Rakamin Academy

Hakuna Matata Travel Insurance

Dokumen Laporan Final Project

Machine Learning Evaluation Unsupervised-Clustering

Standarization

Rakamin Academy

```
from sklearn.preprocessing import StandardScaler

feature = ['Age','Employment Type','GraduateOrNot','AnnualIncome','ChronicDiseases','TravelExperience','FamilyMembers']
X = df_new[feature].values

X_std = StandardScaler().fit_transform(X)

df_std = pd.DataFrame(data = X_std, columns = feature)
df_std.describe()
```

	Age	Employment Type	GraduateOrNot	AnnualIncome	ChronicDiseases	TravelExperience	FamilyMembers
count	1.987000e+03	1.987000e+03	1.987000e+03	1.987000e+03	1.987000e+03	1.987000e+03	1.987000e+03
mean	-6.333915e-16	2.324372e-17	-7.151915e-17	3.575957e-17	-1.162186e-17	6.257925e-17	-1.609181e-16
std	1.000252e+00	1.000252e+00	1.000252e+00	1.000252e+00	1.000252e+00	1.000252e+00	1.000252e+00
min	-1.596603e+00	-6.342384e-01	-2.394910e+00	-1.679482e+00	-6.202169e-01	-6.038873e-01	-1.710675e+00
25%	-5.665868e-01	-6.342384e-01	4.175523e-01	-8.832207e-01	-6.202169e-01	-6.038873e-01	-4.678554e-01
50%	-2.232480e-01	-6.342384e-01	4.175523e-01	-8.695957e-02	-6.202169e-01	-6.038873e-01	1.535541e-01
75%	8.067684e-01	1.576694e+00	4.175523e-01	8.420117e-01	1.612339e+00	4.155894e-01	7.749637e-01
max	1.836785e+00	1.576694e+00	4.175523e-01	2.301824e+00	1.612339e+00	2.454543e+00	2.639192e+00

StandardScaler dilakukan sebagai proses pre-processing data bertujuan untuk memastikan bahwa data dapat digunakan untuk pelatihan model machine learning dengan baik dengan berbagai algoritma

