

Nama : Stephen Prasetya Chrismawan

NIM : H1D021025

UTS Machine Learning (Regresi Linear)

Penjelasan kode, sebagai berikut.

1. Mengimpor library yang dibutuhkan

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression

import numpy as np

from sklearn.metrics import r2_score
```

2. Mengimpor dataset dan memasukkan ke array

```
df = pd.read_csv("DailyDelhiClimateTrain.csv")
```

3. Membuat array msk, yang berisi random 1 dan 0 sebanyak panjang dataset, dengan kemungkinan muncul biner 1 sebanyak 0.8. Tujuannya adalah untuk memilih secara acak array df, sebanyak 80% akan dijadikan data training dan 20% sisanya dijadikan data test

```
msk = np.random.rand(len(df)) < 0.8

train = df[msk]

test = df[~msk]

print(f"\nSelamat DATANG. Aplikasi Regresi untuk menghitung
prediksi Temperatur Rata-Rata berdasarkan nilai Humidity ")
```

4. Membuat fungsi mean_absolute_error untuk menghitung rata rata data yang menyimpan dibanding data aslinya.

```
def mean_absolute_error(y, y_predict):

    n = len(y)

    res = 0

    for i in range(n):

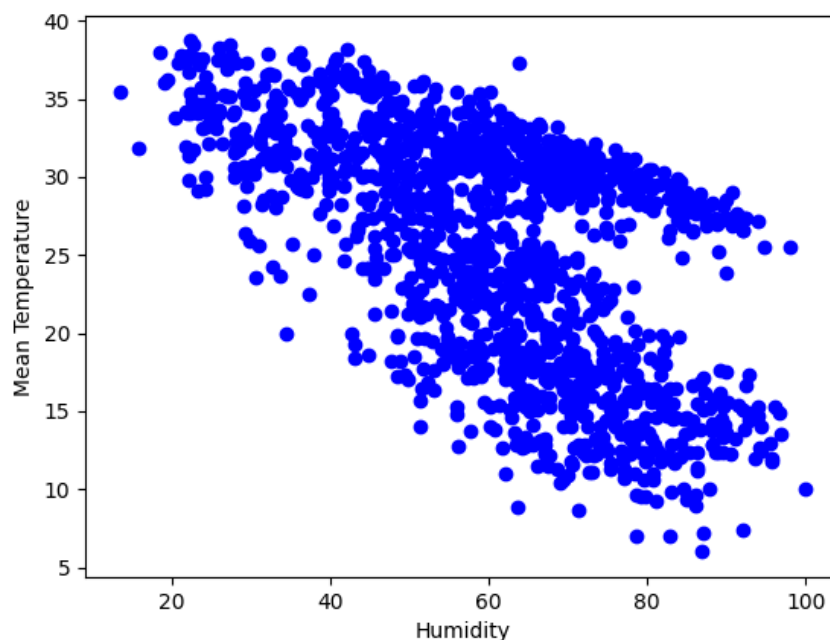
        res += abs(y[i] - y_predict[i])

    return res / n
```

5. Membuat fungsi `mean_absolute_percentage_error` untuk melihat berapa persen rata rata error atau ketidak cocokan data prediksi dengan data asli.

```
def mean_absolute_percentage_error(y, y_predict):  
    n = len(y)  
    res = 0  
    for i in range(n):  
        res += abs((y[i] - y_predict[i])/y[i])  
    return res / n
```

6. Selanjutnya adalah membuat fitting data ke dalam fungsi regresi. Pertama, membuat variabel `regr` sebagai representasi `LinearRegression`. Kemudian saya memilih train x yaitu humidity dan train y yaitu meantemp karena menurut saya memiliki keterkaitan antar data, semakin tinggi humidity semakin rendah rata rata temperature



Langkah selanjutnya yaitu masukkan data x train dan y train ke dalam fungsi fitting di regresi linear, supaya dapat ditemukan persamaan linear yang sesuai. Kemudian melakukan prediksi terhadap data train.

