NEXUS SOFTWARE
DATA ANALYSIS-PROJECT_2

Project Title : Weather Analysis

Data Preparation with Python:

Data Cleaning:

```
import pandas as pd
import numpy as np
import folium
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder
import seaborn as sns
```

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

DATA CLEANING AND DATA PREPROCESSING

```
df.head()
```

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	...
0	8.0	24.3	0.0	3.4	6.3	NW	30.0	SW	NW	6.0	...
1	14.0	26.9	3.6	4.4	9.7	ENE	39.0	E	W	4.0	...
2	13.7	23.4	3.6	5.8	3.3	NW	85.0	N	NNE	6.0	...
3	13.3	15.5	39.8	7.2	9.1	NW	54.0	WNW	W	30.0	...
4	7.6	16.1	2.8	5.6	10.6	SSE	50.0	SSE	ESE	20.0	...

5 rows × 22 columns

```
[10] df.tail()
```

	MinTemp	MaxTemp	Rainfall	Evaporation	Sunshine	WindGustDir	WindGustSpeed	WindDir9am	WindDir3pm	WindSpeed9am	..
361	9.0	30.7	0.0	7.6	12.1	NNW	76.0	SSE	NW	7.0	
362	7.1	28.4	0.0	11.6	12.7	N	48.0	NNW	NNW	2.0	
363	12.5	19.9	0.0	8.4	5.3	ESE	43.0	ENE	ENE	11.0	
364	12.5	26.9	0.0	5.0	7.1	NW	46.0	SSW	WNW	6.0	
365	12.3	30.2	0.0	6.0	12.6	NW	78.0	NW	WNW	31.0	

5 rows × 22 columns

```python
df.isnull().sum()
```

```
MinTemp           0
MaxTemp           0
Rainfall          0
Evaporation       0
Sunshine          3
WindGustDir       3
WindGustSpeed     2
WindDir9am       31
WindDir3pm        1
WindSpeed9am      7
WindSpeed3pm      0
Humidity9am       0
Humidity3pm       0
Pressure9am       0
Pressure3pm       0
Cloud9am          0
Cloud3pm          0
Temp9am           0
Temp3pm           0
RainToday         0
RISK_MM           0
RainTomorrow      0
dtype: int64
```

```python
[14] df_cleaned = df.dropna()
```

```python
[15] numeric_cols = df.select_dtypes(include=np.number).columns
     df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())
```

```python
categorical_cols = df.select_dtypes(include='object').columns
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().iloc[0])
```

```python
label_encoders = {}
for col in categorical_cols:
    label_encoders[col] = LabelEncoder()
    df[col] = label_encoders[col].fit_transform(df[col])
```

```
print("\nPreprocessed Dataset:")
print(df.head())
```

```
Preprocessed Dataset:
   MinTemp  MaxTemp  Rainfall  Evaporation  Sunshine  WindGustDir  \
0      8.0     24.3       0.0          3.4       6.3            7
1     14.0     26.9       3.6          4.4       9.7            1
2     13.7     23.4       3.6          5.8       3.3            7
3     13.3     15.5      39.8          7.2       9.1            7
4      7.6     16.1       2.8          5.6      10.6           10

   WindGustSpeed  WindDir9am  WindDir3pm  WindSpeed9am  ...  Humidity3pm  \
0           30.0          12           7           6.0  ...           29
1           39.0           0          13           4.0  ...           36
2           85.0           3           5           6.0  ...           69
3           54.0          14          13          30.0  ...           56
4           50.0          10           2          20.0  ...           49

   Pressure9am  Pressure3pm  Cloud9am  Cloud3pm  Temp9am  Temp3pm  RainToday  \
0       1019.7       1015.0         7         7     14.4     23.6          0
1       1012.4       1008.4         5         3     17.5     25.7          1
2       1009.5       1007.2         8         7     15.4     20.2          1
3       1005.5       1007.0         2         7     13.5     14.1          1
4       1018.3       1018.5         7         7     11.1     15.4          1

   RISK_MM  RainTomorrow
0      3.6             1
1      3.6             1
2     39.8             1
3      2.8             1
4      0.0             0

[5 rows x 22 columns]
```

