Machine Learning (Praktikum) TI –C4

FAKULTTAS VOKASI
UNIVERSITAS AIRLANGGA

[152111283042] | [Nela Anjani] | [9 November 2023]

Import library

```
# import tools
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
```

import numpy as np: Ini mengimpor pustaka NumPy dengan alias 'np'. NumPy digunakan untuk operasi numerik dan komputasi ilmiah.

import pandas as pd: Ini mengimpor pustaka Pandas dengan alias 'pd'. Pandas digunakan untuk manipulasi dan analisis data.

import matplotlib.pyplot as plt: Ini mengimpor pustaka Matplotlib dengan alias 'plt'. Matplotlib digunakan untuk membuat visualisasi dan plot data.

import seaborn as sns: Ini mengimpor pustaka Seaborn dengan alias 'sns'. Seaborn adalah pustaka untuk membuat visualisasi data yang lebih menarik dan informatif.

from sklearn.cluster import KMeans: Ini mengimpor pustaka Scikit-Learn untuk algoritma clustering K-Means.

import data

```
# import data
df = pd.read_csv(r'Mall_Customers.csv')
df.head()
```

ini membaca data dari file CSV dengan nama 'Mall_Customers.csv' dan memuatnya ke dalam sebuah dataframe Pandas yang disebut 'df'.

df.head() : ini menampilkan lima baris pertama dari dataframe df untuk memberikan pandangan awal tentang data

Eksplorasi Data (pengamatan data)

```
df.shape
df.describe()
```

df.shape: Ini mencetak jumlah baris dan kolom dalam dataframe 'df'.

df.describe(): Ini memberikan statistik deskriptif untuk data seperti rata-rata, standar deviasi, kuartil, dll.

Pengecekan data null

```
# cek null data
df.isnull().sum()
```

df.isnull().sum(): Ini menghitung jumlah data kosong (NaN) dalam setiap kolom dataframe 'df' dan mencetak hasilnya. Ini membantu untuk mengetahui apakah ada data yang hilang dalam dataset.

Visualisasi Data

```
# tingkatkan visualisasi data
plt.style.use('fivethirtyeight')
```

plt.style.use('fivethirtyeight'): Ini mengatur gaya plot Matplotlib menjadi "fivethirtyeight", yang merupakan salah satu gaya plot yang telah ditentukan sebelumnya untuk memberikan tampilan yang lebih menarik.

Visualisasi masing masing fitur

```
# amati masing-masing fitur
plt.figure(1 , figsize = (15 , 6))
n = 0
for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
    n += 1
    plt.subplot(1 , 3 , n)
    plt.subplots_adjust(hspace =0.5 , wspace = 0.5)
    sns.distplot(df[x] , bins = 20)
    plt.title('Distplot of {}'.format(x))
    plt.show()
```

plt.figure(1, figsize=(15, 6)): Kode ini membuat sebuah gambar (figure) Matplotlib dengan nomor indeks 1 dan ukuran 15x6 inch. Gambar ini akan berisi subplot-subplot yang menampilkan distribusi masing-masing fitur.

n = 0: Variabel n digunakan untuk melacak indeks subplot yang akan digunakan. Dimulai dari 0.

Loop for x in ['Age', 'Annual Income (k$)', 'Spending Score (1-100)']:: Ini adalah loop yang akan melakukan iterasi untuk setiap fitur dalam list yang diberikan, yaitu 'Age', 'Annual Income (k$)', dan 'Spending Score (1-100)'.

n += 1: Setiap kali loop dijalankan, nilai n akan ditambahkan 1, sehingga menginkremenkan indeks subplot yang digunakan.

plt.subplot(1, 3, n): Kode ini memilih subplot ke-n dalam gambar (1 baris, 3 kolom) untuk menampilkan distribusi fitur yang sedang diproses.

plt.subplots_adjust(hspace=0.5, wspace=0.5): Ini mengatur jarak antara subplot dalam gambar untuk menghindari tumpang tindih.

sns.distplot(df[x], bins=20): Ini menggunakan Seaborn untuk membuat plot distribusi (histogram) dari fitur yang sedang diproses. df[x] adalah kolom dalam dataframe 'df' yang mewakili fitur yang ingin divisualisasikan.

plt.title('Distplot of {}'.format(x)): Kode ini memberikan judul pada subplot yang sesuai dengan fitur yang sedang diproses. Judul ini akan mencakup nama fitur, seperti "Distplot of Age".

plt.show(): Kode ini digunakan untuk menampilkan subplot yang telah selesai diproses. Setelah itu, loop akan melanjutkan ke fitur berikutnya.

