Project name: Submission [Clustering]_Submission_Akhir_BMLP_Fiko_Rahardito_Baskoro

Project Summary

- Implementasi analisis clustering yang menyeluruh, dimulai dari import library hingga evaluasi hasil. Mencakup preprocessing data, rekayasa fitur, pengembangan model, visualisasi, dan interpretasi hasil.
- Pada bagian pengolahan Data, Implementasi preprocessing lengkap termasuk one-hot encoding untuk variabel kategorikal, penanganan nilai yang hilang, dan standardisasi fitur menggunakan StandardScaler.
- Penggunaan multiple metode seleksi fitur (SelectKBest, analisis korelasi) dan implementasi PCA untuk visualisasi.
- Implementasi K-Means clustering dengan optimasi parameter menggunakan Metode Elbow dan Analisis Silhouette, serta multiple evaluasi performa.
- Visualisasi & Interpretasi: Implementasi berbagai teknik visualisasi (heatmap, scatter plot, PCA) dan analisis karakteristik cluster secara mendalam.

Error Notes

Pada project yang diperiksa terjadi sebuah error karna Dataset dibaca berkali-kali di berbagai bagian kode. dan saya mengatasinya dengan cara Mengunakan pipeline atau fungsi untuk menghindari pembacaan ulang. Selanjutnya

Code Review

```
# 1. Import Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
```

Beberapa library tidak digunakan (RandomForestClassifier, train_test_split)
sebaiknya perlu pengelompokan import berdasarkan fungsi

```
#Type your code here
# Set display options for better readability (optional)
pd.set_option('display.max_columns', None)

# Load the dataset
file_path = "/content/marketing_campaign.csv"
data_v1 = pd.read_csv(file_path)

# Display the first few rows to understand the dataset
print("Dataset Loaded Successfully!")
print(data_v1.head())
```

Untuk lebih effisien tidak perlu menggunakan fungsi print untuk menampilkan data, cukup gunakan “data_v1.head()” untuk menampilkan data, dan juga tidak ada validasi path file serta tidak ada error handling untuk pembacaan file

```
# Lakukan One-Hot Encoding untuk kolom 'Marital_Status'
data_v1 = pd.get_dummies(data_v1, columns=['Marital_Status'], prefix='Marital')

# Lakukan One-Hot Encoding untuk kolom 'Education'
data_v1 = pd.get_dummies(data_v1, columns=['Education'], prefix='Education')

# Convert True/False to 1/0 in the relevant columns
for column in data_v1.select_dtypes(include=['bool']).columns:
    data_v1[column] = data_v1[column].astype(int)  # Convert True to 1, False to 0

# Tampilkan beberapa baris pertama dari data yang telah di-encode
print(data_v1.head())
```

Proses encoding dilakukan terpisah untuk setiap kolom dan perlu validasi hasil transformasi

```
X = data_v1[features]
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Optimasi jumlah cluster
inertia = []
silhouette_avg = []
for n_clusters in range(2, 11):
    kmeans = KMeans(n_clusters=n_clusters, random_state=42)
    kmeans.fit(X_scaled)
    inertia.append(kmeans.inertia_)
    silhouette_avg.append(silhouette_score(X_scaled, kmeans.labels_))
```

Scaling dilakukan sebelum analisis missing values , Tidak ada progress tracking dalam loop dan perlu error handling

```
selector = SelectKBest(score_func=f_classif, k=10)
X_new = selector.fit_transform(X, kmeans.labels_)
selected_features = X.columns[selector.get_support()]

# Model Evaluation
kmeans_selected = KMeans(n_clusters=optimal_n_clusters, random_state=42)
kmeans_selected.fit(X_new)

silhouette_avg_before = silhouette_score(X_scaled, kmeans.labels_)
silhouette_avg_after = silhouette_score(X_new, kmeans_selected.labels_)
```

Pemilihan k=10 perlu justifikasi, perlu validasi hasil seleksi fitur dan dokumentasi perbandingan performa kurang lengkap

```
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X_scaled)
plt.figure(figsize=(8, 6))
for i in range(optimal_n_clusters):
    plt.scatter(X_pca[data_v1['cluster'] == i, 0],
                X_pca[data_v1['cluster'] == i, 1],
                label=f'Cluster {i}')
```

```
# Analisis karakteristik cluster
for i in range(optimal_n_clusters):
    print(f"Karakteristik Cluster {i}:")
    cluster_data = data_v1[data_v1['cluster'] == i]
    print(cluster_data.describe())
```

Visualisasi PCA tidak menjelaskan variance explained, Format output analisis tidak terstruktur sehingga perlu ringkasan karakteristik yang lebih jelas

Rekomendasi Perbaikan

Disarankan untuk menggunakan pipeline untuk menghindari pembacaan data berulang, menambahkan error handling saat memuat data, dan menghapus library yang tidak digunakan. Lakukan one-hot encoding sekaligus dengan validasi hasil, analisis missing values sebelum scaling, serta tambahkan progress tracking saat iterasi K-Means. Justifikasi pemilihan k=10, sertakan variance explained pada PCA, dan tampilkan hasil analisis cluster dalam format yang lebih terstruktur.

Kesimpulan akhir

Secara keseluruhan, perbaikan yang direkomendasikan bertujuan untuk meningkatkan efisiensi, akurasi, dan keterbacaan kode. Dengan implementasi pipeline, validasi data, serta optimasi proses analisis dan visualisasi, diharapkan workflow menjadi lebih terstruktur, andal, dan mudah diinterpretasi, sehingga menghasilkan model clustering yang lebih optimal dan insight yang lebih jelas.