NIM	: 2111521015	Tanggal	: Minggu, 07 Mei 2023
Nama	: Muhammad Irsyadul Fikri	Asisten	: 1. Boby Darmawan
			2. Dwisuci Insani Karimah
			3. Iqbal Fitrahul Ramadhan
			4. Iqbal Manazil Yuni
			5. Muhammad Afif
			6. M. Rayhan Rizaldi
Mata Kuliah	: Praktikum Data Mining		
Modul	: 07		
Kelas	: 01		

Resume dan Tugas "K-Means"

Pengertian

K-means merupakan salah satu algoritma yang bersifat unsupervised learning. K-Means memiliki fungsi untuk mengelompokkan data kedalam data cluster. Algoritma ini dapat menerima data tanpa ada label kategori. K-Means Clustering Algoritma juga merupakan metode non-hierarchy. Metode Clustering Algoritma adalah mengelompokkan beberapa data ke dalam kelompok yang menjelaskan data dalam satu kelompok memiliki karakteristik yang sama dan memiliki karakteristik yang berbeda dengan data yang ada di kelompok lain. Cluster Sampling adalah teknik pengambilan sampel di mana unit-unit populasi dipilih secara acak dari kelompok yang sudah ada yang disebut 'cluster, nah Clustering atau klasterisasi adalah salah satu masalah yang menggunakan teknik *unsupervised learning*.

K-Means Clustering adalah suatu metode penganalisaan data atau metode Data Mining yang melakukan proses pemodelan unssupervised learning dan menggunakan metode yang mengelompokan data berbagai partisi.

K Means Clustering memiliki objective yaitu meminimalisasi object function yang telah di atur pada proses clasterisasi. Dengan cara minimalisasi variasi antar 1 cluster dengan maksimalisasi variasi dengan data di cluster lainnya.

K means clustering merupakan metode algoritma dasar,yang diterapkan sebagai berikut

- Menentukan jumlah cluster
- Secara acak mendistribusikan data cluster
- Menghitung rata rata dari data yang ada di cluster.
- Menggunakan langkah baris 3 kembali sesuai nilai treshold
- Menghitung jarak antara data dan nilai centroid(K means clustering)
- Distance space dapat diimplementasikan untuk menghitung jarak data dan centroid. Contoh penghitungan jarak yang sering digunakan adalah manhattan/city blok distance

Tujuan

Clustering Algoritma (K-Means) memiliki tujuan untuk meminimalisasikan fungsi objective yang telah di set dalam proses clustering. Tujuan tersebut dilakukan dengan cara meminimalikan variasi data yang ada didalam cluster dan memaksimalikan variasi data yang ada di cluster lainnya.

Contoh Clustering Algoritma:

– Segmentasi customer bank atau segmentasi berita-berita online.

– Menentukan Parameter Jumlah data, Cluster, dan Atribut dalam penjurusan Siswa

Karakteristik dari K-Means Cluster:

- Cepat dalam proses clustering
- Sensitif terhadap nilai centroid
- Hasil dari Kmeans selalu berubah ubah(dikarenakan tidak unik)
- Sulit meraih global optimum

Kekurangan dari K-Means clustering

- cluster model berbeda ditemukan
- sulit untuk memilih jumlah cluster yang tepat
- Overlapping
- Kegagalan dalam konverge

Penerapan K-means dengan codingan python

Dengan menggunakan dataset family_income_expand.csv, praktikan diminta untuk
1. Import dataset family_income_expand.csv.

2. Lakukan data cleaning (jika diperlukan).

3. Tentukan dua variabel feature yang akan digunakan. Praktikan harus memilih kolom Total Household Income dan memilih salah satu kolom diantara kolom berikut:

● Total Food Expenditure

● Bread and Cereals Expenditure

● Total Rice Expenditure

● Meat Expenditure

● Total Fish and marine products Expenditure

● Fruit Expenditure

● Vegetables Expenditure

● Restaurant and hotels Expenditure

● Clothing, Footwear and Other Wear Expenditure

4. Tentukan nilai K.

5. Membuat model Algoritma K-Means.

6. Lakukan visualisasi cluster yang terbentuk menggunakan scatter plot.

7. Buat sebuah analisis yang menjelaskan baik itu proses dan output dari tugas yang dikerjakan (Jelaskan dengan rinci).

```
In [34]: import pandas as pd
         df = pd.read_csv('family_income_expend.csv', delimiter = ';')
         df.head()
```

Out[34]:

	Total Household Income	Region	Total Food Expenditure	Main Source of Income	Agricultural Household indicator	Bread and Cereals Expenditure	Total Rice Expenditure	Meat Expenditure	Total Fish and marine products Expenditure	Fruit Expenditure	...	Clothing, Footwear and Other Wear Expenditure	Housing and water Expenditure
0	480332	CAR	117848	Wage/Salaries	0	42140	38300	24676	16806	3325	...	4607	63636
1	198235	CAR	67766	Wage/Salaries	0	17329	13008	17434	11073	2035	...	8230	41370
2	82785	CAR	61609	Wage/Salaries	1	34182	32001	7783	2590	1730	...	2735	14340
3	107589	CAR	78189	Wage/Salaries	0	34030	28659	10914	10812	690	...	1390	16638
4	189322	CAR	94625	Wage/Salaries	0	34820	30167	18391	11309	1395	...	4620	31122

5 rows × 24 columns

```
In [35]: dt = df[['Total Household Income','Total Rice Expenditure']]
         dt.head()
```

Out[35]:

	Total Household Income	Total Rice Expenditure
0	480332	38300
1	198235	13008
2	82785	32001
3	107589	28659
4	189322	30167

```
In [36]: dt.isnull().sum()
```

Out[36]:
```
Total Household Income    0
Total Rice Expenditure    0
dtype: int64
```

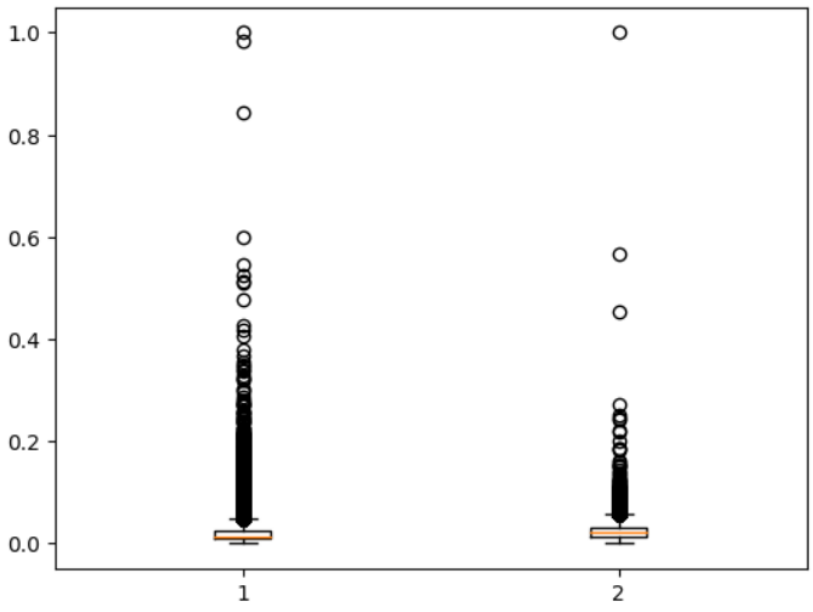
In [37]:
```
data = dt
data['Total Household Income'] = data['Total Household Income'] / data['Total
data['Total Rice Expenditure'] = data['Total Rice Expenditure'] / data['Total
data
```

Out[37]:

	Total Household Income	Total Rice Expenditure
0	0.040651	0.050506
1	0.016777	0.017154
2	0.007006	0.042200
3	0.009105	0.037792
4	0.016023	0.039781
...	...	...
41539	0.010137	0.028407
41540	0.011622	0.001679
41541	0.011270	0.036052
41542	0.010960	0.035150
41543	0.010883	0.054337

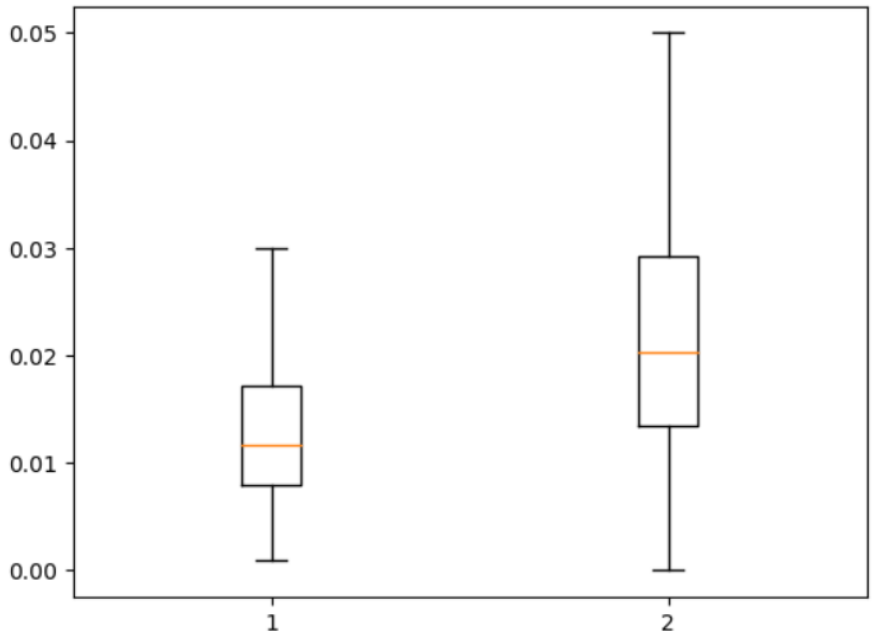
In [38]:
```
import matplotlib.pyplot as plt
plt.boxplot(dt)
plt.show
```