Plotting

```
# Ploting untuk mencari relasi antara Age , Annual Income and Spending Score
plt.figure(1 , figsize = (15 , 7))
n = 0
for x in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
    for y in ['Age' , 'Annual Income (k$)' , 'Spending Score (1-100)']:
        n += 1
        plt.subplot(3 , 3 , n)
        plt.subplots_adjust(hspace = 0.5 , wspace = 0.5)
        sns.regplot(x = x , y = y , data = df)
        plt.ylabel(y.split()[0]+' '+y.split()[1] if len(y.split()) > 1 else y
)
        plt.show()
```

kode diatas bertujuan untuk mencari hubungan antara tiga fitur yang berbeda dalam dataset, yaitu 'Age' (Usia), 'Annual Income (k$)' (Pendapatan Tahunan), dan 'Spending Score (1-100)' (Skor Pengeluaran), dengan cara membuat sejumlah scatter plots menggunakan Seaborn untuk menganalisis hubungan di antara mereka.

Plot age dan annual income

```
# plot Age dan Annual Income
plt.figure(1 , figsize = (15 , 6))
for gender in ['Male' , 'Female']:
    plt.scatter(x = 'Annual Income (k$)',y = 'Spending Score (1-100)' ,
    data = df[df['Gender'] == gender] ,s = 200 , alpha = 0.5 ,
    label = gender)
    plt.xlabel('Annual Income (k$)'), plt.ylabel('Spending Score (1-100)')
    plt.title('Annual Income vs Spending Score')
    plt.legend()
    plt.show()
```

Kode ini bertujuan untuk membuat plot scatter (sebaran) yang membandingkan 'Annual Income (k$)' (Pendapatan Tahunan) dengan 'Spending Score (1-100)' (Skor Pengeluaran) untuk dua kelompok gender yang berbeda, yaitu 'Male' (Laki-laki) dan 'Female' (Perempuan), dalam dataset. Kode ini juga menggambarkan hubungan antara pendapatan tahunan dan skor pengeluaran, serta membedakan antara laki-laki dan perempuan dengan warna yang berbeda dalam plot.

Merancang K-Means

```
# rancang K-Means untuk spending score vs annual income
# Kmeans, menentukan jumlah kluster dengan elbow
X1 = df[['Annual Income (k$)' , 'Spending Score (1-100)']].iloc[: ,
:].values
inertia = []
for n in range(1 , 11):
    algorithm = (KMeans(n_clusters = n ,init='k-means++', n_init = 10
    ,max_iter=300,
    random_state= 111) )
```

```
    algorithm.fit(X1)
    inertia.append(algorithm.inertia_)
```

Kode ini bertujuan mengelompokkan data berdasarkan dua fitur, yaitu 'Annual Income (k$)' (Pendapatan Tahunan) dan 'Spending Score (1-100)' (Skor Pengeluaran). Selain itu, kode ini juga digunakan untuk menentukan jumlah kluster yang optimal menggunakan metode "elbow."

Plot elbow

```
# plot elbow
plt.figure(1 , figsize = (15 ,6))
plt.plot(np.arange(1 , 11) , inertia , 'o')
plt.plot(np.arange(1 , 11) , inertia , '-' , alpha = 0.5)
plt.xlabel('Number of Clusters') , plt.ylabel('Inertia')
plt.show()
```

Kode ini bertujuan untuk membuat plot "elbow" (siku) yang membantu menentukan jumlah kluster yang optimal untuk model K-Means. Plot elbow digunakan untuk mencari titik di mana penurunan inersia berkurang drastis, dan ini menandakan jumlah kluster yang optimal.

