COURSE FINAL PROJECT REPORT

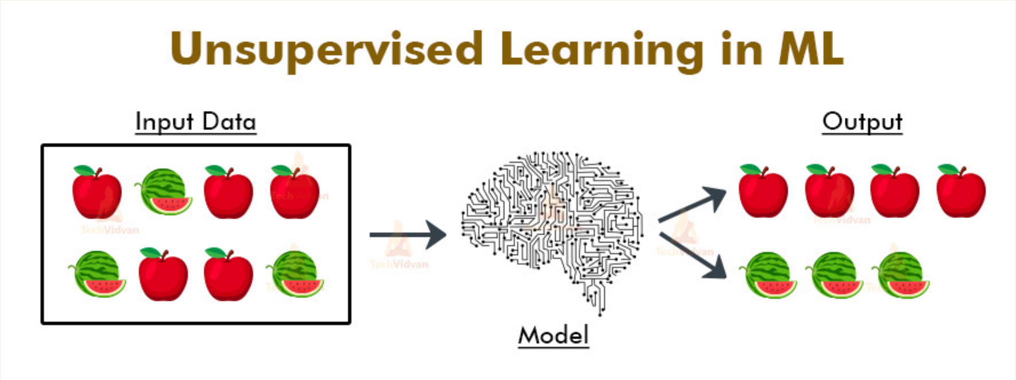
MALL CUSTOMER CLUSTERING

Xu Wang

2022.02.14

Content

- Main objectives of project
- Brief description of data
- Data cleaning & Explanatory data analysis
- Clustering models results
  - *K-means*
  - *Hierarchy agglomerative clustering*
  - *DBSCAN*
  - *Mean shift*
- Summary of models comparison
- Key findings and insights
- Suggestions of next move
- References



Main objectives of project - Mall customer data

- This project will be focused on clustering.
- From the algorithms of clustering comparison, there are several advices and implications will be provided in business prospective, which may help stakeholders for business development and customer retention.
- By the end of this case study, it will achieve customer segmentation, target customer classification with marketing strategies.

Brief description of data - Mall customer data

- The data is from a small part of supermarket mall and through its membership cards data for the purpose of market basket analysis.
- The data has 200 observations (rows) and 5 columns.
- The columns include *Customer ID*, *Gender*, *Age*, *Annual Income and Spending score*.
- *Spending Score* is something you assign to the customer based on your defined parameters like customer behavior and purchasing data.
- Through this data, it will help to **understand the customers like who can be easily converge (valuable targeted customers), so that the sense can be given to marketing team and plan the strategy accordingly.**
- Data source: https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python/version/1

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77

```
data.shape
```

```
(200, 5)
```

