

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
from sklearn import tree
import matplotlib.pyplot as plt
import seaborn as sns

from google.colab import drive
drive.mount('/content/gdrive')

import os

path = "/content/gdrive/MyDrive/praktikum_ml/praktikum07"
try:
    print(os.listdir(path))
except FileNotFoundError:
    print(f"Directory not found: {path}")

data = pd.read_csv(path + "/data/dataset_satelit.csv")
data

Drive already mounted at /content/gdrive; to attempt to forcibly
remount, call drive.mount("/content/gdrive", force_remount=True).
['data', 'notebooks']

{"type": "dataframe", "variable_name": "data"}

```

Kode di atas adalah skrip Python yang dirancang untuk persiapan dan implementasi model Machine Learning Klasifikasi, khususnya menggunakan Decision Tree, di lingkungan Google Colab. Penjelasan Singkat Kode Skrip dimulai dengan mengimpor pustaka utama yang dibutuhkan untuk sains data: pandas untuk manipulasi data, numpy untuk operasi numerik, serta modul sklearn untuk membangun model Decision Tree (DecisionTreeClassifier), membagi data (train\_test\_split), dan mengevaluasi kinerja model (accuracy\_score, classification\_report, confusion\_matrix). Dilanjutkan dengan mengimpor matplotlib.pyplot dan seaborn untuk visualisasi data dan hasil model. Bagian berikutnya melakukan otorisasi untuk mengakses Google Drive menggunakan drive.mount('/content/gdrive'), yang merupakan langkah wajib saat bekerja dengan data yang tersimpan di Drive melalui Colab. Setelah mounting, kode menentukan jalur (path) ke direktori data spesifik di Google Drive. Skrip kemudian mencoba mencetak daftar file di dalam direktori tersebut untuk verifikasi. Akhirnya, data dimasukkan ke dalam DataFrame Pandas dengan membaca file dataset\_satelit.csv dari jalur yang ditentukan, dan menampilkan data tersebut (data), menandai kesiapan data untuk tahap pemrosesan dan pelatihan model selanjutnya.

```

data.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

```

Output dari `data.info()` menunjukkan bahwa *DataFrame* berisi **594 entri** dan **34 kolom**. Data ini didominasi oleh tipe **numerik (float64)**, yang merupakan kabar baik untuk pemodelan Machine Learning. Namun, terdapat dua masalah utama yang perlu ditangani dalam tahap pra-pemrosesan data: **nilai yang hilang** dan **tipe data yang salah**. Kolom **K** memiliki **satu nilai yang hilang** (hanya 593 non-null), yang harus diisi atau dihapus. Lebih krusial lagi, kolom **Mg** (Magnesium), yang seharusnya numerik, terdeteksi sebagai tipe **object**. Ini mengindikasikan adanya data non-numerik (seperti teks atau karakter khusus) di dalam kolom tersebut, dan **harus dikonversi** ke tipe numerik agar dapat digunakan oleh model *Decision Tree* yang akan dibangun.

```

print(data.columns)

Index(['No', 'Longitude', 'Lattitude', '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')

```

Output `print(data.columns)` menampilkan **daftar semua kolom** atau **fitur** yang ada dalam *Dataframe* Anda. Daftar ini merupakan *Index* objek yang berisi **34 nama kolom** yang akan digunakan untuk pemodelan Machine Learning. Kolom-kolom tersebut terdiri dari: **identifikasi** (No), **lokasi geografis** (Longitude, Lattitude), **kadar unsur hara tanah** (seperti N, P, K, Mg, Fe, dll.), **data spektral optik satelit** (b1 hingga b12), dan **data radar satelit** (Sigma\_VV, Sigma\_VH, gamma0\_vv, beta0\_vh, dll.). Mengetahui nama-nama kolom ini sangat penting karena membantu Anda mengidentifikasi fitur target (jika ada), fitur yang perlu dihilangkan, serta fitur-fitur masukan yang akan digunakan untuk melatih model klasifikasi *Decision Tree*.

```

data.describe()
{"type": "dataframe"}

```

`data.describe()` adalah metode dalam library Pandas Python yang digunakan untuk menghasilkan ringkasan statistik deskriptif dari sebuah DataFrame atau Series. Fungsi ini memberikan gambaran cepat tentang karakteristik data numerik, termasuk jumlah data (count), rata-rata (mean), standar deviasi (std), nilai minimum (min), kuartil (persentil ke-25, ke-50, dan ke-75), serta nilai maksimum (max).

```

# Identify feature columns (all except 'No', 'Longitude', 'Lattitude', and 'b' columns)
feature_columns = [col for col in data.columns if col not in ['No', 'Longitude', 'Lattitude'] and not col.startswith('b')]

# Create the feature DataFrame X
X = data[feature_columns]

# Create the target DataFrame y (all columns starting with 'b')
target_columns = [col for col in data.columns if col.startswith('b')]
y = data[target_columns]

# Display the first few rows of X and y to verify
display(X.head())
display(y.head())

{
  "summary": {
    "name": "display(y)",
    "rows": 5,
    "fields": [
      {
        "column": "N",
        "properties": {
          "dtype": "number",
          "std": null
        }
      }
    ]
  }
}

```

```

0.406657103712698,\n          \"min\": 1.77,\n          \"max\": 2.75,\n          \"num_unique_values\": 5,\n          \"samples\": [\n            2.75,\n            2.05,\n            1.77\n          ],\n          \"semantic_type\": \"\",\n          \"description\": \"\"\n        },\n        {\n          \"column\": \"P\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.018165902124584955,\n            \"min\": 0.12,\n            \"max\": 0.17,\n            \"num_unique_values\": 4,\n            \"samples\": [\n              0.17,\n              0.14,\n              0.15\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"K\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.10321579336516284,\n            \"min\": 0.308,\n            \"max\": 0.568,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              0.568,\n              0.308,\n              0.339\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Ca\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.12557866060760484,\n            \"min\": 0.49,\n            \"max\": 0.76,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              0.76,\n              0.64,\n              0.49\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Mg\",\n          \"properties\": {\n            \"dtype\": \"string\",\n            \"num_unique_values\": 5,\n            \"samples\": [\n              \"0.58\",,\n              \"0.72\",,\n              \"0.6\"\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Fe\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 12.353824104300656,\n            \"min\": 87.01,\n            \"max\": 119.96,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              102.63,\n              87.01,\n              107.37\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Mn\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 64.48439253028596,\n            \"min\": 338.17,\n            \"max\": 493.81,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              493.81,\n              384.33,\n              460.93\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Cu\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 1.8735981426122303,\n            \"min\": 0.65,\n            \"max\": 5.6,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              0.65,\n              1.51,\n              2.82\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"Zn\",\n          \"properties\": {\n            \"dtype\": \"number\",\n            \"std\": 0.8851271095159161,\n            \"min\": 14.47,\n            \"max\": 16.69,\n            \"num_unique_values\": 5,\n            \"samples\": [\n              14.47,\n              15.82,\n              14.75\n            ],\n            \"semantic_type\": \"\",,\n            \"description\": \"\"\n          },\n          \"column\": \"B\"}

```

```

    "number",\n          "std": 6.492336251304302,\n          "min":\n13.6,\n          "max": 29.31,\n          "num_unique_values": 5,\n        ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "Sigma_VV",\n      "properties": {\n          "dtype": "number",\n          "min": 0.14769,\n          "max": 0.22079,\n          "num_unique_values": 5,\n          "samples": [\n              0.18205,\n              0.18926\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "Sigma_VH",\n      "properties": {\n          "dtype": "number",\n          "min": 0.03622,\n          "max": 0.0464,\n          "num_unique_values": 5,\n          "samples": [\n              0.03797,\n              0.03992\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "plia",\n      "properties": {\n          "dtype": "number",\n          "std": 1.3293903292186227,\n          "min": 32.68855,\n          "max": 36.08078,\n          "num_unique_values": 5,\n          "samples": [\n              35.12096,\n              32.68855,\n              35.07724\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "lia",\n      "properties": {\n          "dtype": "number",\n          "std": 1.3373631662828185,\n          "min": 32.69293,\n          "max": 36.08469,\n          "num_unique_values": 5,\n          "samples": [\n              35.14591,\n              32.69293,\n              35.0773\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "iafe",\n      "properties": {\n          "dtype": "number",\n          "std": 0.0022897969342277674,\n          "min": 35.41135,\n          "max": 35.41592,\n          "num_unique_values": 5,\n          "samples": [\n              35.41592,\n              35.41135\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "gamma0_vv",\n      "properties": {\n          "dtype": "number",\n          "min": 0.18138,\n          "max": 0.27116,\n          "num_unique_values": 5,\n          "samples": [\n              0.22359,\n              0.23242\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      },\n      {"column": "gamma0_vh",\n      "properties": {\n          "dtype": "number",\n          "std": 0.0053335569744777265,\n          "min": 0.04448,\n          "max": 0.05699,\n          "num_unique_values": 5,\n          "samples": [\n              0.04664,\n              0.04902\n          ],\n          "semantic_type": "\",\n          "description": \"\"\n      }\n    ]\n},\n  "type": "dataframe"
}

{"summary":{\n  "name": "display(y",\n  "rows": 5,\n  "fields": [\n    {"column": "b12",\n
```

```
"properties": {  
    "b1": {"dtype": "number", "min": 0.0725, "max": 0.1057, "num_unique_values": 5, "samples": [0.0725, 0.0836]},  
    "b2": {"semantic_type": "\\", "description": "", "column": "b11", "properties": {"dtype": "number", "std": 0.01424682420752078, "min": 0.1598, "max": 0.1941, "num_unique_values": 5, "samples": [0.1598, 0.175, 0.1941]}, "semantic_type": "\\", "description": "", "column": "b9", "properties": {"dtype": "number", "std": 0.014761707218340293, "min": 0.4603, "max": 0.4981, "num_unique_values": 5, "samples": [0.4603, 0.4811, 0.4655, 0.4981]}], "semantic_type": "\\", "description": "", "column": "b8a", "properties": {"dtype": "number", "std": 0.020604441268813875, "min": 0.3208, "max": 0.3671, "num_unique_values": 5, "samples": [0.3208, 0.36, 0.3655]}, "semantic_type": "\\", "description": "", "column": "b8", "properties": {"dtype": "number", "std": 0.018191976253282654, "min": 0.2828, "max": 0.3256, "num_unique_values": 5, "samples": [0.2828, 0.3166, 0.3256, 0.2918, 0.3346]}, "semantic_type": "\\", "description": "", "column": "b7", "properties": {"dtype": "number", "std": 0.019308547330133357, "min": 0.2953, "max": 0.3386, "num_unique_values": 5, "samples": [0.2953, 0.3324, 0.3346]}, "semantic_type": "\\", "description": "", "column": "b6", "properties": {"dtype": "number", "std": 0.0027754278949380035, "min": 0.0866, "max": 0.0939, "num_unique_values": 5, "samples": [0.0939, 0.0882, 0.0866]}, "semantic_type": "\\", "description": "", "column": "b5", "properties": {"dtype": "number", "std": 0.011544609131538416, "min": 0.2385, "max": 0.2656, "num_unique_values": 5, "samples": [0.2385, 0.2625, 0.2599]}, "semantic_type": "\\", "description": "", "column": "b4", "properties": {"dtype": "number", "std": 0.01007114690588912, "min": 0.0364, "max": 0.0626, "num_unique_values": 5}}]
```