Membagun model k-means

```
# bangun K-Means
algorithm = (KMeans(n_clusters = 5 ,init='k-means++', n_init = 10
,max_iter=300,
tol=0.0001, random_state= 111 , algorithm='elkan') )
algorithm.fit(X1)
labels2 = algorithm.labels_
centroids2 = algorithm.cluster_centers_
```

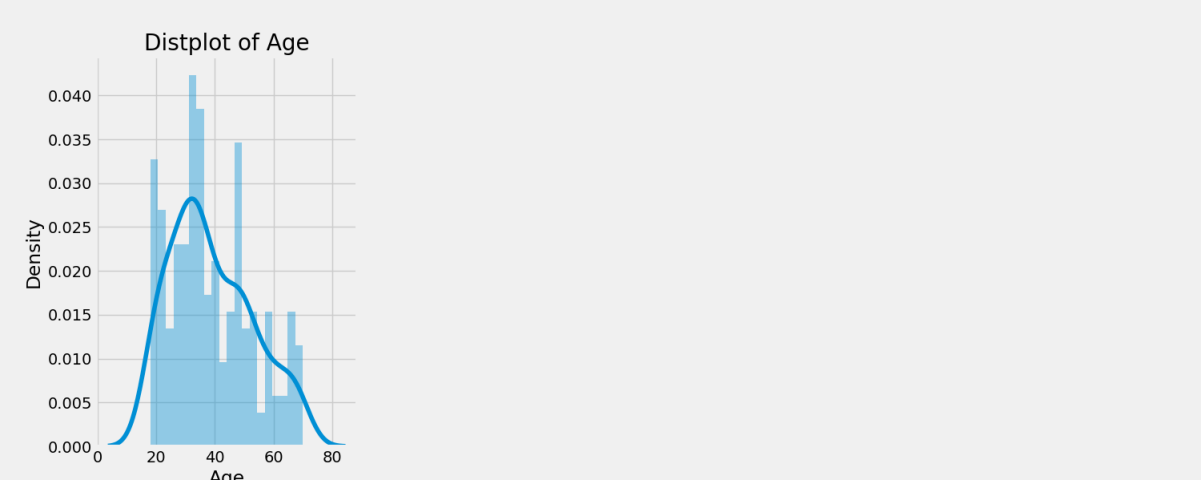
Kode ini digunakan untuk membangun model K-Means dengan parameter yang telah ditentukan.Dengan menghasilkan label kluster dan pusat kluster, dapat menggunakan informasi ini untuk melakukan analisis lebih lanjut, seperti memahami bagaimana data terkelompok dalam kluster atau melakukan prediksi ke kluster mana suatu data tertentu akan masuk.

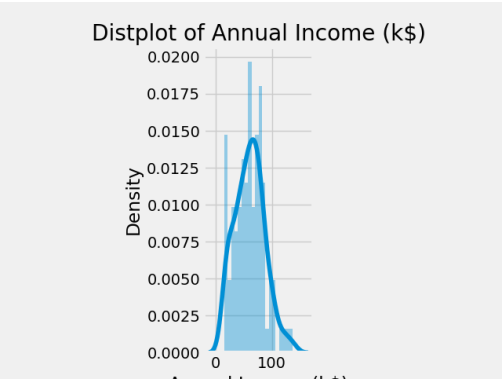
```
# siapkan data untuk plot dan imshow
labels2 = algorithm.labels_
centroids2 = algorithm.cluster_centers_
step = 0.02
x_min, x_max = X1[:, 0].min() - 1, X1[:, 0].max() + 1
y_min, y_max = X1[:, 1].min() - 1, X1[:, 1].max() + 1
xx, yy = np.meshgrid(np.arange(x_min, x_max, step), np.arange(y_min,
y_max, step))
Z1 = algorithm.predict(np.c_[xx.ravel(), yy.ravel()]) # array diratakan 1D
```

