Praktikum Data Preprocessing

Praktikum Cluster Analysis

Ulima Inas Shabrina(2110181048)

Assignment

1. dataset ← transaction.csv, dan tampilkan
2. country ← berapa kemunculan tiap negeri pada dataset, dan tampilkan
3. transaksi ← hitunglah banyaknya rata-rata jumlah barang (Qty) per transaksi pada tiap negara (1 kode InvoiceNo = 1 transaksi)
4. cluster_i[1-10], cluster_val[1-10] ← lakukan clustering pada transaksi dengan K-Means, dengan k=3, sebanyak 10 kali. Setiap kali selesai clustering, lakukan cluster analysis dengan SSE.
5. cluster ← ambil cluster_i yang mempunyai cluster_val terkecil
6. centroid ← tentukan posisi centroid dari setiap cluster
7. sorted ← lakukan pengurutan posisi centroid secara ascending
8. Indeks terdepan dari centroid setelah pengurutan, mengindikasikan cluster transanksi rendah. Indeks terakhir dari centroid setelah pengurutan, mengindikasikan cluster transanksi tinggi. Indeks di antaranya, mengindikasikan cluster transanksi sedang. Tampilkan negara mana saja yang transaksinya rendah, sedang dan tinggi.
9. Visualisasi dengan warna yang berbeda untuk hasil cluster (no. 8), dimana sumbu x=urutan country dan sumbu y=transaksi

In [1]:
```
import pandas as pd
import numpy as np
#1
from sklearn.cluster import KMeans
dataset = pd.read_csv('transaction.csv')
dataset
```

Out[1]:

	InvoiceNo	StockCode	Qty	InvoiceDate	CustomerID	Country
0	537626	22725	830	12/7/2010 14:57	12347	Iceland
1	537626	22729	948	12/7/2010 14:57	12347	Iceland
2	537626	22195	695	12/7/2010 14:57	12347	Iceland
3	542237	22725	636	1/26/2011 14:30	12347	Iceland
4	542237	22729	536	1/26/2011 14:30	12347	Iceland
...	...	...	...	...	...	...
10541	543911	21700	455	2/14/2011 12:46	17829	United Arab Emirates
10542	543911	22111	578	2/14/2011 12:46	17829	United Arab Emirates
10543	543911	22112	163	2/14/2011 12:46	17829	United Arab Emirates
10544	564428	23296	545	8/25/2011 11:27	17844	Canada
10545	564428	23294	643	8/25/2011 11:27	17844	Canada

10546 rows × 6 columns

In [2]:
```
country = pd.DataFrame(dataset['Country'].value_counts())
country = country.sort_index()
country
```

Out[2]:

	Country
Australia	356
Austria	88
Bahrain	3
Belgium	486
Brazil	8
Canada	36
Channel Islands	184
Cyprus	113
Czech Republic	4
Denmark	98
EIRE	1620
European Community	5
Finland	152
France	2109
Germany	2269
Greece	33
Iceland	35
Israel	61
Italy	190
Japan	92
Lebanon	5
Lithuania	8
Malta	15
Netherlands	634
Norway	239
Poland	80
Portugal	367
RSA	14
Saudi Arabia	1
Singapore	45
Spain	539
Sweden	109
Switzerland	434
USA	47
United Arab Emirates	23
Unspecified	44

In [3]:
```
invoice_qty = pd.DataFrame(dataset.groupby(['InvoiceNo'])['Qty'].sum())
invoice_qty
```

Out[3]:

InvoiceNo	Qty
536370	5133
536389	2800
536527	4176
536532	10000
536540	1976
...	...
581494	3791
581570	1063
581574	1361
581578	5470
581587	1395

1565 rows × 1 columns

In [4]:
```
invoice_country = dataset.drop_duplicates(subset = 'InvoiceNo', keep = 'first')
invoice_country = invoice_country.set_index('InvoiceNo')
invoice_country = invoice_country[['Country']]
invoice_country
```

Out[4]:

InvoiceNo	Country
537626	Iceland
542237	Iceland
549222	Iceland
556201	Iceland
562032	Iceland
...	...
559557	Canada
545579	Greece
555931	Malta
543911	United Arab Emirates
564428	Canada

1565 rows × 1 columns

In [5]:
```
invoice_qty_country = pd.concat([invoice_qty, invoice_country], axis = 1)
invoice_qty_country
```

Out[5]:

InvoiceNo	Qty	Country
536370	5133	France
536389	2800	Australia
536527	4176	Germany
536532	10000	Norway
536540	1976	EIRE
...	...	...
581494	3791	Germany
581570	1063	Germany
581574	1361	Germany
581578	5470	Germany
581587	1395	France

1565 rows × 2 columns

In [6]:
```
transaksi = pd.DataFrame(dataset.groupby(['Country'])['Qty'].mean())
transaksi
```

Out[6]:

Country	Qty
Australia	497.632022
Austria	466.397727
Bahrain	490.000000
Belgium	524.172840
Brazil	548.625000
Canada	537.472222
Channel Islands	521.543478
Cyprus	502.778761
Czech Republic	619.750000
Denmark	554.489796
EIRE	523.892593
European Community	553.000000
Finland	533.342105
France	521.553817
Germany	518.356985
Greece	526.212121
Iceland	560.171429
Israel	462.065574
Italy	519.836842
Japan	470.195652
Lebanon	567.200000
Lithuania	503.250000
Malta	501.133333
Netherlands	527.796530
Norway	548.179916
Poland	537.312500
Portugal	504.697548
RSA	469.714286
Saudi Arabia	592.000000
Singapore	559.000000
Spain	539.035250
Sweden	516.330275
Switzerland	535.119816
USA	531.340426
United Arab Emirates	538.869565
Unspecified	486.295455

In [7]:
```
cluster_i = []
cluster_val = []
for i in range(10):
    kmeans = KMeans(n_clusters = 3, init = 'random', n_init=1, max_iter=5).fit(transaksi)
    cluster_i.append(kmeans)
    cluster_val.append(kmeans.inertia_)
    print(kmeans, kmeans.inertia_)
```

```
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8650.598826148962
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8759.282563946897
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8110.970246046107
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8257.151454676508
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8095.124159634649
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 9099.07839747103
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8143.314803447484
```

In [8]:
```
index_cluster = cluster_val.index(min(cluster_val))
cluster = cluster_i[index_cluster]
print(cluster, cluster.inertia_)
```

```
KMeans(init='random', max_iter=5, n_clusters=3, n_init=1) 8095.124159634649
```

In [9]:
```
centroid = cluster.cluster_centers_
centroid
```

Out[9]:
```
array([[486.74185073],
       [530.49957122],
       [572.23017493]])
```

In [10]:
```
idx = np.argsort(centroid.sum(axis=1))
lut = np.zeros_like(idx)
lut[idx] = np.arange(3)

sorted_centroid = centroid[lut]
sorted_label = lut[cluster.labels_]
print('Centroid', sorted_centroid)
print('Label', sorted_label)
```

```
Centroid [[486.74185073]
 [530.49957122]
 [572.23017493]]
Label [0 0 0 1 1 1 1 0 2 2 1 2 1 1 1 1 2 0 1 0 2 0 0 1 1 1 0 0 2 2 1 1 1 1 1 0]
```

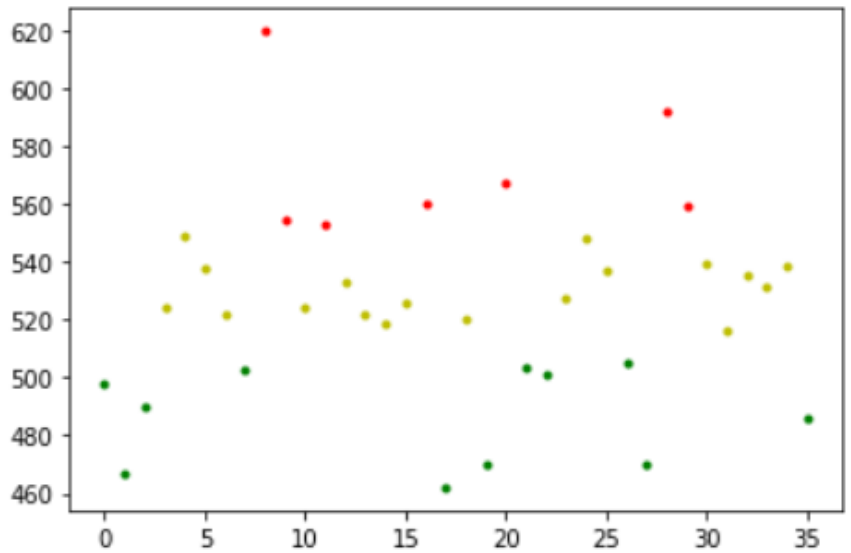
In [11]:
```
label_index_0 = (sorted_label == 0).nonzero()
label_country_0 = transaksi.index[label_index_0]
label_index_1 = (sorted_label == 1).nonzero()
label_country_1 = transaksi.index[label_index_1]
label_index_2 = (sorted_label == 2).nonzero()
label_country_2 = transaksi.index[label_index_2]

print('Transaksi Rendah', label_country_0)
print('Transaksi Sedang', label_country_1)
print('Transaksi Tinggi', label_country_2)
```

```
Transaksi Rendah Index(['Australia', 'Austria', 'Bahrain', 'Cyprus', 'Israel', 'Japan',
       'Lithuania', 'Malta', 'Portugal', 'RSA', 'Unspecified'],
      dtype='object', name='Country')
Transaksi Sedang Index(['Belgium', 'Brazil', 'Canada', 'Channel Islands', 'EIRE', 'Finland',
       'France', 'Germany', 'Greece', 'Italy', 'Netherlands', 'Norway',
       'Poland', 'Spain', 'Sweden', 'Switzerland', 'USA',
       'United Arab Emirates'],
      dtype='object', name='Country')
Transaksi Tinggi Index(['Czech Republic', 'Denmark', 'European Community', 'Iceland', 'Lebanon',
       'Saudi Arabia', 'Singapore'],
      dtype='object', name='Country')
```

In [12]:
```
import matplotlib.pyplot as plt

plt.plot(label_index_2, transaksi.iloc[label_index_2].to_numpy().reshape((1,-1)),'r.')
plt.plot(label_index_1, transaksi.iloc[label_index_1].to_numpy().reshape((1,-1)),'y.')
plt.plot(label_index_0, transaksi.iloc[label_index_0].to_numpy().reshape((1,-1)),'g.')
plt.show()
```



Terimakasih