021-02-26 NaN

... ... ... ... ...

86507 Zimbabwe ZWE 2022-03-25 8691642.0

86508 Zimbabwe ZWE 2022-03-26 8791728.0

86509 Zimbabwe ZWE 2022-03-27 8845039.0

86510 Zimbabwe ZWE 2022-03-28 8934360.0

86511 Zimbabwe ZWE 2022-03-29 9039729.0

people\_vaccinated people\_fully\_vaccinated daily\_vaccinations\_raw \

0 0.0 NaN NaN

1 NaN NaN NaN

2 NaN NaN NaN

3 NaN NaN NaN

4 NaN NaN NaN

... ... ... ...

86507 4814582.0 3473523.0 139213.0

86508 4886242.0 3487962.0 100086.0

86509 4918147.0 3493763.0 53311.0

86510 4975433.0 3501493.0 89321.0

86511 5053114.0 3510256.0 105369.0

daily\_vaccinations total\_vaccinations\_per\_hundred \

0 NaN 0.00

1 1367.0 NaN

2 1367.0 NaN

3 1367.0 NaN

4 1367.0 NaN

... ... ...

86507 69579.0 57.59

86508 83429.0 58.25

86509 90629.0 58.61

86510 100614.0 59.20

86511 103751.0 59.90

people\_vaccinated\_per\_hundred people\_fully\_vaccinated\_per\_hundred \

0 0.00 NaN

1 NaN NaN

2 NaN NaN

3 NaN NaN

4 NaN NaN

... ... ...

86507 31.90 23.02

86508 32.38 23.11

86509 32.59 23.15

86510 32.97 23.20

86511 33.48 23.26

daily\_vaccinations\_per\_million \

0 NaN

1 34.0

2 34.0

3 34.0

4 34.0

... ...

86507 4610.0

86508 5528.0

86509 6005.0

86510 6667.0

86511 6874.0

vaccines \

0 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

1 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

2 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

3 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

4 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

... ...

86507 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

86508 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

86509 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

86510 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

86511 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

source\_name \

0 World Health Organization

1 World Health Organization

2 World Health Organization

3 World Health Organization

4 World Health Organization

... ...

86507 Ministry of Health

86508 Ministry of Health

86509 Ministry of Health

86510 Ministry of Health

86511 Ministry of Health

source\_website

0 https://covid19.who.int/

1 https://covid19.who.int/

2 https://covid19.who.int/

3 https://covid19.who.int/

4 https://covid19.who.int/

... ...

86507 https://www.arcgis.com/home/webmap/viewer.html...

86508 https://www.arcgis.com/home/webmap/viewer.html...

86509 https://www.arcgis.com/home/webmap/viewer.html...

86510 https://www.arcgis.com/home/webmap/viewer.html...

86511 https://www.arcgis.com/home/webmap/viewer.html...

[86512 rows x 15 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 86512 entries, 0 to 86511

DATA CLEANING:

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 country 86512 non-null object

1 iso\_code 86512 non-null object

2 date 86512 non-null object

3 total\_vaccinations 43607 non-null float64

4 people\_vaccinated 41294 non-null float64

5 people\_fully\_vaccinated 38802 non-null float64

6 daily\_vaccinations\_raw 35362 non-null float64

7 daily\_vaccinations 86213 non-null float64

8 total\_vaccinations\_per\_hundred 43607 non-null float64

9 people\_vaccinated\_per\_hundred 41294 non-null float64

10 people\_fully\_vaccinated\_per\_hundred 38802 non-null float64

11 daily\_vaccinations\_per\_million 86213 non-null float64

12 vaccines 86512 non-null object

13 source\_name 86512 non-null object

14 source\_website 86512 non-null object

dtypes: float64(9), object(6)

memory usage: 9.9+ MB

Data point starts from 2020-12-02 00:00:00 n

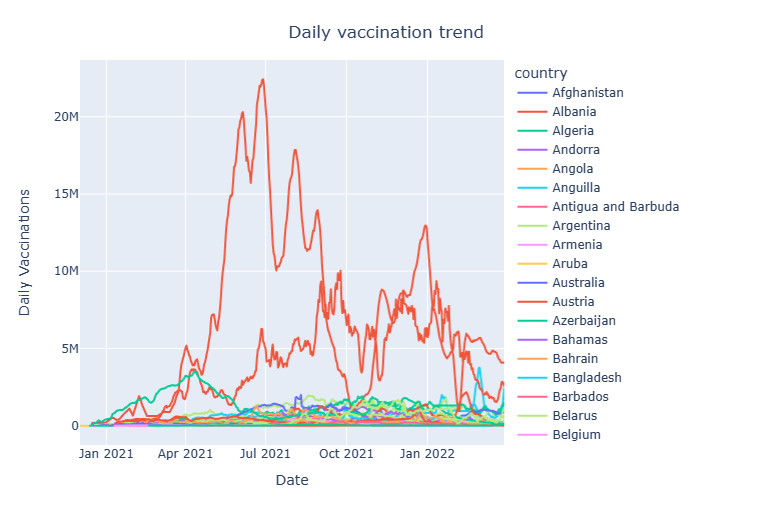
Data point ends at 2022-03-29 00:00:00 n

Total no of countries in the data set 223 n

Total no of unique vaccines in the data set 84 n

DATA VISUALIZATION:





Conclusion:

In conclusion, this analysis has demonstrated the pivotal role of data preprocessing in COVID vaccine research. Through meticulous data cleaning, transformation, and feature engineering, researchers can extract meaningful insights that inform vaccine distribution, efficacy assessments, and public health policies. As we navigate the complexities of the ongoing pandemic, it is imperative that we continue to invest in advanced data preprocessing techniques. By doing so, we can ensure that the analyses conducted on COVID vaccine-related data are not only accurate but also instrumental in guiding evidence-based decisions. The future of our fight against COVID-19 lies in the hands of researchers who harness the power of well-preprocessed data to drive innovation, foster understanding, and ultimately save lives.

TEAMMATES

K.S.SRINITHI

S.LEENA

C.ELAKKIYA

N.LOGESWARI

P.ANUCIYA