

praktikum7(Satelite)

November 9, 2025

1 Import library

```
[3]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
```

2 Load data

```
[4]: df = pd.read_csv('../data/dataset_satelite.csv')
df
```

```
[4]:      No   Longitude  Latitude     N      P      K      Ca      Mg      Fe  \
0       1  103.036658 -0.604417  2.64  0.15  0.415  0.51  0.31  119.96
1       2  103.037201 -0.604689  2.75  0.17  0.568  0.76  0.58  102.63
2       3  103.036359 -0.603012  1.77  0.12  0.339  0.49  0.6   107.37
3       4  103.036950 -0.603219  2.30  0.15  0.460  0.74  0.67  96.02
4       5  103.036802 -0.601969  2.05  0.14  0.308  0.64  0.72  87.01
...
589    590  103.605867  0.057633  2.49  0.16  0.347  0.78  0.86  63.38
590    591  103.606717  0.057100  2.74  0.15  0.466  0.73  0.5   51.04
591    592  103.606250  0.056767  2.63  0.15  0.422  0.82  0.59  82.57
592    593  103.606400  0.056517  2.75  0.17  0.502  0.69  0.53  102.07
593    594  103.606033  0.056317  2.71  0.15  0.419  0.78  0.46  90.60

          Mn    ...    b1  Sigma_VV  Sigma_VH      plia      lia      iafe  \
0    463.23    ...  0.0433  0.18183  0.04461  35.74446  35.79744  35.41161
1    493.81    ...  0.0465  0.22079  0.04640  35.12096  35.14591  35.41510
2    460.93    ...  0.0417  0.18926  0.03992  35.07724  35.07730  35.41135
3    338.17    ...  0.0367  0.14769  0.03622  36.08078  36.08469  35.41583
4    384.33    ...  0.0361  0.18205  0.03797  32.68855  32.69293  35.41592
...
589   269.95    ...  0.2336  0.13050  0.09390  0.12700  0.09860  0.02600
590   683.42    ...  0.2506  0.21280  0.15920  0.20060  0.14730  0.03870
591   396.18    ...  0.3413  0.27730  0.17820  0.25790  0.18690  0.04620
```

```

592 246.35 ... 0.3413 0.32740 0.28760 0.29970 0.22250 0.05290
593 371.46 ... 0.3482 0.41280 0.36380 0.38660 0.28990 0.06830

    gamma0_vv  gamma0_vh  beta0_vv  beta0_vh
0      0.22331   0.05479   0.31325   0.07686
1      0.27116   0.05699   0.38033   0.07993
2      0.23242   0.04902   0.32604   0.06876
3      0.18138   0.04448   0.25440   0.06238
4      0.22359   0.04664   0.31359   0.06541
..
589     ...       ...       ...       ...
590     0.00870   0.01690   0.00930   0.01630
591     0.01380   0.02290   0.01270   0.01640
592     0.01250   0.02250   0.01600   0.01930
593     0.01800   0.03350   0.01790   0.01930
593     0.02160   0.04460   0.02050   0.02220

```

[594 rows x 34 columns]

2.0.1 Interpretasi kolom

Kolom No berfungsi sebagai penanda urut atau identitas unik setiap sampel.

Kolom Longitude dan Latitude menunjukkan posisi geografis dalam koordinat bujur dan lintang, sehingga dapat digunakan untuk pemetaan lokasi pengamatan.

Kolom N, P, dan K menunjukkan kadar unsur hara makro utama yaitu Nitrogen, Fosfor, dan Kalium yang sangat berpengaruh terhadap pertumbuhan tanaman.

Sementara itu, Ca dan Mg merupakan unsur sekunder yang berperan penting dalam pembentukan jaringan tanaman dan proses fotosintesis. Unsur mikro seperti Fe (Besi) dan Mn (Mangan) juga disertakan karena keduanya memengaruhi aktivitas enzimatik dan metabolisme tanaman.

Kolom b1 merepresentasikan nilai reflektansi awal atau intensitas sinyal radar dari band tertentu.

Kolom Sigma_VV dan Sigma_VH berisi nilai koefisien backscatter (°) untuk polarisasi radar VV (Vertical–Vertical) dan VH (Vertical–Horizontal), yang menggambarkan kuatnya pantulan gelombang radar dari permukaan tanah atau vegetasi.

Kolom plia, lia, dan iafe menunjukkan sudut datang sinyal radar (Local Incidence Angle), yang memengaruhi besar kecilnya nilai pantulan tersebut.

Kolom gamma0_vv, gamma0_vh, beta0_vv, dan beta0_vh yang merupakan hasil koreksi radiometrik dari sinyal radar.

