

Tugas 1: Tugas Praktikum 3 Machine Learning

Jamilatun Khoerunnisa - 010222254

Teknik Informatika, STT Terpadu Nurul Fikri, Depok

*E-mail: Jami22254ti@student.nurulfikri.ac.id

1. Langkah 1

```
Import pandas as pd

# read the CSV file with a comma delimiter
df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum03/data/day.csv')
df.head()
```

	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

Langkah kedua yaitu membaca data (file csv) menggunakan pandas, **df = pd.read_csv('/content/drive/MyDrive/praktikum_ml/praktikum03/data/day.csv')** untuk membaca file csv yang diberikan dan menyimpan file pada dataframe (df). **Df** untuk menampilkan isi dari dataframe.

2. Langkah 2

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype  
---  -
 0   No                   5 non-null     int64  
 1   Nama                 5 non-null     object  
 2   tinggi_cm           5 non-null     int64  
 3   actual_berat_kg     5 non-null     float64
dtypes: float64(1), int64(2), object(1)
memory usage: 292.0+ bytes
```

digunakan untuk melihat gambaran umum struktur dataset, seperti jumlah baris dan kolom, nama kolom, tipe data tiap kolom, dan apakah ada data yang kosong atau tidak.

3. Langkah 3

df.describe()

	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	306.000000	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	0.077496	686.622488	1590.256377	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	0.529000	0.134950	315.500000	2497.000000	3152.000000
50%	306.000000	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	0.608802	0.730209	0.233214	1096.000000	4778.500000	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

Df.describe() perintah ini digunakan untuk menampilkan hasil dari statistic deskriptif dari kolom numerik pada dataframe.

4. Langkah 4

```

from sklearn.model_selection import train_test_split
y = df["cnt"]
X = df[["season", "yr", "mnth", "holiday", "weekday", "workingday", "weathersit", "temp", "atemp", "hum", "windspeed"]]

# Bagi data jadi train dan test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=7)
X_train.head()

```

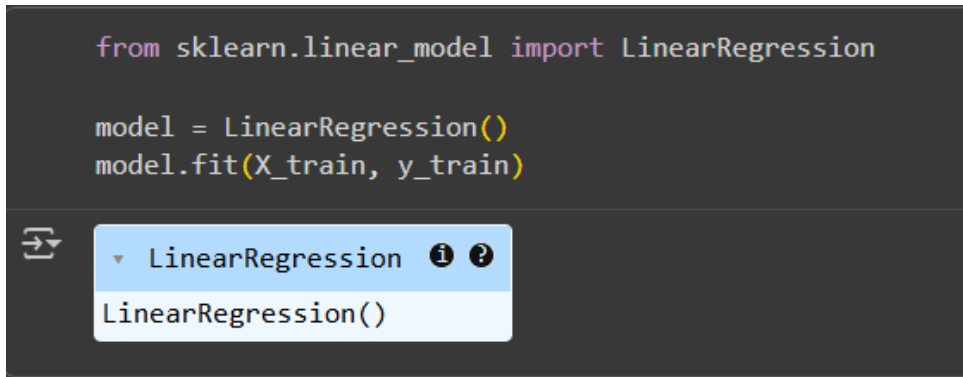
	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed
548	3	1	7	0	1	1	1	0.781667	0.702038	0.447083	0.195267
115	2	0	4	0	2	1	1	0.631667	0.594083	0.729167	0.326500
272	4	0	9	0	5	1	1	0.564167	0.544829	0.647500	0.206475
65	1	0	3	0	1	1	1	0.261739	0.238461	0.551304	0.341352
232	3	0	8	0	0	0	1	0.710833	0.675525	0.770000	0.248754

Langkah ini untuk Membagi dataset menjadi data latih dan uji agar model bisa dilatih dan dievaluasi dengan data yang berbeda supaya hasil prediksi tidak bias.

5. Langkah 5

```
from sklearn.linear_model import LinearRegression

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



Gambar ini menunjukkan tahap Melatih model regresi linear menggunakan data train (X_{train} , y_{train}) agar model dapat mempelajari hubungan antara fitur-fitur dengan target cnt.