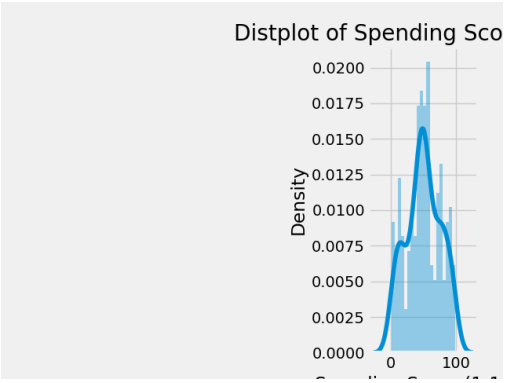
```
plt.figure(1 , figsize = (15 , 7) )
plt.clf()
Z1 = Z1.reshape(xx.shape)
plt.imshow(Z1 , interpolation='nearest',
extent=(xx.min(), xx.max(), yy.min(), yy.max()),
cmap = plt.cm.Pastel2, aspect = 'auto', origin='lower')
plt.scatter( x = 'Annual Income (k$)' ,y = 'Spending Score (1-100)' , data
= df , c = labels2 ,
s = 200 )
plt.scatter(x = centroids2[: , 0] , y = centroids2[: , 1] , s = 300 , c =
'red' ,
alpha = 0.5)
plt.ylabel('Spending Score (1-100)') , plt.xlabel('Annual Income (k$)')
plt.show()
```

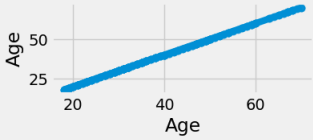
Kode ini bertujuan untuk menampilkan visualisasi dari hasil clustering menggunakan model K-Means yang telah dibangun sebelumnya.

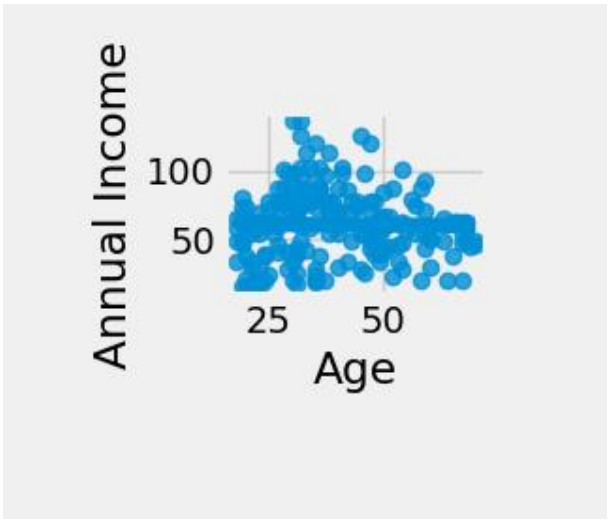
Hasil

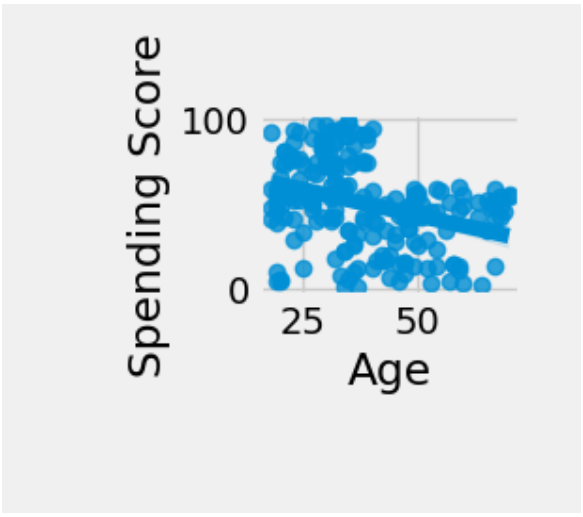


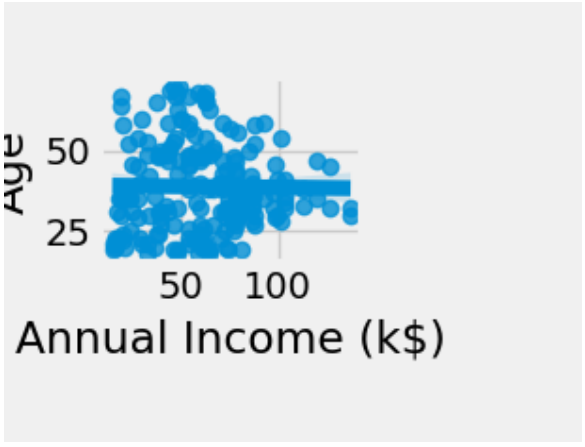


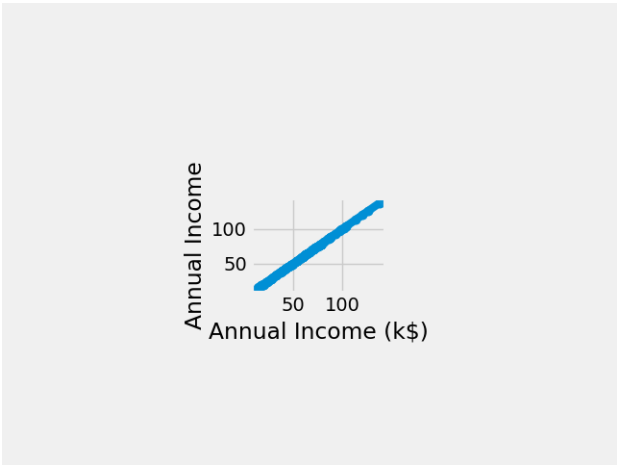


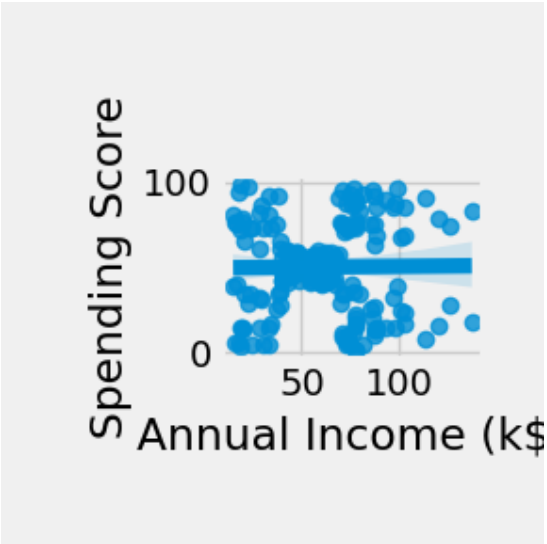

Age
50
25
20
40
60
Age

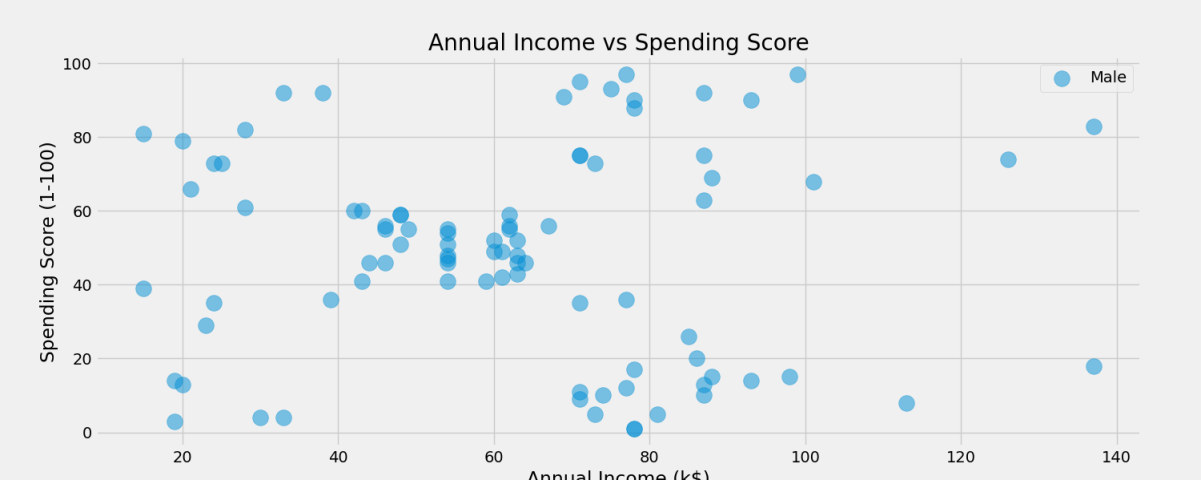

Annual Income
100
50
25
50
