Tugas Mandiri 3 : Analisis Multiple Regresi Linear Pada Dataset Bike Sharing

Raffa Yuda Pratama - 0110224081

Teknik Informatika, STT Terpadu Nurul Fikri, Depok
E-mail: 0110224081@student.nurulfikri.ac.id

Abstract. Laporan ini menerapkan regresi linear sederhana dan berganda pada dataset Bike Sharing (day.csv) untuk memprediksi jumlah penyewaan (cnt). Pada regresi sederhana digunakan satu fitur (temperatur), sedangkan pada regresi berganda digunakan delapan fitur (temperatur, temperatur_terasa, kelembaban, kecepatan_angin, musim, tahun, kondisi_cuaca, hari_kerja). Evaluasi dilakukan dengan metrik R^2, MAE, MSE, dan RMSE. Hasil menunjukkan model regresi berganda memberikan R^2 lebih tinggi dan error lebih rendah dibanding regresi sederhana, sehingga lebih cocok untuk tugas prediksi sederhana ini. Nama kolom dalam notebook juga telah diperbarui ke bahasa Indonesia untuk memudahkan pembacaan.

1. Persiapan Data dan Analisis

1.1. Import Library dan Load Dataset

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import statsmodels.api as sm

# Membaca dataset
df_raw = pd.read_csv('../../data/day.csv')

# Rename kolom ke bahasa Indonesia
df = df_raw.rename(columns={
    'temp': 'temperatur',
    'atemp': 'temperatur_terasa',
    'hum': 'kelembaban',
    'windspeed': 'kecepatan_angin',
    'season': 'musim',
    'yr': 'tahun',
    'weathersit': 'kondisi_cuaca',
    'workingday': 'hari_kerja',
    'casual': 'pengguna_kasual',
    'registered': 'pengguna_terdaftar'
})

print("Dataset Shape:", df.shape)
print("\n5 Data Pertama:")
df.head()
```

✓ 0.5s

```
Dataset Shape: (731, 16)

5 Data Pertama:
```

	instant	dteday	musim	tahun	mnth	holiday	weekday	hari_kerja	kondisi_cuaca	temperatur	temperatur_terasa	kelembaban	kecepatan_angin	pengguna_kasual	pengguna_terdaftar	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 1 Import Library dan Load Dataset

Bagian ini merupakan tahap persiapan awal dimana semua library yang diperlukan diimport dan dataset bike sharing dibaca dari file CSV dengan melakukan rename kolom ke bahasa Indonesia.

- **Import pandas dan numpy**: Library untuk manipulasi data dan operasi numerik
- **Import matplotlib dan seaborn**: Library untuk visualisasi data dalam bentuk grafik
- **Import sklearn modules**: Untuk split data, membuat model regresi, dan menghitung metrik evaluasi
- **Import statsmodels**: Untuk analisis regresi yang lebih detail dengan output statistik lengkap

- **Membaca dataset**: Menggunakan pd.read_csv() untuk load file day.csv ke dalam df_raw
- **Rename kolom**: Menggunakan .rename() untuk mengubah nama kolom ke bahasa Indonesia
    - temp → temperatur
    - atemp → temperatur_terasa
    - hum → kelembaban
    - windspeed → kecepatan_angin
    - season → musim
    - yr → tahun
    - weathersit → kondisi_cuaca
    - workingday → hari_kerja
    - casual → pengguna_kasual
    - registered → pengguna_terdaftar
- **Menampilkan shape**: Melihat dimensi dataset (731 baris dan jumlah kolom)
- **Menampilkan 5 data pertama**: Menggunakan df.head() untuk preview struktur dan isi dataset

1.2. Eksplorasi Data dan Statistik Deskriptif