K-means Modeling - Elbow Method

Rakamin Academy

Modeling

kmeans modeling - Elbow Method

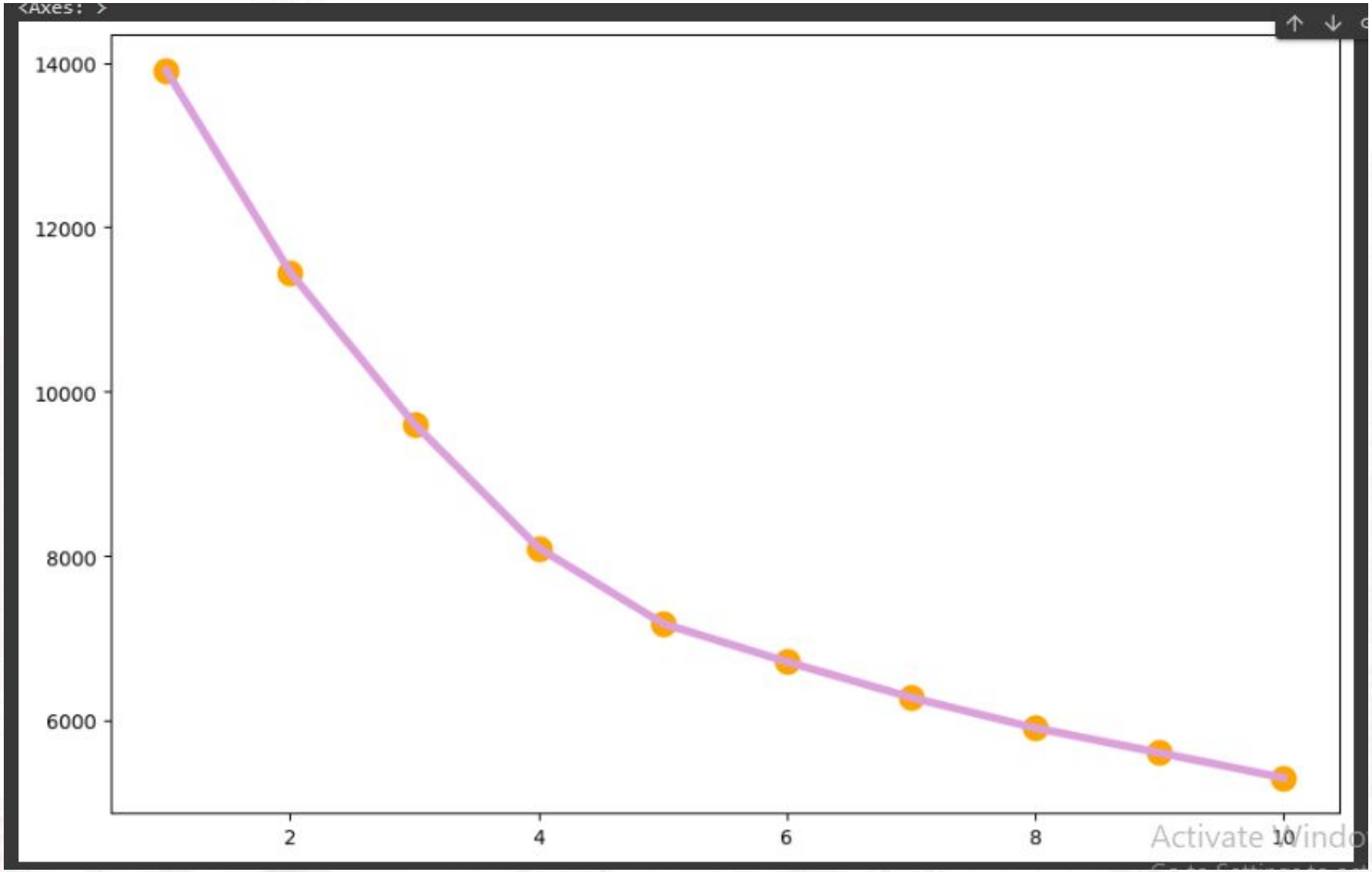
```
[ ]  #Elbow Method
     from sklearn.cluster import KMeans

     inertia = []

     for i in range(1,11):
       kmeans = KMeans(n_clusters=i, random_state=0)
       kmeans.fit(df_std)
       inertia.append(kmeans.inertia_)
```

```
plt.figure(figsize=(11,7))
sns.lineplot(x= range(1,11), y=inertia, color='plum', linewidth=4)
sns.scatterplot(x= range(1,11), y=inertia, color='orange', s=200)
```

Menghitung nilai inersia pada cluster dalam algoritma K-means. Nilai inersia mengukur seberapa jauh titik-titik dalam suatu cluster berada dari pusat clusternya (centroid). Semakin kecil nilai inersia, semakin padat clusternya, yang menunjukkan bahwa titik-titik dalam cluster berada lebih dekat dengan pusat clusternya.



Berdasarkan visualisasi nilai inersia tersebut elbow method cluster yang optimal adalah 4

K-means Clustering

Rakamin Academy

```
[ ]  from sklearn.cluster import KMeans
     kmeans = KMeans(n_clusters=4, random_state=0)
     kmeans.fit(X_std)

     # add column clusters to standardize data
     df_std['clusters'] = kmeans.labels_

     # add column clusters to raw data
     df_new['clusters'] = kmeans.labels_
```

```
[ ]  df_new.sample(5)
```

	Age	Employment Type	GraduateOrNot	AnnualIncome	FamilyMembers	ChronicDiseases	TravelInsurance	TravelExperience	clusters
976	26	0	1	1300000	5	0	0	0	1
1506	28	0	1	1150000	6	1	0	0	1
217	28	0	1	800000	6	0	0	0	1
222	28	0	1	1200000	3	0	0	0	1
1502	28	0	1	750000	6	1	0	0	1

