THE QUESTION

```
import pandas as pd
data=pd.read_csv("C:/Users/jjgj/Downloads/Data.csv")
data.head(5)
```

	Year	Month	Day	TEM	MXT	MNT	HUM	SLP	WIS	RIN	SSH	CLD
0	1980	1	1	12.5	26.0	12.2	69	1013.5	3.3	0.0	7.9	0
1	1980	1	2	14.4	21.2	9.5	85	1014.0	8.0	0.0	0.8	6
2	1980	1	3	15.3	22.2	14.7	84	1015.0	5.0	8.0	4.1	3
3	1980	1	4	12.4	24.0	10.0	73	1016.8	4.0	0.0	8.4	1
4	1980	1	5	12.0	23.9	12.0	72	1016.1	8.0	0.0	9.0	2

TEM=temperature

MXT=maximum temperature

MNT=minimum temperature

HUM=humidity

SLP=sea level pressure

WIS= wind speed

RIN=rainfall

SSH= sunshine hour

CLD=Cloud

The dataset **Data.csv** contains daily atmospheric information (TEM, MXT, MNT, HUM, SLP, WIS, RIN, SSH, CLD) for the period **January 1980 to March 2022** for the location **Jessore, Bangladesh**.

Tasks:

1. **Perform exploratory data analysis (EDA)** on the dataset after taking necessary actions to handle missing data.
2. **Fit a neural network model** to predict **maximum temperature (MXT)** based on the other atmospheric parameters.
3. Before fitting the model, **normalize all variables** using the transformation

$$\text{Scaled Value} = 0.1 + 0.8 \times \frac{\text{Actual} - \text{Minimum}}{\text{Maximum} - \text{Minimum}}$$

4. Use **80% of the data as training data**, and the rest of data as the test data
5. Present the **prediction accuracy** of the neural network model **separately for the training and test data**.

Represents the prediction accuracy results of the models for training data and test data independently. Also, represent the scatter plot for the actual maximum rainfall and the predicted maximum rainfall for training data,test data, overall data. Discuss your results.

Use your exam roll as a seed point (example: 122) to draw the training data.

##################Python Code:

```
# ================================
# IMPORT LIBRARIES
# ================================
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# ================================
# LOAD DATA
# ================================
data = pd.read_csv("C:/Users/jjgj/Downloads/Data.csv")

# ================================
# 1. EXPLORATORY DATA ANALYSIS (EDA)
# ================================
print("\n===== BASIC INFO =====")
print(data.info())

print("\n===== SUMMARY STATISTICS =====")
print(data.describe())

print("\n===== MISSING VALUES =====")
```

```
print(data.isnull().sum())


# ---- Handle missing values (Impute with mean) ----
data.fillna(data.mean(numeric_only=True), inplace=True)

print("\n===== AFTER IMPUTATION =====")
print(data.isnull().sum())



# ================================
# 2. NORMALIZATION USING GIVEN SCALING
#     Scaled = 0.1 + 0.8*(x - xmin)/(xmax - xmin)
# ================================
scaled = data.copy()

min_vals = scaled.min()
max_vals = scaled.max()

scaled = 0.1 + 0.8 * (scaled - min_vals) / (max_vals - min_vals)



# ================================
# 3. TRAIN–TEST SPLIT (80–20)
#    Seed = exam roll number → example: 122
# ================================
seed = 122   # change to your exam roll
```

```
X = scaled.drop(columns=['MXT'])
y = scaled['MXT']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=seed
)



# ================================
# 4. NEURAL NETWORK MODEL
# ================================
model = MLPRegressor(hidden_layer_sizes=(20, 20),
                     activation='relu',
                     solver='adam',
                     max_iter=1000,
                     random_state=seed)

model.fit(X_train, y_train)

# ================================
# 5. PREDICTIONS & ACCURACY
# ================================
y_pred_train = model.predict(X_train)
y_pred_test  = model.predict(X_test)
```

```
# ---- Training accuracy ----
rmse_train = np.sqrt(mean_squared_error(y_train, y_pred_train))
mae_train  = mean_absolute_error(y_train, y_pred_train)
r2_train   = r2_score(y_train, y_pred_train)

# ---- Test accuracy ----
rmse_test = np.sqrt(mean_squared_error(y_test, y_pred_test))
mae_test  = mean_absolute_error(y_test, y_pred_test)
r2_test   = r2_score(y_test, y_pred_test)

print("\n========= TRAINING ACCURACY =========")
print("RMSE:", rmse_train)
print("MAE :", mae_train)
print("R²  :", r2_train)

print("\n========= TEST ACCURACY =========")
print("RMSE:", rmse_test)
print("MAE :", mae_test)
print("R²  :", r2_test)

# ================================
# 6. SCATTER PLOTS
# ================================
plt.figure(figsize=(15,4))

# ---- Train ----
```

```
plt.subplot(1,3,1)
plt.scatter(y_train, y_pred_train, s=8, color='blue')
plt.xlabel("Actual MXT")
plt.ylabel("Predicted MXT")
plt.title("Training Data")

# ---- Test ----
plt.subplot(1,3,2)
plt.scatter(y_test, y_pred_test, s=8, color='red')
plt.xlabel("Actual MXT")
plt.ylabel("Predicted MXT")
plt.title("Test Data")

# ---- Overall ----
plt.subplot(1,3,3)
plt.scatter(y, model.predict(X), s=8, color='green')
plt.xlabel("Actual MXT")
plt.ylabel("Predicted MXT")
plt.title("Overall Data")

plt.tight_layout()
plt.show()
```