**DISASTER RECOVERY WITH IBM CLOUD VIRTUAL SERVERS**

**INTRODUCTION:**

The Energy Consumption Data Analysis and Visualization program is designed for the analysis and

visualization of energy consumption data. This program processes and visualizes data from multiple sources,

enabling insights into energy consumption patterns and trends.

**DATA SOURCES:**

The program loads disaster recovery data from various CSV files, each representing datetime

Recovery for different regions and time.

**DATA ANALYSIS:**

**Data Merge:**

1. Once you have the necessary data backups, the next step is to merge the data into a coherent and usable form. This can be a complex process, as it often involves reconciling changes made to data since the last backup. The merge process may include dealing with conflicts, duplicate data, and ensuring data integrity.

**2. Data Preprocessing:**

**a.** It converts the 'Datetime' column to a datetime format and handles missing values, preparing the data for analysis**.**

**Map Recovery Strategies to RTO and RPO:**

1. Develop recovery strategies and procedures that align with the RTO and RPO requirements. The choice of recovery strategy, such as backups, failover, or other mechanisms, will depend on these objectives. For example, if your RTO is 2 hours, you may choose to implement a hot standby server that can take over quickly in case of a failure.

**Critical Database: RTO = 4 hours, RPO = 15 minutes**

**Email System: RTO = 2 hours, RPO = 1 hour**

**VISUALIZATION:**

The program offers the following visualization tools:

**1. Histogram:**

**a**. Shows the distribution of backup(‘x’), offering a frequency-based

view of recovery levels

**2. Scatter Plot:**

**a**. Depicts the relationship between 'x' and ’y', helping identify

patterns or correlations.

**3.pH value:**

pH is effectively a measure of the concentration of hydrogen ions (that is, protons) in a substance.

**PROGRAM:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

from datetime import datetime as dt

import warnings

from matplotlib import style

from sklearn.decomposition import PCA

from sklearn.cluster import KMeans

warnings.filterwarnings('ignore')

df=pd.read\_csv('../input/weather-prediction/weather\_prediction\_dataset.csv')

df.head()

df['MONTH'].unique()

style.use('dark\_background')

plt.scatter(df['MONTH'],df['BASEL\_global\_radiation'])

style.use('Solarize\_Light2')

plt.scatter(df['BASEL\_global\_radiation'],df['BASEL\_humidity'])

df=df.drop(columns=['DATE'])

corr=df.corr()

style.use('grayscale')

plt.figure(figsize=[12,12])

sns.heatmap(corr,cmap='bwr',square=True)