Menggunakan algoritma K-Means untuk melakukan clusterisasi pada data yang telah diskalakan (X_std). Kemudian, menambahkan kolom clusters ke dalam data yang telah diskalakan (df_std) dan data asli (df_new) untuk menandai cluster mana yang termasuk pada setiap titik data

```
[ ] df_std['clusters'].value_counts()

1    845
3    502
2    360
0    280
Name: clusters, dtype: int64
```

Setelah menambahkan kolom clusters. Kemudian, melakukan valua_counts pada setiap clusters, berdasarkan hasil dari kode di atas dapat dilihat bahwa terdapat empat cluster yang diberi label (0, 1, 2, 3) dengan menunjukkan jumlah titik data yang terdapat pada masing-masing cluster. Semakin merata jumlahnya, semakin seimbang clusternya. Dalam kasus ini, jumlah titik data dalam setiap cluster tidak merata, dengan beberapa cluster memiliki lebih banyak titik data daripada yang lain. Hal ini menunjukkan adanya ketidakseimbangan dalam distribusi cluster

Agglomerative Clustering

Rakamin Academy

```
[ ] from sklearn.cluster import AgglomerativeClustering

[ ] ac = AgglomerativeClustering(n_clusters=4, affinity='euclidean')
    ac.fit(df_std2)

    AgglomerativeClustering
    AgglomerativeClustering(affinity='euclidean', n_clusters=4)

[ ] ac.labels_

    array([1, 0, 0, ..., 0, 2, 0])

[ ] df_new2 = df_new.copy()

[ ] # add column clusters to standardize data
    df_std2['clusters'] = ac.labels_

    # add column clusters to raw data
    df_new2['clusters'] = ac.labels_
```

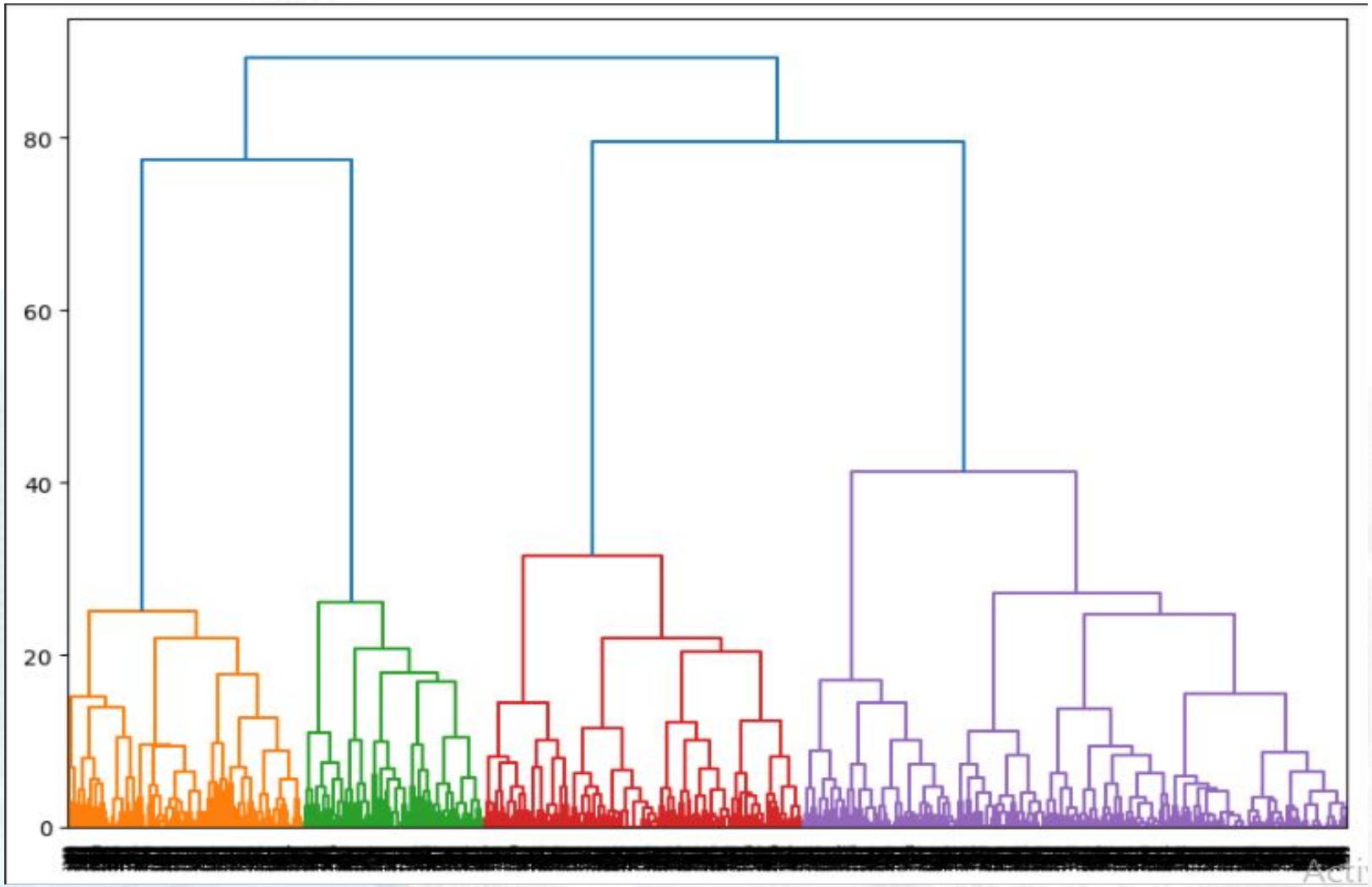
```
[ ] df_std2['clusters'].value_counts()

    0    844
    1    495
    2    367
    3    281
    Name: clusters, dtype: int64

[ ] import scipy.cluster.hierarchy as shc
    from scipy.cluster.hierarchy import linkage, dendrogram

[ ] plt.figure(figsize=(10,7))
    dend = shc.dendrogram(shc.linkage(df_std2, method='ward'))
```

