

Praktikum 3 : Laporan Praktikum Mandiri Pertemuan 3

Sintia Sari - 0110222135 ^{1*}

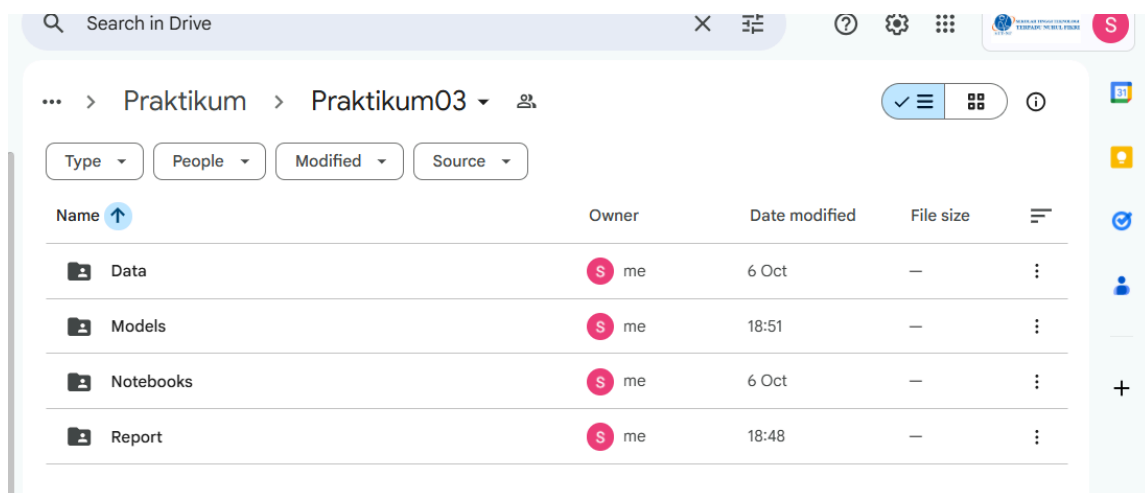
¹ Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: sint22135ti@student.nurulfikri.ac.id

Abstract. Pada praktikum ini membahas analisis data peminjaman sepeda harian menggunakan metode Multiple Linear Regression. Dataset berisi 731 data dengan 16 variabel, meliputi faktor cuaca, musim, hari kerja, serta jumlah peminjaman sepeda. Hasil analisis menunjukkan bahwa variabel suhu, kelembaban, kecepatan angin, musim, kondisi cuaca, dan hari kerja berpengaruh terhadap jumlah peminjaman. Model yang dibangun mampu menjelaskan sekitar 51% variasi data dengan tingkat kesalahan prediksi (MAE 1144.87, RMSE 1349.93). Visualisasi nilai aktual dan prediksi menunjukkan model cukup baik mengikuti pola data, meskipun masih terdapat error pada beberapa titik. Secara keseluruhan, Multiple Linear Regression dapat menggambarkan hubungan antara faktor lingkungan dan sosial terhadap jumlah peminjaman sepeda, namun akurasi masih perlu ditingkatkan.

Praktikum Mandiri Pertemuan 3

1. Direktori program



Gambar 1. Direktori Program

Dalam pengerjaan tugas maupun praktikum, semua coding dikerjakan dalam folder notebooks dengan menggunakan file Python bernama `PraktikumMandiri3_Sintia Sari_0110222135_ML-Pagi.ipynb`. Dataset mentah ditempatkan pada folder data, adapun laporan maupun hasil visualisasi disimpan pada folder reports

2. Menghubungkan dengan Google Drive

```
# Menghubungkan colab dengan google drive
from google.colab import drive
drive.mount('/content/gdrive')
import os

Mounted at /content/gdrive

# Memanggil dataset melalui gdrive
path = "gdrive/MyDrive/Machine Learning/Praktikum/Praktikum03/Data/"
```

Gambar 2. Menghubungkan colab dengan Gdrive

Penjelasan kode :

- `from google.colab import drive & drive.mount('/content/gdrive')`
Menghubungkan Google Colab dengan Google Drive supaya file dataset bisa diakses langsung dari Drive.
- `import os`
Digunakan untuk mengatur direktori/file di sistem (opsional, untuk navigasi folder).
- `path = "gdrive/MyDrive/Machine Learning/Praktikum/Praktikum03/Data/"`
Menyimpan alamat folder tempat file dataset berada.

