Praktikum 3: Machine Learning – Regresi Dan Evaluasi Model

Muhammad Shiddiq 1 - 0110222199 [1]

[1] Teknik Informatika, STT Terpadu Nurul Fikri, Depok

E-mail: muhammadshiddiq785@gmail.com

Abstract. Regresi merupakan salah satu teknik dalam pembelajaran mesin dan statistika yang digunakan untuk memodelkan hubungan antara variabel independen (prediktor) dan variabel dependen (target). Tujuan utamanya adalah untuk memprediksi nilai output berdasarkan input yang diberikan dengan membangun fungsi terbaik yang menggambarkan pola data. Proses regresi dapat dilakukan menggunakan berbagai metode seperti regresi linear, regresi polinomial, dan regresi non-linear lainnya. Setelah model regresi dibangun, tahap evaluasi model dilakukan untuk menilai seberapa baik model tersebut mampu memprediksi data. Evaluasi biasanya menggunakan metrik seperti Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan R-squared (R^2). Metrik-metrik ini membantu menentukan akurasi, efisiensi, serta kemampuan generalisasi model terhadap data baru, sehingga model yang dihasilkan tidak hanya cocok pada data pelatihan tetapi juga memiliki performa baik pada data uji.

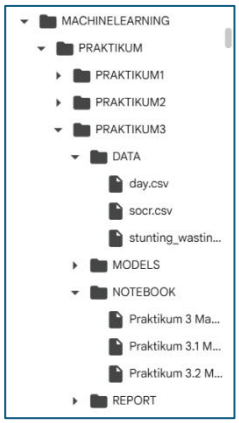
1. Praktikum Mandiri – Membuat Model Prediksi Jumlah Penyewaan Sepeda

Model prediksi jumlah penyewaan sepeda dibuat untuk memperkirakan berapa banyak sepeda yang akan disewa pada waktu tertentu, misalnya per jam atau per hari. Tujuan utamanya adalah membantu pengelola layanan sepeda agar dapat mengatur jumlah sepeda yang tersedia sesuai dengan kebutuhan pengguna. Dalam pembuatannya, data historis penyewaan sepeda digunakan, yang biasanya berisi informasi seperti tanggal, suhu, kelembapan, kecepatan angin, dan kondisi cuaca.

Metode yang umum digunakan untuk membuat model ini adalah regresi, karena metode ini dapat menggambarkan hubungan antara faktor-faktor tersebut dengan jumlah sepeda yang disewa. Setelah model dibangun, langkah selanjutnya adalah melakukan evaluasi untuk mengetahui seberapa baik model tersebut dalam memprediksi data baru. Beberapa ukuran yang digunakan antara lain MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), dan R^2 (R-squared). Nilai MAE dan RMSE yang kecil menunjukkan bahwa prediksi model cukup akurat, sedangkan nilai R^2 yang tinggi menunjukkan bahwa model mampu menjelaskan sebagian besar variasi data. Dengan hasil ini, model dapat digunakan sebagai dasar pengambilan keputusan untuk mengoptimalkan penyediaan sepeda di berbagai kondisi.

1.1 Membuat Folder

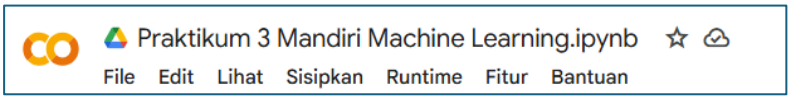
Langkah pertama kita harus membuat folder yang terstruktur dan juga rapih di google drive.



Gambar 1. Membuat folder di google drive, agar mudah untuk diakses

1.2 Membuat file notebook google colab

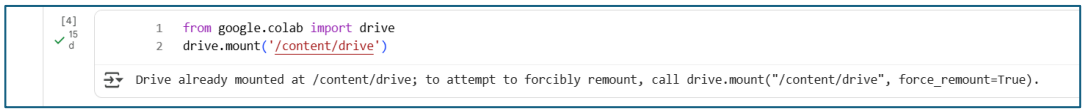
Selanjutnya membuat file notebook di google colab untuk praktikum.



Gambar 2. Membuat file google colab

1.3 *Menghubungkan google colab dengan google drive*

Selanjutnya menghubungkan google colab dengan google drive menggunakan perintah "From google.colab import drive
Drive,mount('/content/drive')".



Gambar 3. Menghubungkan google colab dengan google drive

1.4 Meng install pandas

Selanjutnya meng install library pandas dengan perintah "!pip install pandas".

Gambar 4. Meng install pandas.

1.5 Meng import library pandas

Selanjutnya meng import library pandas dengan perintah "import pandas as pd". Pandas adalah perpustakaan Python sumber terbuka yang banyak digunakan untuk analisis dan manipulasi data. Perpustakaan ini menyediakan struktur data yang kuat dan fleksibel, terutama Series dan DataFrame, yang dirancang untuk menangani data terstruktur dengan efisien.

```
[1]
✓ 1 d
1  import pandas as pd
```

Gambar 5. Mengimport library pandas

1.6 Membaca dataset

Selanjutnya membaca dataset day.csv yang ada di google drive menggunakan perintah "df = pd.read_csv('/content/drive/MyDrive/MACHINELEARNING/PRAKTIKUM/PRAKTIKUM 3/DATA/day.csv') df"

```
1  df = pd.read_csv(path + "day.csv")
2  df
```

Gambar 6. Membaca dataset day.csv.