```

\"samples\": [\n          0.0626,\n          0.0364,\n0.0444\n        ],\n        \"semantic_type\": \"\",\n      \"description\": \"\"\n      },\n      {\n        \"column\":\n        \"b3\",\n        \"properties\": {\n          \"dtype\": \"number\",\n          \"std\": 0.005204325124355701,\n          \"min\": 0.0548,\n          \"max\": 0.0677,\n          \"num_unique_values\": 5,\n          \"samples\": [\n            0.0677,\n            0.0559,\n            0.0622\n          ],\n          \"semantic_type\": \"\",,\n          \"description\": \"\"\n          },\n          {\n            \"column\":\n            \"b2\",\n            \"properties\": {\n              \"dtype\": \"number\",\n              \"std\": 0.004631630382489518,\n              \"min\": 0.0336,\n              \"max\": 0.0445,\n              \"num_unique_values\": 5,\n              \"samples\": [\n                0.0445,\n                0.0336,\n                0.0403\n              ],\n              \"semantic_type\": \"\",,\n              \"description\": \"\"\n              },\n              {\n                \"column\":\n                \"b1\",\n                \"properties\": {\n                  \"dtype\": \"number\",\n                  \"std\": 0.004428092139962762,\n                  \"min\": 0.0361,\n                  \"max\": 0.0465,\n                  \"num_unique_values\": 5,\n                  \"samples\": [\n                    0.0465,\n                    0.0361,\n                    0.0417\n                  ],\n                  \"semantic_type\": \"\",,\n                  \"description\": \"\"\n                  },\n                  {\n                    \"column\":\n                    \"beta0_vv\",\n                    \"properties\": {\n                      \"dtype\": \"number\",\n                      \"std\": 0.04482047155039759,\n                      \"min\": 0.2544,\n                      \"max\": 0.38033,\n                      \"num_unique_values\": 5,\n                      \"samples\": [\n                        0.38033,\n                        0.31359,\n                        0.32604\n                      ],\n                      \"semantic_type\": \"\",,\n                      \"description\": \"\"\n                      },\n                      {\n                        \"column\":\n                        \"beta0_vh\",\n                        \"properties\": {\n                          \"dtype\": \"number\",\n                          \"std\": 0.007485043086048337,\n                          \"min\": 0.06238,\n                          \"max\": 0.07993,\n                          \"num_unique_values\": 5,\n                          \"samples\": [\n                            0.07993,\n                            0.06541,\n                            0.06876\n                          ],\n                          \"semantic_type\": \"\",,\n                          \"description\": \"\"\n                          }\n                        }\n                      }\n                    ],\n                    \"type\":\"dataframe\"}\n
```

Kode di atas adalah langkah penting dalam **persiapan data** untuk pemodelan Machine Learning, di mana data dibagi menjadi fitur input dan variabel target. Pertama, kolom-kolom yang berfungsi sebagai fitur input (X) diidentifikasi dengan **mengecualikan** kolom identifikasi (No), lokasi geografis (**Longitude**, **Lattitude**), dan semua kolom yang dimulai dengan huruf 'b'. Kolom fitur input yang tersisa ini (N, P, K, Ca, Mg, Fe, Mn, Cu, Zn, B, dan semua data radar) kemudian dikelompokkan ke dalam *DataFrameX*. Kedua, kolom-kolom yang dimulai dengan huruf 'b' (yang mewakili data spektral optik satelit seperti b1, b2, hingga b12) diidentifikasi dan dikelompokkan sebagai variabel target (y), menunjukkan bahwa tujuannya adalah memprediksi atau menganalisis nilai-nilai spektral ini berdasarkan fitur unsur hara dan radar. Terakhir, *X.head()* dan *y.head()* ditampilkan untuk **memverifikasi** pemisahan yang benar antara fitur input dan output sebelum melangkah ke pelatihan model.

```

X_train, X_test, y_train, y_test = train_test_split(\n    X, y, test_size=0.2, random_state=7\n)

```

Kode ini adalah langkah penting dalam proses *Machine Learning* yang dikenal sebagai **pemisahan data (data splitting)**. Dengan menggunakan fungsi `train_test_split` dari pustaka Scikit-learn, `DataFrame` fitur input (`X`) dan target output (`y`) **dibagi secara acak** menjadi dua *subset* utama: **data pelatihan (`X_train, y_train`)** dan **data pengujian (`X_test, y_test`)**. Argumen `test_size=0.2` menetapkan bahwa **20%** dari total data akan dialokasikan untuk pengujian, sementara sisanya (80%) digunakan untuk pelatihan model. Penggunaan `random_state=7` memastikan bahwa pemisahan ini **konsisten dan dapat direplikasi** setiap kali kode dijalankan, yang sangat penting untuk perbandingan hasil model yang adil. *Subset* pelatihan (`X_train, y_train`) akan digunakan untuk mengajarkan model *Decision Tree*, sedangkan *subset* pengujian (`X_test, y_test`) akan digunakan untuk mengevaluasi kinerja model pada data yang belum pernah dilihat sebelumnya.

```
# Replace the non-numeric character in 'Mg' and convert to float
X_train['Mg'] = X_train['Mg'].astype(str).str.replace("'", ''),
regex=False)
X_test['Mg'] = X_test['Mg'].astype(str).str.replace("'", ''),
regex=False)

X_train['Mg'] = pd.to_numeric(X_train['Mg'], errors='coerce')
X_test['Mg'] = pd.to_numeric(X_test['Mg'], errors='coerce')

# Fill missing values in 'K' with the mean
mean_k = X_train['K'].mean()
X_train['K'].fillna(mean_k, inplace=True)
X_test['K'].fillna(mean_k, inplace=True)

# Re-train the model
from sklearn.linear_model import LinearRegression # Import
LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Missing values in X\_train after filling:

N	0
P	0
K	0
Ca	0
Mg	0
Fe	0
Mn	0
Cu	0
Zn	0
B	0
Sigma_VV	0
Sigma_VH	0
plia	0
lia	0
iafe	0
gamma0_vv	0

```

gamma0_vh      0
dtype: int64

Missing values in X_test after filling:
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64

LinearRegression()

```

Kode di atas menyelesaikan langkah-langkah pra-pemrosesan data yang diperlukan dan melanjutkan ke inisiasi serta pelatihan model. Tahap pertama berfokus pada perbaikan dua masalah yang teridentifikasi: kolom **Mg** dan **K**. Untuk kolom Mg, karakter non-numerik (tanda kurung tutup )) dihapus, dan kolom tersebut dikonversi paksa menjadi tipe data **numerik** (`float`). Setelah itu, **nilai yang hilang** pada kolom **K** diatasi dengan teknik **imputasi**, di mana nilai-nilai yang hilang pada kedua *subset* (pelatihan dan pengujian) diisi menggunakan **nilai rata-rata (mean)** dari kolom K pada data pelatihan. Setelah data dipersiapkan dan dibersihkan, model *Machine Learning* yaitu **Regresi Linier (LinearRegression)** diimpor dan diinisiasi, dan kemudian model tersebut **dilatih** (`model.fit`) menggunakan data fitur pelatihan (`X_train`) dan data target pelatihan (`y_train`). Langkah ini menandai transisi dari persiapan data ke fase *fitting* model.

```

print(X_train.info())
for col in X_train.columns:
    if X_train[col].dtype == 'object':
        print(f"Column '{col}' has object dtype. Unique values:")
        print(X_train[col].unique())

<class 'pandas.core.frame.DataFrame'>
Index: 475 entries, 142 to 175
Data columns (total 17 columns):
 #   Column     Non-Null Count Dtype
 ---  ---          ---           ---
 0   N           475 non-null   float64

```

```

1 P          475 non-null    float64
2 K          475 non-null    float64
3 Ca         475 non-null    float64
4 Mg         475 non-null    float64
5 Fe         475 non-null    float64
6 Mn         475 non-null    float64
7 Cu         475 non-null    float64
8 Zn         475 non-null    float64
9 B          475 non-null    float64
10 Sigma_VV  475 non-null    float64
11 Sigma_VH  475 non-null    float64
12 plia      475 non-null    float64
13 lia       475 non-null    float64
14 iafe      475 non-null    float64
15 gamma0_vv 475 non-null    float64
16 gamma0_vh 475 non-null    float64
dtypes: float64(17)
memory usage: 66.8 KB
None

```

Kode ini bertujuan untuk **memeriksa status akhir** dari *DataFrame* fitur pelatihan (*X\_train*) setelah dilakukan pembersihan dan penanganan nilai yang hilang pada langkah sebelumnya. Pertama, `print(X_train.info())` menampilkan ringkasan struktural data, termasuk memastikan semua kolom sudah memiliki **tipe data numerik** (float64 atau int64) dan **tidak ada nilai yang hilang** (semua Non-Null Count sama dengan total baris). Kedua, perulangan (`for loop`) secara spesifik melakukan **verifikasi tambahan** dengan mencari kolom yang masih bertipe `object` (non-numerik). Dengan kode sebelumnya yang sudah mengkonversi Mg dan mengisi K, hasil yang diharapkan dari langkah ini adalah konfirmasi bahwa **semua kolom kini telah siap** untuk proses pelatihan model regresi linier, dengan semua fitur input bertipe numerik dan tidak memiliki data yang hilang.

```

print("Nilai yang Hilang di X_train setelah cleaning:")
print(X_train.isnull().sum())

print("\nNilai yang Hilang di X_test setelah cleaning:")
print(X_test.isnull().sum())

# Re-fill missing values in 'K' using the updated inplace method
mean_k_train = X_train['K'].mean()
X_train['K'] = X_train['K'].fillna(mean_k_train)

mean_k_test = X_test['K'].mean()
X_test['K'] = X_test['K'].fillna(mean_k_test)

# Verify missing values are handled
print("\nNilai yang Hilang di X_Train setelah mengisi ulang:")
print(X_train.isnull().sum())

print("\nNilai yang Hilang di X_test setelah mengisi ulang:")

```

```
print(X_test.isnull().sum())

# Re-train the model
model = LinearRegression()
model.fit(X_train, y_train)

Nilai yang Hilang di X_train setelah cleaning:
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64

Nilai yang Hilang di X_test setelah cleaning:
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64

Nilai yang Hilang di X_Train setelah mengisi ulang:
N          0
P          0
```

```
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64
```

Nilai yang Hilang di  $X_{test}$  setelah mengisi ulang:

```
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64
```