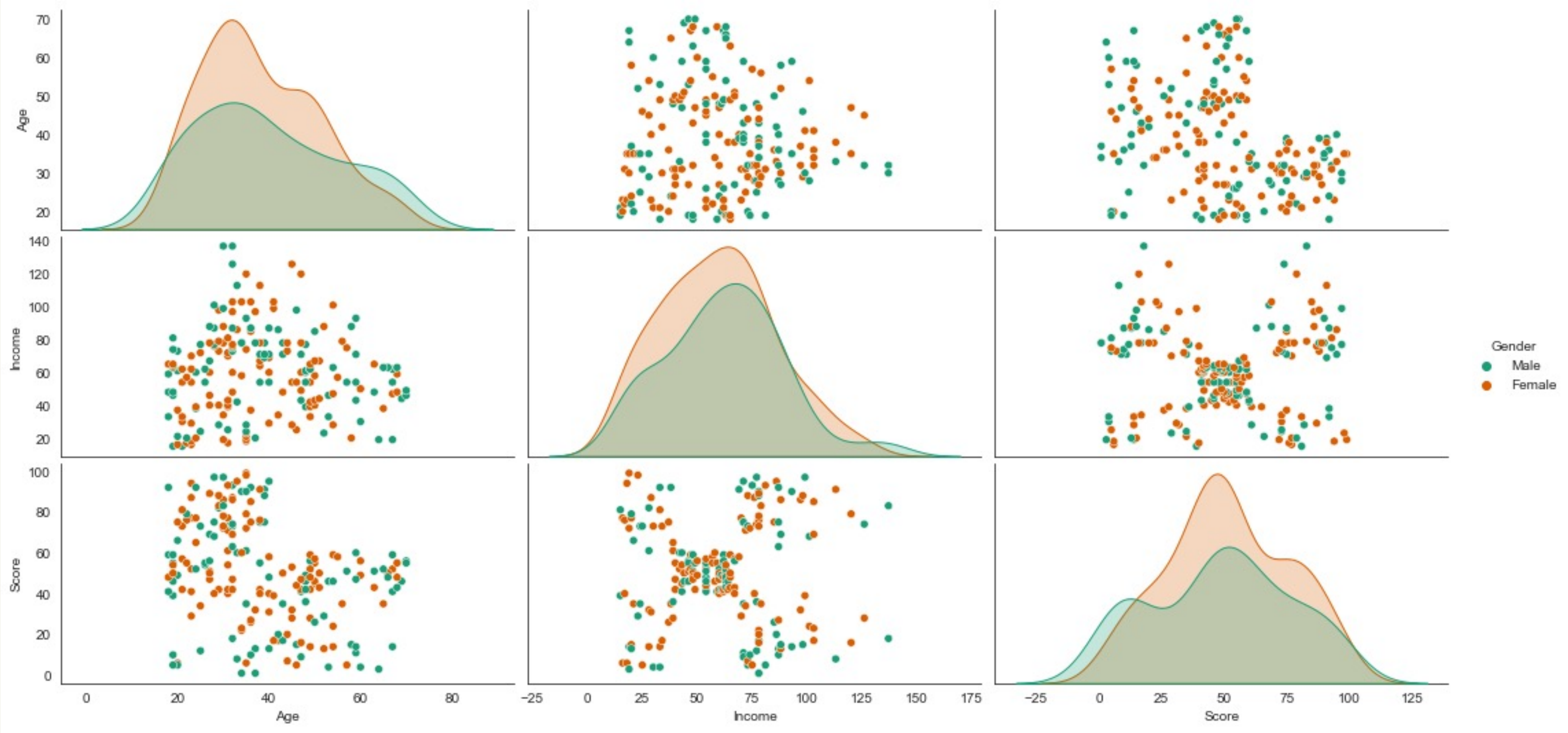
Data cleaning & Explanatory data analysis

- Check null values and Dtypes.
- Understand the columns and rename them.

```
data.rename(index=str, columns={'Annual Income (k$)':'Income',
                                'Spending Score (1-100)':'Score'}, inplace=True)
data
```

	CustomerID	Gender	Age	Income	Score
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

- Check gender distribution of other features & their skewness.
  - *Its distribution is not very clear for clustering. So ‘Gender‘ is irrelevant here.*



- Prepare relevant features for model training.

```
#Since CustomerID and Gender are useless in model training
df = data.drop(['CustomerID','Gender'], axis=1)
df.head()
```