1. Agglomerative Clustering digunakan untuk mengelompokkan titik-titik data ke dalam cluster berdasarkan tingkat kemiripan atau kedekatan antara data
2. berdasarkan kode pada slide, algoritma Agglomerative Clustering untuk melakukan clusterisasi pada data (df_std2) dengan 4 cluster. Kemudian, label cluster yang diperoleh disimpan dalam atribut labels_ dari objek AgglomerativeClustering, lalu menambahkan kolom clusters pada data
3. Value_counts digunakan untuk menghitung jumlah kemunculan setiap nilai unik dalam kolom clusters.



Dendrogram menunjukkan representasi grafis dari hasil klasterisasi hierarkis berdasarkan kode pada slide sebelumnya, di mana sumbu-x menunjukkan titik data atau cluster, sedangkan sumbu-y menunjukkan tingkat kedekatan atau kemiripan antara mereka. Garis-garis vertikal pada dendrogram menunjukkan penggabungan cluster, sedangkan tinggi garis horizontal menunjukkan tingkat kesamaan antara klaster yang digabungkan.

Model Evaluation

Rakamin Academy

Model Evaluation

```
[ ]  # Evaluasi Hierarchical Clustering
     from sklearn.metrics.cluster import adjusted_rand_score

     hierarchical_score = adjusted_rand_score(df_std2['clusters'], df_std['cluster'])
     print("Adjusted Rand Score for KMeans and Hierarchical Clustering:", hierarchical_score)
```

```
Adjusted Rand Score for KMeans and Hierarchical Clustering: 0.98467312767906l6
```

Nilai dari evaluasi model menunjukkan Rand Score 0.98…, hal ini menunjukkan adanya kesesuaian yang baik antara kedua clusterisasi yang sudah dilakukan, yaitu K-means dan Hierarchical Clustering. Hal tersebut dikarenakan Semakin dekat nilai Adjusted Rand Score ke 1, semakin baik kesesuaian antara kedua clusterisasi