```
print("Statistik Deskriptif:")
df.describe()
```
✓ 0.6s

Statistik Deskriptif:

	instant	musim	tahun	mnth	holiday	weekday	hari_kerja	kondisi_cuaca	temperatur	temperatur_terasa	kelembaban	kecepatan_angin	peng
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	
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	
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	
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	
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	
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	
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	
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	

Gambar 2 Eksplorasi Data dan Statistik Deskriptif

Tahap ini melakukan eksplorasi awal untuk memahami karakteristik dan struktur dataset sebelum melakukan pemodelan.

- **df.describe()**: Menghitung statistik deskriptif (mean, std, min, max, quartiles) untuk semua kolom numerik
- **Identifikasi missing values**: Mengecek apakah ada data yang hilang atau null
- **Memahami rentang data**: Melihat nilai minimum dan maksimum setiap variabel
- **Mendeteksi potensi outlier**: Dari nilai min/max dan quartiles dapat mengidentifikasi data yang tidak normal

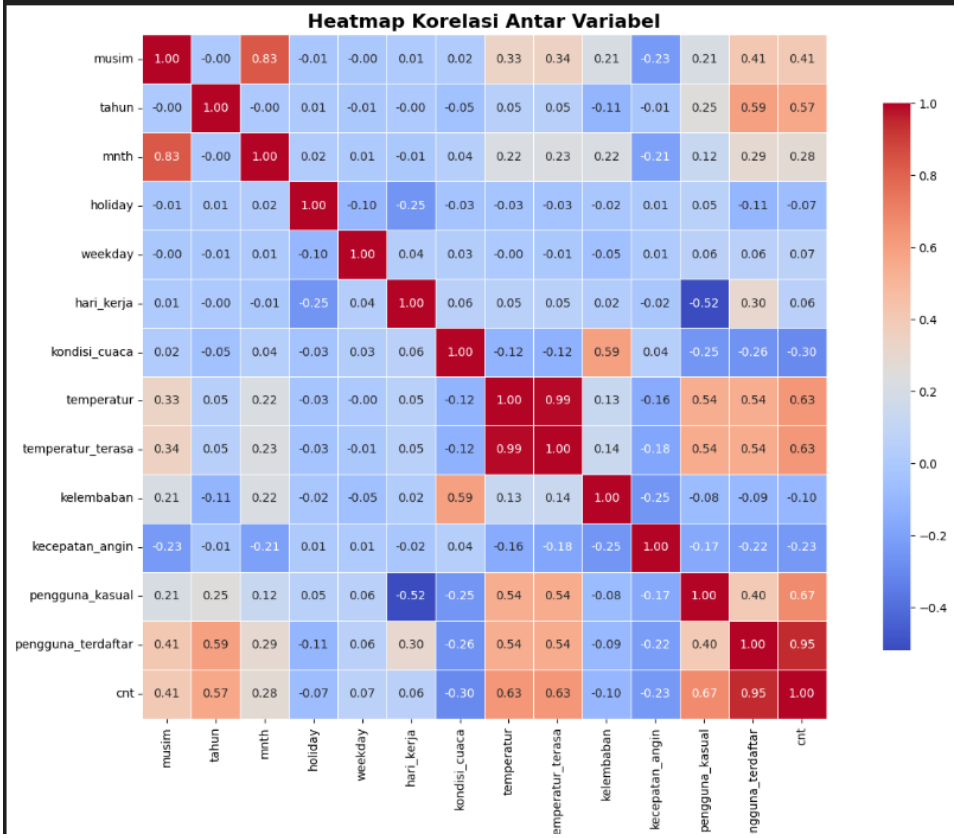
1.3. Analisis Korelasi

```python
# Pilih kolom numerik untuk analisis korelasi
numeric_cols = ['musim', 'tahun', 'mnth', 'holiday', 'weekday', 'hari_kerja',
                'kondisi_cuaca', 'temperatur', 'temperatur_terasa', 'kelembaban',
                'kecepatan_angin', 'pengguna_kasual', 'pengguna_terdaftar', 'cnt']