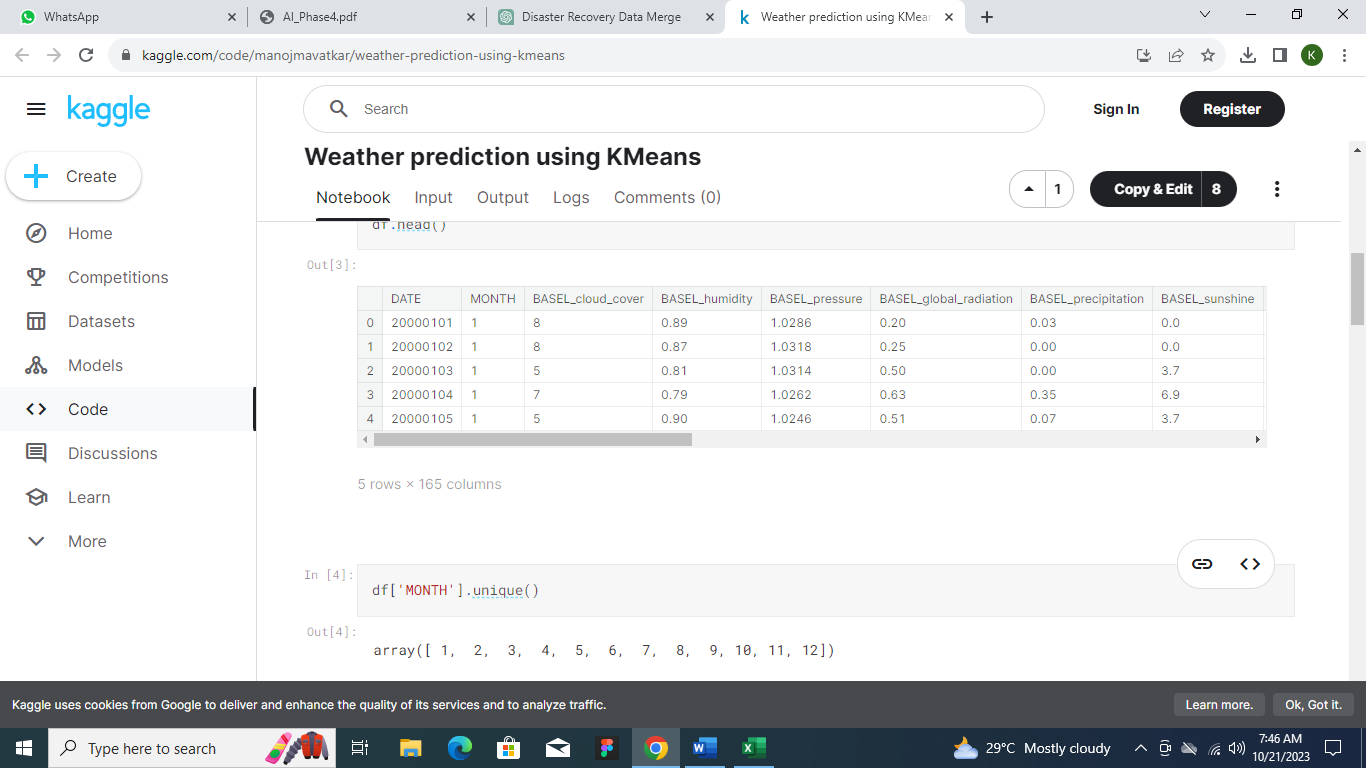
pca=PCA(n\_components=3)

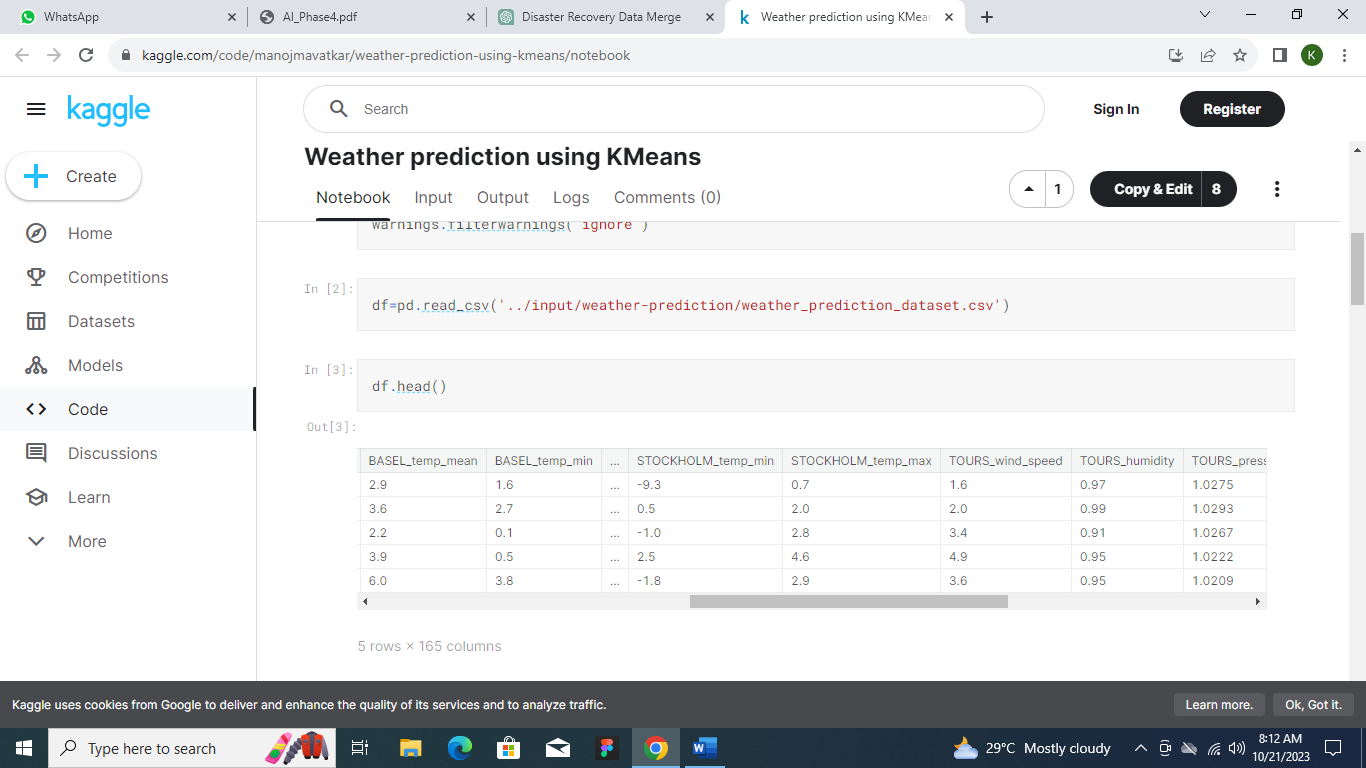
x=pca.fit\_transform(df)

