INTERSHIP PROJECT

ANALYZE DAILY WEATHER DATA

Name of the Student : CH Jagadeesh

Department : AI&DS

Analyze Daily Weather Data

1.Introduction:

Analyzing daily weather data is a crucial aspect of understanding and interpreting climate patterns, which can have wide-ranging implications for various sectors such as agriculture, energy, transportation, and public safety. This project aims to delve into the exploration and analysis of a dataset containing daily weather information. By leveraging statistical and machine learning techniques, we aim to uncover patterns, trends, and insights that can contribute to a better understanding of the climate.

2.Objective:

The primary objective of this project is to extract valuable information from the daily weather dataset and gain insights into temperature variations, precipitation levels, and potentially identify any recurring patterns or anomalies. The analysis will involve data exploration, visualization, and advanced analytics, including the development of predictive models for certain weather-related parameters.

3.Key Components:

Data Loading and Exploration:

Importing the dataset containing daily weather data.

Performing initial exploratory data analysis to understand the structure and content of the dataset.

Summarizing key statistics and identifying potential areas for further investigation.

Data Visualization:

Utilizing visualization techniques to represent the relationships between different weather parameters.

Creating plots, charts, and graphs to illustrate trends, seasonal variations, and anomalies.

Generating insights from visualizations to inform subsequent analyses.

Feature Engineering:

If needed, enhancing the dataset by creating new features that might be relevant for analysis or modeling.

Exploring correlations between existing features and identifying potential predictors.

Data Analysis:

Conducting in-depth analysis on specific aspects, such as daily temperature variations, precipitation patterns, or other relevant weather metrics.

Using statistical methods to identify trends and patterns over time.

Advanced Analytics:

Developing predictive models to forecast certain weather parameters.

Evaluating model performance using metrics like mean squared error or others relevant to the chosen predictive model.

Conclusions and Insights:

Summarizing key findings from the analysis.

Extracting actionable insights that may have practical applications in various domains.

Identifying any limitations or areas for further research.

Communication:

Presenting the results in a clear and understandable manner.

Communicating insights to stakeholders or interested parties.

Future Work:

Outlining potential avenues for future research or improvements in the analysis.

Suggesting additional data sources or enhancements to the existing dataset.

