

Project Summary

- Proyek ini berfokus pada penerapan clustering menggunakan K-Means untuk mengelompokkan data spesies berdasarkan beberapa fitur yaitu fitur panjang (length), berat (weight), dan rasio berat terhadap panjang (w_l_ratio). Proses mencakup Exploratory Data Analysis (EDA), preprocessing, feature selection, model training, evaluasi, dan visualisasi hasil clustering.
- Struktur workflow sudah sistematis mulai dari EDA → Preprocessing → Clustering → Evaluasi.
- Penggunaan K-Means dan evaluasi dengan Elbow Method & Silhouette Score.
- Pembersihan data duplikat dan normalisasi menggunakan MinMaxScaler.
- Saya menambahkan Eksperimen dengan Feature Selection menggunakan PCA.
- Visualisasi hasil clustering menggunakan scatter plot.
- Kode memiliki redundansi yang bisa diminimalkan dengan fungsi modular.
- Feature selection (PCA) menurunkan kualitas clustering.
- Belum ada eksplorasi dengan metode clustering lain seperti DBSCAN.

Error Notes

Pada project yang diperiksa terjadi sebuah error saat saat normalisasi dan penambahan kolom cluster Penyebabnya ada pada **fixsdataset["lengthNorm"] = scaler_length.fit_transform(fixsdataset[["length"]])** menyebabkan SettingWithCopyWarning. dan saya mengatasinya dengan cara mengunakan **.loc[]** untuk memastikan operasi dilakukan dengan benar.

Code Review

```
pip install --upgrade package-name
```

Selalu gunakan versi terbaru dari library Python karena biasanya versi terbaru lebih stabil dan lebih kompatibel. Gunakan kode tersebut untuk memperbarui liblary python yang akan digunakan.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import LabelEncoder, StandardScaler, MinMaxScaler
from sklearn.cluster import KMeans, DBSCAN
from sklearn.metrics import silhouette_score
```

Pastikan hanya mengimpor pustaka yang digunakan (DBSCAN diimpor tetapi tidak digunakan dalam implementasi kodennya).
Coba Pertimbangkan untuk penggunaan “tqdm” untuk progress monitoring jika dataset yang akan diolah lebih besar.

```
# Tampilkan DataFrame untuk memastikan telah dibaca dengan benar
datasetori.head(10)
```

Untuk membaca dan memastikan data benar dibaca, cukup gunakan kode “datasetori.head()”.
Tambahkan Angka di dalam kurung jika ingin membaca dataset sesuai dengan format atau jumlah yang ingin ditampilkan.

```
dataset.isnull().sum()  
dataset = dataset.drop_duplicates()
```

Pastikan untuk melakukan pengecekan missing value dilakukannya preprocessing data lainnya. Dan anda bisa juga untuk menambahkan "reset_index(drop=True)" setelah menghapus duplikat agar indexing tetap rapi.

Contoh : `dataset = dataset.drop_duplicates().reset_index(drop=True)`

```
le = LabelEncoder()  
dataset["species_Encoded"] = le.fit_transform(dataset["species"])  
  
scaler = MinMaxScaler()  
dataset[["length", "weight", "w_l_ratio"]] = scaler.fit_transform(dataset[["length", "weight", "w_l_ratio"]])
```

Akan lebih baik untuk menghindari mutasi langsung pada DataFrame saat normalisasi, gunakan `.loc[]` untuk menghindari SettingWithCopyWarning.

```
kmeans = KMeans(n_clusters=3, random_state=42)  
clusters = kmeans.fit_predict(dataset[["length", "weight", "w_l_ratio"]])  
dataset["Cluster"] = clusters
```

Tampilan kode diatas merupakan kode yang sudah disesuaikan. gunakan `n_init=10` pada KMeans untuk hasil lebih stabil.
Pertimbangkan juga eksperimen dengan metode lain seperti DBSCAN untuk perbandingan hasil.

```
plt.scatter(dataset['length'], dataset['weight'], c=dataset['Cluster'], cmap='rainbow')  
plt.xlabel('Length')  
plt.ylabel('Weight')  
plt.title('Scatter Plot of Length vs Weight with Clusters')  
plt.colorbar(label='Cluster')  
plt.show()
```

Anda bisa menggunakan `sns.pairplot()` untuk melihat hubungan variabel lebih baik.
Tambahkan juga visualisasi Silhouette Score per klaster.

Notes :

Perbaiki SettingWithCopyWarning dengan `.loc[]` untuk menghindari error.

Modularisasi kode dengan fungsi agar lebih reusable (misalnya preprocessing & model training).

Eksperimen dengan DBSCAN dan Davies-Bouldin Index untuk evaluasi tambahan.

Gunakan warna lebih kontras pada scatter plot agar visualisasi lebih jelas.

Tambahkan logging atau tqdm jika dataset besar untuk monitoring proses.

Kesimpulan Akhir:

Proyek ini sudah sesuai dengan instruksi dan memiliki workflow yang jelas.

Namun, ada beberapa aspek yang bisa dioptimalkan, seperti handling DataFrame dengan lebih baik, eksperimen metode lain, dan memperbaiki visualisasi clustering.