# Hitung korelasi
corr_matrix = df[numeric_cols].corr()

# Visualisasi heatmap korelasi
plt.figure(figsize=(14, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f',
            square=True, cbar_kws={"shrink": .8}, linewidths=0.5)
plt.title('Heatmap Korelasi Antar Variabel', fontsize=16, fontweight='bold')
plt.tight_layout()
plt.show()
```

✓ 4.3s



Gambar 3 Load dan Preview Data

Analisis korelasi sangat penting untuk menentukan variabel mana yang memiliki hubungan kuat dengan target prediksi (cnt).

- **Mendefinisikan kolom numerik**: Membuat list berisi nama-nama kolom yang bertipe numerik untuk analisis
- **Menghitung correlation matrix**: Menggunakan df.corr() untuk menghitung korelasi pearson antar semua variabel
- **Sorting korelasi dengan cnt**: Mengurutkan nilai korelasi dari tertinggi ke terendah untuk melihat variabel paling berpengaruh
- **Membuat heatmap**: Visualisasi matriks korelasi dengan warna, dimana merah = korelasi positif tinggi, biru = korelasi negatif
- **Set figure size**: Membuat ukuran plot 14x10 inch agar semua variabel terlihat jelas
- **Menambahkan anotasi**: Parameter annot=True menampilkan nilai korelasi di setiap cell
- **Interpretasi hasil**: Registered (0.95) dan casual (0.69) berkorelasi tinggi tapi tidak digunakan karena data leakage

2. Multiple Linear Regression

2.1. Split Data untuk Multiple Linear Regression

```python
# Variabel independent (X) - multiple features
X_multiple = df[['temperatur', 'temperatur_terasa', 'kelembaban', 'kecepatan_angin',
                 'musim', 'tahun', 'kondisi_cuaca', 'hari_kerja']]

# Variabel dependent (Y) tetap sama
# y sudah didefinisikan sebelumnya

# Split data: 80% training, 20% testing
X_train_mult, X_test_mult, y_train_mult, y_test_mult = train_test_split(
    X_multiple, y, test_size=0.2, random_state=42
)

print(f"Jumlah data training: {len(X_train_mult)}")
print(f"Jumlah data testing : {len(X_test_mult)}")
print(f"Jumlah fitur        : {X_train_mult.shape[1]}")
print(f"\nContoh data training:")
X_train_mult.head()
```

✓ 0.0s

```
Jumlah data training: 584
Jumlah data testing : 147
Jumlah fitur        : 8

Contoh data training:
```

	temperatur	temperatur_terasa	kelembaban	kecepatan_angin	musim	tahun	kondisi_cuaca	hari_kerja
682	0.343333	0.323225	0.662917	0.342046	4	1	2	1
250	0.633913	0.555361	0.939565	0.192748	3	0	3	1
336	0.299167	0.310604	0.612917	0.095783	4	0	1	0
260	0.507500	0.490537	0.695000	0.178483	3	0	1	0
543	0.697500	0.640792	0.360000	0.271775	3	1	1	1

Gambar 4 method describe()

Mempersiapkan data untuk model regresi berganda dengan menggunakan 8 variabel independent yang berkorelasi signifikan dengan target.

- **Mendefinisikan X_multiple**: Memilih 8 kolom fitur (temp, atemp, hum, windspeed, season, yr, weathersit, workingday)
- **Pemilihan fitur berdasarkan korelasi**: Variabel yang dipilih memiliki korelasi signifikan dengan cnt
- **Menghindari data leakage**: Tidak menggunakan 'registered' dan 'casual' meskipun korelasinya tinggi
- **Variabel y tetap sama**: Target prediksi masih cnt (jumlah penyewaan sepeda)
- **Split data 80:20**: Membagi data menjadi training dan testing dengan proporsi yang sama
- **random_state=42**: Menggunakan seed yang sama untuk konsistensi hasil
- **Print informasi**: Menampilkan jumlah data training, testing, dan jumlah fitur yang digunakan
- **Verifikasi data**: Menampilkan 5 baris pertama data training untuk memastikan split berhasil

2.2. Pemodelan Multiple Linear Regression

```python
# Tambahkan konstanta untuk statsmodels
X_train_const = sm.add_constant(X_train_mult)

# Buat model OLS (Ordinary Least Squares)
model_multiple = sm.OLS(y_train_mult, X_train_const).fit()

# Tampilkan summary model
print("="*60)
print("SUMMARY MODEL MULTIPLE LINEAR REGRESSION")
print("="*60)
print(model_multiple.summary())