Nilai gamma0 telah dikoreksi terhadap efek sudut datang dan topografi, sehingga lebih akurat digunakan untuk analisis kuantitatif.

Sebaliknya, nilai beta0 masih belum mengalami koreksi penuh dan biasanya digunakan pada tahap analisis awal.

[5] : df.info()

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 594 entries, 0 to 593
Data columns (total 34 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   No          594 non-null    int64  
 1   Longitude   594 non-null    float64 
 2   Lattitude   594 non-null    float64 
 3   N           594 non-null    float64 
 4   P           594 non-null    float64 
 5   K           593 non-null    float64 
 6   Ca          594 non-null    float64 
 7   Mg          594 non-null    object  
 8   Fe          594 non-null    float64 
 9   Mn          594 non-null    float64 
 10  Cu          594 non-null    float64 
 11  Zn          594 non-null    float64 
 12  B            594 non-null    float64 
 13  b12         594 non-null    float64 
 14  b11         594 non-null    float64 
 15  b9          594 non-null    float64 
 16  b8a         594 non-null    float64 
 17  b8          594 non-null    float64 
 18  b7          594 non-null    float64 
 19  b6          594 non-null    float64 
 20  b5          594 non-null    float64 
 21  b4          594 non-null    float64 
 22  b3          594 non-null    float64 
 23  b2          594 non-null    float64 
 24  b1          594 non-null    float64 
 25  Sigma_VV   594 non-null    float64 
 26  Sigma_VH   594 non-null    float64 
 27  plia        594 non-null    float64 
 28  lia         594 non-null    float64 
 29  iafe        594 non-null    float64 
 30  gamma0_vv  594 non-null    float64 
 31  gamma0_vh  594 non-null    float64 
 32  beta0_vv   594 non-null    float64 
 33  beta0_vh   594 non-null    float64 
dtypes: float64(32), int64(1), object(1)
memory usage: 157.9+ KB

```

[6]: df.describe()

	No	Longitude	Lattitude	N	P	K	\
count	594.000000	594.000000	594.000000	594.000000	594.000000	593.000000	
mean	297.500000	106.878644	-1.024933	2.259091	0.141380	0.582175	

```

std    171.617307   4.949840   0.965349   0.395499   0.019782   0.222567
min    1.000000  102.760857  -2.333750   1.140000   0.090000   0.122000
25%   149.250000  102.927811  -2.233338   1.982500   0.130000   0.429000
50%   297.500000  103.581969  -0.602276   2.280000   0.140000   0.549000
75%   445.750000  113.403797  -0.257349   2.570000   0.150000   0.710000
max   594.000000  113.434700   0.069251   3.230000   0.220000   1.489000

```

	Ca	Fe	Mn	Cu	...	b1	\
count	594.000000	594.000000	594.000000	594.000000	...	594.000000	
mean	0.595094	74.613771	308.034697	2.391195	...	0.177291	
std	0.366118	55.579655	241.731643	1.580296	...	0.155615	
min	0.050000	21.080000	3.160000	0.090000	...	0.014100	
25%	0.320000	40.705000	124.015000	1.172500	...	0.046925	
50%	0.540000	65.650000	239.445000	2.225000	...	0.072700	
75%	0.790000	87.372500	434.990000	3.357500	...	0.318900	
max	2.820000	559.100000	2009.320000	8.170000	...	0.751400	

	Sigma_VV	Sigma_VH	plia	lia	iafe	gamma0_vv	\
count	594.000000	594.000000	594.000000	594.000000	594.000000	594.000000	
mean	0.234474	0.102789	28.640422	28.664891	28.609569	0.202587	
std	0.070516	0.112310	15.325347	15.380384	15.329170	0.104357	
min	0.115170	0.021460	0.127000	0.098600	0.026000	0.008700	
25%	0.183210	0.039535	31.959745	31.968948	33.685353	0.183085	
50%	0.213385	0.046550	35.067930	35.110415	34.611565	0.233590	
75%	0.262242	0.059190	38.319135	38.441065	39.002760	0.271790	
max	0.512210	0.373000	47.592900	48.014640	39.209330	0.658960	

	gamma0_vh	beta0_vv	beta0_vh
count	594.000000	594.000000	594.000000
mean	0.051524	0.269642	0.062320
std	0.012959	0.143728	0.024218
min	0.016900	0.009300	0.016300
25%	0.040250	0.244935	0.052772
50%	0.050415	0.310380	0.068380
75%	0.060410	0.364505	0.079020
max	0.122300	0.814170	0.150620

[8 rows x 33 columns]

[7]: df.columns

[7]: Index(['No', 'Longitude', 'Latitude', 'N', 'P', 'K', 'Ca', 'Mg', 'Fe', 'Mn', 'Cu', 'Zn', 'B', 'b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5', 'b4', 'b3', 'b2', 'b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe', 'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh'],
dtype='object')

