Project name: Submission [Clustering] Submission Akhir BMLP_Muhammad Fikry Rizal

Project Summary

- Dimulai dari persiapan data (missing values handling, outlier detection), split data untuk analisis, pelatihan model clustering, hingga evaluasi dengan silhouette score dan elbow method
- Menggunakan K-Means clustering dengan optimasi parameter melalui elbow method
- Implementasi berbagai visualisasi menggunakan matplotlib dan seaborn untuk EDA dan hasil clustering
- Implementasi fungsi-fungsi terpisah untuk analisis dan evaluasi cluster
- Fokus pada fitur Income dan MntMeatProducts dengan pertimbangan korelasi dan relevansi untuk segmentasi

Error Notes

Pada project yang diperiksa terjadi sebuah error saat Data leakage karena tidak ada pemisahan data dan validasi silang, Kolom Income dan MntMeatProducts perlu divalidasi korelasinya dengan target segmentasi, Potensi bias pada hasil clustering karena outlier treatment yang terlalu agresif, Missing values handling yang hanya menggunakan mean tanpa mempertimbangkan distribusi data. dan saya mengatasinya dengan cara memperbaiki struktur kode dan menambahkan penyempurnaan pada kode seperti melakukan pemisahan data dan validasi silang hingga menghandle missing value dengan mempertimbangkan distribusi data. (bisa menggunakan mean, median dllnya sesuai dengan sutuasi dari data itu sendiri)

Code Review

```
import pandas as pd
import matplotlib.pyplot as plt
from yellowbrick.cluster import KElbowVisualizer
from sklearn.cluster import KMeans, DBSCAN
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import silhouette_score
```

Sebaiknya menghapus code yang tidak diperlukan seperti pada bagian Import DBSCAN dan MinMaxScaler tidak digunakan serta Tambahkan import seaborn (sns) karna digunakan dalam code tersebut namun tidak di import disini

```
data_numerik = data.select_dtypes(include=['int64', 'float64']).copy()

num_vars = data_numerik.shape[1]
n_cols = 4
n_rows = -(-num_vars // n_cols)

fig, axes = plt.subplots(n_rows, n_cols, figsize=(20, n_rows * 4))
axes = axes.flatten()

for i, column in enumerate(data_numerik.columns):
    data_numerik[column].hist(ax=axes[i], bins=20, edgecolor='black')
    axes[i].set_title(column)
    axes[i].set_xlabel('Value')
```

```
    axes[i].set_ylabel('Frequency')

for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()

plt.figure(figsize=(12, 10))
correlation_matrix = data_numerik.corr()
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix')
plt.show()
```

Pada bagian ini anda bisa membuat fungsi terpisah untuk visualisasi, Tambahkan error handling untuk plotting, Optimasi figure size dan layout dan Dokumentasikan insight dari EDA secara jelas berdasarkan output nya

```
for feature in data_numerik.drop(columns=["ID"]).columns:
    Q1 = data_selection[feature].quantile(0.25)
    Q3 = data_selection[feature].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    data_selection[feature] = data_selection[feature].apply(
        lambda x: lower_bound if x < lower_bound else upper_bound if x > upper_bound else x
    )

    plt.figure(figsize=(10, 6))
    sns.boxplot(x=data_selection[feature])
    plt.title(f'Box Plot of {feature}')
    plt.show()
```

Pada bagian ini bisa dibuat fungsi khusus outlier detection, Tambahkan juga multiple detection methods serta Validasi hasil treatment

```
data_cluster = data_selection[["Income", "MntMeatProducts"]].copy()
data_scaled = data_cluster.values

k = 3
kmeans = KMeans(
    n_clusters=k,
    init='k-means++',
    max_iter=500,
    random_state=42
)
kmeans.fit(data_scaled)
labels = kmeans.labels_
```

Justifikasi pemilihan k dan Tambahkan parameter tuning

```
kmeans = KMeans()
visualizer = KElbowVisualizer(kmeans, k=(1, 10))
visualizer.fit(data_scaled)
visualizer.show()
```

```
silhouette_avg = silhouette_score(data_scaled, labels)
print(f"\nSilhouette Score: {silhouette_avg:.2f}")
```

Anda bisa menambahkan multiple evaluation metrics, Implementasi cross-validation serta Dokumentasikan evaluation results

```
centroids = kmeans.cluster_centers_

plt.figure(figsize=(12, 8))
plt.scatter(data_scaled[:, 0], data_scaled[:, 1], c=labels, cmap='viridis', s=50, alpha=0.6, edgecolors='w',
marker='o')
plt.scatter(centroids[:, 0], centroids[:, 1], c='red', s=200, marker='X', label='Centroids')

for i, centroid in enumerate(centroids):
    plt.text(centroid[0], centroid[1], f'Centroid {i+1}', color='red', fontsize=12, ha='center', va='center')

plt.title('Visualisasi Cluster dengan Centroid')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Amount spent on meat')
plt.legend()
plt.show()

print("Nilai Centroids:")
for i, centroid in enumerate(centroids):
    print(f"Centroid {i+1}: Annual Income = {centroid[0]:.2f}, Amount Spent on Meat = {centroid[1]:.2f}")
```

Pada bagian ini anda bisa mengimprove visualization customization, Tambahkan juga statistical analysis Serta comprehensive reporting

Rekomendasi:

- Terapkan Normalisasi/Standarisasi sebelum clustering
- Lakukan Cross-Validation untuk stabilitas model
- Cek Kebutuhan Fitur dengan feature importance analysis
- Modularisasi Kode untuk reusability
- Eksplorasi metode clustering lain untuk perbandingan

Kesimpulan Akhir:

Proyek ini sudah mencakup tahapan penting dalam analisis clustering, mulai dari preprocessing, implementasi model, hingga evaluasi dan visualisasi hasil. Namun masih ada ruang peningkatan dalam hal validasi, error handling, dan dokumentasi agar lebih baik lagi