# Tampilkan koefisien
print("\n" + "="*60)
print("KOEFISIEN MODEL")
print("="*60)
for idx, coef in enumerate(model_multiple.params):
    print(f"{model_multiple.params.index[idx]:15s}: {coef:10.4f}")
```

✓ 0.1s

```
============================================================
SUMMARY MODEL MULTIPLE LINEAR REGRESSION
============================================================
                            OLS Regression Results
==============================================================================
Dep. Variable:                    cnt   R-squared:                       0.782
Model:                            OLS   Adj. R-squared:                  0.779
Method:                 Least Squares   F-statistic:                     258.4
Date:                Sat, 11 Oct 2025   Prob (F-statistic):          7.54e-185
Time:                        16:15:32   Log-Likelihood:                -4796.8
No. Observations:                 584   AIC:                             9612.
Df Residuals:                     575   BIC:                             9651.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       1482.1561    266.750      5.556      0.000     958.233    2006.080
temp        2272.4747   1504.004      1.511      0.131    -681.536    5226.486
atemp       3369.3318   1701.957      1.980      0.048      26.522    6712.142
hum        -1058.1635    357.856     -2.957      0.003   -1761.029    -355.298
windspeed  -2107.1061    527.687     -3.993      0.000   -3143.536   -1070.677
season       425.1802     36.925     11.515      0.000     352.656     497.705
yr          2013.4631     75.242     26.760      0.000    1865.680    2161.246
weathersit  -603.2804     89.055     -6.774      0.000    -778.194    -428.367
...
season         :   425.1802
yr             :  2013.4631
weathersit     :  -603.2804
workingday     :   211.1847
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

Gambar 5 Pemodelan Multiple Linear Regression

Membuat model regresi berganda menggunakan statsmodels OLS untuk mendapatkan informasi statistik yang lebih detail.

- **Menambahkan konstanta**: sm.add_constant() menambahkan kolom konstanta (intercept) ke data training
- **Membuat model OLS**: Ordinary Least Squares, metode standar untuk regresi linear
- **Fitting model**: .fit() melatih model dengan data training yang sudah ditambah konstanta
- **Menampilkan summary**: Output lengkap berisi R^2, Adjusted R^2, F-statistic, koefisien, p-value, confidence interval
- **Interpretasi p-value**: Nilai $p < 0.05$ menunjukkan variabel signifikan secara statistik
- **Cek multikolinearitas**: Dari condition number di summary dapat mendeteksi

multikolinearitas

- **Loop koefisien**: Menampilkan semua koefisien model dalam format yang lebih mudah dibaca
- **Analisis koefisien**: Koefisien positif = meningkatkan cnt, negatif = menurunkan cnt

2.3. Persamaan Regresi Berganda

```
# Buat persamaan regresi
const = model_multiple.params['const']
params = model_multiple.params.drop('const')

print("PERSAMAAN REGRESI LINEAR BERGANDA:")
print("="*60)
equation = f"cnt = {const:.2f}"
for feature, coef in params.items():
    sign = "+" if coef >= 0 else "-"
    equation += f" {sign} {abs(coef):.2f}*{feature}"
print(equation)
print("="*60)
```

✓ 0.0s Python

```
PERSAMAAN REGRESI LINEAR BERGANDA:
============================================================
cnt = 1482.16 + 2272.47*temp + 3369.33*atemp - 1058.16*hum - 2107.11*windspeed + 425.18*season + 2013.46*yr - 603.28*weathersit + 211.18*workingda
============================================================
```

Gambar 6 Persamaan Regresi Berganda

Membuat persamaan matematis dari model regresi berganda untuk memudahkan interpretasi dan penggunaan model.