`LinearRegression()`

Kode ini menjalankan serangkaian **verifikasi dan pembersihan data ulang** yang kritis sebelum melatih model Regresi Linier. Awalnya, kode memeriksa **jumlah nilai yang hilang** (`isnull().sum()`) pada *subset* pelatihan ( $X_{train}$ ) dan pengujian ( $X_{test}$ ) setelah proses pembersihan sebelumnya. Kemudian, terdapat langkah **imputasi ulang** yang mengisi nilai yang hilang pada kolom **K** dengan **nilai rata-rata (mean)** yang dihitung secara terpisah untuk  $X_{train}$  dan  $X_{test}$ , dan ini diverifikasi lagi untuk memastikan semua nilai hilang telah hilang. Setelah data dipastikan bersih, model **Regresi Linier** diinisiasi kembali dan **dilatih** menggunakan data fitur ( $X_{train}$ ) dan target ( $y_{train}$ ) yang sudah *clean*. Terakhir, kode melakukan pemeriksaan akhir menggunakan  $X_{train}.info()$  dan mencari sisa kolom **object** untuk mengkonfirmasi

bahwa **semua data fitur input sudah dalam format numerik** dan siap sepenuhnya untuk pemodelan.

```
# Fill missing values in 'Mg' with the mean
mean_mg_train = X_train['Mg'].mean()
X_train['Mg'] = X_train['Mg'].fillna(mean_mg_train)

mean_mg_test = X_test['Mg'].mean()
X_test['Mg'] = X_test['Mg'].fillna(mean_mg_test)

# Verify missing values are handled in Mg
print("\nMissing values in X_train after filling Mg:")
print(X_train.isnull().sum())

print("\nMissing values in X_test after filling Mg:")
print(X_test.isnull().sum())

# Re-train the model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
Missing values in X_train after filling Mg:
```

```
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
Cu         0
Zn         0
B          0
Sigma_VV   0
Sigma_VH   0
plia       0
lia        0
iafe       0
gamma0_vv  0
gamma0_vh  0
dtype: int64
```

```
Missing values in X_test after filling Mg:
```

```
N          0
P          0
K          0
Ca         0
Mg         0
Fe         0
Mn         0
```

```
Cu      0
Zn      0
B       0
Sigma_VV 0
Sigma_VH 0
plia    0
lia     0
iafe    0
gamma0_vv 0
gamma0_vh 0
dtype: int64
```

```
LinearRegression()
```

Kode ini melanjutkan tahap pra-pemrosesan data dengan secara spesifik menangani nilai-nilai yang masih hilang di kolom **Mg** (Magnesium). Proses ini menggunakan teknik **imputasi** dengan mengisi setiap nilai yang hilang pada Mg baik di *subset* pelatihan (*X\_train*) maupun pengujian (*X\_test*) dengan **nilai rata-rata (mean)** dari kolom Mg masing-masing. Setelah imputasi selesai, kode segera melakukan **verifikasi** (*X\_train.isnull().sum()* dan *X\_test.isnull().sum()*) untuk memastikan bahwa tidak ada lagi nilai yang hilang pada kolom Mg di kedua *subset*. Setelah data dipastikan **bersih dan lengkap**, model **Regresi Linier** diinisiasi ulang, dan model tersebut kemudian **dilatih** (*model.fit*) menggunakan fitur pelatihan yang sudah bersih (*X\_train*) dan target pelatihan (*y\_train*), menandakan model siap untuk dievaluasi.

```
from sklearn.metrics import mean_squared_error, r2_score

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R2): {r2}")

Mean Squared Error (MSE): 0.3552381004200159
R-squared (R2): 0.46035633350431915
```

Kode ini adalah langkah penting dalam **evaluasi kinerja** model Regresi Linier yang telah dilatih. Pertama, model digunakan untuk **memprediksi** nilai target (*y\_pred*) berdasarkan data fitur pengujian yang belum pernah dilihat sebelumnya (*X\_test*). Kemudian, dua metrik evaluasi utama dihitung: **Mean Squared Error (MSE)**, yang mengukur rata-rata kuadrat dari kesalahan (perbedaan antara nilai aktual dan prediksi), memberikan indikasi seberapa jauh rata-rata prediksi dari nilai sebenarnya; dan **R-squared ( $R^2$ ) Score**, yang mengukur proporsi varians dalam variabel dependen yang dapat dijelaskan dari variabel independen, dengan nilai mendekati 1 menunjukkan kecocokan model yang sangat baik. Kedua hasil metrik ini kemudian **dicetak** untuk memberikan ringkasan kuantitatif tentang seberapa akurat model Regresi Linier dalam memprediksi data spektral satelit berdasarkan fitur unsur hara dan radar.

```

# Get the coefficients and intercept from the trained model
coefficients = model.coef_
intercept = model.intercept_

# Get feature names from X_train
feature_names = X_train.columns

# Create a DataFrame to display coefficients - Transpose the
# coefficients array
coefficients_df = pd.DataFrame(coefficients.T,
columns=y_train.columns, index=feature_names)

# Print the intercept
print("Intersep:")
print(intercept)

# Print the coefficients DataFrame
print("\nKoefisien:")
display(coefficients_df)

# Interpretation of coefficients
print("\nInterpretasi:")
print(f"Nilai intersep mewakili nilai prediksi untuk setiap variabel 'b' ketika semua variabel fitur (N, P, K, dll.) adalah nol.")
print("\nInterpretasi koefisien terpilih:")
for target_col in coefficients_df.columns:
    print(f"\nUntuk variabel target '{target_col}':")
    # Select a few coefficients for interpretation
    selected_features =
coefficients_df[target_col].abs().nlargest(3).index.tolist()
    for feature in selected_features:
        coeff_value = coefficients_df.loc[feature, target_col]
        print(f"- Peningkatan satu unit pada '{feature}' dikaitkan dengan perubahan sebesar {coeff_value:.4f} pada prediksi '{target_col}', dengan asumsi semua fitur lain dianggap konstan.")

```

Intersep:

```
[2.39798405e-01 1.06871731e-01 3.72238112e+01 3.57867761e+01
3.56518469e+01 2.40284466e-01 2.19440955e-01 5.01782575e-01
3.24110750e-01 3.73418542e-01 4.53879011e-01 5.07578229e-01
2.85974790e-02 3.05932035e-03]
```

Koefisien:

```
{"summary": {\n    "name": "coefficients_df",\n    "rows": 17,\n    "fields": [\n        {"\n            "column": "b12",\n            "properties": {\n                "dtype": "number",\n                "std": 0.14188468991278494,\n                "max": 0.2621801131279404,\n                "samples": [\n                    -0.020257803115113226,\n
```

```
0.057832764072044136,\n          0.0001624415638839465\n      ],\n  \"semantic_type\": \"\",\\n      \"description\": \"\"\n    },\\n    {\n      \"column\": \"b11\",\\n      \"properties\": {\n        \"dtype\": \"number\",\\n        \"std\": 0.07566216541915832,\n        \"min\": -0.2300779172782112,\n        \"max\": 0.04920298248264011,\n        \"num_unique_values\": 17,\n        \"samples\": [\n          -0.019091262144604847,\n          -0.059443032270585935,\n          0.00012567942678616128\n        ],\n        \"semantic_type\": \"\",\\n        \"description\": \"\"\n      },\\n      {\n        \"column\": \"b9\",\\n        \"properties\": {\n          \"dtype\": \"number\",\\n          \"std\": 7.0625725834395805,\n          \"min\": -1.6433476066369532,\n          \"max\": 22.530915766387185,\n          \"num_unique_values\": 17,\n          \"samples\": [\n            -1.6433476066369532,\n            10.521445736397201,\n            -0.008654094534965448\n          ],\n          \"semantic_type\": \"\",\\n          \"description\": \"\"\n        },\\n        {\n          \"column\": \"b8a\",\\n          \"properties\": {\n            \"dtype\": \"number\",\\n            \"std\": 8.296989945845937,\n            \"min\": -1.8699386427301727,\n            \"max\": 26.939234179914408,\n            \"num_unique_values\": 17,\n            \"samples\": [\n              -1.8699386427301727,\n              10.507828662647,\n              -0.010784974632020874\n            ],\n            \"semantic_type\": \"\",\\n            \"description\": \"\"\n          },\\n          {\n            \"column\": \"b8\",\\n            \"properties\": {\n              \"dtype\": \"number\",\\n              \"std\": 8.674459299548422,\n              \"min\": -2.669204620292374,\n              \"max\": 24.793644250275886,\n              \"num_unique_values\": 17,\n              \"samples\": [\n                -2.11060068157761,\n                12.829603686077958,\n                -0.010154319628786894\n              ],\n              \"semantic_type\": \"\",\\n              \"description\": \"\"\n            },\\n            {\n              \"column\": \"b7\",\\n              \"properties\": {\n                \"dtype\": \"number\",\\n                \"std\": 0.17762591723558258,\n                \"min\": -0.5918193574161796,\n                \"max\": 0.29614289351190537,\n                \"num_unique_values\": 17,\n                \"samples\": [\n                  0.003670503795802375,\n                  -0.139648040676523,\n                  0.00020598868797326\n                ],\n                \"semantic_type\": \"\",\\n                \"description\": \"\"\n              },\\n              {\n                \"column\": \"b6\",\\n                \"properties\": {\n                  \"dtype\": \"number\",\\n                  \"std\": 0.1951186957434077,\n                  \"min\": -0.2934846767289381,\n                  \"max\": 0.5973694857011361,\n                  \"num_unique_values\": 17,\n                  \"samples\": [\n                    -0.05748381808907633,\n                    0.42837788033790136,\n                    0.0002628325756189769\n                  ],\n                  \"semantic_type\": \"\",\\n                  \"description\": \"\"\n                },\\n                {\n                  \"column\": \"b5\",\\n                  \"properties\": {\n                    \"dtype\": \"number\",\\n                    \"std\": 0.38974048803752004,\n                    \"min\": -1.1759341192899742,\n                    \"max\": 0.4687292722548779,\n                    \"num_unique_values\": 17,\n                    \"samples\": [\n                      0.027649977085066443,\n                      -0.3770008531139194,\n                      0.0004474763122182123\n                    ],\n                    \"semantic_type\": \"\",\\n                    \"description\": \"\"\n                  },\\n                  {\n                    \"column\": \"b4\",\\n                    \"properties\": {\n                      \"dtype\": \"number\",\\n                      \"std\": 0.0001624415638839465,\n                      \"min\": -0.008654094534965448,\n                      \"max\": 0.0002628325756189769,\n                      \"num_unique_values\": 17,\n                      \"samples\": [\n                        -0.0001624415638839465,\n                        0.0002628325756189769,\n                        -0.008654094534965448\n                      ],\n                      \"semantic_type\": \"\",\\n                      \"description\": \"\"\n                    }\n                  }\n                }\n              }\n            }\n          }\n        }\n      }\n    }\n  }\n}\n
```

```
\"std\": 0.18400796526867108,\n          \"min\": -0.6907084893563987,\n\"max\": 0.07366544908033248,\n          \"num_unique_values\": 17,\n\"samples\": [\n            -0.01774214013739778,\n            0.0004376731730674932\n          ],\n        \"semantic_type\": \"\",\n          \"description\": \"\"\n        },\n      {\n        \"column\": \"b3\",\n          \"properties\": {\n            \"dtype\": \"number\",\n              \"std\": 0.19932054132019936,\n            \"min\": -0.48817357902094316,\n              \"max\": 0.37700028666192925,\n            \"num_unique_values\": 17,\n            \"samples\": [\n              -0.020209340539054887,\n                0.061347865639627006,\n                0.0004932263987555375\n              ],\n              \"semantic_type\": \"\",\n                \"description\": \"\"\n              },\n            {\n              \"column\": \"b2\",\n                \"properties\": {\n                  \"dtype\": \"number\",\n                    \"std\": 0.18901917674367008,\n                    \"min\": -0.49872416362354244,\n                    \"max\": 0.330125160962027,\n                    \"num_unique_values\": 17,\n                    \"samples\": [\n                      -0.030521007430479998,\n                      0.1703264078133773,\n                        0.0005693937651161129\n                      ],\n                      \"semantic_type\": \"\",,\n                        \"description\": \"\"\n                      },\n                    {\n                      \"column\": \"b1\",\n                        \"properties\": {\n                          \"dtype\": \"number\",\n                            \"std\": 0.21502462069034667,\n                            \"min\": -0.7006498680508486,\n                            \"max\": 0.3705858936258019,\n                            \"num_unique_values\": 17,\n                            \"samples\": [\n                              -0.02671494460550312,\n                                0.15242169436391534,\n                                0.0004574843117468084\n                              ],\n                              \"semantic_type\": \"\",,\n                                \"description\": \"\"\n                              },\n                            {\n                              \"column\": \"beta0_vv\",\n                                \"properties\": {\n                                  \"dtype\": \"number\",\n                                    \"std\": 0.34518558471749466,\n                                    \"min\": -0.3001717871700705,\n                                    \"max\": 1.3269935266008719,\n                                    \"num_unique_values\": 17,\n                                    \"samples\": [\n                                      -0.02450583355948737,\n                                        0.02450583355948737,\n                                          0.0001308462431197341\n                                        ],\n                                        \"semantic_type\": \"\",,\n                                          \"description\": \"\"\n                                        },\n                                          {\n                                            \"column\": \"beta0_vh\",\n                                              \"properties\": {\n                                                \"dtype\": \"number\",\n                                                  \"std\": 0.3039289500939507,\n                                                  \"min\": -0.09347453906768599,\n                                                  \"max\": 1.2412447935746151,\n                                                  \"num_unique_values\": 17,\n                                                  \"samples\": [\n                                                    -0.0051665704736395075,\n                                                      2.850228645043846e-05\n                                                    ],\n                                                    \"semantic_type\": \"\",,\n                                                      \"description\": \"\"\n                                                    }\n                                                },\n                                                \"type\": \"dataframe\", \"variable_name\": \"coefficients_df\"}\n                                              ]\n                                            ]\n                                          ]\n                                        ]\n                                      ]\n                                    ]\n                                  ]\n                                ]\n                              ]\n                            ]\n                          ]\n                        ]\n                      ]\n                    ]\n                  ]\n                ]\n              ]\n            ]\n          ]\n        ]\n      ]\n    ]\n  ]\n}
```