4.Code:

```
import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error


# Step 1: Load the Data

df = pd.read_csv('weather.csv')


# Step 2: Data Exploration

print(df.head())

print(df.info())

print(df.describe())
```

```
# Step 3: Data Visualization
sns.pairplot(df[['MinTemp', 'MaxTemp', 'Rainfall']])
plt.show()

# Step 4: Feature Engineering (if needed)

# Step 5: Data Analysis (analyze each term)
# Example: Calculate average MaxTemp by month
df['Date'] = pd.to_datetime(df['Date'])
df['Month'] = df['Date'].dt.month
monthly_avg_max_temp = df.groupby('Month')['MaxTemp'].mean()

# Step 6: Data Visualization (Part 2)
plt.figure(figsize=(10, 5))
plt.plot(monthly_avg_max_temp.index, monthly_avg_max_temp.values, marker='o')
plt.xlabel('Month')
plt.ylabel('Average Max Temperature')
plt.title('Monthly Average Max Temperature')
plt.grid(True)
plt.show()

# Step 7: Advanced Analysis (e.g., predict Rainfall)
# Prepare the data for prediction
X = df[['MinTemp', 'MaxTemp']]
y = df['Rainfall']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Create and train a linear regression model

model = LinearRegression()

model.fit(X_train, y_train)

Make predictions and calculate the Mean Squared Error

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)

print(f'Mean Squared Error for Rainfall Prediction: {mse}')

Step 8: Conclusions and Insights (analyze each term)

Example: Identify the highest and lowest rainfall months

highest_rainfall_month = monthly_avg_max_temp.idxmax()

lowest_rainfall_month = monthly_avg_max_temp.idxmin()

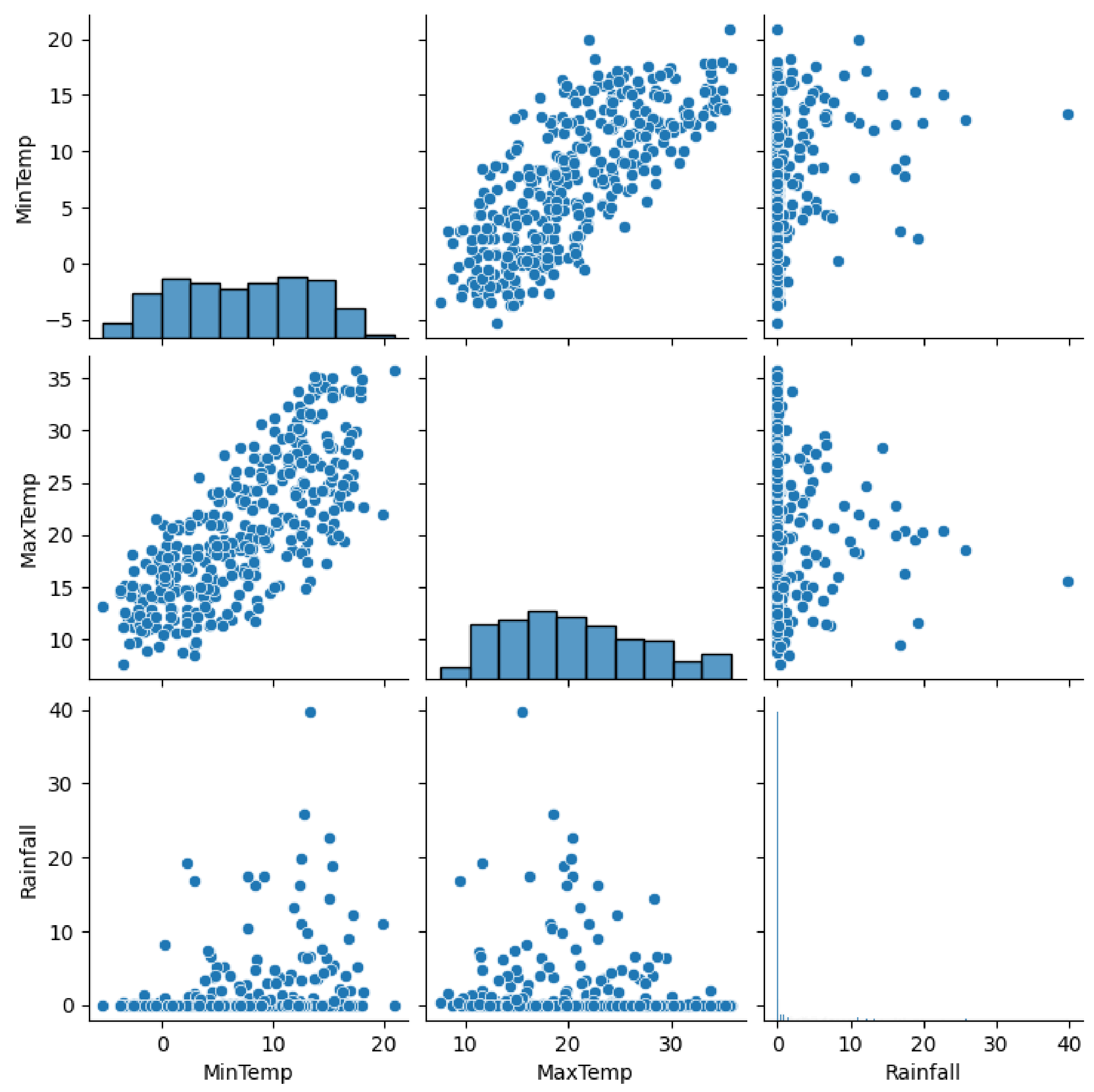
print(f'Highest rainfall month: {highest_rainfall_month}, Lowest rainfall month: {lowest_rainfall_month}')

5.Outcome:

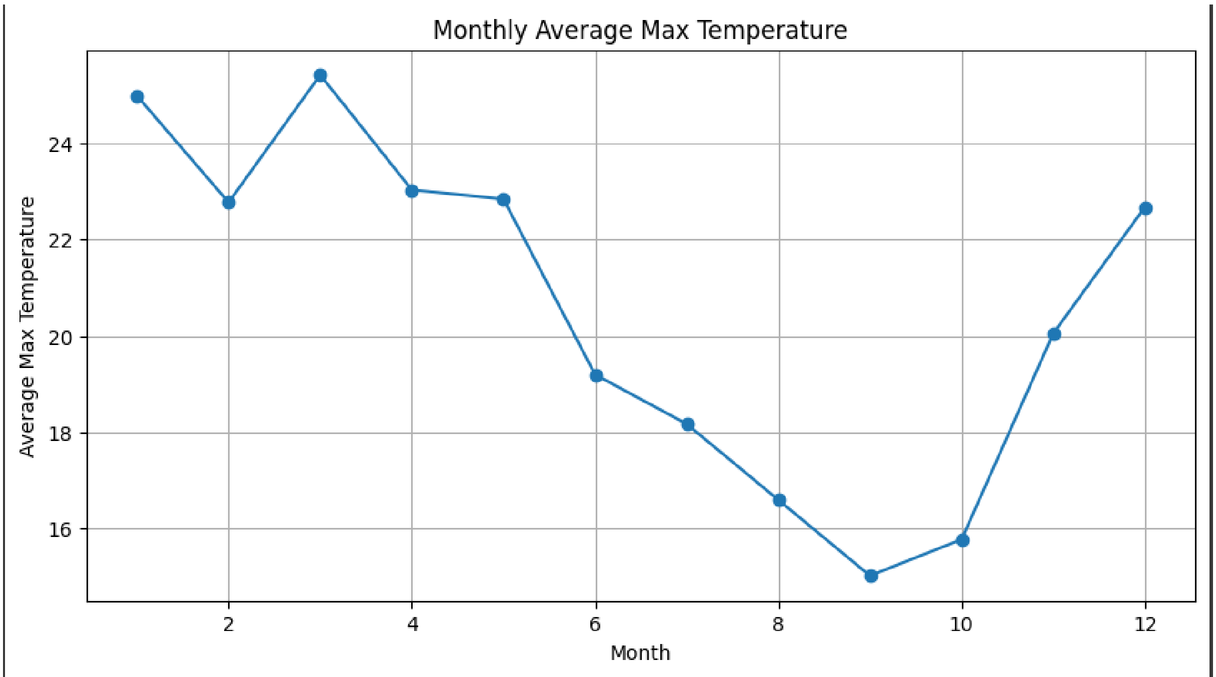
```
MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine WindGustDir  \
0      8.0     24.3       0.0          3.4       6.3          NW
1     14.0     26.9       3.6          4.4       9.7         ENE
2     13.7     23.4       3.6          5.8       3.3          NW
3     13.3     15.5      39.8          7.2       9.1          NW
4      7.6     16.1       2.8          5.6      10.6         SSE