Age

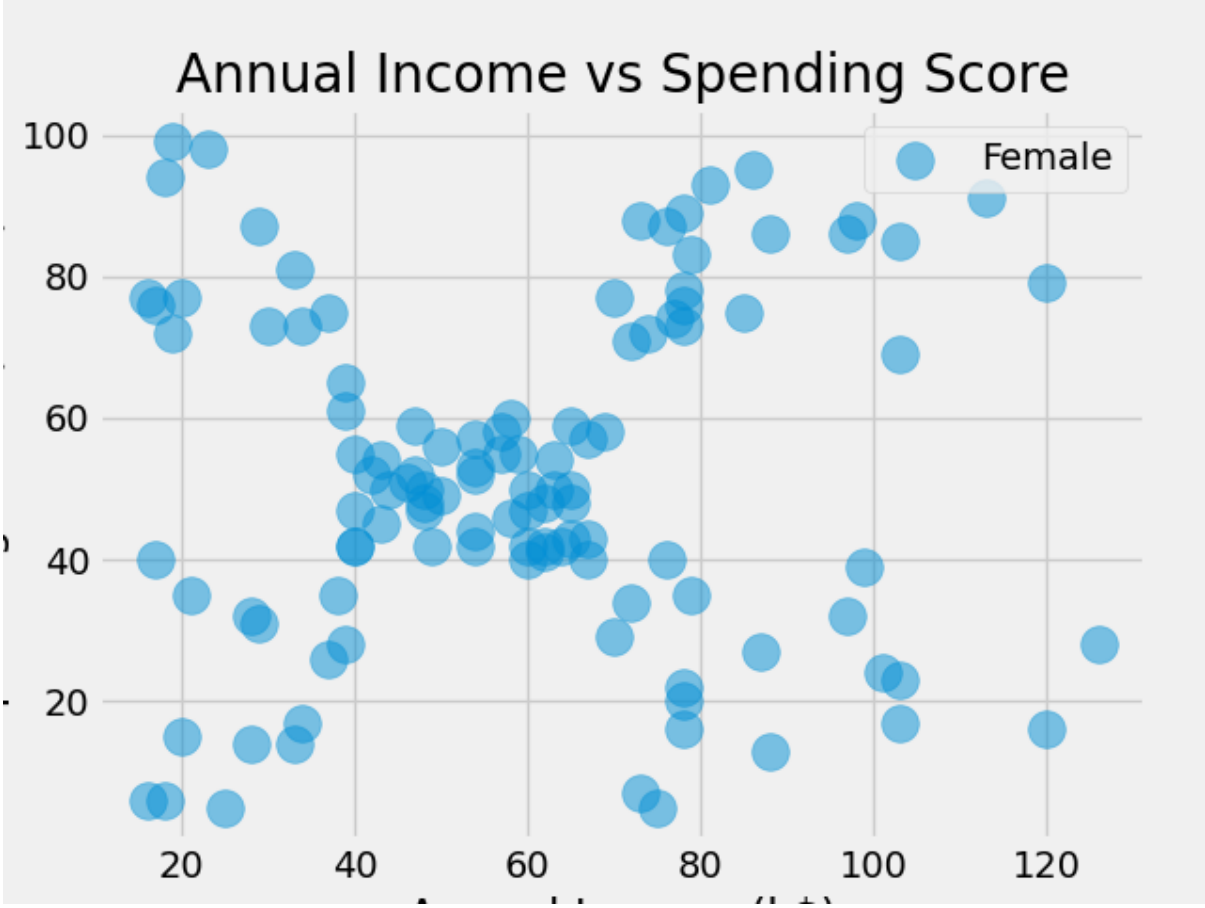

Spending Score
100
0
25
50
Age

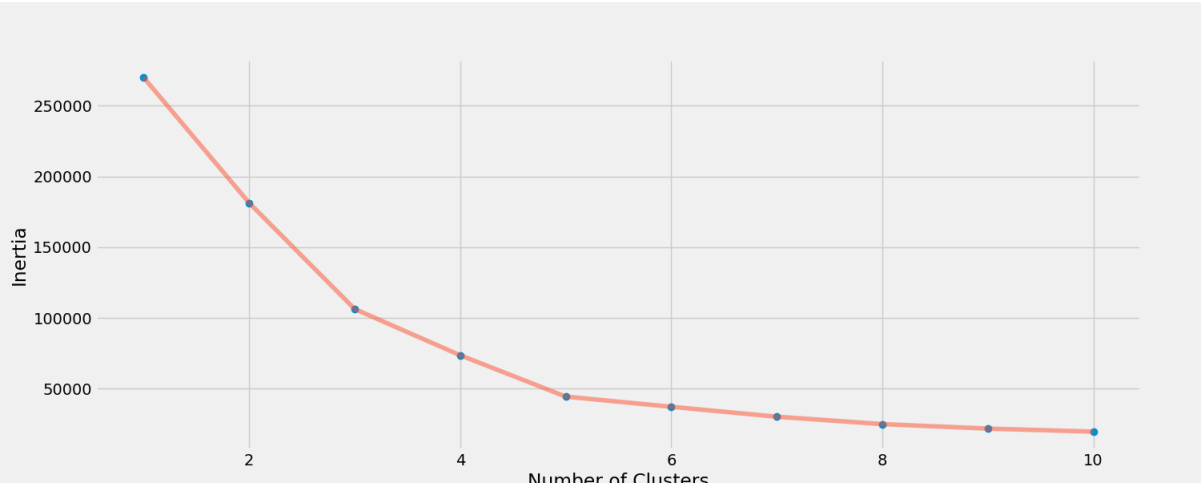

Age
50
25
50
100
Annual Income (k$)

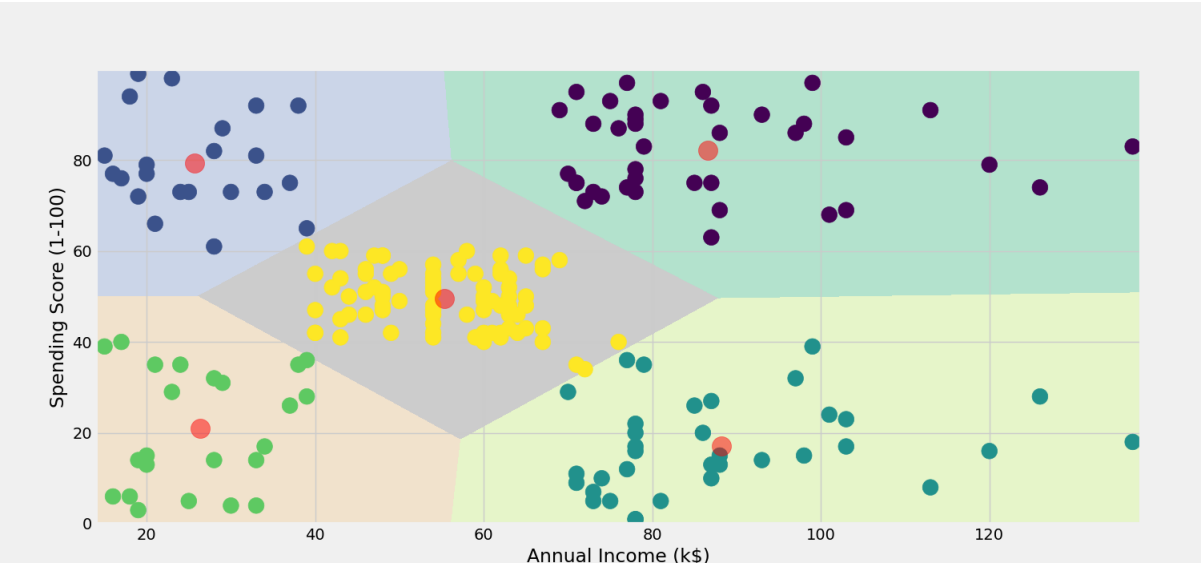

Annual Income
100
50
50
100
Annual Income (k$)


Spending Score
100
0
50
100
Annual Income (k$


Annual Income vs Spending Score
Male
Spending Score (1-100)
Annual Income (k$)
0
20
40
60
80
100
120
140


Annual Income vs Spending Score
Female
100
80
60
40
20
20
40
60
80
100
120


250000
200000
150000
100000
50000
Inertia
2
4
6
8
10
Number of Clusters


80
60
40
20
0
Spending Score (1-100)
20
40
60
80
100
120
Annual Income (k$)