6. Langkah 6

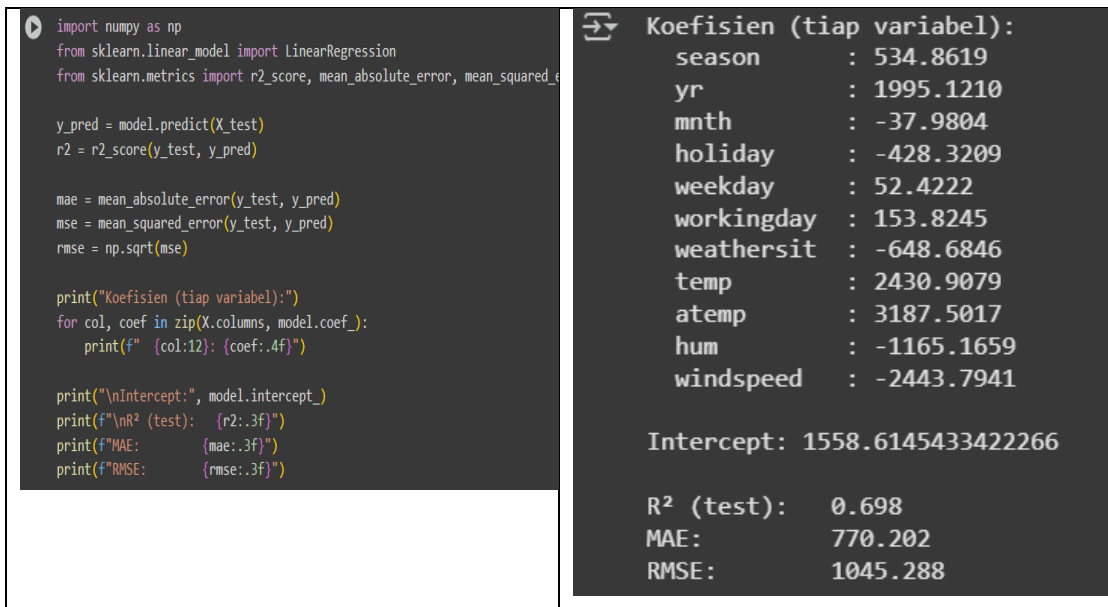
```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

y_pred = model.predict(X_test)
r2 = r2_score(y_test, y_pred)

mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)

print("Koefisien (tiap variabel):")
for col, coef in zip(X.columns, model.coef_):
    print(f" {col:12}: {coef:.4f}")

print("\nIntercept:", model.intercept_)
print(f"\nR2 (test): {r2:.3f}")
print(f"MAE: {mae:.3f}")
print(f"RMSE: {rmse:.3f}")
```



Koefisien (tiap variabel):	
season	: 534.8619
yr	: 1995.1210
mnth	: -37.9804
holiday	: -428.3209
weekday	: 52.4222
workingday	: 153.8245
weathersit	: -648.6846
temp	: 2430.9079
atemp	: 3187.5017
hum	: -1165.1659
windspeed	: -2443.7941
Intercept: 1558.6145433422266	
R ² (test):	0.698
MAE:	770.202
RMSE:	1045.288

Pada Langkah Model regresi linear menjelaskan sekitar 70% variasi jumlah pengguna sepeda (cnt), dengan error rata-rata sekitar 770–1045 sepeda. Variabel seperti musim, suhu, dan atemp berpengaruh positif, sedangkan kelembapan dan kecepatan angin berpengaruh negatif terhadap jumlah pengguna sepeda.

7. Langkah 7

```
# Buat tabel hasil prediksi vs aktual
hasil = pd.DataFrame({
    'Aktual': y_test,
    'Prediksi': y_pred.round(2),
    'Selisih': (y_test - y_pred).round(2)
})
hasil.head()
```

	Aktual	Prediksi	Selisih
708	3228	4676.50	-1448.50
696	3959	4346.03	-387.03
624	7333	6488.17	844.83
304	4068	4002.38	65.62
333	3613	3347.46	265.54

Pada Langkah ini menunjukkan tabel perbandingan hasil prediksi model regresi dengan data sebenarnya. Dari kolom “Selisih”, bisa dilihat sejauh mana model melakukan kesalahan apakah prediksinya terlalu tinggi atau terlalu rendah dibandingkan nilai aktual.