Setelah dijalankan, Colab akan menampilkan pesan "Mounted at /content/gdrive", artinya Google Drive berhasil terhubung ke Colab dan bisa digunakan untuk membaca atau menyimpan file. Selain itu, variabel path menyimpan lokasi dataset sehingga bisa digunakan nanti.

3. Membaca file CSV

```
# Membaca file csv menggunakan pandas
import pandas as pd

df = pd.read_csv(path + 'day.csv')
df.head()
```

Gambar 3. Membaca dan Menampilkan Dataset

Penjelasan kode :

- `import pandas as pd`

Mengimpor library pandas dengan alias `pd`, digunakan untuk membaca dan mengolah data dalam bentuk tabel (DataFrame).

- `df = pd.read_csv(path + 'day.csv')`

Membaca file CSV bernama `day.csv` yang ada di folder `path`, lalu menyimpannya ke dalam DataFrame dengan nama variabel `df`.

- `df.head()`

Menampilkan isi DataFrame lima baris teratas.

	instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
0	1	2011-01-01	1	0	1	0	6	0	2	0.344167	0.363625	0.805833	0.160446	331	654	985
1	2	2011-01-02	1	0	1	0	0	0	2	0.363478	0.353739	0.696087	0.248539	131	670	801
2	3	2011-01-03	1	0	1	0	1	1	1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
3	4	2011-01-04	1	0	1	0	2	1	1	0.200000	0.212122	0.590435	0.160296	108	1454	1562
4	5	2011-01-05	1	0	1	0	3	1	1	0.226957	0.229270	0.436957	0.186900	82	1518	1600

Gambar 4. Isi dari Dataset day.csv

Dataset berhasil dimuat dengan total data sebanyak $731 \text{ baris} \times 16 \text{ kolom}$, yaitu:

Tabel 1. Nama dan Deskripsi Dataset

Kolom	Deskripsi
instant	Nomor urut (ID) unik setiap record
dteday	Tanggal pencatatan (format YYYY-MM-DD)
season	Musim (1: Winter, 2: Spring, 3: Summer, 4: Fall)
yr	Tahun (0: 2011, 1: 2012)
mnth	Bulan
holiday	Status hari libur (0: bukan hari libur, 1: hari libur resmi)
weekday	Hari dalam seminggu
workingday	Status hari kerja (0: bukan hari kerja/akhir pekan/libur, 1: hari kerja)

weathersit	Kondisi cuaca (1: Clear, 2: Mist/Cloudy, 3: Light Snow/Rain, 4: Heavy Rain/Snow/Fog)
temp	Suhu (normalized, nilai riil suhu °C dibagi 41)
atemp	Suhu yang dirasakan (“feels like”), normalized
hum	Kelembaban relatif (normalized, 0–1)
windspeed	Kecepatan angin (normalized)
casual	Jumlah peminjaman oleh pengguna casual (tanpa registrasi)
registered	Jumlah peminjaman oleh pengguna terdaftar (registered user)
cnt	Total jumlah peminjaman (casual + registered) per hari

4. Melihat informasi umum data

```
# Mencari info data pada file (tipe datanya, non null count data, nama kolom)
df.info()

<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
3   yr          731 non-null    int64
4   mnth        731 non-null    int64
5   holiday     731 non-null    int64
6   weekday     731 non-null    int64
7   workingday  731 non-null    int64
8   weathersit   731 non-null    int64
9   temp        731 non-null    float64
10  atemp       731 non-null    float64
11  hum         731 non-null    float64
12  windspeed   731 non-null    float64
13  casual      731 non-null    int64
14  registered  731 non-null    int64
15  cnt         731 non-null    int64
dtypes: float64(4), int64(11), object(1)
memory usage: 91.5+ KB
```

Gambar 5. Melihat Informasi Umum Data

Fungsi `df.info()` adalah untuk menampilkan informasi ringkas tentang DataFrame, termasuk jumlah baris, jumlah kolom, nama kolom, jumlah data non-null, tipe data tiap kolom, serta estimasi penggunaan memori

Dataset memiliki total data sebanyak 731 baris dan 16 kolom. Tipe data dari kolom `instant`, `season`, `yr`, `mnth`, `holiday`, `weekday`, `workingday`, `weathersit`, `casual`, `registered`, `cnt` bertipe integer, kolom `dteday` bertipe string/object, dan kolom `temp`, `atemp`, `hum`, `windspeed` bertipe float.