Hyperparameter Tuning

Rakamin Academy

```
[ ] from sklearn.model_selection import GridSearchCV

    kmeans = KMeans(random_state=0)

    # Daftar hyperparameter yang akan diuji
    param_grid_kmeans = {
        'n_clusters': [2, 3, 4, 5, 6],
        'init': ['k-means++', 'random'],
        'max_iter': [300, 500, 1000],
        'tol': [1e-4, 1e-5, 1e-6]
    }

    # Lakukan GridSearchCV untuk KMeans
    grid_search_kmeans = GridSearchCV(estimator=kmeans, param_grid=param_grid_kmeans, cv=3)
    grid_search_kmeans.fit(df_std)

    # Tampilkan parameter terbaik
    print("Best parameters for KMeans:", grid_search_kmeans.best_params_)
```

```
Best parameters for KMeans: {'init': 'k-means++', 'max_iter': 300, 'n_clusters': 6, 'tol': 0.0001}
```

1. Menggunakan GridSearchCV dari sklearn.model_selection untuk melakukan pencarian parameter terbaik untuk model K-Means
2. Berdasarkan hasil kode menunjukkan bahwa inisialisasi (‘init’) terbaik pada kasus dataset ini adalah 'k-means++'. Metode ini merupakan salah satu metode inisialisasi yang umum digunakan dalam algoritma K-Means untuk menentukan posisi awal pusat cluster yang lebih baik secara acak.
3. Untuk iterasi maksimum yang terbaik adalah 300 kali pada setiap cluster
4. ‘n_clusters’ atau jumlah cluster yang terbaik K-means pada dataset ini adalah 6 cluster
5. Nilai toleransi yang optimal adalah 0.0001. Artinya, algoritma K-Means akan berhenti ketika perubahan nilai inersia antara dua iterasi berturut-turut kurang dari 0.0001.

```python
from sklearn.metrics import silhouette_score

ac = AgglomerativeClustering()

# Daftar hyperparameter yang akan diuji
param_grid_ac = {
    'n_clusters': [2, 3, 4, 5, 6],
    'linkage': ['ward', 'complete', 'average', 'single']
}

# Fungsi untuk menghitung Silhouette Score sebagai metrik evaluasi
def silhouette_scorer(estimator, X):
    clusters = estimator.fit_predict(X)
    return silhouette_score(X, clusters)

# Lakukan GridSearchCV untuk Agglomerative Clustering
grid_search_ac = GridSearchCV(estimator=ac, param_grid=param_grid_ac, cv=3, scoring=silhouette_scorer)
grid_search_ac.fit(df_std2)

# Tampilkan parameter terbaik
print("Best parameters for Agglomerative Clustering:", grid_search_ac.best_params_)
```

```
Best parameters for Agglomerative Clustering: {'linkage': 'ward', 'n_clusters': 4}
```

Metrik Silhouette score digunakan untuk mengevaluasi kualitas klasterisasi, dalam kasus ini adalah AgglomerativeClustering. Berdasarkan hasil kode yang sudah dijalankan, menunjukkan hasil bahwa parameter metode penggabungan data (‘linkage’) yang optimal adalah 'ward'. Ward adalah salah satu metode yang umum digunakan dalam Agglomerative Clustering. Metode ini menggunakan metode dalam hierarki klasterisasi untuk meminimalkan varians dalam setiap klaster yang digabungkan. Sedangkan untuk jumlah cluster terbaik adalah 4 cluster