## Interpretasi:

Nilai intersep mewakili nilai prediksi untuk setiap variabel 'b' ketika semua variabel fitur (N, P, K, dll.) adalah nol.

Interpretasi koefisien terpilih:

Untuk variabel target 'b12':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan

sebesar -0.3964 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.2622 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.2340 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b11':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2301 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.1760 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.1036 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b9':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 22.5309 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 17.3803 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 10.5214 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b8a':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 26.9392 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 20.7853 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 10.5078 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b8':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 24.7936 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 24.6061 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar

12.8296 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b7':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.5918 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.2961 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.2234 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b6':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 0.5974 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 0.4284 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.2935 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b5':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -1.1759 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.9285 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.4687 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b4':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.6907 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.3131 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 0.0737 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b3':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4882 pada prediksi 'b3', dengan asumsi semua fitur lain

diangap konstan.

- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.4433 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3770 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b2':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4987 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.3748 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3301 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b1':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.7006 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.3706 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2295 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'beta0\_vv':

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 1.3270 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.3002 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2442 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'beta0\_vh':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 1.2412 pada prediksi 'beta0\_vh', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.0935 pada prediksi 'beta0\_vh', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan

```
sebesar -0.0372 pada prediksi 'beta0_vh', dengan asumsi semua fitur lain dianggap konstan.
```

Kode ini bertujuan untuk **menganalisis hasil dari model Regresi Linier** yang telah dilatih, yang memprediksi nilai data spektral satelit (kolom 'b') berdasarkan fitur unsur hara dan radar. Pertama, skrip **mengambil nilai Intersep dan Koefisien** dari model. Nilai **Intersep** adalah perkiraan nilai prediksi untuk setiap kolom target 'b' ketika semua fitur input bernilai nol. **Koefisien** dikelompokkan dalam *DataFrame* untuk menunjukkan dampak setiap fitur input terhadap setiap variabel target. Interpretasi koefisien kemudian diberikan, yang menjelaskan bahwa nilai koefisien menunjukkan **perubahan yang diharapkan** pada variabel target untuk setiap **peningkatan satu unit** pada fitur input terkait, dengan asumsi fitur lainnya tetap konstan. Terakhir, skrip mengulangi langkah evaluasi dengan menghitung **Mean Squared Error (MSE)** dan **R-squared ( $R^2$ )** pada data pengujian untuk mengukur **akurasi dan kecocokan** model secara keseluruhan.

```
coefficients = model.coef_
intercept = model.intercept_

feature_names = X_train.columns

coefficients_df = pd.DataFrame(coefficients.T,
columns=y_train.columns, index=feature_names)

print("Intersep:")
print(intercept)

print("\nKoefisien:")
display(coefficients_df)

print("\nInterpretasi:")
print(f"Nilai intersep mewakili nilai prediksi untuk setiap variabel 'b' ketika semua variabel fitur (N, P, K, dll.) adalah nol.")
print("\nInterpretasi koefisien terpilih:")
for target_col in coefficients_df.columns:
    print(f"\nUntuk variabel target '{target_col}':")
    selected_features =
coefficients_df[target_col].abs().nlargest(3).index.tolist()
    for feature in selected_features:
        coeff_value = coefficients_df.loc[feature, target_col]
        print(f"- Peningkatan satu unit pada '{feature}' dikaitkan dengan perubahan sebesar {coeff_value:.4f} pada prediksi '{target_col}', dengan asumsi semua fitur lain dianggap konstan.")

Intersep:
[2.39798405e-01 1.06871731e-01 3.72238112e+01 3.57867761e+01
 3.56518469e+01 2.40284466e-01 2.19440955e-01 5.01782575e-01
 3.24110750e-01 3.73418542e-01 4.53879011e-01 5.07578229e-01
 2.85974790e-02 3.05932035e-03]
```

Koeffisien:

```
{"summary": {"name": "coefficients_df", "rows": 17, "fields": [{"column": "b12", "properties": {"dtype": "number", "std": 0.14188468991278494, "min": -0.3963780371997336, "max": 0.2621801131279404, "num_unique_values": 17, "samples": [-0.020257803115113226, 0.057832764072044136, 0.0001624415638839465], "semantic_type": "\", "description": "\n"}, {"column": "b11", "properties": {"dtype": "number", "std": 0.07566216541915832, "min": -0.2300779172782112, "max": 0.04920298248264011, "num_unique_values": 17, "samples": [-0.019091262144604847, 0.00012567942678616128, 10.521445736397201, 0.008654094534965448], "semantic_type": "\", "description": "\n"}, {"column": "b9", "properties": {"dtype": "number", "std": 7.0625725834395805, "min": -1.6433476066369532, "max": 22.530915766387185, "num_unique_values": 17, "samples": [-1.6433476066369532, 1.8699386427301727, 10.507828662647, 0.010784974632020874, 1.8699386427301727, 12.829603686077958, 0.003670503795802375, 0.00020598868797326], "semantic_type": "\", "description": "\n"}, {"column": "b8a", "properties": {"dtype": "number", "std": 8.296989945845937, "min": -1.8699386427301727, "max": 26.939234179914408, "num_unique_values": 17, "samples": [10.507828662647, -2.11060068157761, -0.010154319628786894, 0.0002628325756189769], "semantic_type": "\", "description": "\n"}, {"column": "b7", "properties": {"dtype": "number", "std": 0.17762591723558258, "min": -0.5918193574161796, "max": 0.29614289351190537, "num_unique_values": 17, "samples": [-0.139648040676523, 0.0002628325756189769], "semantic_type": "\", "description": "\n"}, {"column": "b6", "properties": {"dtype": "number", "std": 0.1951186957434077, "min": -0.2934846767289381, "max": 0.5973694857011361, "num_unique_values": 17, "samples": [-0.05748381808907633, 0.42837788033790136], "semantic_type": "\", "description": "\n"}]}
```

```
n },\n      {\n        \"column\": \"b5\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.38974048803752004, \n          \"min\": -1.1759341192899742, \n          \"max\": 0.4687292722548779, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            0.027649977085066443, \n            -0.3770008531139194, \n            0.0004474763122182123\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"b4\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.18400796526867108, \n          \"min\": -0.6907084893563987, \n          \"max\": 0.07366544908033248, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            -0.01774214013739778, \n            0.07366544908033248, \n            0.0004376731730674932\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"b3\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.19932054132019936, \n          \"min\": -0.48817357902094316, \n          \"max\": 0.37700028666192925, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            -0.020209340539054887, \n            0.061347865639627006, \n            0.0004932263987555375\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"b2\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.18901917674367008, \n          \"min\": -0.49872416362354244, \n          \"max\": 0.330125160962027, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            -0.030521007430479998, \n            0.1703264078133773, \n            0.0005693937651161129\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"b1\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.21502462069034667, \n          \"min\": -0.7006498680508486, \n          \"max\": 0.3705858936258019, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            -0.02671494460550312, \n            0.15242169436391534, \n            0.0004574843117468084\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"beta0_vv\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.34518558471749466, \n          \"min\": -0.3001717871700705, \n          \"max\": 1.3269935266008719, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            0.02450583355948737, \n            -0.14836336213803097, \n            0.0001308462431197341\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }, \n      {\n        \"column\": \"beta0_vh\", \n        \"properties\": {\n          \"dtype\": \"number\", \n          \"std\": 0.3039289500939507, \n          \"min\": -0.09347453906768599, \n          \"max\": 1.2412447935746151, \n          \"num_unique_values\": 17, \n          \"samples\": [\n            0.0051665704736395075, \n            -0.022675727373445757, \n            2.850228645043846e-05\n          ], \n          \"semantic_type\": \"\", \n          \"description\": \"\"\n        }\n      }\n    }, \n    \"type\": \"dataframe\", \n    \"variable_name\": \"coefficients_df\"}
```

Interpretasi:

Nilai intersep mewakili nilai prediksi untuk setiap variabel 'b' ketika semua variabel fitur (N, P, K, dll.) adalah nol.

Interpretasi koefisien terpilih:

Untuk variabel target 'b12':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.3964 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.2622 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.2340 pada prediksi 'b12', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b11':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2301 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.1760 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.1036 pada prediksi 'b11', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b9':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 22.5309 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 17.3803 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 10.5214 pada prediksi 'b9', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b8a':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 26.9392 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 20.7853 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 10.5078 pada prediksi 'b8a', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b8':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar 24.7936 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 24.6061 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 12.8296 pada prediksi 'b8', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b7':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.5918 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.2961 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar -0.2234 pada prediksi 'b7', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b6':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 0.5974 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 0.4284 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.2935 pada prediksi 'b6', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b5':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -1.1759 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.9285 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.4687 pada prediksi 'b5', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b4':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.6907 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan

sebesar -0.3131 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 0.0737 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b3':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4882 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.4433 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3770 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b2':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4987 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.3748 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3301 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b1':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.7006 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.3706 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2295 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'beta0\_vv':

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 1.3270 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.3002 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2442 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.

```

Untuk variabel target 'beta0_vh':
- Peningkatan satu unit pada 'gamma0_vh' dikaitkan dengan perubahan sebesar 1.2412 pada prediksi 'beta0_vh', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma_VH' dikaitkan dengan perubahan sebesar -0.0935 pada prediksi 'beta0_vh', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma_VV' dikaitkan dengan perubahan sebesar -0.0372 pada prediksi 'beta0_vh', dengan asumsi semua fitur lain dianggap konstan.