5. Menampilkan ringkasan statistik deskriptif

```
# Menampilkan ringkasan statistik deskriptif
df.describe()
```

Gambar 6. Menampilkan statistik deskriptif

Fungsi `df.describe()` dari pandas digunakan untuk menampilkan statistik deskriptif dari setiap kolom numerik dalam dataset.

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
count	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000	731.000000
mean	366.000000	2.496580	0.500684	6.519836	0.028728	2.997264	0.683995	1.395349	0.495385	0.474354	0.627894	0.190486	848.176471	3656.172367	4504.348837
std	211.165812	1.110807	0.500342	3.451913	0.167155	2.004787	0.465233	0.544894	0.183051	0.162961	0.142429	0.077498	686.622488	1560.256377	1937.211452
min	1.000000	1.000000	0.000000	1.000000	0.000000	0.000000	0.000000	1.000000	0.059130	0.079070	0.000000	0.022392	2.000000	20.000000	22.000000
25%	183.500000	2.000000	0.000000	4.000000	0.000000	1.000000	0.000000	1.000000	0.337083	0.337842	0.520000	0.134950	315.500000	2497.000000	3152.000000
50%	366.000000	3.000000	1.000000	7.000000	0.000000	3.000000	1.000000	1.000000	0.498333	0.486733	0.626667	0.180975	713.000000	3662.000000	4548.000000
75%	548.500000	3.000000	1.000000	10.000000	0.000000	5.000000	1.000000	2.000000	0.655417	0.608602	0.730209	0.233214	1096.000000	4776.500000	5956.000000
max	731.000000	4.000000	1.000000	12.000000	1.000000	6.000000	1.000000	3.000000	0.861667	0.840896	0.972500	0.507463	3410.000000	6946.000000	8714.000000

Gambar 7. Statistik Deskriptif

Hasilnya menunjukkan bahwa dataset memiliki 731 hari pengamatan dengan rata-rata peminjaman sepeda harian sekitar 4.504 unit, minimum 22, dan maksimum 8.714. Sebagian besar peminjaman berasal dari pengguna `registered` dengan rata-rata 3.656, sedangkan pengguna `casual` rata-ratanya hanya 848. Dari sisi cuaca, suhu rata-rata sekitar 20°C, kelembaban 62%, dan kecepatan angin 19%. Secara keseluruhan, data ini menggambarkan bahwa jumlah peminjaman sepeda cukup berfluktuasi dan dipengaruhi oleh faktor cuaca, musim, serta hari kerja atau libur.

6. Analisis Heatmap Korelasi

```
import seaborn as sns

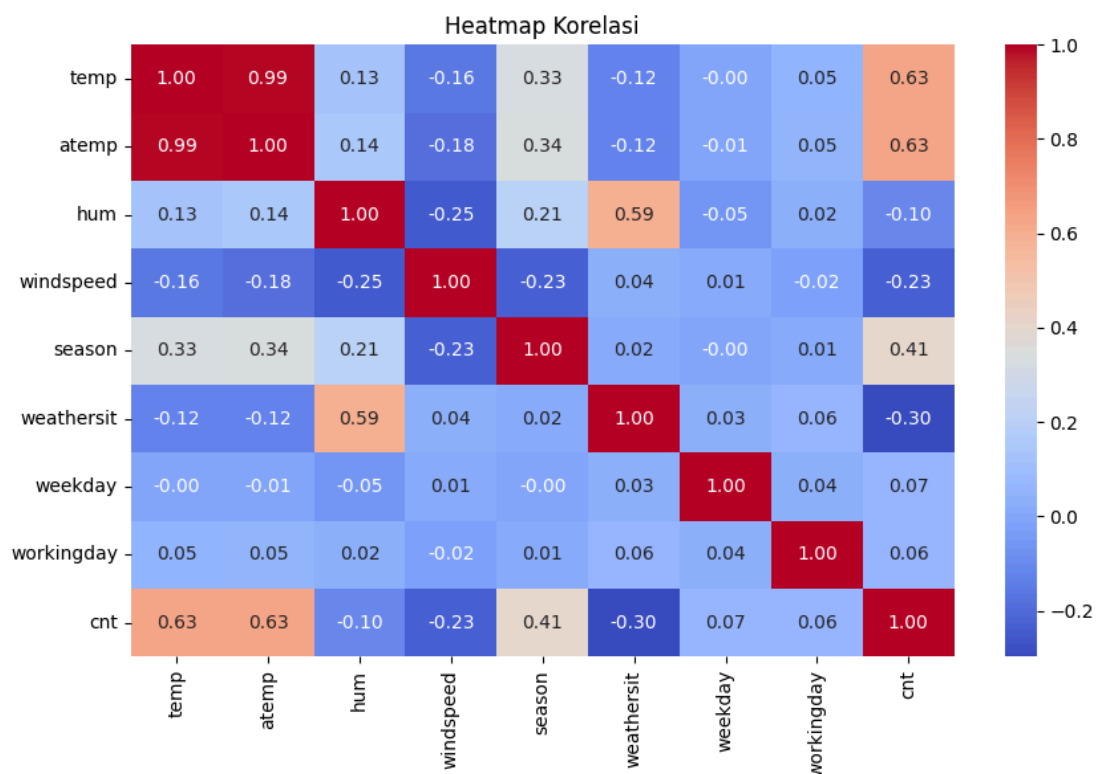
# Memilih kolom yang ingin dilihat korelasinya
kolom_korelasi = ["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday", "cnt"]

# Menghitung korelasi antar kolom tersebut
corr_matrix = df[kolom_korelasi].corr()

# Visualisasi heatmap korelasi
plt.figure(figsize=(10,6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", fmt=".2f")
plt.title("Heatmap Korelasi")
plt.show()
```