	Age	Income	Score
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40

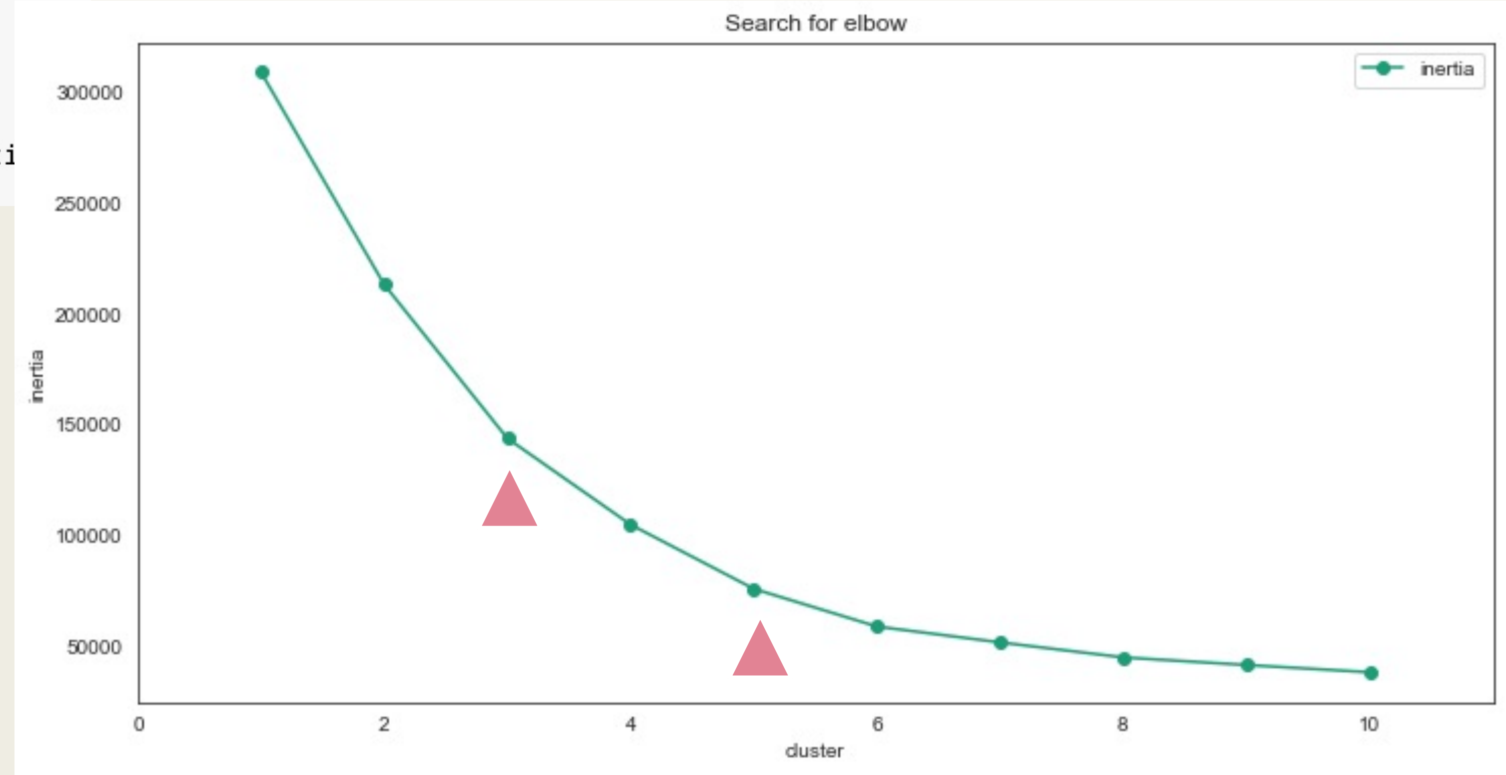
Summary of data cleaning & EDA

- As the data only has 200 customers data which include their gender, income and spending scores, there is no null and noisy values to process.
- We aim to experiment in cluster models training later, so the Customer ID is irrelevant and it will be removed.
- From the above pair plot, Gender also has no direct relation to customer segmentation, as result Gender will be removed as well.

