## 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	A CLE	ANING A	ND DATA F	PREPROCES	SING							
0	df.h	ead()										
글	М	linTemp	MaxTemp	Rainfall E	Evaporation :	Sunshine W	VindGustDir	WindGustSpeed \	WindDir9am \	WindDir3pm \	WindSpeed9am	
	0	8.0	24.3	0.0	3.4	6.3	NW	30.0	sw	NW	6.0	
		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		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	
		s×22 colu ail()	mns									
		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	
		12.5	26.9	0.0	5.0	7.1	NW	46.0	ssw	WNW	6.0	
	364	12.5	20.0									
	364 365	12.3			6.0	12.6	NW	78.0	NW	WNW	31.0	

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

→ MinTemp

    MaxTemp
                      0
    Rainfall
    Evaporation
                    0
                      3
    Sunshine
    WindGustDir
                      3
                     2
    WindGustSpeed
    WindDir9am
    WindDir3pm
    WindSpeed9am
    WindSpeed3pm
                      0
    Humidity9am
                      0
    Humidity3pm
                      0
    Pressure9am
                      0
    Pressure3pm
                      0
    Cloud9am
                      0
    Cloud3pm
                     0
    Temp9am
                     0
                      0
    Temp3pm
    RainToday
                      0
    RISK_MM RainTomorrow
                      0
                      0
    dtype: int64
[14] df_cleaned = df.dropna()
[15] numeric_cols = df.select_dtypes(include=np.number).columns
    df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())
```

```
categorical_cols = df.select_dtypes(include='object').columns
df[categorical_cols] = df[categorical_cols].fillna(df[categorical_cols].mode().i
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 \
      8.0
               24.3
                        0.0 3.4
                                                      6.3
1
      14.0
                26.9
                             3.6
                                           4.4
                                                      9.7
      13.7
                23.4
                             3.6
                                          5.8
                                                      3.3
               15.5
                                          7.2
3
      13.3
                           39.8
                                                      9.1
4
                           2.8
                                         5.6
       7.6
               16.1
                                                     10.6
                                                                      10
   WindGustSpeed WindDir9am WindDir3pm WindSpeed9am ... Humidity3pm \
30.0 12 7 6.0 ... 29
39.0 0 13 4.0 ... 36
85.0 3 5 6.0 ... 69
54.0 14 13 30.0 ... 56
50.0 10 2 20.0 ... 49
0
2
   Pressure9am Pressure3pm Cloud9am Cloud3pm Temp9am Temp3pm RainToday \
        1019.7
                     1015.0
                                                          14.4
0
                                                                   23.6
                                                                                   0
                                                                    25.7
1
         1012.4
                       1008.4
                                                          17.5
2
3
         1009.5
                                                          15.4
                                                                    20.2
                       1007.2
                                      8
        1005.5
                       1007.0
                                                         13.5
                                                                   14.1
        1018.3
                       1018.5
                                                                   15.4
                                                          11.1
   RISK_MM RainTomorrow
0
       3.6
1
2
3
        3.6
      39.8
       2.8
        0.0
                         0
[5 rows x 22 columns]
```

#### **Correlation and Regression Analysis:**

```
CORRELATION AND REGRESSION ANALYSIS

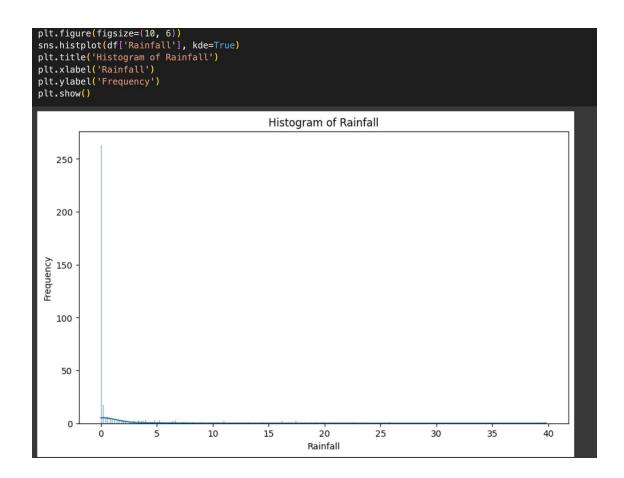
Correlation_matrix = df.corr()