Gambar 8. Heatmap Korelasi

Kode tersebut digunakan untuk menghitung korelasi antar variabel pada dataset dan menampilkannya dalam bentuk heatmap. Pertama, dipilih beberapa kolom penting seperti suhu (temp), suhu yang dirasakan (atemp), kelembapan (hum), kecepatan angin (windspeed), musim (season), kondisi cuaca (weathersit), hari kerja (workingday), hari dalam seminggu (weekday), serta jumlah penyewaan sepeda (cnt). Setelah itu, dibuat matriks korelasi yang menunjukkan hubungan antar variabel dengan nilai antara -1 hingga 1.



Gambar 9. Heatmap Korelasi

Heatmap divisualisasikan menggunakan warna merah untuk korelasi positif dan biru untuk korelasi negatif. Dari hasilnya terlihat bahwa temp dan atemp memiliki korelasi sangat tinggi (0.99), suhu dan jumlah penyewaan (cnt) juga berkorelasi cukup kuat (0.63), musim (season) berpengaruh positif terhadap jumlah penyewaan (0.41), sedangkan kondisi cuaca buruk (weathersit) justru menurunkan jumlah penyewaan dengan korelasi negatif (-0.30). Hal ini menunjukkan bahwa suhu, musim, dan cuaca menjadi faktor yang cukup penting terhadap jumlah penyewaan sepeda.

7. Pembagian dataset, menentukan variabel dependen dan independen

```
from sklearn.model_selection import train_test_split

# Tentukan Variabel Dependen dan Independen
X = df[["temp", "atemp", "hum", "windspeed", "season", "weathersit", "weekday", "workingday"]]
y = df["cnt"]

# Bagi Data Menjadi Train dan Test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)
```

Gambar 10. Pembagian Dataset, Variabel Dependen dan Independen

Pada kode diatas adalah melakukan pembagian dataset untuk keperluan pemodelan machine learning. Pertama, data dibagi menjadi dua bagian utama, yaitu variabel independen atau fitur (X) yang berisi faktor-faktor seperti suhu, kelembapan, kecepatan angin, musim, kondisi cuaca, hari kerja, dan lain-lain, serta variabel dependen atau target (y) yang berisi jumlah peminjaman sepeda (cnt). Setelah itu, dataset dipisahkan menjadi data latih dan data uji, di mana data latih digunakan untuk membangun dan melatih model, sedangkan data uji digunakan untuk mengukur serta mengevaluasi kinerja model agar dapat memprediksi dengan baik pada data baru.