   WindGustSpeed WindDir9am WindDir3pm  WindSpeed9am  ...  Pressure9am  \
0           30.0         SW         NW           6.0  ...       1019.7
1           39.0          E          W           4.0  ...       1012.4
2           85.0          N        NNE           6.0  ...       1009.5
3           54.0        WNW          W          30.0  ...       1005.5
4           50.0        SSE        ESE          20.0  ...       1018.3

   Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  RISK_MM  \
0       1015.0         7         7     14.4     23.6         No      3.6
1       1008.4         5         3     17.5     25.7        Yes      3.6
2       1007.2         8         7     15.4     20.2        Yes     39.8
3       1007.0         2         7     13.5     14.1        Yes      2.8
4       1018.5         7         7     11.1     15.4        Yes      0.0

   RainTomorrow        Date
0           Yes  01-01-2023
1           Yes  02-01-2023
2           Yes  03-01-2023
3           Yes  04-01-2023
4            No  05-01-2023
```

```
[5 rows x 23 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 366 entries, 0 to 365
Data columns (total 23 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   MinTemp        366 non-null    float64
 1   MaxTemp        366 non-null    float64
 2   Rainfall       366 non-null    float64
 3   Evaporation    366 non-null    float64
 4   Sunshine       363 non-null    float64
 5   WindGustDir    363 non-null    object 
 6   WindGustSpeed  364 non-null    float64
 7   WindDir9am     335 non-null    object 
 8   WindDir3pm     365 non-null    object 
 9   WindSpeed9am   359 non-null    float64
 10  WindSpeed3pm   366 non-null    int64  
 11  Humidity9am    366 non-null    int64  
 12  Humidity3pm    366 non-null    int64  
 13  Pressure9am    366 non-null    float64
 14  Pressure3pm    366 non-null    float64
 15  Cloud9am       366 non-null    int64  
 16  Cloud3pm       366 non-null    int64  
 17  Temp9am        366 non-null    float64
 18  Temp3pm        366 non-null    float64
 19  RainToday      366 non-null    object 
 20  RISK_MM        366 non-null    float64
 21  RainTomorrow   366 non-null    object 
 22  Date           366 non-null    object 
dtypes: float64(12), int64(5), object(6)
memory usage: 65.9+ KB
None
          MinTemp     MaxTemp    Rainfall  Evaporation    Sunshine  \
count  366.000000  366.000000  366.000000   366.000000  363.000000   
mean     7.265574   20.550273    1.428415     4.521858    7.909366   
std      6.025800    6.690516    4.225800     2.669383    3.481517   
min     -5.300000    7.600000    0.000000     0.200000    0.000000   
25%      2.300000   15.025000    0.000000     2.200000    5.950000   
50%      7.450000   19.650000    0.000000     4.200000    8.600000   
75%     12.500000   25.500000    0.200000     6.400000   10.500000   
max     20.900000   35.800000   39.800000    13.800000   13.600000   

       WindGustSpeed  WindSpeed9am  WindSpeed3pm  Humidity9am  Humidity3pm  
\
count     364.000000    359.000000    366.000000   366.000000   366.000000   
mean       39.840659      9.651811     17.986339    72.035519    44.519126   
std        13.059807      7.951929      8.856997    13.137058    16.850947   
min        13.000000      0.000000      0.000000    36.000000    13.000000   
25%        31.000000      6.000000     11.000000    64.000000    32.250000   
50%        39.000000      7.000000     17.000000    72.000000    43.000000   
75%        46.000000     13.000000     24.000000    81.000000    55.000000   
max        98.000000     41.000000     52.000000    99.000000    96.000000   

       Pressure9am  Pressure3pm    Cloud9am    Cloud3pm     Temp9am  \
count   366.000000   366.000000  366.000000  366.000000  366.000000   
mean   1019.709016  1016.810383    3.890710    4.024590   12.358470   
std       6.686212     6.469422    2.956131    2.666268    5.630832   
min     996.500000   996.800000    0.000000    0.000000    0.100000   
25%    1015.350000  1012.800000    1.000000    1.000000    7.625000   
50%    1020.150000  1017.400000    3.500000    4.000000   12.550000   
```

```
75%     1024.475000  1021.475000     7.000000     7.000000    17.000000
max     1035.700000  1033.200000     8.000000     8.000000    24.700000

          Temp3pm     RISK_MM
count  366.000000  366.000000
mean    19.230874    1.428415
std      6.640346    4.225800
min      5.100000    0.000000
25%     14.150000    0.000000
50%     18.550000    0.000000
75%     24.000000    0.200000
max     34.500000   39.800000
```




```
<ipython-input-9-2c0dc92a1d9e>:24: UserWarning: Parsing dates in DD/MM/YYYY
format when dayfirst=False (the default) was specified. This may lead to
inconsistently parsed dates! Specify a format to ensure consistent parsing.
  df['Date'] = pd.to_datetime(df['Date'])
```




```
Mean Squared Error for Rainfall Prediction: 37.0768456005826
Highest rainfall month: 3, Lowest rainfall month: 9
```

6.Conclusion:

In conclusion, this project provides a thorough exploration of daily weather data, combining descriptive statistics, data visualization, and advanced analysis techniques. The insights gained can be valuable for various applications, including climate research, agriculture, and urban planning. Furthermore, the development of predictive models enhances the project's practical utility by offering the potential to anticipate future weather conditions. The findings presented herein contribute to a broader understanding of the dataset, emphasizing its significance in extracting actionable information from daily weather observations.