Tabel 1. Berikut adalah hasil dataset yang telah dibaca.

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
726	727	2012-12-27	1	1	12	0	4	1	2	0.254167	0.226642	0.652917	0.350133	247	1867	2114
727	728	2012-12-28	1	1	12	0	5	1	2	0.253333	0.255046	0.590000	0.155471	644	2451	3095
728	729	2012-12-29	1	1	12	0	6	0	2	0.253333	0.242400	0.752917	0.124383	159	1182	1341
729	730	2012-12-30	1	1	12	0	0	0	1	0.255833	0.231700	0.483333	0.350754	364	1432	1796
730	731	2012-12-31	1	1	12	0	1	1	2	0.215833	0.223487	0.577500	0.154846	439	2290	2729

731 rows × 16 columns

1.7 Mengecek informasi dataset

Selanjutnya mengecek informasi dataset yang dibaca, dari total, jumlah kolum, missing value, dan type data menggunakan perintah "df.info()"

```
1  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 7. Mengecek informasi dataset.

1.8 Mencari nilai statistik deskriptif secara cepat

Selanjutnya mencari nilai statistik dari dataset dengan cepat menggunakan perintah, df.describe().

	instant	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual
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
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
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
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
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
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
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
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

Gambar 8. Mencari nilai statistik deskriptif.

1.9 Menentukan variable independent dan dependent

Selanjutnya menentukan variable independent dan dependent.

```
X = df.drop(['instant', 'dteday', 'casual', 'registered', 'cnt'], axis=1)
y = df['cnt']

print("Shape X:", X.shape)
print("Shape y:", y.shape)

```

```
Shape X: (731, 11)
Shape y: (731,)
```

Gambar 9. Mencari nilai korelasi.

1.10 Membagi data testing dan training

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Training data:", X_train.shape)
print("Testing data:", X_test.shape)
```

```
Training data: (584, 11)
Testing data: (147, 11)
```

Gambar 10. Membagi data.

1.11 Menginstall model linear regresi

```
from sklearn.linear_model import LinearRegression

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

```
LinearRegression()
```

Gambar 11. Menginstall model linear regresi.

1.12 Menyiapkan model prediksi

```
y_pred = model.predict(X_test)
```

Gambar 12. Menyiapkan model prediksi.

1.13 Evaluasi model

```python
from sklearn.metrics import r2_score, mean_squared_error
import numpy as np

r2 = r2_score(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print("R² Score:", r2)
print("RMSE:", rmse)
```

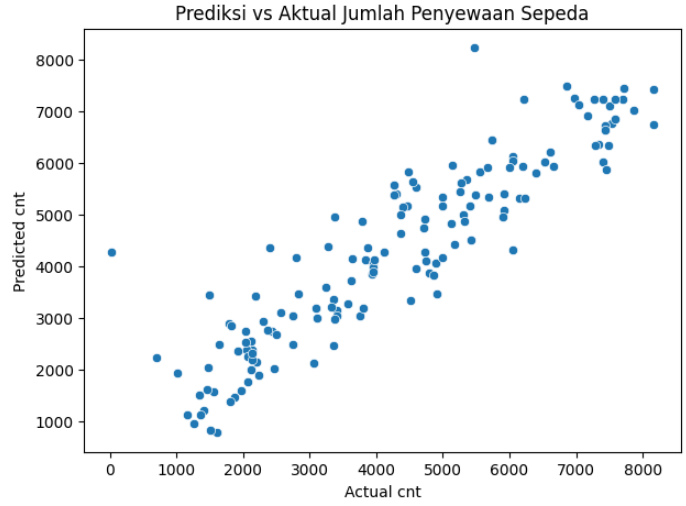
```
R² Score: 0.8276670090367212
RMSE: 831.2851545662686
```

Gambar 13. Mengevaluasi model.

1.14 Visualisasi Data

```python
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(7,5))
sns.scatterplot(x=y_test, y=y_pred)
plt.xlabel("Actual cnt")
plt.ylabel("Predicted cnt")
plt.title("Prediksi vs Aktual Jumlah Penyewaan Sepeda")
plt.show()
```



Gambar 14. Hasil Visualisasi.

Link Github : https://github.com/Shid2iq/Machine-Learning

Referensi:

Munir, S., Seminar, K. B., Sudradjat, Sukoco, H., & Buono, A. (2022). The Use of Random Forest Regression for Estimating Leaf Nitrogen Content of Oil Palm Based on Sentinel 1-A Imagery. *Information*, *14*(1), 10. https://doi.org/10.3390/info14010010

Seminar, K. B., Imantho, H., Sudradjat, Yahya, S., Munir, S., Kaliana, I., Mei Haryadi, F., Noor Baroroh, A., Supriyanto, Handoyo, G. C., Kurnia Wijayanto, A., Ijang Wahyudin, C., Liyantono, Budiman, R., Bakir Pasaman, A., Rusiawan, D., & Sulastri. (2024). PreciPalm: An Intelligent System for Calculating Macronutrient Status and Fertilizer Recommendations for Oil Palm on Mineral Soils Based on a Precision Agriculture Approach. *Scientific World Journal*, *2024*(1). https://doi.org/10.1155/2024/1788726