```

Kode ini berfungsi untuk **mengekstraksi dan menginterpretasikan parameter yang dipelajari** oleh model Regresi Linier, yang merupakan kunci untuk memahami hubungan antara fitur input dan variabel target. Pertama, nilai **Intersep** (titik potong) dan **Koefisien** (bobot) dari model diambil. Intersep menunjukkan nilai yang diprediksi untuk semua variabel target spektral ('b') ketika semua fitur input unsur hara dan radar bernilai nol. Koefisien, yang disajikan dalam *DataFrame*, adalah metrik utama: setiap nilai koefisien menunjukkan **magnitudo dan arah pengaruh** fitur input terkait terhadap setiap variabel target. Interpretasi disajikan dengan menyoroti **tiga fitur dengan pengaruh terbesar** (koefisien absolut terbesar) untuk setiap variabel target 'b', menjelaskan bahwa **peningkatan satu unit pada fitur tersebut akan menyebabkan perubahan sebesar nilai koefisien** pada variabel target yang diprediksi, dengan asumsi semua fitur lain dipertahankan konstan (*ceteris paribus*). Hal ini memberikan wawasan tentang fitur unsur hara dan radar mana yang paling dominan dalam memengaruhi nilai spektral satelit.

```

import matplotlib.pyplot as plt

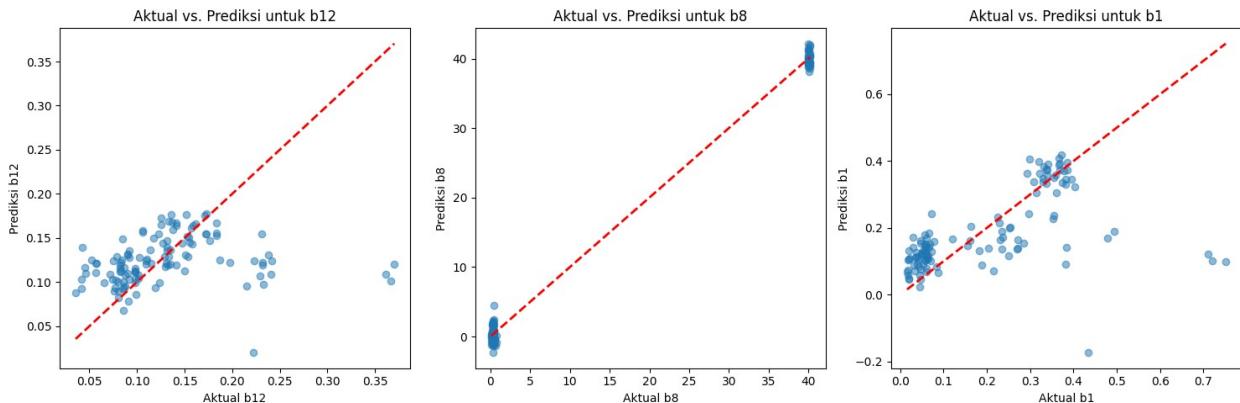
# Pilih beberapa variabel target untuk divisualisasikan
selected_targets = ['b12', 'b8', 'b1'] # Anda bisa mengganti ini dengan variabel target lain yang menarik

plt.figure(figsize=(15, 5))

for i, target_col in enumerate(selected_targets):
    plt.subplot(1, len(selected_targets), i + 1)
    plt.scatter(y_test[target_col], y_pred[:, y_test.columns.get_loc(target_col)], alpha=0.5)
    plt.plot([y_test[target_col].min(), y_test[target_col].max()],
             [y_test[target_col].min(), y_test[target_col].max()],
             'r--', lw=2)
    plt.xlabel(f"Aktual {target_col}")
    plt.ylabel(f"Prediksi {target_col}")
    plt.title(f"Aktual vs. Prediksi untuk {target_col}")

plt.tight_layout()
plt.show()

```



Kode ini bertujuan untuk **memvisualisasikan kinerja** model Regresi Linier dengan membandingkan nilai aktual dan nilai prediksi dari variabel target terpilih menggunakan **Scatter Plot**. Skrip memilih tiga variabel target, yaitu **b12**, **b8**, dan **b1**, lalu membuat *figure* dengan tiga *subplot* yang berdampingan. Pada setiap *subplot*, titik-titik data diplot di mana sumbu X mewakili nilai aktual dari data pengujian (*y\_test*) dan sumbu Y mewakili nilai yang diprediksi oleh model (*y\_pred*). Selain itu, sebuah **garis diagonal putus-putus merah** ( $y=x$ ) diletakkan sebagai referensi ideal; titik-titik yang jatuh **dekat dengan garis ini** menunjukkan bahwa model memprediksi dengan akurat. Visualisasi ini sangat penting untuk menilai secara grafis **berapa baik model menyesuaikan** data dan mengidentifikasi potensi *bias* atau anomali dalam prediksi.

```
print("\nInterpretasi koefisien terpilih (lanjutan):")

target_cols_remaining = [col for col in coefficients_df.columns if col not in ['b12', 'b11', 'b9', 'b8a', 'b8', 'b7', 'b6', 'b5']]

for target_col in target_cols_remaining:
    print(f"\nUntuk variabel target '{target_col}':")
    selected_features =
coefficients_df[target_col].abs().nlargest(3).index.tolist()
    for feature in selected_features:
        coeff_value = coefficients_df.loc[feature, target_col]
        print(f"- Peningkatan satu unit pada '{feature}' dikaitkan dengan perubahan sebesar {coeff_value:.4f} pada prediksi '{target_col}', dengan asumsi semua fitur lain dianggap konstan.")
```

Interpretasi koefisien terpilih (lanjutan):

Untuk variabel target 'b4':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.6907 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.3131 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'P' dikaitkan dengan perubahan sebesar 0.0737 pada prediksi 'b4', dengan asumsi semua fitur lain dianggap

konstan.

Untuk variabel target 'b3':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4882 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.4433 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3770 pada prediksi 'b3', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b2':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.4987 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.3748 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 0.3301 pada prediksi 'b2', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'b1':

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.7006 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar 0.3706 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2295 pada prediksi 'b1', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'beta0\_vv':

- Peningkatan satu unit pada 'gamma0\_vv' dikaitkan dengan perubahan sebesar 1.3270 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.3002 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar -0.2442 pada prediksi 'beta0\_vv', dengan asumsi semua fitur lain dianggap konstan.

Untuk variabel target 'beta0\_vh':

- Peningkatan satu unit pada 'gamma0\_vh' dikaitkan dengan perubahan sebesar 1.2412 pada prediksi 'beta0\_vh', dengan asumsi semua fitur lain dianggap konstan.

- Peningkatan satu unit pada 'Sigma\_VH' dikaitkan dengan perubahan sebesar -0.0935 pada prediksi 'beta0\_vh', dengan asumsi semua fitur lain dianggap konstan.
- Peningkatan satu unit pada 'Sigma\_VV' dikaitkan dengan perubahan sebesar -0.0372 pada prediksi 'beta0\_vh', dengan asumsi semua fitur lain dianggap konstan.

Kode ini melanjutkan langkah **interpretasi model Regresi Linier** yang sebelumnya telah dilakukan, dengan memfokuskan analisis pada variabel target spektral yang **belum dianalisis** (kolom 'b' yang tersisa, seperti b4, b3, b2, dan b1). Skrip ini mengiterasi melalui kolom target yang tersisa, dan untuk setiap kolom target, ia mengidentifikasi **tiga fitur input** (dari unsur hara dan data radar) yang memiliki **koefisien absolut terbesar**—yang menunjukkan pengaruh paling dominan terhadap target tersebut. Hasilnya dicetak sebagai serangkaian pernyataan interpretatif, di mana setiap pernyataan menjelaskan bahwa **peningkatan satu unit** pada fitur input yang terpilih akan menghasilkan **perubahan sebesar nilai koefisien** pada nilai prediksi variabel target spektral yang spesifik, dengan mengasumsikan faktor-faktor lain tetap konstan. Tujuannya adalah untuk mengungkap fitur-fitur yang paling berpengaruh dalam memprediksi setiap pita spektral yang belum dianalisis.

```
y_pred_test = model.predict(X_test)

hasil = pd.DataFrame(X_test.copy())

for i, col in enumerate(y_test.columns):
    hasil[f"Aktual {col}"] = y_test[col].to_numpy()
    hasil[f"Prediksi {col}"] = y_pred_test[:, i]
    epsilon = 1e-8
    hasil[f"Percentase Error {col}"] = (
        ((hasil[f"Prediksi {col}"] - hasil[f"Aktual {col}"]).abs() /
        (hasil[f"Aktual {col}"].abs() + epsilon)) * 100
    )

display(hasil)

{"type": "dataframe", "variable_name": "hasil"}
```

Kode ini bertujuan untuk **menganalisis hasil prediksi model Regresi Linier** pada data pengujian secara rinci, khususnya dengan menghitung **persentase kesalahan** untuk setiap variabel target spektral. Pertama, model digunakan untuk menghasilkan prediksi pada data fitur pengujian (`X_test`), disimpan dalam `y_pred_test`. Selanjutnya, sebuah `DataFrame` baru bernama `hasil` dibuat, yang menyalin fitur-fitur dari `X_test`. Kemudian, melalui sebuah perulangan, kolom-kolom baru ditambahkan ke `DataFrame hasil` untuk setiap variabel target ('b'), yang berisi: **Nilai Aktual** (`y_test`), **Nilai Prediksi** (`y_pred_test`), dan yang paling penting, **Persentase Error**. Persentase error ini dihitung sebagai nilai absolut dari perbedaan antara prediksi dan aktual, dibagi dengan nilai aktual (ditambah  $\epsilon$  kecil untuk menghindari pembagian dengan nol), lalu dikalikan 100. `DataFrame hasil` akhir kemudian ditampilkan, memungkinkan analisis baris demi baris mengenai seberapa akurat model memprediksi setiap nilai spektral dan persentase kesalahannya.

```

import statsmodels.api as sm
from sklearn.metrics import mean_squared_error, r2_score

# Add a constant to the feature variables (intercept)
X_train_sm = sm.add_constant(X_train)
X_test_sm = sm.add_constant(X_test)

# Fit the OLS model for each target variable and print the summary
for target_col in y_train.columns:
    print(f"\n-----")
    print(f"OLS Regression Results for target variable: {target_col}")
    print(f"-----")
    model_sm = sm.OLS(y_train[target_col], X_train_sm).fit()
    print(model_sm.summary())

# Evaluate the model using MSE and R-squared
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"\n-----")
print("Model Evaluation Metrics:")
print(f"-----")
print(f"Mean Squared Error (MSE): {mse}")
print(f"R-squared (R2): {r2}")

```

-----

OLS Regression Results for target variable: b12

-----

OLS Regression Results

-----

=====

Dep. Variable:	b12	R-squared:	
0.188			
Model:	OLS	Adj. R-squared:	
0.158			
Method:	Least Squares	F-statistic:	
6.239			
Date:	Sat, 08 Nov 2025	Prob (F-statistic):	
3.31e-13			
Time:	08:41:35	Log-Likelihood:	
726.41			
No. Observations:	475	AIC:	
-1417.			
Df Residuals:	457	BIC:	
-1342.			