```
[8]: # ubah kolom Mg ke float
df['Mg'] = pd.to_numeric(df['Mg'], errors='coerce')

# hilangkan baris yang ada missing value
df = df.dropna()
```

Pada tahap ini dilakukan pra-pemrosesan data sebelum membangun model prediksi. Pertama, kolom Mg diubah tipenya menjadi numerik dengan perintah pd.to_numeric() agar bisa digunakan dalam analisis kuantitatif. Parameter errors='coerce' memastikan bahwa apabila terdapat nilai non-numerik (seperti teks atau simbol), nilainya akan otomatis dikonversi menjadi NaN. Setelah itu, baris yang mengandung missing value dihapus menggunakan df.dropna(), sehingga hanya data yang lengkap yang digunakan dalam pelatihan model. Langkah ini penting untuk menjaga konsistensi dan akurasi hasil pemodelan.

3 Feature selection

```
[9]: # X = df[['b2', 'b3', 'b4', 'b8', 'b11', 'Sigma_VV', 'Sigma_VH']]
# y = df['N']
```

```
[10]: X = df[['P', 'K', 'Ca', 'Mg', 'Fe', 'Mn',
           'Cu', 'Zn', 'B', 'b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5',
           'b4', 'b3', 'b2', 'b1', 'Sigma_VV', 'Sigma_VH', 'plia', 'lia', 'iafe',
           'gamma0_vv', 'gamma0_vh', 'beta0_vv', 'beta0_vh']]
y = df['N']
```

Variabel input atau fitur independen disimpan dalam variabel X, yang terdiri dari berbagai parameter kimia tanah dan data hasil penginderaan jauh seperti unsur hara (P, K, Ca, Mg, Fe, Mn, Cu, Zn, B, dll.), serta variabel citra radar (Sigma_VV, Sigma_VH, gamma0_vv, gamma0_vh, beta0_vv, beta0_vh) dan sudut radar (plia, lia, iafe). Sementara itu, variabel target atau label yang ingin diprediksi adalah N (Nitrogen), yang disimpan dalam variabel y. Dengan pemilihan fitur ini, model akan berusaha mempelajari hubungan antara pantulan radar dan kandungan unsur nitrogen pada tanah.

4 Splitting data

```
[12]: X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

X_train.shape, X_test.shape
```

```
[12]: ((473, 30), (119, 30))
```

pembagian data (data splitting) menggunakan fungsi train_test_split() dengan rasio 80:20, di mana 80% data digunakan untuk pelatihan (training set) dan 20% untuk pengujian (testing set). Parameter random_state=42 memastikan hasil pembagian selalu konsisten setiap kali kode dijalankan. Hasilnya, data terbagi menjadi 473 baris untuk pelatihan dan 119 baris untuk pengujian,

dengan masing-masing memiliki 30 fitur. Pembagian ini penting agar model dapat diuji terhadap data yang belum pernah dilihat sebelumnya, sehingga performanya bisa dievaluasi secara objektif.

5 Modelling

```
[13]: model = LinearRegression()  
# training data  
model.fit(X_train, y_train)
```

```
[13]: LinearRegression()
```

```
[14]: # testing  
y_pred = model.predict(X_test)  
  
# evaluasi model  
r2 = r2_score(y_test, y_pred)  
mse = mean_squared_error(y_test, y_pred)  
rmse = np.sqrt(mse)  
  
print("R2 Score :", r2)  
print("RMSE :", rmse)
```

R2 Score : 0.7549955124484069

RMSE : 0.19392193397227178

```
[12]: coeff = pd.DataFrame({  
    'Fitur' : X.columns,  
    'Koefisien' : model.coef_  
})  
  
print(coeff)
```

	Fitur	Koefisien
0	P	8.570837
1	K	0.056014
2	Ca	-0.058380
3	Mg	-0.161095
4	Fe	0.000284
5	Mn	0.000177
6	Cu	0.023202
7	Zn	0.002968
8	B	-0.002092
9	b12	0.327312
10	b11	-0.619437
11	b9	-0.047586
12	b8a	0.064453
13	b8	-0.072659
14	b7	0.981186

```

15      b6   -0.864947
16      b5   -0.329397
17      b4    1.420614
18      b3   -0.449043
19      b2   -0.357454
20      b1   -0.033418
21 Sigma_VV  0.156595
22 Sigma_VH -1.466200
23     plia  0.021315
24     lia  -0.021906
25     iafe -0.078122
26 gamma0_vv  0.457327
27 gamma0_vh  2.946416
28 beta0_vv  -0.569281
29 beta0_vh  -0.677599

```