```
#membuat model regresi
```

```

regr = LinearRegression()

train_x = np.asanyarray(train[['humidity']])

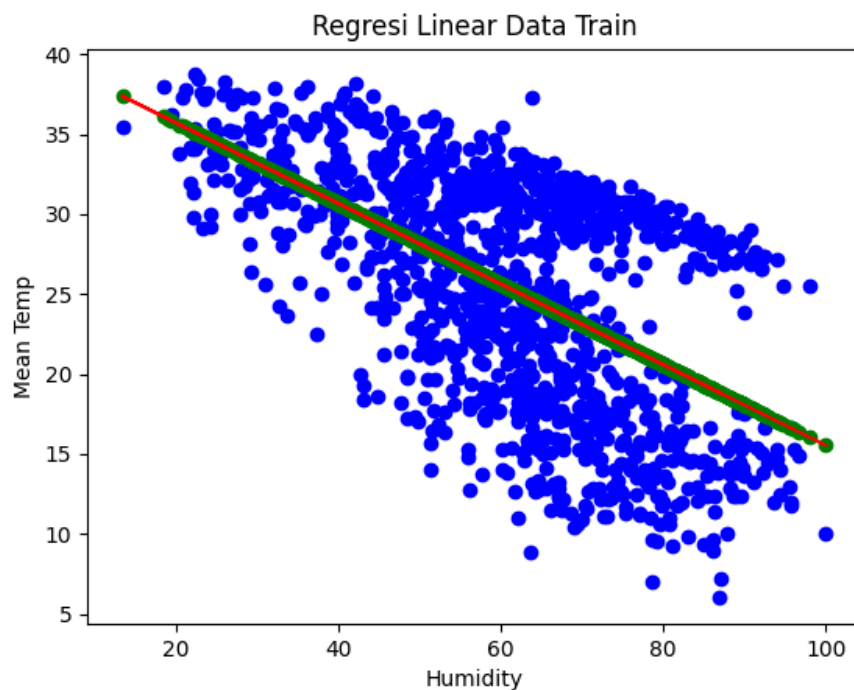
train_y = np.asanyarray(train[['meantemp']])

regr.fit(train_x, train_y)

train_y_predict = regr.predict(train_x)

```

7. Selanjutnya saya, menampilkan grafik untuk menampilkan garis prediksi yang dibuat oleh Regresi Linear dan menampilkan dan Pengaruh Humidity terhadap Mean Temp.



```

plt.scatter(train.humidity, train.meantemp,color="blue")

plt.scatter(train_x, train_y_predict, color="green",
label="Prediksi Model")

plt.plot(train_x,regr.coef_[0][0]*train_x+regr.intercept_[0],'-r')

plt.xlabel("Humidity")

plt.ylabel("Mean Temp")

plt.title("Regresi Linear Data Train")

plt.show()

```

8. Selanjutnya saya menampilkan nilai MAE, MAPE, dan juga R2score untuk mengetahui seberapa baik akurasi.

```
MAE Data TRAIN : 5.107436744842092
MAPE Data TRAIN : 24.52204956609239%
Dengan R2Score/Akurasi data TRAIN = 0.3236527333454944
```

Dengan nilai yang diperoleh, menunjukkan presentase errornya sangat tinggi yaitu 24% error dari data aslinya. R2Score nya juga masih kurang baik yaitu 0.3, masih sangat jauh dari angka 1.

```
maeTrain = mean_absolute_error(train_y,train_y_predict)

mapeTrain

=mean_absolute_percentage_error(train_y,train_y_predict)*100

print(f"\nMAE Data TRAIN : {maeTrain[0]}")

print(f"MAPE Data TRAIN : {mapeTrain[0]}%")

print(f"Dengan R2Score/Akurasi data TRAIN =

{r2_score(train_y,train_y_predict)}")
```

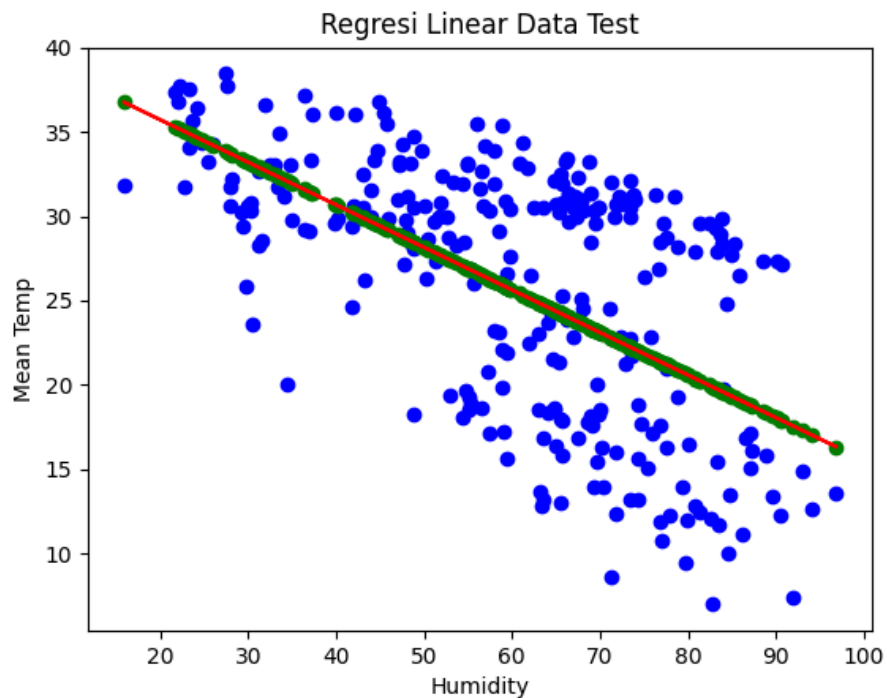
9. Membuat nilai prediksi untuk data test

```
test_x = np.asanyarray(test[['humidity']])

test_y = np.asanyarray(test[['meantemp']])

test_y_predict = regr.predict(test_x)
```

10. Menampilkan grafik data test dan juga garis prediksinya.



```
plt.scatter(test.humidity, test.meantemp,color="blue")

plt.scatter(test_x, test_y_predict,color="green")

plt.plot(test_x,regr.coef_[0][0]*test_x+regr.intercept_[0],'-r')

plt.xlabel("Humidity")

plt.ylabel("Mean Temp")

plt.title("Regresi Linear Data Test")

plt.show()
```

11. Selanjutnya saya menampilkan nilai MAE, MAPE, dan juga R2score untuk mengetahui seberapa baik akurasi.

```
MAE Data TEST : 5.227830755093491
MAPE Data TEST : 25.081750834538013%
Dengan R2Score/Akurasi data TEST = 0.3385
098534792098
```

Dengan nilai yang diperoleh, menunjukkan presentase errornya sangat tinggi yaitu 25% error dari data aslinya. R2Score nya juga masih kurang baik yaitu 0.3, masih sangat jauh dari angka 1.

```
maeTest = mean_absolute_error(test_y,test_y_predict)

mapeTest =

mean_absolute_precentage_error(test_y,test_y_predict)*100
```

```

print(f"\nMAE Data TEST : {maeTest[0]}")

print(f"MAPE Data TEST : {mapeTest[0]}%")

print(f"Dengan R2Score/Akurasi data TEST =
{r2_score(test_y, test_y_predict)}")

```

12. Kode dibawah ini dimaksudkan untuk bisa menerima inputan pengguna. Jadi, pengguna akan menginputkan nilai X yaitu Humidity, dan akan keluar prediksi Y yaitu Rata rata Temperature.

```

#inputan User =

ulang = "a"

while ulang == "y" or "a" :

    def input_angka():

        while True:

            user_input = input("\nUntuk memprediksi temperatur,
Masukkan nilai Humidity: ")

            # Periksa apakah setiap karakter adalah angka atau koma
            if all(char.isdigit() or char == '.' for char in
user_input):

                return float(user_input) # Kembalikan input
sebagai float

            else:

                print("\nInput tidak valid. Mohon masukkan angka
saja.")

        # Contoh penggunaan

    x_user = input_angka()

    x_user = np.array([[x_user]])

    user_y_predict = regr.predict(x_user)

    print(f"\nHasil Prediksi Data Humidity = {x_user[0][0]} , Rata
Rata Temperature adalah = {user_y_predict[0][0]}")

```

```

ulang = input("Ingin Mengulangi Prediksi? Y/N \n").lower()

if ulang == "n" :

    break

```

13. Berikut hasil perbandingan jika menginputkan data yang ada di dataset dengan data prediksi. Dengan menginputkan manual.

Nilai Humidity	Nilai Prediksi Rata rata Temperature	Nilai Asli Rata Rata Temperature Dataset
63.71428571	24.69037186982404	8.857142857
92.16666667	17.51575220632467	13.83333333
74.33333333	22.01264771137146	29.66666667

Dengan ini, kesimpulannya adalah algoritma regresi linear kurang cocok apabila dataset asli masih terlalu tidak teratur seperti gambar dibawah, karena akan sangat memengaruhi garis regresi linear.