Df Model:

17

Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025
0.975]					
const	0.2398	0.056	4.302	0.000	0.130
0.349	-0.0203	0.012	-1.674	0.095	-0.044
N	0.004				
P	0.0578	0.176	0.328	0.743	-0.289
0.405					
K	0.0353	0.015	2.395	0.017	0.006
0.064					
Ca	-0.0012	0.008	-0.161	0.872	-0.016
0.014					
Mg	0.0079	0.018	0.436	0.663	-0.028
0.043					
Fe	0.0002	4.73e-05	3.435	0.001	6.95e-05
0.000					
Mn	-5.76e-05	1.29e-05	-4.460	0.000	-8.3e-05
3.22e-05					
Cu	-0.0006	0.002	-0.338	0.735	-0.004
0.003					
Zn	0.0006	0.000	1.537	0.125	-0.000
0.001					
B	-0.0010	0.000	-2.270	0.024	-0.002
-0.000					
Sigma_VV	0.2622	0.257	1.020	0.308	-0.243
0.767					
Sigma_VH	-0.3964	0.260	-1.523	0.129	-0.908
0.115					
plia	0.0208	0.022	0.967	0.334	-0.021
0.063					
lia	-0.0213	0.021	-0.992	0.322	-0.063
0.021					
iafe	-0.0013	0.002	-0.735	0.463	-0.005
0.002					
gamma0_vv	-0.2340	0.213	-1.098	0.273	-0.653
0.185					
gamma0_vh	-0.2265	0.363	-0.625	0.533	-0.939
0.486					

Omnibus:

183.259 Durbin-Watson:

```
2.047  
Prob(Omnibus):          0.000   Jarque-Bera (JB):  
632.415  
Skew:                  1.793   Prob(JB):  
4.71e-138  
Kurtosis:              7.370   Cond. No.  
7.92e+04  
=====
```

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
-----  
OLS Regression Results for target variable: b11  
-----
```

OLS Regression Results

```
=====  
=====  
Dep. Variable:          b11    R-squared:  
0.597  
Model:                 OLS    Adj. R-squared:  
0.582  
Method:                Least Squares    F-statistic:  
39.86  
Date:                  Sat, 08 Nov 2025    Prob (F-statistic):  
4.88e-79  
Time:                  08:41:35    Log-Likelihood:  
726.66  
No. Observations:      475    AIC:  
-1417.  
Df Residuals:          457    BIC:  
-1342.  
Df Model:              17  
  
Covariance Type:       nonrobust  
=====
```

```
=====  
=====  
            coef    std err         t      P>|t|      [0.025  
0.975]  
-----  
const      0.1069     0.056      1.918      0.056      -0.003  
0.216
```

N	-0.0191	0.012	-1.578	0.115	-0.043
0.005					
P	-0.0594	0.176	-0.337	0.736	-0.406
0.287					
K	0.0275	0.015	1.865	0.063	-0.001
0.056					
Ca	0.0061	0.008	0.796	0.426	-0.009
0.021					
Mg	0.0110	0.018	0.611	0.542	-0.024
0.046					
Fe	0.0001	4.73e-05	2.659	0.008	3.28e-05
0.000					
Mn	-1.077e-05	1.29e-05	-0.835	0.404	-3.61e-05
1.46e-05					
Cu	-0.0012	0.002	-0.679	0.498	-0.005
0.002					
Zn	0.0008	0.000	2.159	0.031	7.15e-05
0.002					
B	-0.0007	0.000	-1.604	0.109	-0.002
0.000					
Sigma_VV	0.0492	0.257	0.192	0.848	-0.456
0.554					
Sigma_VH	-0.1760	0.260	-0.676	0.499	-0.687
0.335					
plia	0.0446	0.021	2.076	0.038	0.002
0.087					
lia	-0.0441	0.021	-2.058	0.040	-0.086
-0.002					
iafe	0.0031	0.002	1.673	0.095	-0.001
0.007					
gamma0_vv	-0.1036	0.213	-0.486	0.627	-0.522
0.315					
gamma0_vh	-0.2301	0.363	-0.635	0.526	-0.942
0.482					
<hr/>					
<hr/>					
Omnibus:		91.503	Durbin-Watson:		
1.965					
Prob(Omnibus):		0.000	Jarque-Bera (JB):		
435.089					
Skew:		0.747	Prob(JB):		
3.32e-95					
Kurtosis:		7.444	Cond. No.		
7.92e+04					
<hr/>					
<hr/>					
Notes:					
[1] Standard Errors assume that the covariance matrix of the errors is					

correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

OLS Regression Results for target variable: b9

OLS Regression Results

=====

Dep. Variable: b9 R-squared: 0.993

Model: OLS Adj. R-squared: 0.993

Method: Least Squares F-statistic: 4089.

Date: Sat, 08 Nov 2025 Prob (F-statistic): 0.00

Time: 08:41:35 Log-Likelihood: -801.37

No. Observations: 475 AIC: 1639.

Df Residuals: 457 BIC: 1714.

Df Model: 17

Covariance Type: nonrobust

=====

=====

	coef	std err	t	P> t	[0.025
0.975]					
const	37.2238	1.390	26.777	0.000	34.492
39.956					
N	-1.6433	0.302	-5.444	0.000	-2.237
-1.050					
P	10.5214	4.401	2.391	0.017	1.874
19.169					
K	0.1709	0.368	0.464	0.643	-0.552
0.894					
Ca	-0.1770	0.190	-0.930	0.353	-0.551
0.197					
Mg	-0.8032	0.449	-1.789	0.074	-1.686
0.079					
Fe	-0.0087	0.001	-7.339	0.000	-0.011
-0.006					

Mn	-0.0015	0.000	-4.797	0.000	-0.002
-0.001					
Cu	0.1583	0.044	3.612	0.000	0.072
0.244					
Zn	-0.0315	0.009	-3.425	0.001	-0.050
-0.013					
B	0.0132	0.011	1.162	0.246	-0.009
0.036					
Sigma_VV	-0.6520	6.408	-0.102	0.919	-13.245
11.941					
Sigma_VH	22.5309	6.492	3.471	0.001	9.773
35.289					
plia	-0.9150	0.536	-1.706	0.089	-1.969
0.139					
lia	0.8860	0.535	1.656	0.098	-0.165
1.937					
iafe	-0.9234	0.046	-20.288	0.000	-1.013
-0.834					
gamma0_vv	0.2782	5.317	0.052	0.958	-10.170
10.727					
gamma0_vh	17.3803	9.045	1.921	0.055	-0.395
35.156					
<hr/>					
<hr/>					
Omnibus:		49.790	Durbin-Watson:		
2.011					
Prob(Omnibus):		0.000	Jarque-Bera (JB):		
199.950					
Skew:		0.360	Prob(JB):		
3.81e-44					
Kurtosis:		6.096	Cond. No.		
7.92e+04					
<hr/>					
<hr/>					

#### Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: b8a  
-----

#### OLS Regression Results

=====  
=====  
Dep. Variable: b8a R-squared:

0.992  
 Model: OLS Adj. R-squared:  
 0.991  
 Method: Least Squares F-statistic:  
 3251.  
 Date: Sat, 08 Nov 2025 Prob (F-statistic):  
 0.00  
 Time: 08:41:35 Log-Likelihood:  
 -859.21  
 No. Observations: 475 AIC:  
 1754.  
 Df Residuals: 457 BIC:  
 1829.  
 Df Model: 17  
 Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025
0.975]					
const	35.7868	1.570	22.792	0.000	32.701
38.872					
N	-1.8699	0.341	-5.485	0.000	-2.540
-1.200					
P	10.5078	4.970	2.114	0.035	0.740
20.276					
K	0.1777	0.416	0.427	0.669	-0.639
0.995					
Ca	-0.1393	0.215	-0.648	0.517	-0.562
0.283					
Mg	-0.7907	0.507	-1.559	0.120	-1.787
0.206					
Fe	-0.0108	0.001	-8.097	0.000	-0.013
-0.008					
Mn	-0.0009	0.000	-2.598	0.010	-0.002
-0.000					
Cu	0.1480	0.049	2.991	0.003	0.051
0.245					
Zn	-0.0321	0.010	-3.095	0.002	-0.053
-0.012					
B	0.0202	0.013	1.567	0.118	-0.005
0.045					
Sigma_VV	0.5237	7.238	0.072	0.942	-13.700
14.748					
Sigma_VH	26.9392	7.333	3.674	0.000	12.530
41.349					

plia	-0.9050	0.606	-1.494	0.136	-2.095
0.285					
lia	0.8742	0.604	1.447	0.149	-0.313
2.061					
iafe	-0.8884	0.051	-17.280	0.000	-0.989
-0.787					
gamma0_vv	-0.8943	6.005	-0.149	0.882	-12.696
10.907					
gamma0_vh	20.7853	10.217	2.034	0.042	0.708
40.863					

---

Omnibus:	41.615	Durbin-Watson:
2.022		
Prob(Omnibus):	0.000	Jarque-Bera (JB):
139.968		
Skew:	0.327	Prob(JB):
4.04e-31		
Kurtosis:	5.577	Cond. No.
7.92e+04		

---

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: b8  
-----

OLS Regression Results

---



---

Dep. Variable:	b8	R-squared:
0.994		
Model:	OLS	Adj. R-squared:
0.994		
Method:	Least Squares	F-statistic:
4814.		
Date:	Sat, 08 Nov 2025	Prob (F-statistic):
0.00		
Time:	08:41:35	Log-Likelihood:
-761.89		
No. Observations:	475	AIC:
1560.		
Df Residuals:	457	BIC:

1635.

Df Model:

17

Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025 0.975]
const	35.6518	1.279	27.869	0.000	33.138
N	-2.1106	0.278	-7.599	0.000	-2.656
P	12.8296	4.050	3.168	0.002	4.871
K	0.1502	0.339	0.444	0.658	-0.515
Ca	-0.0166	0.175	-0.095	0.925	-0.361
Mg	-0.7113	0.413	-1.721	0.086	-1.523
Fe	-0.0102	0.001	-9.357	0.000	-0.012
Mn	-0.0007	0.000	-2.353	0.019	-0.001
Cu	0.1143	0.040	2.834	0.005	0.035
Zn	-0.0341	0.008	-4.027	0.000	-0.051
B	0.0060	0.010	0.573	0.567	-0.015
Sigma_VV	2.5121	5.897	0.426	0.670	-9.077
Sigma_VH	24.7936	5.974	4.150	0.000	13.054
plia	-0.9095	0.493	-1.843	0.066	-1.879
lia	0.8769	0.492	1.781	0.076	-0.090
iafe	-0.8781	0.042	-20.964	0.000	-0.960
gamma0_vv	-2.6692	4.893	-0.546	0.586	-12.284
gamma0_vh	24.6061	8.324	2.956	0.003	8.248

```
Omnibus:                      32.228    Durbin-Watson:  
1.944  
Prob(Omnibus):                0.000    Jarque-Bera (JB):  
90.744  
Skew:                          0.270    Prob(JB):  
1.97e-20  
Kurtosis:                      5.072    Cond. No.  
7.92e+04  
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
OLS Regression Results for target variable: b7
```

OLS Regression Results

```
=====  
=====  
Dep. Variable:                  b7      R-squared:  
0.425  
Model:                          OLS      Adj. R-squared:  
0.403  
Method: Least Squares          F-statistic:  
19.86  
Date: Sat, 08 Nov 2025          Prob (F-statistic):  
8.76e-45  
Time: 08:41:35                  Log-Likelihood:  
528.84  
No. Observations:               475      AIC:  
-1022.  
Df Residuals:                  457      BIC:  
-946.7  
Df Model:                      17  
  