Penjelasan kode :

- `from sklearn.model_selection import train_test_split`
Mengimpor fungsi `train_test_split` dari library scikit-learn, yang digunakan untuk membagi dataset menjadi data latih (train) dan data uji (test).
- X (Independent Variable / fitur)
Berisi kolom-kolom prediktor seperti temp, atemp, hum, windspeed, season, weathersit, weekday, workingday. Fitur ini adalah faktor yang memengaruhi hasil (contohnya cuaca, musim, hari kerja, dsb.).
- y (Dependent Variable / target)
Adalah kolom cnt, yaitu jumlah total sepeda yang dipinjam (target yang ingin diprediksi).
- `X_train, y_train`, data latih (80% dari data).
- `X_test, y_test`, data uji (20% dari data).
- `test_size=0.2`, 20% data digunakan untuk pengujian.

- `random_state=42`, angka acak tetap, supaya pembagian data selalu sama (reproducible).

8. Pembuatan dan pelatihan model regresi linear

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import numpy as np

# Inisialisasi dan latih model regresi linear
model = LinearRegression()
model.fit(X_train, y_train)

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

Gambar 11. Pembuatan dan Pelatihan Model Regresi Linear

Kode ini sedang melakukan pembuatan dan pelatihan model regresi linear. Pertama, model dibuat dan dilatih menggunakan data latih, kemudian digunakan untuk memprediksi nilai target pada data uji. Hasil prediksi nantinya dapat dibandingkan dengan nilai aktual untuk mengevaluasi kinerja model.

Penjelasan kode :

- `LinearRegression`
Digunakan untuk membuat model regresi linear.
- `mean_absolute_error, mean_squared_error, r2_score`
Digunakan untuk metrik evaluasi model untuk mengukur performa prediksi.
- `numpy (np)`
Library untuk perhitungan numerik.
- `model = LinearRegression(), model.fit(X_train, y_train)`
Untuk membuat objek model regresi linear, lalu melatihnya menggunakan data latih `X_train` (fitur) dan `y_train` (target).
- `y_pred = model.predict(X_test)`
Menggunakan model yang sudah dilatih untuk memprediksi nilai target pada data uji (`X_test`) dan hasil prediksi disimpan di `y_pred`.

9. Evaluasi Performa Model

```
# Evaluasi performa model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

# Tampilkan hasil evaluasi
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R-squared (R²): {r2:.3f}")

Mean Absolute Error (MAE): 1144.87
Mean Squared Error (MSE): 1822308.96
Root Mean Squared Error (RMSE): 1349.93
R-squared (R²): 0.546
```

Gambar 12. Evaluasi Performa Model

Kode ini sedang melakukan evaluasi performa model regresi linear dengan menggunakan metrik error (MAE, MSE, RMSE) dan goodness of fit (R^2). Tujuannya adalah menilai seberapa akurat model dalam memprediksi data uji dibandingkan dengan nilai aktual. Berikut ini penjelasan dari masing-masing metrik evaluasi.

- MAE (Mean Absolute Error)
Rata-rata selisih absolut antara nilai aktual dan prediksi. Semakin kecil nilainya, semakin baik.
- MSE (Mean Squared Error)
Rata-rata kuadrat selisih antara nilai aktual dan prediksi. Memberi penalti lebih besar pada error yang besar.
- RMSE (Root Mean Squared Error)
Akar dari MSE, sehingga hasilnya kembali ke satuan data aslinya.
- R^2 (R-squared)
Menunjukkan seberapa baik model menjelaskan variasi data (0–1, makin mendekati 1 semakin baik).

Hasil evaluasi menunjukkan bahwa rata-rata kesalahan prediksi model sekitar 1145–1350 unit (MAE dan RMSE), dengan R^2 sebesar 0.546 yang berarti model hanya mampu menjelaskan sekitar 54,6% variasi data. Artinya, model cukup baik tetapi masih perlu ditingkatkan agar prediksinya lebih akurat.