8. Langkah 8

```
#ambil slope (coeffisien) dan intercept dari model
intercept = model.intercept_
coef = model.coef_

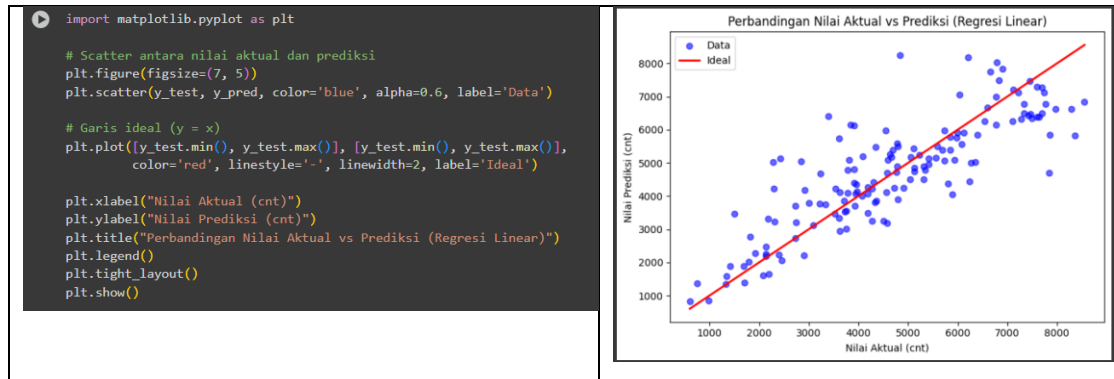
# tampilkan persamaan regresi dengan semua variabel x
print("Persamaan Regresi Linear Berganda:")
persamaan = f"int = {intercept:.3f}"
for var, c in zip(X.columns, coef):
    persamaan = f"{persamaan} + {(c:.3f)} * {var}"
print(persamaan)
```

Persamaan Regresi Linear Berganda:

$$\text{cnt} = 1558.615 + (534.862 \times \text{season}) + (1995.121 \times \text{yr}) + (-37.980 \times \text{mth}) + (-428.321 \times \text{holiday}) + (52.422 \times \text{weekday}) + (153.824 \times \text{workingday}) + (-648.685 \times \text{registered}) + (2430.908 \times \text{temp}) + (3187.502 \times \dots$$

Gambar ini menampilkan proses dan hasil pembentukan persamaan regresi linear berganda yang digunakan untuk memprediksi nilai target berdasarkan variabel-variabel input.

9. Langkah 9



Langkah ini menunjukkan visualisasi hasil prediksi model regresi linear dibandingkan dengan nilai aktual dari dataset.

10. Langkah 10

```
y_pred_test = model.predict(X_test)

hasil = X_test.copy()
hasil["Jumlah Aktual (cnt)"] = y_test.values
hasil["Jumlah Prediksi (cnt)"] = y_pred_test
hasil["Selisih error (cnt)"] = hasil["Jumlah Prediksi (cnt)"] - hasil["Jumlah Aktual (cnt)"]

denom = hasil["Jumlah Aktual (cnt)"].replace(0, np.nan) # antisipasi pembagi nol
hasil["Akurasi (%)"] = (1 - (hasil["Selisih error (cnt)"].abs() / denom)).clip(lower=0, upper=1) * 100

hasil = hasil[[
    "season", "yr", "mnth", "holiday", "weekday", "workingday",
    "weathersit", "temp", "atemp", "hum", "windspeed",
    "Jumlah Aktual (cnt)", "Jumlah Prediksi (cnt)", "Selisih error (cnt)", "Akurasi (%)"
]]

# Tampilkan 10 data pertama
hasil.head(10)
```

	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	Jumlah Aktual (cnt)	Jumlah Prediksi (cnt)	Selisih error (cnt)	Akurasi (%)
708	4	1	12	0	0	0	2	0.384167	0.390146	0.905417	0.157975	3228	4676.496109	1448.496109	55.127134
696	4	1	11	0	2	1	2	0.291667	0.281558	0.766667	0.237562	3959	4346.031055	387.031055	90.224020
624	3	1	9	0	0	0	1	0.580000	0.563125	0.570000	0.090183	7333	6488.168079	-844.831921	88.479041
304	4	0	11	0	2	1	1	0.400833	0.397088	0.683750	0.135571	4068	4002.379569	-65.620431	98.386912
333	4	0	11	0	3	1	1	0.325000	0.311221	0.613333	0.271158	3613	3347.458267	-265.541733	92.650381
138	2	0	5	0	4	1	2	0.530833	0.510742	0.829583	0.108213	4575	3191.925605	-1383.074395	69.768866
687	4	1	11	0	0	0	1	0.342500	0.337746	0.692917	0.227612	4669	5172.266585	503.266585	89.221105
346	4	0	12	0	2	1	1	0.282500	0.301138	0.590000	0.140550	3523	3467.968455	-55.031545	98.437935
563	3	1	7	0	2	1	1	0.818333	0.755083	0.505833	0.114429	6786	8031.455815	1245.455815	81.646687
492	2	1	5	0	1	1	2	0.537500	0.527142	0.664167	0.230721	6273	4991.614763	-1281.385237	79.573008

Langkah di atas untuk menampilkan hasil prediksi model regresi linear secara detail, termasuk perbandingan antara nilai aktual, nilai prediksi, error, dan akurasi tiap baris data.

Link Github:

https://github.com/Jamilatun/ti03_Mila_01101222254/tree/main/praktikum03