Feature Importance

Rakamin Academy

```
[ ] from scipy.stats import f_oneway

    # Memisahkan data menjadi kelompok berdasarkan cluster
    cluster0 = df_std2[df_std2['cluster'] == 0]
    cluster1 = df_std2[df_std2['cluster'] == 1]
    cluster2 = df_std2[df_std2['cluster'] == 2]
    cluster3 = df_std2[df_std2['cluster'] == 3]
```

```
[ ] # Uji ANOVA untuk setiap fitur
    for feature in df_std2.columns[:-1]:  # Exclude 'cluster' column
        f_statistic, p_value = f_oneway(cluster0[feature], cluster1[feature], cluster2[feature], cluster3[feature])
        print(f"ANOVA for {feature}: F-statistic={f_statistic}, p-value={p_value}")
```

```
ANOVA for Age: F-statistic=8.72063163007818, p-value=9.549531743280643e-06
ANOVA for Employment Type: F-statistic=3855.8494711300727, p-value=0.0
ANOVA for GraduateOrNot: F-statistic=10889.960898010993, p-value=0.0
ANOVA for AnnualIncome: F-statistic=392.87006444497564, p-value=2.973260500477045e-200
ANOVA for ChronicDiseases: F-statistic=0.8040037982542952, p-value=0.49156182049231556
ANOVA for TravelExperience: F-statistic=1865.088123703384, p-value=0.0
ANOVA for FamilyMembers: F-statistic=0.777240818764334, p-value=0.5066195421026924
ANOVA for cluster: F-statistic=inf, p-value=0.0
/usr/local/lib/python3.10/dist-packages/scipy/stats/_stats_py.py:4167: ConstantInputWarning: Each of the input arrays is constant;the F statis
  warnings.warn(stats.ConstantInputWarning(msg))
```

Hasil uji ANOVA menunjukkan bahwa terdapat perbedaan signifikan antara setidaknya dua kelompok klaster untuk semua fitur kecuali 'ChronicDiseases' dan 'FamilyMembers'.

Untuk fitur 'Age', 'Employment Type', 'GraduateOrNot', 'AnnualIncome', dan 'TravelExperience', nilai p-value yang sangat kecil menunjukkan bahwa terdapat perbedaan signifikan antara setidaknya dua kelompok klaster untuk fitur-fitur tersebut.

Namun, untuk fitur 'ChronicDiseases' dan 'FamilyMembers', nilai p-value yang lebih besar dari 0.05 menunjukkan bahwa tidak ada perbedaan signifikan antara kelompok klaster untuk fitur-fitur tersebut.

Hasil uji ANOVA untuk kolom 'cluster' menunjukkan bahwa terdapat perbedaan signifikan antara setidaknya dua kelompok klaster, seperti yang diharapkan.

Ini menunjukkan bahwa fitur-fitur seperti 'Age', 'Employment Type', 'GraduateOrNot', 'AnnualIncome', dan 'TravelExperience' mungkin menjadi penentu penting dalam membedakan antara klaster-klaster yang telah diidentifikasi. Sedangkan fitur-fitur seperti 'ChronicDiseases' dan 'FamilyMembers' mungkin tidak memberikan kontribusi signifikan dalam membedakan klaster-klaster tersebut.

Insight of the dataset

Cluster 0:

Karakteristik:

Rata-rata pelanggan bekerja di bidang swasta/wiraswasta.
Rata-rata bukan lulusan universitas.
Memiliki pendapatan di bawah rata-rata.
Jarang bepergian domestik dan internasional.
Interpretasi: Pelanggan di cluster ini dapat dikategorikan sebagai pelanggan berpenghasilan rendah dengan mobilitas rendah. Cluster ini perlu didorong untuk membeli paket asuransi yang sesuai dengan keuangannya dan fokuskan pada perjalanan domestik.

Cluster 1:

Karakteristik:

Rata-rata pelanggan bekerja di bidang swasta/wiraswasta.
Rata-rata adalah lulusan universitas.
Memiliki pendapatan rata-rata.
Jarang bepergian domestik dan internasional.
Interpretasi: Pelanggan di cluster ini dapat dikategorikan sebagai pelanggan berpenghasilan sedang dengan mobilitas rendah. Cluster ini memiliki potensi untuk menjadi pelanggan berulang (repeat customer).

Cluster 2:

Karakteristik:

Rata-rata pelanggan bekerja di bidang swasta/wiraswasta.
Rata-rata adalah lulusan universitas.
Memiliki pendapatan di atas rata-rata.
Sering bepergian domestik dan internasional.
Interpretasi: Pelanggan di cluster ini dapat dikategorikan sebagai pelanggan berpenghasilan tinggi dengan mobilitas tinggi. Cluster ini perlu dipertahankan dan ditingkatkan penjualannya maupun nilai pelanggannya.