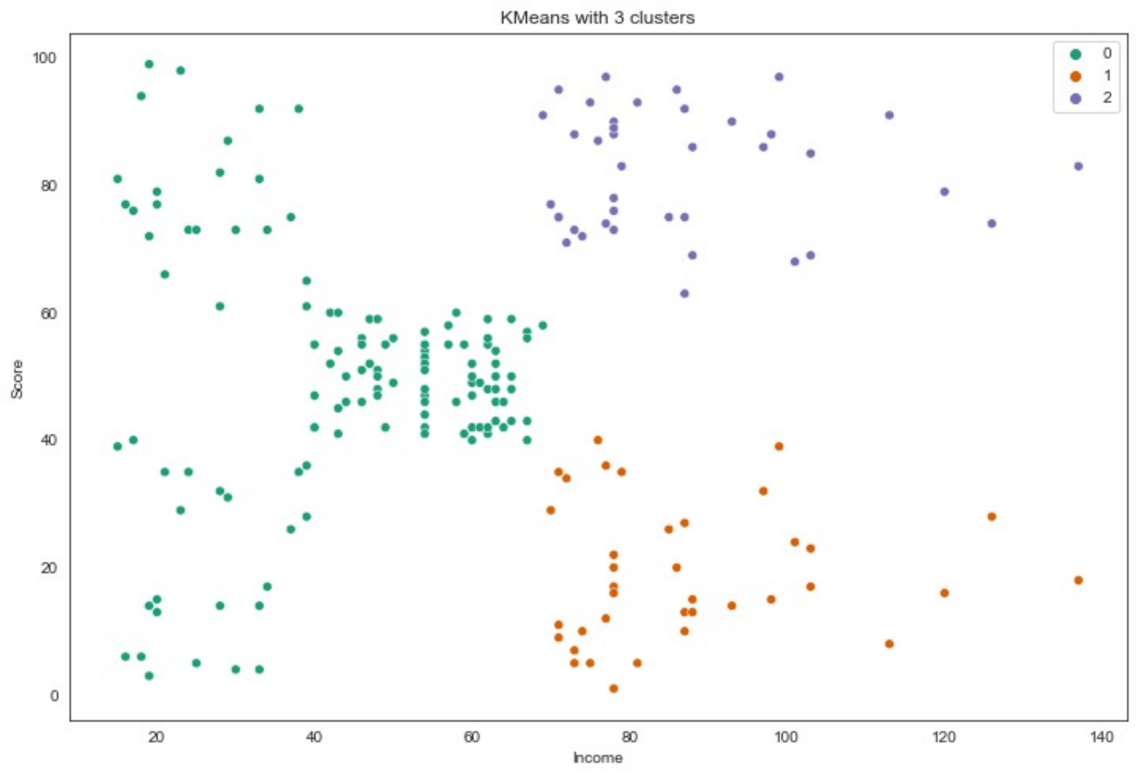
Clustering models - KMeans

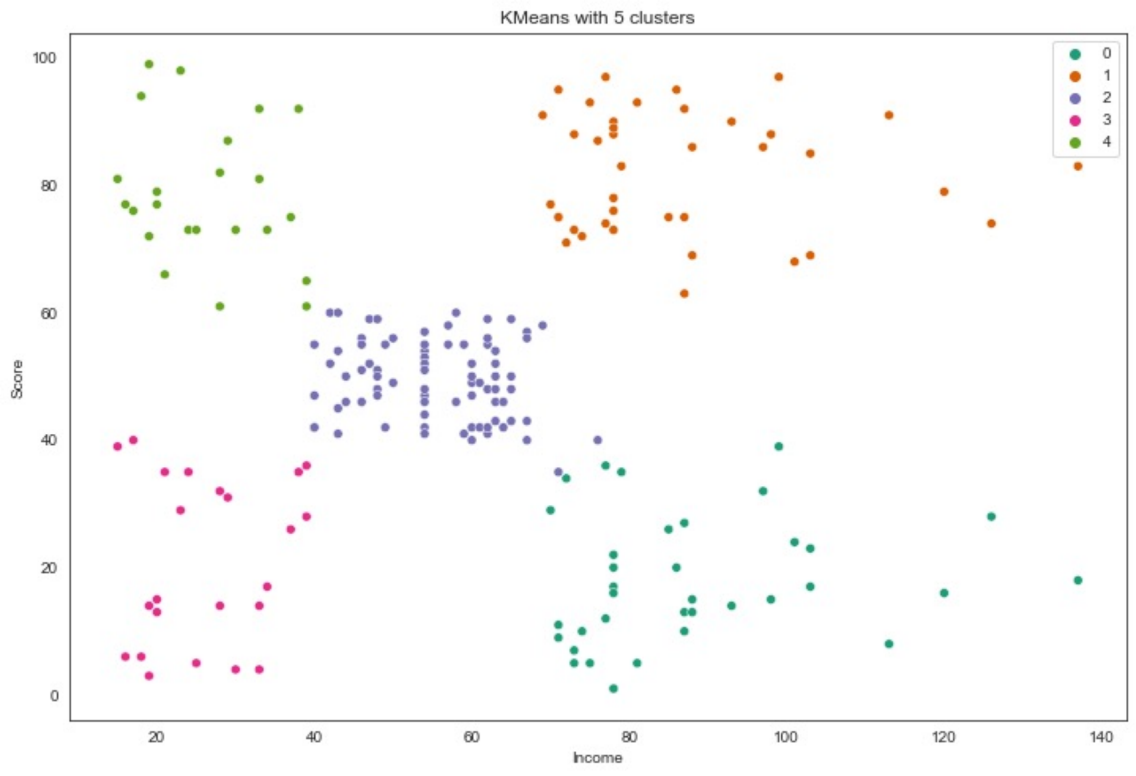
- To find out the ‘right’ k of Kmeans, the inertia shows the elbow point which is very considerable. From the figure, it seems k=3, or k=5 may be better.