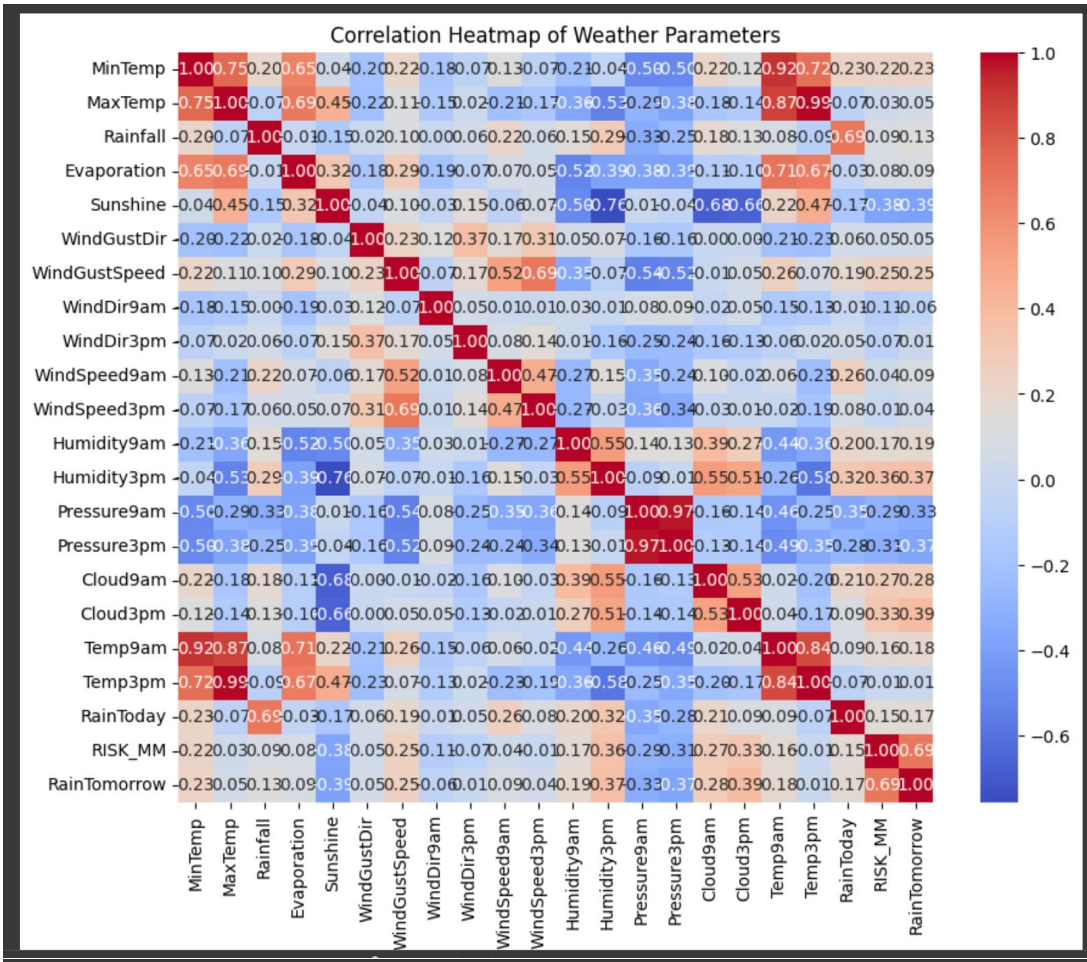
Correlation and Regression Analysis:

CORRELATION AND REGRESSION ANALYSIS

```
correlation_matrix = df.corr()
```

Plotting correlation heatmap

```
plt.figure(figsize=(10, 8))
sns.heatmap(correla Loading... x, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of Weather Parameters')
plt.show()
```




```
[27] X = df.drop(columns=['MinTemp'])
     y = df['MinTemp']
```

```
[30] X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[31] model = LinearRegression()
     model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

```
y_pred = model.predict(X_test)
```

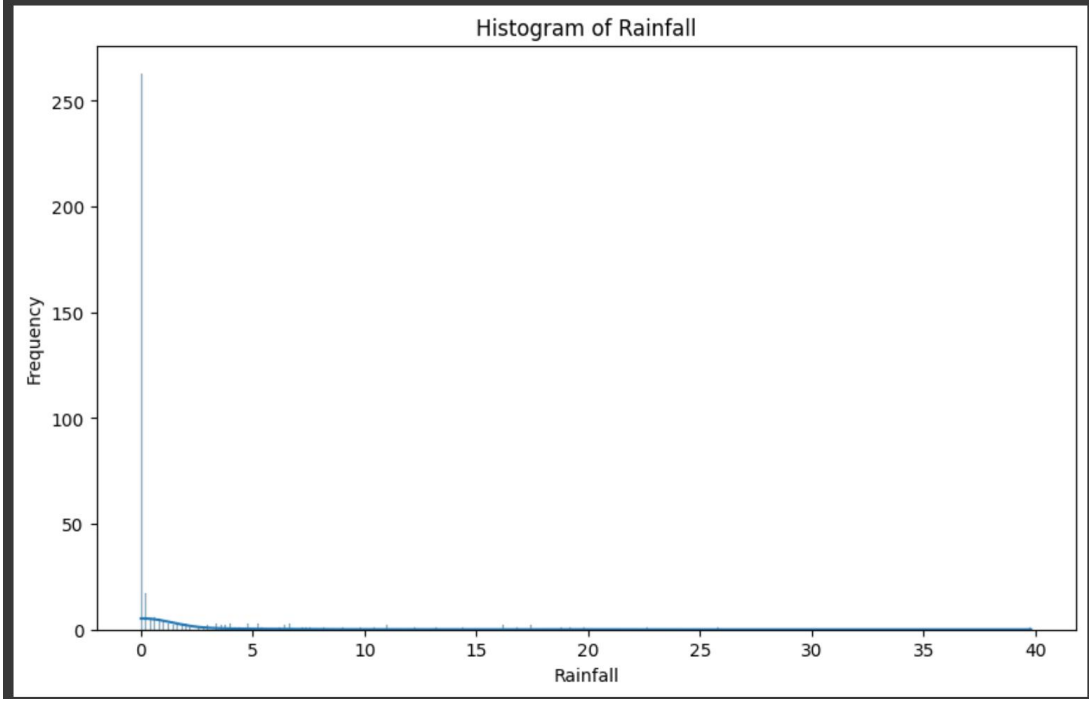
```
[33] mse = mean_squared_error(y_test, y_pred)
     r2 = r2_score(y_test, y_pred)
```

```
[35] print("\nRegression Analysis Results:")
     print("Mean Squared Error (MSE):", mse)
     print("R-squared (R2):", r2)
```

```
Regression Analysis Results:
Mean Squared Error (MSE): 3.0786292078126634
R-squared (R2): 0.9152462967416777
```

Histogram of Rainfall:

```
plt.figure(figsize=(10, 6))
sns.histplot(df['Rainfall'], kde=True)
plt.title('Histogram of Rainfall')
plt.xlabel('Rainfall')
plt.ylabel('Frequency')
plt.show()
```



Barplot of Raintoday:

```
plt.figure(figsize=(8, 6))
sns.countplot(df, x='RainToday')
plt.title('Bar Plot of RainToday')
plt.xlabel('Rain Today')
plt.ylabel('Count')
plt.show()
```