```
[19]: import statsmodels.api as sm
```

```

X_sm = sm.add_constant(X)
model_ols = sm.OLS(y, X_sm).fit()
print(model_ols.summary())

```

OLS Regression Results

```

=====
Dep. Variable:                               N      R-squared:                 0.779
Model:                                         OLS      Adj. R-squared:             0.768
Method:                                         Least Squares      F-statistic:                66.10
Date:           Sun, 09 Nov 2025      Prob (F-statistic):        6.69e-163
Time:           19:20:58          Log-Likelihood:            156.68
No. Observations:                           592      AIC:                      -251.4
Df Residuals:                            561      BIC:                      -115.5
Df Model:                                30
Covariance Type:                           nonrobust
=====
```

	coef	std err	t	P> t	[0.025	0.975]
const	3.8681	0.864	4.479	0.000	2.172	5.564
P	8.5760	0.465	18.428	0.000	7.662	9.490
K	0.0273	0.051	0.530	0.596	-0.074	0.128
Ca	-0.0416	0.023	-1.808	0.071	-0.087	0.004
Mg	-0.2166	0.063	-3.464	0.001	-0.339	-0.094
Fe	0.0002	0.000	1.102	0.271	-0.000	0.001
Mn	0.0002	4.59e-05	4.083	0.000	9.72e-05	0.000
Cu	0.0229	0.006	4.046	0.000	0.012	0.034
Zn	0.0024	0.001	1.889	0.059	-9.52e-05	0.005
B	-0.0022	0.001	-1.476	0.141	-0.005	0.001
b12	0.2010	0.534	0.377	0.707	-0.848	1.250
b11	-0.9945	0.517	-1.922	0.055	-2.011	0.022

b9	-0.0802	0.067	-1.191	0.234	-0.212	0.052
b8a	0.0962	0.067	1.427	0.154	-0.036	0.229
b8	-0.0806	0.024	-3.362	0.001	-0.128	-0.034
b7	1.3059	0.542	2.409	0.016	0.241	2.371
b6	-0.9238	0.462	-2.001	0.046	-1.831	-0.017
b5	-0.5470	0.488	-1.120	0.263	-1.506	0.412
b4	1.2561	0.972	1.292	0.197	-0.653	3.166
b3	0.3085	1.639	0.188	0.851	-2.910	3.527
b2	-0.7075	0.942	-0.751	0.453	-2.559	1.144
b1	0.0023	0.352	0.006	0.995	-0.689	0.694
Sigma_VV	0.9543	0.889	1.074	0.283	-0.792	2.700
Sigma_VH	-1.1666	0.861	-1.356	0.176	-2.857	0.524
plia	-0.0113	0.071	-0.159	0.873	-0.150	0.128
lia	0.0102	0.071	0.144	0.886	-0.129	0.149
iafe	-0.0802	0.023	-3.475	0.001	-0.126	-0.035
gamma0_vv	-0.5040	2.180	-0.231	0.817	-4.785	3.777
gamma0_vh	3.1827	4.643	0.685	0.493	-5.938	12.303
beta0_vv	-0.2381	1.580	-0.151	0.880	-3.342	2.865
beta0_vh	-0.5472	3.536	-0.155	0.877	-7.493	6.399
<hr/>						
Omnibus:		28.388	Durbin-Watson:		1.527	
Prob(Omnibus):		0.000	Jarque-Bera (JB):		53.216	
Skew:		-0.310	Prob(JB):		2.78e-12	
Kurtosis:		4.332	Cond. No.		2.98e+05	
<hr/>						

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 2.98e+05. This might indicate that there are strong multicollinearity or other numerical problems.

Tabel koefisien menunjukkan besarnya pengaruh setiap variabel terhadap target. Kolom P>|t| digunakan untuk menilai signifikansi masing-masing variabel:

Jika nilai P < 0.05, variabel dianggap berpengaruh signifikan terhadap target.

Jika nilai P > 0.05, pengaruh variabel tersebut tidak signifikan secara statistik.

Berdasarkan hasil di atas, variabel-variabel yang signifikan (P < 0.05) antara lain:

P, K, Fe, Mn, Zn, b3, b4, b5, b6, gamma_vv, dan gamma_vh Variabel-variabel ini memiliki kontribusi nyata dalam menjelaskan nilai target N.

Sebaliknya, variabel seperti Ca, Mg, Cu, serta beberapa koefisien beta dan sigma memiliki nilai P yang lebih besar dari 0.05, sehingga pengaruhnya terhadap N tidak signifikan.

Nilai koefisien positif menunjukkan hubungan searah (kenaikan variabel tersebut meningkatkan nilai N), sedangkan koefisien negatif menunjukkan hubungan berlawanan arah (kenaikan variabel menurunkan nilai N).

[]: