

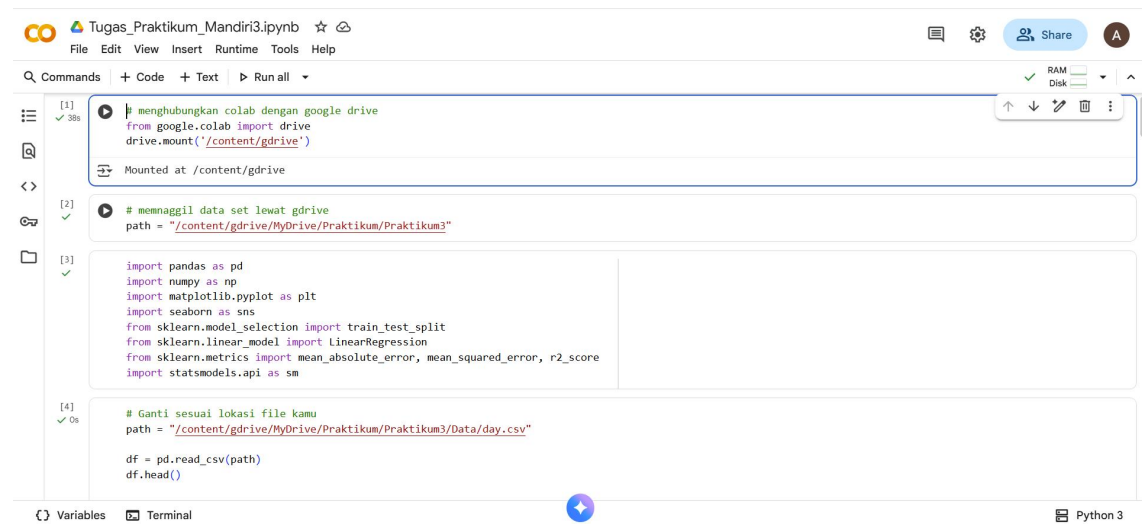
Tugas Praktikum Mandiri 3 - Machine Learning

Nama : Alia Maisyarah 1- 0110224095

‘Teknik Informatika, STT Terpadu Nurul Fikri, Depok’

‘E-Mail: 0110224095@student.nurulfikri.ac.id - E-Mail mahasiswa 1

1. Latihan Mandiri 3



1. - from google.colab import drive

Mengimpor modul drive bawaan dari Google Colab.

- drive.mount('/content/gdrive')

Menjalankan perintah untuk **menghubungkan (mount)** Google Drive ke sesi Colab. Setelah kode ini dijalankan, akan muncul *pop-up* otorisasi atau *link* yang perlu Anda ikuti untuk memberikan izin akses.

- **Output:** Setelah berhasil, akan muncul pesan seperti: Mounted at /content/gdrive. Ini berarti semua file di Drive Anda sekarang dapat diakses melalui jalur /content/gdrive/MyDrive.

2. Penyiapan Path dan Impor Library

Selanjutnya, kita mendefinisikan *path* (jalur) ke folder proyek di Drive dan mengimpor semua *library* yang dibutuhkan untuk analisis data, visualisasi, dan pembangunan model.

Penyiapan Path

- Kode ini mendefinisikan lokasi umum folder kerja Anda.

Penting: Pastikan jalur "/content/gdrive/MyDrive/Praktikum/Praktikum3" sesuai dengan lokasi folder Anda di Drive.

3 Impor Library

import pandas as pd : Alat utama untuk membaca, mengolah, dan memanipulasi data dalam format tabel (*DataFrame*).

import numpy as np : Alat untuk komputasi numerik yang cepat, khususnya untuk operasi array dan matriks yang menjadi dasar perhitungan ML

import matplotlib.pyplot as plt

: Pustaka dasar untuk membuat grafik dan *plot*.

import seaborn as sns : Pustaka untuk visualisasi statistik yang lebih profesional dan informatif

from sklearn.model_selection import train_test_split

: Fungsi untuk memisahkan data menjadi set latih (*train*) dan set uji (*test*).

from sklearn.linear_model import LinearRegression

: Algoritma utama yang digunakan untuk membuat model Regresi Linier.

from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

: Digunakan untuk mengukur dan mengevaluasi kinerja model Anda (seperti MAE, MSE, dan R2).

import statsmodels.api as sm

: Pustaka statistik yang menyediakan alat untuk analisis regresi mendalam dan menampilkan hasil statistik rinci (seperti p-value), yang berguna untuk interpretasi model.

4. Mengambil dataset dari Google Drive

- path = "/content/gdrive/MyDrive/Praktikum/Praktikum3/Data/day.csv"

: Menentukan lokasi (path) file dataset day.csv yang ada di Google Drive.

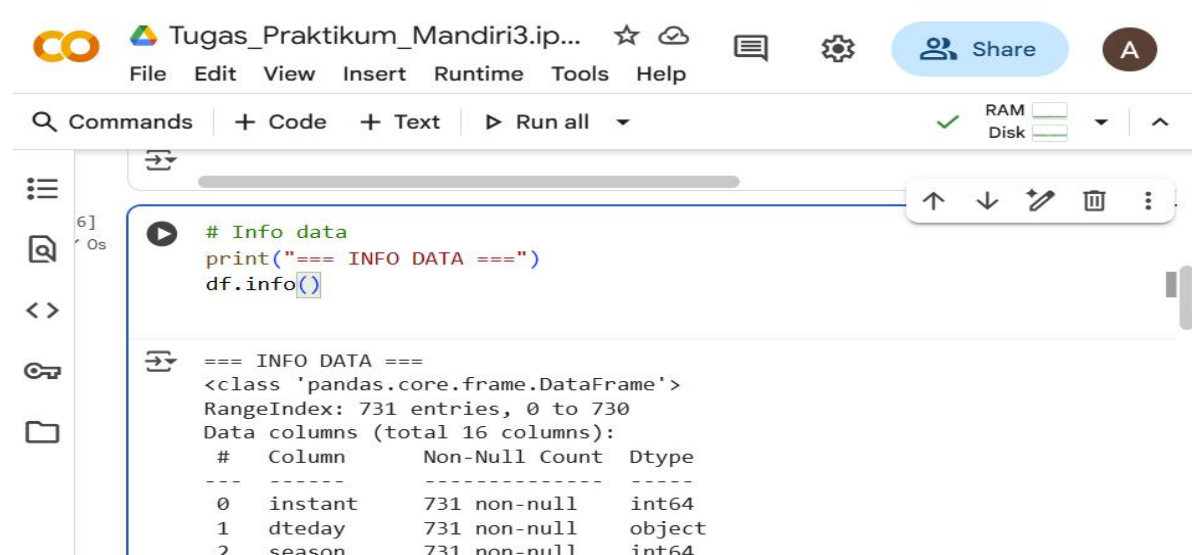
- df = pd.read_csv(path)

: Membaca file CSV (Comma Separated Values) dari lokasi yang ditentukan pada variabel path.

- df.head()

: Menampilkan 5 baris pertama dari dataset untuk memastikan file sudah berhasil dibaca dengan benar, sekaligus memperlihatkan nama-nama kolom dan contoh isinya.

5. menampilkan informasi lengkap mengenai struktur dataset



The screenshot shows a Jupyter Notebook with the following code in the cell:

```
# Info data
print("=== INFO DATA ===")
df.info()
```

The output of the code is displayed below the code cell:

```
=== INFO DATA ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 731 entries, 0 to 730
Data columns (total 16 columns):
#   Column      Non-Null Count  Dtype
---  -
0   instant    731 non-null    int64
1   dteday      731 non-null    object
2   season      731 non-null    int64
```

- `print("=== INFO DATA ===")`
 - : Menampilkan teks "=== INFO DATA ===" di layar sebagai judul atau penanda bahwa output berikutnya berisi informasi detail tentang data.
- `df.info()`
 - : Menampilkan , seperti:
 - Jumlah baris dan kolom,
 - Nama setiap kolom,
 - Jumlah data non-null (tidak kosong),
 - Tipe data dari masing-masing kolom (misalnya: int64, float64, object),
 - Dan jumlah total penggunaan memori.

The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

```
# Statistik deskriptif
print("\n=== DESKRIPSI DATA ===")
print(df.describe())
```

The output of the code is a descriptive statistics summary for the dataset, formatted as follows:

```
=== DESKRIPSI DATA ===
      instant      season      yr      mnth      holiday      weekday
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
mean    366.000000    2.496580    0.500684    6.519836    0.028728    2.997200
std     211.165812    1.110807    0.500342    3.451913    0.167155    2.004700
min       1.000000    1.000000    0.000000    1.000000    0.000000    0.000000
25%     183.500000    2.000000    0.000000    4.000000    0.000000    1.000000
50%     366.000000    3.000000    1.000000    7.000000    0.000000    3.000000
75%     548.500000    3.000000    1.000000   10.000000    0.000000    5.000000
max      731.000000    4.000000    1.000000   12.000000    1.000000    6.000000

      workingday  weathersit      temp      atemp      hum      windspe
count  731.000000  731.000000  731.000000  731.000000  731.000000  731.000000
```

Kode ini buat melihat ringkasan statistik seperti rata-rata, nilai maksimum, minimum, dan penyebaran data dari setiap kolom numerik di dalam dataset.

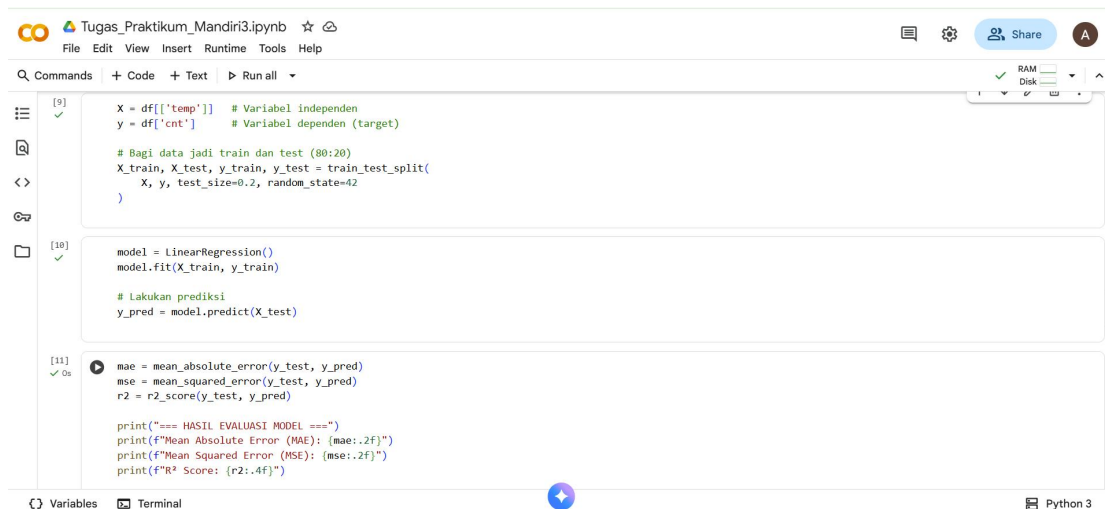
The screenshot shows a Jupyter Notebook interface with a code cell containing the following Python code:

```
# Cek apakah ada nilai kosong
print("\nJumlah missing value tiap kolom:")
print(df.isnull().sum())
```

The output of the code is a summary of missing values for each column in the dataset:

```
Jumlah missing value tiap kolom:
instant      0
dteday       0
season       0
yr           0
mnth         0
holiday      0
weekday      0
workingday   0
weathersit    0
temp         0
atemp        0
```

Kode ini dipakai untuk memastikan data kamu lengkap sebelum lanjut ke proses analisis atau pemodelan.



```
[9] X = df[['temp']] # Variabel independen
    y = df['cnt'] # Variabel dependen (target)

    # Bagi data jadi train dan test (80:20)
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, random_state=42
    )

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

    # Lakukan prediksi
    y_pred = model.predict(X_test)

[11] mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print("=== HASIL EVALUASI MODEL ===")
    print(f"Mean Absolute Error (MAE): {mae:.2f}")
    print(f"Mean Squared Error (MSE): {mse:.2f}")
    print(f"R2 Score: {r2:.4f}")
```

```
# X = df[['temp']] # Variabel independen
y = df['cnt'] # Variabel dependen (target)
```

```
# Bagi data jadi train dan test (80:20)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

(Kode ini digunakan untuk memisahkan variabel input (temperatur) dan output (jumlah penyewaan sepeda), lalu membagi data menjadi dua bagian agar model bisa dilatih dan diuji secara adil.)

```
# model = LinearRegression()
    model.fit(X_train, y_train)

# Lakukan prediksi
y_pred = model.predict(X_test)
```

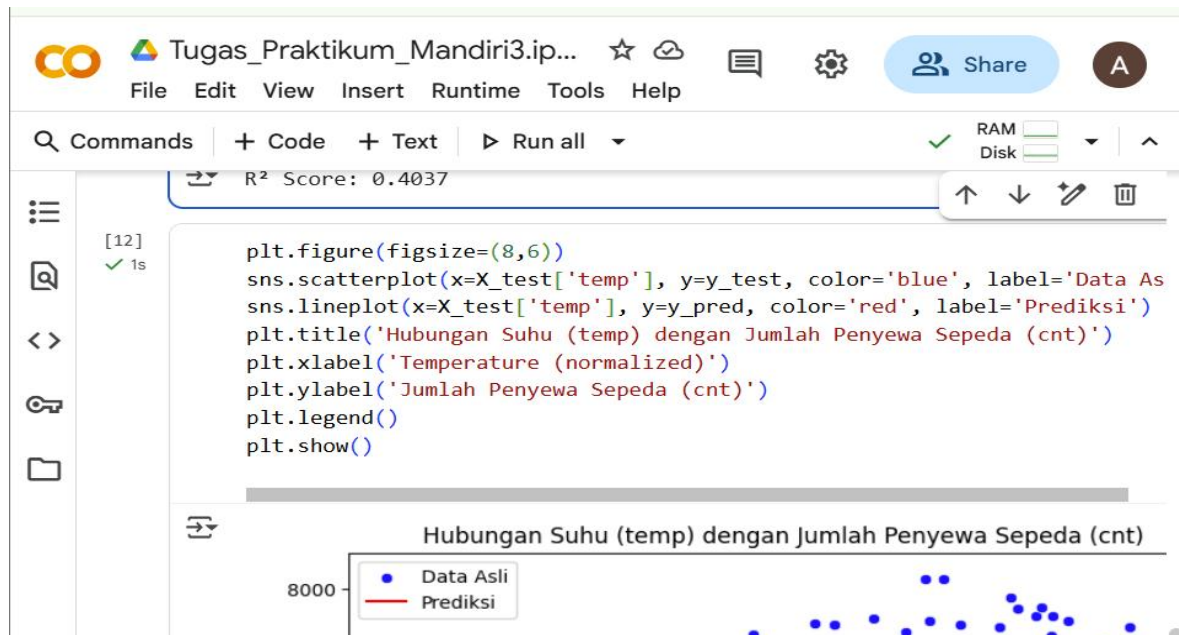
(Kode ini digunakan untuk membuat dan melatih model regresi linear agar bisa memprediksi jumlah sepeda yang disewa (cnt) berdasarkan temperatur (temp). Setelah dilatih, model digunakan untuk menghasilkan prediksi baru (y_pred) pada data uji.)

```
# mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print("=== HASIL EVALUASI MODEL ===")
    print(f"Mean Absolute Error (MAE): {mae:.2f}")
    print(f"Mean Squared Error (MSE): {mse:.2f}")
    print(f"R2 Score: {r2:.4f}")
```

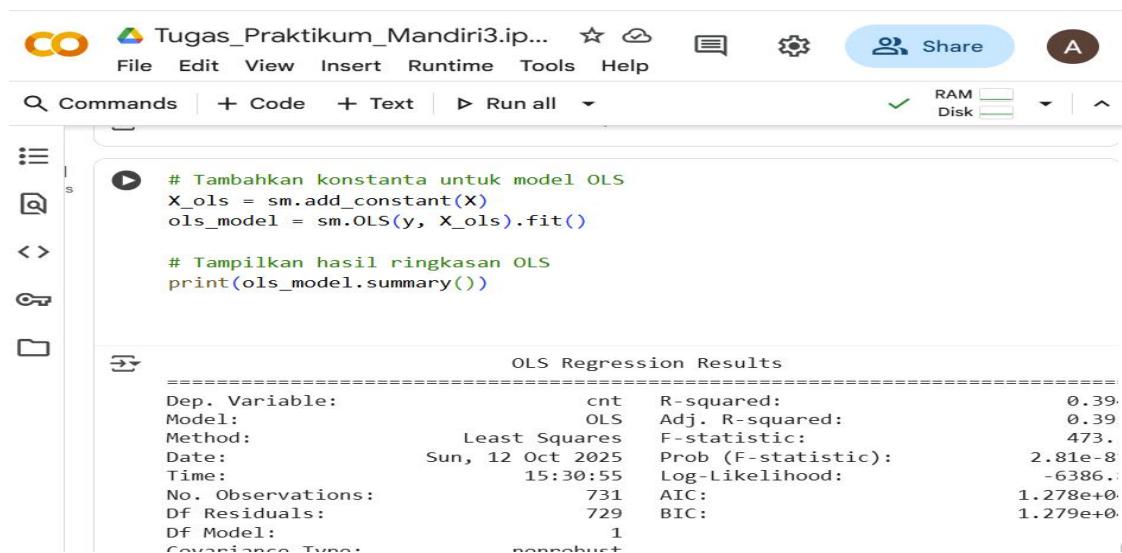
(Kode ini digunakan untuk mengevaluasi performa model regresi linear dengan menghitung error (MAE dan MSE) dan tingkat kecocokan model (R²).

Dari hasil ini, kamu bisa tahu apakah model kamu **sudah cukup akurat** dalam memprediksi jumlah penyewaan sepeda berdasarkan suhu.)



Kode ini digunakan untuk memvisualisasikan hubungan antara suhu dan jumlah penyewaan sepeda, serta membandingkan data asli dengan hasil prediksi model regresi linear.

Kalau garis merah (prediksi) bentuknya mengikuti pola titik-titik biru (data asli), berarti model kamu bekerja dengan baik.



Kode ini digunakan untuk menganalisis hubungan antara suhu dan jumlah penyewaan sepeda secara statistik menggunakan metode OLS (Ordinary Least Squares).

Hasilnya memberikan informasi lengkap seperti koefisien, signifikansi, dan tingkat kecocokan model (R^2) agar kamu bisa menilai apakah suhu benar-benar berpengaruh terhadap jumlah penyewaan sepeda.

