Title :Weather Prediction

Code :

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as seabornInstance

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn import metrics

%matplotlib inline

dataset = pd.read\_csv("C:/Users/HP/Desktop/Weather.csv")

dataset.head()

dataset.describe()

dataset.plot(x='MinTemp', y='MaxTemp', style='o')

plt.title('MinTemp vs MaxTemp')

plt.xlabel('MinTemp')

plt.ylabel('MaxTemp')

plt.show()

plt.figure(figsize=(15,10))

plt.tight\_layout()

seabornInstance.distplot(dataset['MaxTemp'])

X = dataset['MinTemp'].values.reshape(-1,1)

y = dataset['MaxTemp'].values.reshape(-1,1)

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

regressor = LinearRegression()

regressor.fit(X\_train, y\_train) #training the algorithm

#To retrieve the intercept:

print(regressor.intercept\_)

#For retrieving the slope:

print(regressor.coef\_)

y\_pred = regressor.predict(X\_test)

df = pd.DataFrame({'Actual': y\_test.flatten(), 'Predicted': y\_pred.flatten()})

df

df1 = df.head(25)

df1.plot(kind='bar',figsize=(16,10))

plt.grid(which='major', linestyle='-', linewidth='0.5', color='green')

plt.grid(which='minor', linestyle=':', linewidth='0.5', color='black')

plt.show()

plt.scatter(X\_test, y\_test, color='gray')

plt.plot(X\_test, y\_pred, color='red', linewidth=2)

plt.show()

print('Mean Absolute Error:', metrics.mean\_absolute\_error(y\_test, y\_pred))

print('Mean Squared Error:', metrics.mean\_squared\_error(y\_test, y\_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean\_squared\_error(y\_test, y\_pred)))

ScreenShots :