10. Visualisasi prediksi vs aktual

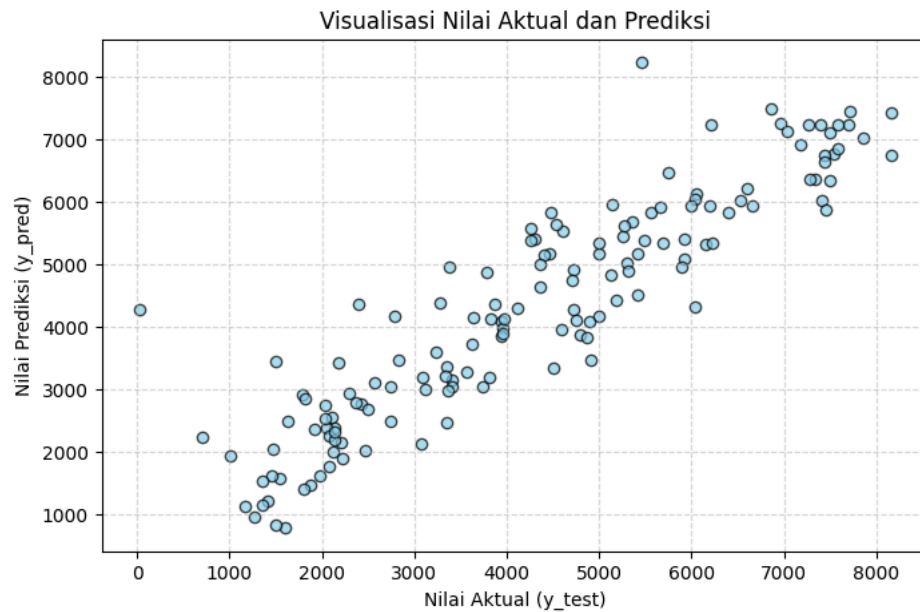
```
import matplotlib.pyplot as plt

# Visualisasi hasil prediksi vs data aktual
plt.figure(figsize=(8, 5))
plt.scatter(y_test, y_pred, color='skyblue', edgecolor='k', alpha=0.7)
plt.title("Visualisasi Nilai Aktual dan Prediksi")
plt.xlabel("Nilai Aktual (y_test)")
plt.ylabel("Nilai Prediksi (y_pred)")
plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

Gambar 13. Kode Visualisasi Prediksi vs Aktual

Kode ini sedang memproses visualisasi hasil prediksi model dibandingkan dengan data aktual menggunakan scatter plot. Tujuannya adalah untuk melihat seberapa baik model mampu memprediksi data uji secara visual. Penjelasan kode :

- `import matplotlib.pyplot as plt`
Mengimpor library matplotlib untuk membuat visualisasi grafik.
- `plt.figure(figsize=(8, 5))`
Menentukan ukuran grafik dengan lebar 8 dan tinggi 5.
- `plt.scatter(y_test, y_pred, color='skyblue', edgecolor='k', alpha=0.7)`
Membuat scatter plot antara nilai aktual (`y_test`) dan nilai prediksi (`y_pred`):
 - `color='skyblue'`, titik berwarna biru muda.
 - `edgecolor='k'`, tepi titik berwarna hitam.
- `plt.title("Visualisasi Nilai Aktual dan Prediksi")`, `plt.xlabel("Nilai Aktual (y_test)")`, `plt.ylabel("Nilai Prediksi (y_pred)")`
Memberi judul grafik serta label pada sumbu X (nilai aktual) dan sumbu Y (nilai prediksi).
- `alpha=0.7`
Befungsi untuk tingkat transparansi agar lebih jelas.
- `plt.grid(True, linestyle='--', alpha=0.5)`, `plt.show()`
Menambahkan garis bantu (grid) bergaya putus-putus agar lebih mudah dibaca, lalu menampilkan grafik.



Gambar 14. Visualisasi Prediksi vs Aktual

Grafik scatter tersebut menunjukkan perbandingan antara nilai aktual (y_{test}) dan nilai prediksi (y_{pred}). Titik-titik data sebagian besar mengikuti pola naik yang searah dengan garis diagonal, artinya model cukup mampu menangkap tren data. Namun, masih ada titik yang menyebar jauh dari pola diagonal, menandakan adanya selisih atau error pada beberapa prediksi. Secara ringkas, model sudah bisa memprediksi dengan kecenderungan yang benar, tetapi akurasi belum sepenuhnya sempurna.

11. Koefisien dan konstanta model

```
# Melihat koefisien dan konstanta model
print("Intercept (Konstanta):", model.intercept_)
print("Koefisien Tiap Variabel:")
for col, coef in zip(X.columns, model.coef_):
    print(f"{col}: {coef}")

Intercept (Konstanta): 2839.0904316245797
Koefisien Tiap Variabel:
temp: 2455.6523519298316
atemp: 3491.039948563661
hum: -2146.395753260919
windspeed: -2464.7206998833485
season: 419.21413853476497
weathersit: -500.5628428339221
weekday: 66.44675753118675
workingday: 154.6296516775804
```

Gambar 15. Koefisien dan Konstanta Model

Kode tersebut digunakan untuk menampilkan intercept dan koefisien regresi tiap variabel. Hasilnya menunjukkan bahwa konstanta model adalah 2839.09, sedangkan variabel temp dan atemp berpengaruh positif besar terhadap jumlah penyewaan, sementara hum, windspeed, dan weathersit berpengaruh negatif. Variabel lain seperti season, weekday, dan workingday juga berpengaruh positif meski lebih kecil. Ini berarti suhu dan musim mendorong peningkatan penyewaan, sedangkan kelembapan, angin, dan cuaca buruk menurunkannya.

12. Analisis statistik dengan OLS

```
import statsmodels.api as sm

# Analisis Statistik dengan OLS
X_train_const = sm.add_constant(X_train)
ols_model = sm.OLS(y_train, X_train_const).fit()
print(ols_model.summary())
```

Gambar 16. Analisis statistik dengan OLS

Kode di atas menggunakan library statsmodels untuk melakukan analisis regresi linear dengan metode Ordinary Least Squares (OLS). Baris `X_train_const = sm.add_constant(X_train)` menambahkan konstanta (intercept) pada model. Selanjutnya, `ols_model = sm.OLS(y_train, X_train_const).fit()` membangun model regresi OLS dengan data latih, lalu `ols_model.summary()` menampilkan ringkasan hasil analisis.

OLS Regression Results

Dep. Variable:cntR-squared:0.516

Model:OLSAdj. R-squared:0.510

Method:Least SquaresF-statistic:76.70

Date:Thu, 09 Oct 2025Prob (F-statistic):1.21e-85

Time:14:29:19Log-Likelihood:-5030.1

No. Observations:584AIC:1.008e+04

Df Residuals:575BIC:1.012e+04

Df Model:8

Covariance Type:nonrobust

	coef	std err	t	P> t	[0.025	0.975]
const	2839.0904	400.244	7.093	0.000	2052.971	3625.210
temp	2455.6524	2244.785	1.094	0.274	-1953.325	6864.630
atemp	3491.0399	2540.096	1.374	0.170	-1497.959	8480.039
hum	-2146.3958	531.279	-4.040	0.000	-3189.880	-1102.912
windspeed	-2464.7207	786.875	-3.132	0.002	-4010.221	-919.220
season	419.2141	55.055	7.615	0.000	311.081	527.347
weathersit	-500.5628	132.909	-3.766	0.000	-761.609	-239.516
weekday	66.4468	27.612	2.406	0.016	12.214	120.679
workingday	154.6297	119.085	1.298	0.195	-79.265	388.525

Omnibus:8.667Durbin-Watson:2.015

Prob(Omnibus):0.013Jarque-Bera (JB):5.735

Skew:0.081Prob(JB):0.0568

Kurtosis:2.543Cond. No.286.

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Gambar 16. Analisis statistik dengan OLS

Hasil output menunjukkan nilai R-squared 0.516 dan Adj. R-squared 0.510, yang berarti model dapat menjelaskan sekitar 51% variasi data. Koefisien pada tabel (coef) menunjukkan arah dan besar pengaruh tiap variabel terhadap target cnt, misalnya temp (+2455) dan season (+419) berpengaruh positif signifikan, sedangkan hum (-2146), windspeed (-2464), dan weathersit (-500) berpengaruh negatif. Kolom P>|t| menunjukkan signifikansi; variabel dengan nilai di bawah 0.05 dianggap signifikan. Dari hasil ini terlihat bahwa temp, hum, windspeed, season, weathersit, weekday, dan workingday signifikan, sementara atemp tidak signifikan. Secara keseluruhan, model cukup mampu menjelaskan hubungan antar variabel meski masih ada keterbatasan karena R^2 tidak terlalu tinggi.

References

Link Github :

1) Praktikum dikelas :

https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2003/notebooks/Praktikum3_Sintia_Sari_0110222135_ML_Pagi.ipynb

2) Praktikum mandiri :

https://github.com/ssintyaaa/MachineLearning/blob/main/praktikum%2003/notebooks/PraktikumMandiri3_Sintia_Sari_0110222135_ML_Pagi.ipynb

Link Gdrive :

 Praktikum03