Plotting correlation heatmap

plt.figure(figsize='1a_oll sns.heatmap(correla_Loading... x, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Heatmap of Weather Parameters')
plt.show()
```

```
Correlation Heatmap of Weather Parameters
                                                                                                                        1.0
        MinTemp -1.000.750.200.650.040.200.220.180.070.130.070.210.040.560.560.220.120.920.720.230.220.23
       MaxTemp -0.751.000.070.690.450.220.110.150.020.210.170.360.530.290.360.180.140.870.990.070.030.05
         Rainfall -0.200.071.000.010.150.020.100.000.060.220.060.150.290.330.250.180.130.080.050.690.090.13
                                                                                                                       - 0.8
     Evaporation -0.650.690.011.000.320.180.290.190.070.070.050.520.390.380.340.110.100.710.670.030.080.09
        Sunshine -0.04<mark>0.45</mark>0.150.32<mark>1.00</mark>0.040.100.030.150.060.070.560.760.010.040.660.66<mark>0.220.47</mark>0.170.380.3
                                                                                                                        0.6
    WindGustDir -0.200.220.020.180.041.000.230.120.370.170.310.050.070.160.160.000.000.210.230.060.050.05
 WindGustSpeed -0.220.110.100.290.100.23<mark>1.00</mark>0.070.170.520.690.350.070.540.520.010.050.260.070.190.250.25
    WindDir9am -0.180.150.000.190.030.120.071.000.050.010.010.030.010.080.090.020.050.150.130.010.110.06
                                                                                                                       - 0.4
    WindDir3pm -0.070.020.060.070.15<mark>0.370.170.05</mark>1.000.080.140.010.160.250.240.160.130.060.020.050.070.01
 WindSpeed9am -0.130.210.220.070.060.170.520.010.081.000.470.270.150.350.240.100.020.060.230.260.040.09
                                                                                                                       - 0.2
 WindSpeed3pm -0.070.170.060.050.070.310.690.010.140.471.000.270.030.350.340.030.010.020.190.080.010.04
   Humidity9am -0.210.3c0.150.520.5c0.050.3c0.030.010.270.271.000.550.140.130.390.270.440.3c0.200.170.19
   Humidity3pm -0.040.5:0.290.390.760.070.070.010.160.150.030.551.060.090.010.550.510.260.50.320.360.37
                                                                                                                       - 0.0
   Pressure9am =0.560.290.330.380.010.160.540.080.250.350.360.140.091.000.970.160.140.460.250.350.290.33
    Pressure3pm -0.560.3e0.250.3e0.040.160.52<mark>0.09</mark>0.240.240.340.130.01<mark>0.971.00</mark>0.130.140.490.350.280.310.3
                                                                                                                        -0.2
      Cloud9am -0.220.180.180.110.680.090.010.020.160.190.030.390.550.160.131.000.530.020.200.210.270.28
      Cloud3pm -0.120.140.130.100.600.000.050.050.130.020.010.270.510.140.140.531.000.040.170.090.330.39
      Temp9am -0.920.870.080.710.220.210.260.150.060.060.020.440.260.450.450.020.041.000.840.090.160.18
                                                                                                                         -0.4
      Temp3pm -0.720.990.090.670.470.230.070.130.020.230.190.360.520.250.350.260.170.841.000.070.010.01
      RainToday -0.230.07<mark>0.69</mark>0.030.170.060.190.010.050.260.080.200.320.350.280.210.090.090.071.000.150.17
                                                                                                                         -0.6
       RISK MM -0.220.030.090.080.380.050.250.110.070.040.010.170.360.290.310.270.330.160.010.151.000.6
  RainTomorrow -<mark>0.23</mark>0.050.130.09<mark>0.35</mark>0.05<mark>0.25</mark>0.060.010.090.040.19<mark>0.37</mark>0.330.370.280.390.180.010.17
                                                                                    Cloud3pm
                                                                    Humidity3pm
                   MinTemp
                                                WindDir9am
                                                   WindDir3pm
                                                                Humidity9am
                                                                                                          RainTomorrow
                                                            VindSpeed3pm
[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:



### **Barplot of Raintoday:**