- **Ekstrak konstanta**: Mengambil nilai konstanta dari model_multiple.params['const']
- **Ekstrak koefisien variabel**: Menggunakan .drop('const') untuk mendapatkan koefisien semua variabel tanpa konstanta
- **Inisialisasi equation**: Memulai string persamaan dengan nilai konstanta
- **Loop semua variabel**: Iterasi setiap feature dan koefisiennya untuk membangun persamaan lengkap
- **Menentukan tanda**: Menggunakan "+" untuk koefisien positif dan "-" untuk negatif
- **Format koefisien**: Menggunakan abs() dan format .2f untuk menampilkan 2 desimal
- **Print persamaan lengkap**: Menampilkan persamaan dalam format cnt = const + coef1×var1 + ... + coef8×var8
- **Kegunaan persamaan**: Dapat digunakan untuk membuat prediksi manual atau memahami kontribusi setiap variable

2.4. Evaluasi Model Multiple Linear Regression

```python
# Tambahkan konstanta ke data testing
X_test_const = sm.add_constant(X_test_mult)

# Prediksi menggunakan data testing
y_pred_mult = model_multiple.predict(X_test_const)

# Hitung metrik evaluasi
r2_mult = r2_score(y_test_mult, y_pred_mult)
mae_mult = mean_absolute_error(y_test_mult, y_pred_mult)
mse_mult = mean_squared_error(y_test_mult, y_pred_mult)
rmse_mult = np.sqrt(mse_mult)

print("="*60)
print("EVALUASI MODEL MULTIPLE LINEAR REGRESSION")
print("="*60)
print(f"R² Score (Koefisien Determinasi): {r2_mult:.4f} ({r2_mult*100:.2f}%)")
print(f"MAE (Mean Absolute Error)       : {mae_mult:.2f}")
print(f"MSE (Mean Squared Error)        : {mse_mult:.2f}")
print(f"RMSE (Root Mean Squared Error)  : {rmse_mult:.2f}")
print("="*60)
```

✓ 0.0s

```
============================================================
EVALUASI MODEL MULTIPLE LINEAR REGRESSION
============================================================
R² Score (Koefisien Determinasi): 0.8183 (81.83%)
MAE (Mean Absolute Error)       : 641.30
MSE (Mean Squared Error)        : 728775.46
RMSE (Root Mean Squared Error)  : 853.68
============================================================
```

Gambar 7 Evaluasi Model Multiple Linear Regression

Mengukur performa model multiple regression pada data testing menggunakan berbagai metrik evaluasi.

- **Menambahkan konstanta ke data testing**: sm.add_constant(X_test_mult) agar konsisten dengan data training
- **Membuat prediksi**: model_multiple.predict() untuk memprediksi cnt pada data testing
- **Menghitung R^2 Score**: Mengukur seberapa baik model menjelaskan variansi data testing
- **Menghitung MAE**: Rata-rata error absolut dalam satuan jumlah penyewaan
- **Menghitung MSE**: Mean squared error untuk mengukur error dengan penalti kuadrat
- **Menghitung RMSE**: Root mean squared error, lebih interpretable karena satuannya sama dengan target
- **Perbandingan dengan simple regression**: R^2 lebih tinggi menunjukkan model lebih baik
- **Print hasil evaluasi**: Menampilkan semua metrik dengan format yang rapi dan mudah dibaca

2.5. Tabel Hasil Prediksi Multiple Regression

```
# Buat tabel hasil prediksi
hasil_multiple = pd.DataFrame({
    'CNT Aktual': y_test_mult.values,
    'CNT Prediksi': y_pred_mult,
    'Selisih Error': y_test_mult.values - y_pred_mult,
    'Akurasi (%)': (1 - abs(y_test_mult.values - y_pred_mult) / y_test_mult.values) * 100
})

# Gabungkan dengan fitur untuk analisis lebih detail
hasil_multiple = pd.concat([
    X_test_mult.reset_index(drop=True),
    hasil_multiple.reset_index(drop=True)
], axis=1)

print("Sample Hasil Prediksi (15 data pertama):")
hasil_multiple.head(15)
```