Cluster 3:

Karakteristik:

Rata-rata pelanggan bekerja di pemerintah.
Rata-rata adalah lulusan universitas.
Memiliki pendapatan di bawah rata-rata.
Jarang bepergian domestik dan internasional.
Interpretasi: Sedikit berbeda dengan cluster 0, cluster 3 adalah Pegawai Negeri Sipil dengan mobilitas rendah. Umumnya, PNS sudah memiliki asuransi sendiri sehingga tidak membutuhkan asuransi dari luar lagi. Tujuan menyasar cluster ini adalah untuk membangun kerjasama dengan instansi pemerintah dan mendorong pembelian paket asuransi pelengkap.

Business Recommendation

Cluster 0: Pelanggan Berpenghasilan Rendah dengan Mobilitas Rendah

- Tawarkan premi yang terjangkau.
- Tawarkan jenis asuransi single trip yang dapat menjamin untuk satu kali perjalanan dalam kurun waktu tertentu.
- Tawarkan jenis asuransi perjalanan domestik.
- Fokus pada manfaat dasar asuransi perjalanan, yaitu mengganti biaya pengobatan dan kecelakaan diri selama perjalanan.
- Jalin kerjasama dengan perusahaan tour & travel dan online booking platform.
- Edukasi pelanggan tentang manfaat asuransi perjalanan dengan Covid cover melalui email, sosial media, dan acara Travel Fair.

Cluster 1: Pelanggan Berpenghasilan Sedang dengan Mobilitas Rendah

- Tawarkan promo dan diskon menarik untuk mendorong pembelian.
- Tambahkan manfaat asuransi perjalanan, seperti penundaan/pembatalan perjalanan, keterlambatan/kehilangan bagasi, dan kerusakan/kehilangan barang berharga.
- Tawarkan jenis asuransi perjalanan domestik dan internasional.
- Jalin kerjasama dengan perusahaan tour & travel dan online booking platform.
- Edukasi pelanggan tentang manfaat asuransi perjalanan dengan Covid cover melalui email, sosial media, dan acara Travel Fair.

Cluster 2: Pelanggan Berpenghasilan Tinggi dengan Mobilitas Tinggi

- Tawarkan asuransi perjalanan dengan manfaat yang lebih luas, seperti perlindungan gadget, perlindungan terhadap rumah apabila meninggalkan rumah dalam jangka waktu yang cukup lama, dan bantuan hukum jika mengalami masalah hukum di lokasi tujuan.
- Tawarkan perlindungan premium tambahan seperti perlindungan visa, resiko olahraga musim dingin, dan perlindungan kapal pesiar.
- Tawarkan jenis asuransi tahunan, yaitu program asuransi yang dapat menjamin perjalanan sepanjang tahun dengan batasan durasi setiap perjalanannya.
- Iklankan penawaran khusus untuk perjalanan bisnis atau liburan eksklusif melalui email dan sosial media.
- Jalin kerjasama dengan maskapai penerbangan, hotel, dan perusahaan tour & travel untuk menawarkan paket wisata dan asuransi perjalanan yang menarik.
- Buat program loyalitas untuk pelanggan setia yang sering bepergian.

Cluster 3: Pegawai Negeri Sipil dengan Mobilitas Rendah

- Jalin kerjasama dengan instansi pemerintah untuk menawarkan asuransi perjalanan kepada para pegawainya.
- Beri manfaat tambahan, peningkatan cakupan, atau proses klaim lebih mudah yang tidak tersedia dalam asuransi pemerintah.
- Tawarkan premi yang terjangkau.
- Jelaskan manfaat asuransi perjalanan yang relevan dengan kebutuhan mereka, seperti perlindungan perjalanan dinas dan wisata keluarga.

Pembagian Tugas Stage 3

Pembagian tugas di stage ini:

Ana Azzahra : Menulis Laporan

Jerio Benediktus Rumagit : Modeling

Mutiara Citra Sari : Model Evaluation

Ahmad Faqih Ulumuddin: Hyperparameter Tuning

Esa Risa Rouli : Feature Importance

Athiya Fathinati Anindya: Insight and business recommendation

Nicken Shidqia Nurahman : Modeling