Covariance Type:               nonrobust
```

```
=====  
=====  
            coef    std err        t     P>|t|      [0.025  
0.975]  
-----  
const      0.2403    0.084     2.844     0.005      0.074
```

0.406						
N	0.0037	0.018	0.200	0.842	-0.032	
0.040						
P	-0.1396	0.267	-0.522	0.602	-0.665	
0.386						
K	0.0687	0.022	3.070	0.002	0.025	
0.113						
Ca	-0.0043	0.012	-0.373	0.709	-0.027	
0.018						
Mg	0.0240	0.027	0.881	0.379	-0.030	
0.078						
Fe	0.0002	7.17e-05	2.874	0.004	6.51e-05	
0.000						
Mn	-6.001e-05	1.96e-05	-3.065	0.002	-9.85e-05	-
2.15e-05						
Cu	-0.0017	0.003	-0.633	0.527	-0.007	
0.004						
Zn	0.0012	0.001	2.133	0.033	9.37e-05	
0.002						
B	-0.0013	0.001	-1.946	0.052	-0.003	
1.34e-05						
Sigma_VV	0.2961	0.390	0.760	0.447	-0.469	
1.062						
Sigma_VH	-0.5918	0.395	-1.500	0.134	-1.367	
0.184						
plia	0.0646	0.033	1.983	0.048	0.001	
0.129						
lia	-0.0619	0.033	-1.905	0.057	-0.126	
0.002						
iafe	-0.0004	0.003	-0.133	0.894	-0.006	
0.005						
gamma0_vv	-0.2234	0.323	-0.691	0.490	-0.858	
0.412						
gamma0_vh	0.0310	0.550	0.056	0.955	-1.049	
1.111						
<hr/>						
<hr/>						
Omnibus:		55.723	Durbin-Watson:			
1.922						
Prob(Omnibus):		0.000	Jarque-Bera (JB):			
82.980						
Skew:		-0.789	Prob(JB):			
9.58e-19						
Kurtosis:		4.304	Cond. No.			
7.92e+04						
<hr/>						
<hr/>						

Notes:

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

correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

OLS Regression Results for target variable: b6

OLS Regression Results

=====

Dep. Variable: b6 R-squared: 0.584  
Model: OLS Adj. R-squared: 0.568  
Method: Least Squares F-statistic: 37.69  
Date: Sat, 08 Nov 2025 Prob (F-statistic): 7.98e-76  
Time: 08:41:35 Log-Likelihood: 481.35  
No. Observations: 475 AIC: -926.7  
Df Residuals: 457 BIC: -851.8  
Df Model: 17  
Covariance Type: nonrobust

=====

	coef	std err	t	P> t	[0.025
0.975]					
const	0.2194	0.093	2.350	0.019	0.036
0.403	-0.0575	0.020	-2.835	0.005	-0.097
N	0.4284	0.296	1.449	0.148	-0.153
-0.018	0.1114	0.025	4.505	0.000	0.063
P	-0.0211	0.013	-1.650	0.100	-0.046
1.009	0.0416	0.030	1.379	0.168	-0.018
K	0.0003	7.92e-05	3.318	0.001	0.000
0.160					
Ca					
0.004					
Mg					
0.101					
Fe					
0.000					

Mn	-0.0002	2.16e-05	-8.480	0.000	-0.000
-0.000					
Cu	0.007	0.0014	0.003	0.488	0.626
0.001					
Zn	0.001	-0.0006	0.001	-0.960	0.338
B	-0.001	-0.0025	0.001	-3.318	0.001
Sigma_VV	0.808	-0.0377	0.430	-0.087	0.930
0.564					
Sigma_VH	0.072	-0.2935	0.436	-0.673	0.501
plia	0.071	0.0012	0.036	0.034	0.973
lia	0.071	0.0008	0.036	0.022	0.982
iafe	0.005	-0.0011	0.003	-0.352	0.725
gamma0_vv	0.765	0.0636	0.357	0.178	0.859
gamma0_vh	1.791	0.5974	0.608	0.983	0.326
Omnibus:			27.544	Durbin-Watson:	
1.988					
Prob(Omnibus):			0.000	Jarque-Bera (JB):	
36.988					
Skew:			0.481	Prob(JB):	
9.29e-09					
Kurtosis:			3.972	Cond. No.	
7.92e+04					
Notes:					

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: b5  
-----

### OLS Regression Results

=====  
=====  
Dep. Variable: b5 R-squared:

0.388  
 Model: OLS Adj. R-squared:  
 0.365  
 Method: Least Squares F-statistic:  
 17.02  
 Date: Sat, 08 Nov 2025 Prob (F-statistic):  
 7.28e-39  
 Time: 08:41:35 Log-Likelihood:  
 508.26  
 No. Observations: 475 AIC:  
 -980.5  
 Df Residuals: 457 BIC:  
 -905.6  
 Df Model: 17  
 Covariance Type: nonrobust

	coef	std err	t	P> t	[0.025
0.975]					
const	0.5018	0.088	5.687	0.000	0.328
0.675	0.0276	0.019	1.443	0.150	-0.010
N	-0.3770	0.279	-1.350	0.178	-0.926
0.065	0.0089	0.023	0.381	0.703	-0.037
P	0.0074	0.012	0.612	0.541	-0.016
0.172	-0.0092	0.028	-0.321	0.748	-0.065
K	0.0004	7.48e-05	5.978	0.000	0.000
0.055	-4.888e-05	2.04e-05	-2.391	0.017	-8.91e-05
Ca	-0.0033	0.003	-1.190	0.235	-0.009
0.031	0.0027	0.001	4.546	0.000	0.002
Mg	-0.0013	0.001	-1.842	0.066	-0.003
0.047	0.4687	0.407	1.152	0.250	-0.331
Fe	-1.1759	0.412	-2.854	0.005	-1.986
0.001					
Mn					
-8.7e-06					
Cu					
0.002					
Zn					
0.004					
B					
8.91e-05					
Sigma_VV					
1.268					
Sigma_VH					
-0.366					

plia	0.0600	0.034	1.762	0.079	-0.007
0.127					
lia	-0.0595	0.034	-1.752	0.080	-0.126
0.007					
iafe	-0.0063	0.003	-2.192	0.029	-0.012
-0.001					
gamma0_vv	-0.3802	0.337	-1.127	0.261	-1.043
0.283					
gamma0_vh	-0.9285	0.574	-1.617	0.107	-2.057
0.200					
=====					
=====					
Omnibus:	84.402	Durbin-Watson:			
2.014					
Prob(Omnibus):	0.000	Jarque-Bera (JB):			
177.371					
Skew:	0.957	Prob(JB):			
3.05e-39					
Kurtosis:	5.302	Cond. No.			
7.92e+04					
=====					
=====					

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: b4  
-----

OLS Regression Results

Dep. Variable:	b4	R-squared:
0.257		
Model:	OLS	Adj. R-squared:
0.229		
Method:	Least Squares	F-statistic:
9.300		
Date:	Sat, 08 Nov 2025	Prob (F-statistic):
5.47e-21		
Time:	08:41:35	Log-Likelihood:
492.75		
No. Observations:	475	AIC:
-949.5		
Df Residuals:	457	BIC:

-874.6					
Df Model:		17			
Covariance Type:		nonrobust			
<hr/>					
<hr/>					
0.975]	coef	std err	t	P> t	[0.025
<hr/>					
const	0.3241	0.091	3.555	0.000	0.145
0.503					
N	-0.0177	0.020	-0.896	0.371	-0.057
0.021					
P	0.0737	0.289	0.255	0.799	-0.493
0.641					
K	0.0590	0.024	2.443	0.015	0.012
0.106					
Ca	-0.0080	0.012	-0.642	0.521	-0.033
0.017					
Mg	0.0260	0.029	0.883	0.378	-0.032
0.084					
Fe	0.0004	7.73e-05	5.659	0.000	0.000
0.001					
Mn	-0.0001	2.11e-05	-6.604	0.000	-0.000
-9.8e-05					
Cu	-1.163e-05	0.003	-0.004	0.997	-0.006
0.006					
Zn	0.0007	0.001	1.241	0.215	-0.000
0.002					
B	-0.0020	0.001	-2.695	0.007	-0.003
-0.001					
Sigma_VV	-0.0233	0.420	-0.055	0.956	-0.849
0.803					
Sigma_VH	-0.6907	0.426	-1.622	0.105	-1.527
0.146					
plia	0.0050	0.035	0.142	0.887	-0.064
0.074					
lia	-0.0058	0.035	-0.166	0.868	-0.075
0.063					
iafe	-0.0037	0.003	-1.237	0.217	-0.010
0.002					
gamma0_vv	0.0277	0.349	0.079	0.937	-0.658
0.713					
gamma0_vh	-0.3131	0.593	-0.528	0.598	-1.479
0.853					
<hr/>					
<hr/>					

```
Omnibus:                      167.463   Durbin-Watson:  
2.081  
Prob(Omnibus):                0.000    Jarque-Bera (JB):  
494.013  
Skew:                          1.697    Prob(JB):  
5.33e-108  
Kurtosis:                      6.665    Cond. No.  
7.92e+04  
=====
```

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

```
OLS Regression Results for target variable: b3
```

OLS Regression Results

```
=====
```

Dep. Variable:	b3	R-squared:
0.267		
Model:	OLS	Adj. R-squared:
0.240		
Method:	Least Squares	F-statistic:
9.814		
Date:	Sat, 08 Nov 2025	Prob (F-statistic):
2.93e-22		
Time:	08:41:35	Log-Likelihood:
479.11		
No. Observations:	475	AIC:
-922.2		
Df Residuals:	457	BIC:
-847.3		
Df Model:	17	
Covariance Type:	nonrobust	

```
=====
```

	coef	std err	t	P> t	[0.025
0.975]					
-----	-----	-----	-----	-----	-----
const	0.3734	0.094	3.980	0.000	0.189

0.558					
N	-0.0202	0.020	-0.992	0.322	-0.060
0.020					
P	0.0613	0.297	0.207	0.836	-0.522
0.645					
K	0.0609	0.025	2.451	0.015	0.012
0.110					
Ca	-0.0073	0.013	-0.572	0.568	-0.033
0.018					
Mg	0.0320	0.030	1.056	0.292	-0.028
0.092					
Fe	0.0005	7.96e-05	6.197	0.000	0.000
0.001					
Mn	-0.0002	2.17e-05	-6.932	0.000	-0.000
-0.000					
Cu	-1.909e-05	0.003	-0.006	0.995	-0.006
0.006					
Zn	0.0007	0.001	1.050	0.294	-0.001
0.002					
B	-0.0022	0.001	-2.876	0.004	-0.004
-0.001					
Sigma_VV	-0.4433	0.433	-1.025	0.306	-1.293
0.407					
Sigma_VH	-0.2829	0.438	-0.646	0.519	-1.144
0.578					
plia	0.0018	0.036	0.050	0.960	-0.069
0.073					
lia	-0.0020	0.036	-0.054	0.957	-0.073
0.069					
iafe	-0.0053	0.003	-1.710	0.088	-0.011
0.001					
gamma0_vv	0.3770	0.359	1.051	0.294	-0.328
1.082					
gamma0_vh	-0.4882	0.610	-0.800	0.424	-1.688
0.712					
<hr/>					
<hr/>					

Omnibus: 153.970 Durbin-Watson:

2.055

Prob(Omnibus): 0.000 Jarque-Bera (JB):

437.668

Skew: 1.565 Prob(JB):

9.15e-96

Kurtosis: 6.510 Cond. No.

7.92e+04

---

---

Notes:

```
[1] Standard Errors assume that the covariance matrix of the errors is  
correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that  
there are  
strong multicollinearity or other numerical problems.
```

```
-----  
OLS Regression Results for target variable: b2
```

```
-----  
OLS Regression Results
```

```
=====
```

```
Dep. Variable: b2 R-squared: 0.269  
Model: OLS Adj. R-squared: 0.242  
Method: Least Squares F-statistic: 9.898  
Date: Sat, 08 Nov 2025 Prob (F-statistic): 1.82e-22  
Time: 08:41:35 Log-Likelihood: 420.38  
No. Observations: 475 AIC: -804.8  
Df Residuals: 457 BIC: -729.8  
Df Model: 17  
Covariance Type: nonrobust
```

```
=====
```

	coef	std err	t	P> t	[0.025
0.975]					
const	0.4539	0.106	4.275	0.000	0.245
0.663	-0.0305	0.023	-1.324	0.186	-0.076
N	0.1703	0.336	0.507	0.613	-0.490
0.015	0.0613	0.028	2.179	0.030	0.006
P	-0.0073	0.015	-0.505	0.614	-0.036
0.831	0.0321	0.034	0.936	0.350	-0.035
K	0.0006	9.01e-05	6.322	0.000	0.000
0.117					
Ca					
0.021					
Mg					
0.099					
Fe					

0.001					
Mn	-0.0002	2.46e-05	-7.085	0.000	-0.000
-0.000					
Cu	0.0008	0.003	0.237	0.813	-0.006
0.007					
Zn	0.0006	0.001	0.855	0.393	-0.001
0.002					
B	-0.0026	0.001	-3.010	0.003	-0.004
-0.001					
Sigma_VV	-0.3748	0.489	-0.766	0.444	-1.337
0.587					
Sigma_VH	-0.2393	0.496	-0.483	0.630	-1.214
0.735					
plia	-0.0091	0.041	-0.223	0.823	-0.090
0.071					
lia	0.0093	0.041	0.228	0.820	-0.071
0.090					
iafe	-0.0078	0.003	-2.250	0.025	-0.015
-0.001					
gamma0_vv	0.3301	0.406	0.813	0.417	-0.468
1.128					
gamma0_vh	-0.4987	0.691	-0.722	0.471	-1.856
0.859					
<hr/>					
<hr/>					
Omnibus:		148.642	Durbin-Watson:		
2.050					
Prob(Omnibus):		0.000	Jarque-Bera (JB):		
405.431					
Skew:		1.527	Prob(JB):		
9.16e-89					
Kurtosis:		6.341	Cond. No.		
7.92e+04					
<hr/>					
<hr/>					

#### Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: b1  
-----

#### OLS Regression Results

=====  
=====

Dep. Variable:	b1	R-squared:			
0.482					
Model:	OLS	Adj. R-squared:			
0.463					
Method:	Least Squares	F-statistic:			
25.01					
Date:	Sat, 08 Nov 2025	Prob (F-statistic):			
9.33e-55					
Time:	08:41:35	Log-Likelihood:			
375.34					
No. Observations:	475	AIC:			
-714.7					
Df Residuals:	457	BIC:			
-639.7					
Df Model:	17				
Covariance Type:	nonrobust				
<hr/>					
<hr/>					
	coef	std err	t	P> t	[0.025
0.975]					
<hr/>					
const	0.5076	0.117	4.348	0.000	0.278
0.737					
N	-0.0267	0.025	-1.054	0.292	-0.077
0.023					
P	0.1524	0.370	0.412	0.680	-0.574
0.879					
K	0.0751	0.031	2.430	0.016	0.014
0.136					
Ca	-0.0053	0.016	-0.333	0.739	-0.037
0.026					
Mg	0.0217	0.038	0.575	0.565	-0.052
0.096					
Fe	0.0005	9.9e-05	4.620	0.000	0.000
0.001					
Mn	-0.0002	2.7e-05	-6.938	0.000	-0.000
-0.000					
Cu	0.0012	0.004	0.328	0.743	-0.006
0.008					
Zn	0.0010	0.001	1.286	0.199	-0.001
0.003					
B	-0.0025	0.001	-2.630	0.009	-0.004
-0.001					
Sigma_VV	0.3706	0.538	0.689	0.491	-0.687
1.428					
Sigma_VH	-0.7006	0.545	-1.285	0.199	-1.772

0.371					
plia	0.0178	0.045	0.395	0.693	-0.071
0.106					
lia	-0.0190	0.045	-0.422	0.673	-0.107
0.069					
iafe	-0.0081	0.004	-2.108	0.036	-0.016
-0.001					
gamma0_vv	-0.2169	0.446	-0.486	0.627	-1.094
0.660					
gamma0_vh	-0.2295	0.760	-0.302	0.763	-1.722
1.263					
=====					
=====					
Omnibus:		179.827	Durbin-Watson:		
2.075					
Prob(Omnibus):		0.000	Jarque-Bera (JB):		
596.511					
Skew:		1.775	Prob(JB):		
2.95e-130					
Kurtosis:		7.188	Cond. No.		
7.92e+04					
=====					
=====					

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: beta0\_vv  
-----

OLS Regression Results

=====					
=====					
Dep. Variable:	beta0_vv	R-squared:			
0.989					
Model:	OLS	Adj. R-squared:			
0.988					
Method:	Least Squares	F-statistic:			
2368.					
Date:	Sat, 08 Nov 2025	Prob (F-statistic):			
0.00					
Time:	08:41:35	Log-Likelihood:			
1321.3					
No. Observations:	475	AIC:			
-2607.					

Df Residuals:	457	BIC:			
-2532.					
Df Model:	17				
Covariance Type:		nonrobust			
<hr/>					
<hr/>					
0.975]	coef	std err	t	P> t	[0.025
<hr/>					
const	0.0286	0.016	1.795	0.073	-0.003
N	0.0245	0.003	7.084	0.000	0.018
P	-0.1484	0.050	-2.942	0.003	-0.247
K	0.0012	0.004	0.287	0.774	-0.007
Ca	0.0003	0.002	0.116	0.908	-0.004
Mg	0.0112	0.005	2.175	0.030	0.001
Fe	0.0001	1.35e-05	9.682	0.000	0.000
Mn	9.086e-06	3.69e-06	2.462	0.014	1.83e-06
Cu	-0.0014	0.001	-2.709	0.007	-0.002
Zn	0.0004	0.000	4.066	0.000	0.000
B	-0.0002	0.000	-1.247	0.213	-0.000
Sigma_VV	0.0113	0.073	0.154	0.878	-0.133
Sigma_VH	-0.3002	0.074	-4.035	0.000	-0.446
plia	0.0082	0.006	1.338	0.181	-0.004
lia	-0.0079	0.006	-1.294	0.196	-0.020
iafe	-0.0017	0.001	-3.348	0.001	-0.003
gamma0_vv	1.3270	0.061	21.778	0.000	1.207
gamma0_vh	-0.2442	0.104	-2.356	0.019	-0.448

```
=====
Omnibus:                 22.698   Durbin-Watson:
1.942
Prob(Omnibus):           0.000    Jarque-Bera (JB):
61.107
Skew:                    -0.103   Prob(JB):
5.38e-14
Kurtosis:                4.745    Cond. No.
7.92e+04
=====
```

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.  
[2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
OLS Regression Results for target variable: beta0\_vh  
-----

### OLS Regression Results

```
=====
Dep. Variable:          beta0_vh   R-squared:
0.973
Model:                  OLS      Adj. R-squared:
0.972
Method:                 Least Squares   F-statistic:
966.4
Date:                   Sat, 08 Nov 2025   Prob (F-statistic):
0.00
Time:                   08:41:35      Log-Likelihood:
1956.4
No. Observations:       475      AIC:
-3877.
Df Residuals:           457      BIC:
-3802.
Df Model:                17
Covariance Type:        nonrobust
=====
```

	coef	std err	t	P> t	[0.025
0.975]					

```
=====
```

const	0.0031	0.004	0.731	0.465	-0.005
0.011					
N	0.0052	0.001	5.687	0.000	0.003
0.007					
P	-0.0227	0.013	-1.712	0.088	-0.049
0.003					
K	-0.0002	0.001	-0.223	0.824	-0.002
0.002					
Ca	-0.0007	0.001	-1.237	0.217	-0.002
0.000					
Mg	0.0011	0.001	0.793	0.428	-0.002
0.004					
Fe	2.85e-05	3.55e-06	8.031	0.000	2.15e-05
3.55e-05					
Mn	1.724e-06	9.69e-07	1.779	0.076	-1.81e-07
3.63e-06					
Cu	-0.0003	0.000	-2.313	0.021	-0.001
4.58e-05					
Zn	0.0001	2.77e-05	4.591	0.000	7.27e-05
0.000					
B	-3.603e-05	3.43e-05	-1.050	0.294	-0.000
3.14e-05					
Sigma_VV	-0.0372	0.019	-1.930	0.054	-0.075
0.001					
Sigma_VH	-0.0935	0.020	-4.784	0.000	-0.132
-0.055					
plia	0.0032	0.002	1.978	0.049	2.01e-05
0.006					
lia	-0.0031	0.002	-1.929	0.054	-0.006
5.88e-05					
iafe	-0.0002	0.000	-1.801	0.072	-0.001
2.24e-05					
gamma0_vv	0.0321	0.016	2.009	0.045	0.001
0.064					
gamma0_vh	1.2412	0.027	45.595	0.000	1.188
1.295					
<hr/>					
<hr/>					
Omnibus:		37.481	Durbin-Watson:		
1.981					
Prob(Omnibus):		0.000	Jarque-Bera (JB):		
149.286					
Skew:		-0.169	Prob(JB):		
3.83e-33					
Kurtosis:		5.726	Cond. No.		
7.92e+04					
<hr/>					
<hr/>					

Notes:

- [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
- [2] The condition number is large, 7.92e+04. This might indicate that there are strong multicollinearity or other numerical problems.

-----  
Model Evaluation Metrics:  
-----

Mean Squared Error (MSE): 0.3552381004200159

R-squared (R2): 0.46035633350431915

Kode ini bertujuan untuk menganalisis hubungan regresi secara lebih mendalam dan statistikal menggunakan pustaka **statsmodels** alih-alih hanya menggunakan **sklearn**. Pertama, ia menambahkan konstanta (intercept) ke *Dataframe* fitur pelatihan (*X\_train\_sm*) dan pengujian (*X\_test\_sm*), yang merupakan kebutuhan khas untuk model **Ordinary Least Squares (OLS)**. Selanjutnya, skrip melakukan **Regresi OLS terpisah** untuk **setiap variabel target spektral ('b')** terhadap semua fitur input. Untuk setiap model OLS, ia mencetak **Ringkasan Model lengkap (model\_sm.summary())**, yang tidak hanya memberikan koefisien dan *R-squared*, tetapi juga statistik inferensial penting seperti nilai-p (p-value) untuk mengidentifikasi fitur mana yang **signifikan secara statistik**. Terakhir, kode mengulang perhitungan **Mean Squared Error (MSE)** dan **R-squared (R<sup>2</sup>)** menggunakan fungsi dari **sklearn.metrics** untuk memberikan metrik kinerja model secara umum pada data pengujian, melengkapi analisis statistik dengan evaluasi prediktif.