✓ 0.0s

Sample Hasil Prediksi (15 data pertama):

	temperatur	temperatur_terasa	kelembaban	kecepatan_angin	musim	tahun	kondisi_cuaca	hari_kerja	CNT Aktual	CNT Prediksi	Selisih Error	Akurasi (%)
0	0.475833	0.469054	0.733750	0.174129	4	1	1	1	6606	6322.625657	283.374343	95.710349
1	0.186957	0.177878	0.437826	0.277752	1	0	1	1	1550	1490.881215	59.118785	96.185885
2	0.330833	0.318812	0.585833	0.229479	4	0	2	1	3747	2910.050204	836.949796	77.663470
3	0.425833	0.417287	0.676250	0.172267	2	1	2	0	6041	4434.524018	1606.475982	73.407118
4	0.550000	0.544179	0.570000	0.236321	4	1	1	1	7538	6786.518468	751.481532	90.030757
5	0.716667	0.650271	0.633333	0.151733	3	1	1	1	7264	7208.763144	55.236856	99.239581
6	0.134783	0.144283	0.494783	0.188839	1	0	2	1	1605	782.923336	822.076664	48.780270
7	0.373333	0.377513	0.686250	0.274246	1	0	1	1	2209	2331.566798	-122.566798	94.451480
8	0.731667	0.667933	0.485833	0.080850	3	1	2	1	7499	7004.516135	494.483865	93.406003
9	0.722500	0.672992	0.684583	0.295400	2	1	1	1	5743	6516.440349	-773.440349	86.532468
10	0.255833	0.231700	0.483333	0.350754	1	1	1	0	1796	3429.045994	-1633.045994	9.073163
11	0.423333	0.426121	0.757500	0.047275	1	0	2	1	3068	2408.544454	659.455546	78.505360
12	0.604167	0.591546	0.507083	0.269283	2	0	1	1	4891	4202.505184	688.494816	85.923230
13	0.296667	0.289762	0.506250	0.210821	4	1	1	1	5260	5474.799494	-214.799494	95.916359
14	0.292500	0.302400	0.420833	0.120650	1	0	1	1	2133	2499.292900	-366.292900	82.827337

Gambar 8 Tabel Hasil Prediksi Multiple Regression

Membuat tabel komprehensif yang menggabungkan semua fitur input dengan hasil prediksi dan metrik evaluasinya.

- **Membuat DataFrame hasil**: Berisi actual, predicted, error, dan akurasi
- **Kolom Actual_cnt**: Nilai jumlah penyewaan aktual dari data testing
- **Kolom Predicted_cnt**: Hasil prediksi model multiple regression
- **Kolom Selisih_Error**: Dihitung sebagai actual - predicted untuk melihat arah error
- **Kolom Akurasi (%)**: Persentase akurasi dihitung dengan (1 - |error|/actual) × 100%
- **Menggabungkan dengan fitur**: pd.concat() untuk menggabungkan X_test_mult dengan hasil prediksi
- **Reset index**: Memastikan index konsisten setelah penggabungan DataFrame
- **Menampilkan 15 data**: .head(15) untuk melihat sample yang cukup representatif
- **Analisis detail**: Tabel ini memungkinkan analisis pengaruh kombinasi fitur terhadap akurasi prediksi

14. Kesimpulan

Model regresi berganda lebih baik daripada regresi sederhana dalam konteks dataset ini: R^2 meningkat dan nilai MAE/MSE/RMSE menurun, artinya prediksi lebih akurat ketika menggunakan beberapa fitur. Temperatur terbukti berpengaruh positif terhadap jumlah penyewaan, namun fitur seperti pengguna_terdaftar dan pengguna_kasual tidak digunakan karena menyebabkan data leakage. Untuk perbaikan selanjutnya bisa dilakukan feature engineering, pemeriksaan multikolinearitas, atau penggunaan regularisasi agar generalisasi lebih baik. Secara praktis, model ini sudah layak untuk latihan forecasting sederhana dan dapat membantu estimasi kebutuhan sepeda berdasarkan kondisi cuaca dan temporal.

Link Github Praktikum :https://github.com/raffayuda/Machine-Learning/blob/main/pertemuan3/Notebook/praktikum/praktikum3.ipynb

Link Github Praktikum Mandiri : https://github.com/raffayuda/Machine-Learning/blob/main/pertemuan3/Notebook/tugas/mandiri3.ipynb