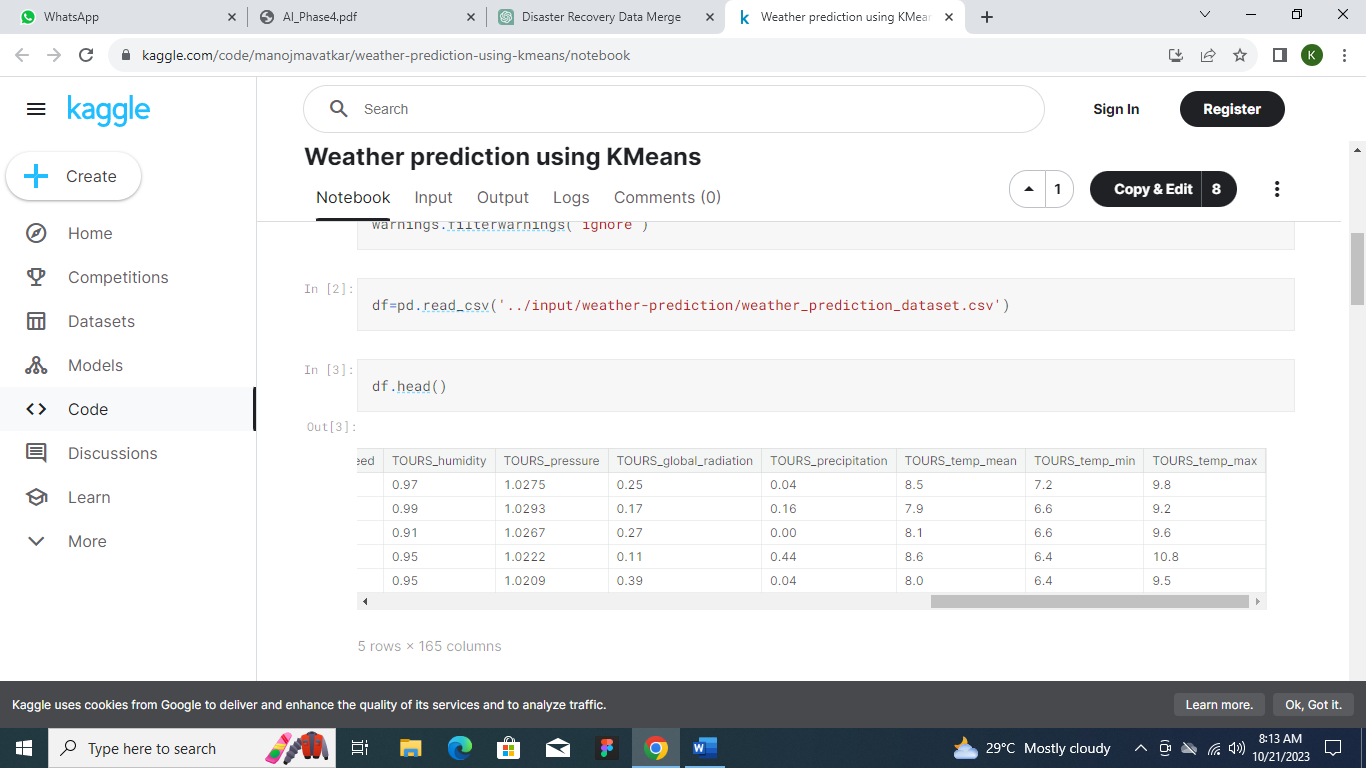
means=KMeans(n\_clusters=2,init='k-means++',max\_iter=300)

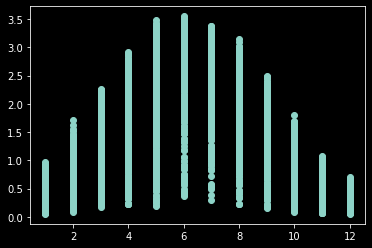
y\_predict=means.fit\_predict(x)

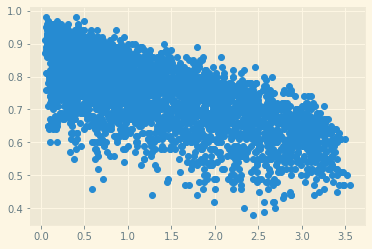
y\_predict

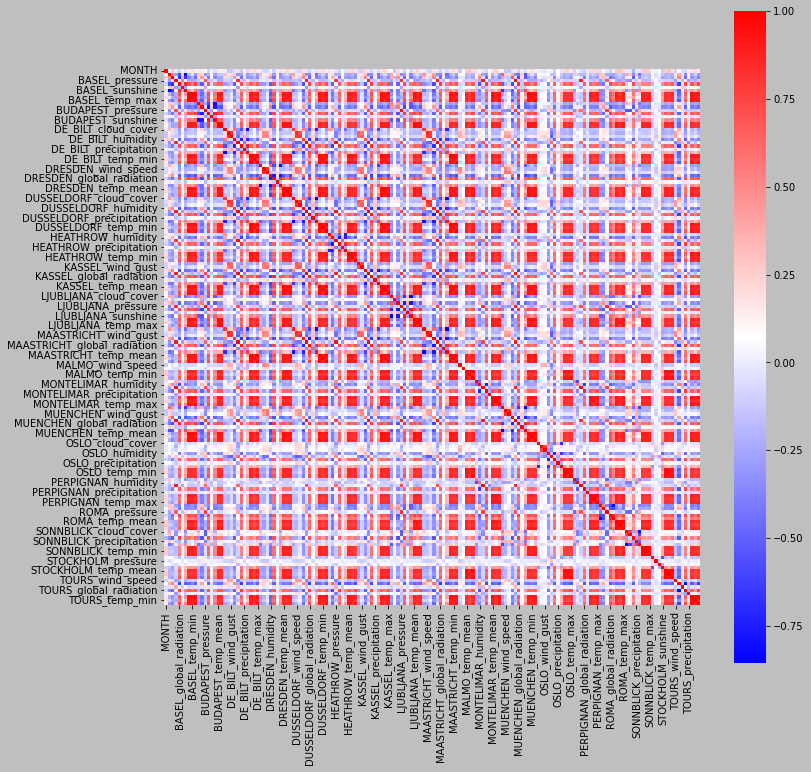
**OUTPUT**

****

****

****

****

****

**CONVERTED BY APPLYING DATASET:**

1: cloud cover in oktas

2: wind direction in degrees

3: wind speed in 1 m/s

4: wind gust in 1 m/s

5: humidity in fraction of 100 %

6: sea level pressure in 1000 h Pa

7: global radiation in 100 W/m2

8: precipitation amount in 10 mm

9: sunshine in 1 Hours

10: mean temperature in 1 &#176; C

11: minimum temperature in 1 &#176; C

12: maximum temperature in 1 &#176; C

**USAGE:**

To use the program, ensure you have the necessary libraries installed, such as pandas, matplotlib, numpy and seaborn.

Customize the program as needed to include additional plots or perform specific data analysis.

**CONCLUSION:**

The **Disaster recovery** Data Analysis and Visualization program serves as a valuable tool to backup professionals and analysts. It allows for the exploration of historical data, trend identification, and

data-informed decision-making in recovery management.