Out[38]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [52]:
```
data = data[data['Total Household Income']<0.03]
data = data[data['Total Rice Expenditure']<0.05]
import matplotlib.pyplot as plt
plt.boxplot(data)
plt.show
```

Out[52]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [58]:
```
x = data.iloc[100:1000, [0, 1]].values
x
```

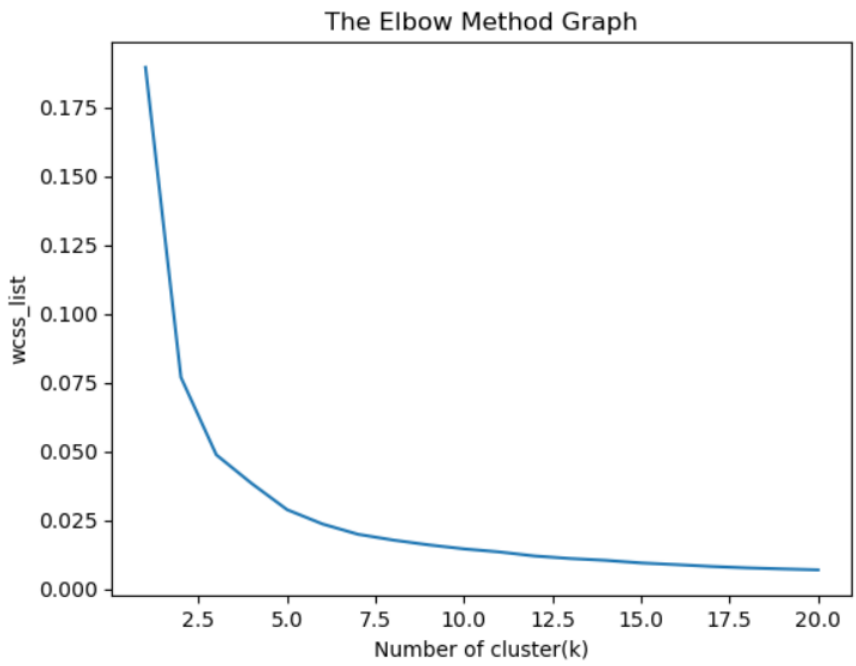
Out[58]:
```
array([[0.00657034, 0.01712324],
       [0.00944229, 0.03042491],
       [0.0102539 , 0.02703323],
       ...,
       [0.01681493, 0.00601324],
       [0.00865395, 0.0180582 ],
       [0.02454953, 0.01174956]])
```

In [59]:
```
import numpy as nm
import matplotlib.pyplot as mtp
```

In [60]:
```
import warnings
warnings.filterwarnings('ignore')
from sklearn.cluster import KMeans
wcss_list = []

for i in range(1, 21):
    kmeans = KMeans(n_clusters=i, init='k-means++', random_state=42)
    kmeans.fit(x)
    wcss_list.append(kmeans.inertia_)
mtp.plot(range(1, 21), wcss_list)
mtp.title('The Elbow Method Graph')
mtp.xlabel('Number of cluster(k)')
mtp.ylabel('wcss_list')
mtp.show
```

Out[60]: <function matplotlib.pyplot.show(close=None, block=None)>



In [61]:
```
kmeans = KMeans(n_clusters=6, init='k-means++', random_state=42)
y_predict =  kmeans.fit_predict(x)
```

In [62]:
```
mtp.scatter(x[y_predict == 0, 0], x[y_predict == 0, 1], s=50, c= 'blue', label
mtp.scatter(x[y_predict == 1, 0], x[y_predict == 1, 1], s=50, c= 'green', labe
mtp.scatter(x[y_predict == 2, 0], x[y_predict == 2, 1], s=50, c= 'red', label
mtp.scatter(x[y_predict == 3, 0], x[y_predict == 3, 1], s=50, c= 'cyan', label
mtp.scatter(x[y_predict == 4, 0], x[y_predict == 4, 1], s=50, c= 'magenta', la
mtp.scatter(x[y_predict == 5, 0], x[y_predict == 5, 1], s=50, c= 'maroon', lab
mtp.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s =
mtp.title('Clusters Of Familiy')
mtp.xlabel('Total Household Income')
mtp.ylabel('Total Rice Expenditure')
mtp.legend()
mtp.show()
```