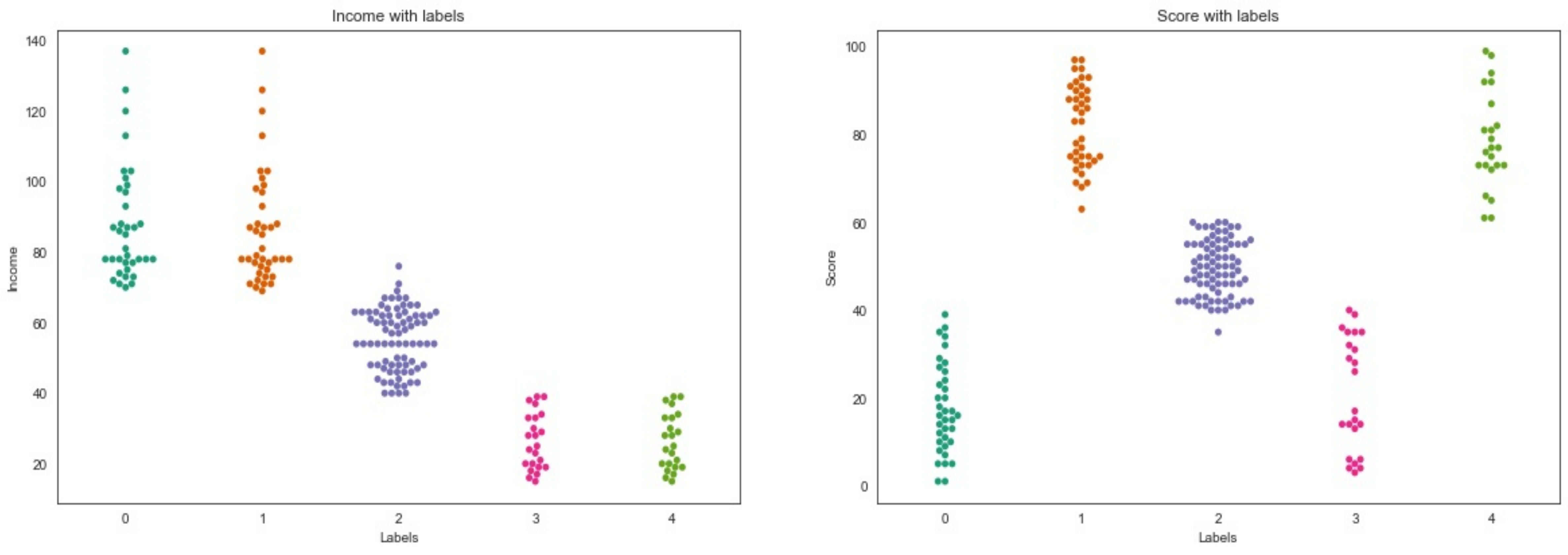
```python
from sklearn.cluster import KMeans

km_list= list()

for i in range(1,11):
    km=KMeans(n_clusters=i, random_state=20)
    km=km.fit(df)
    km_list.append(pd.Series({'cluster':i,
                               'inertia':km.inerti
                              'model':km}))
```



K=5 is clearly better than k=3.







It will be classified as 5 classes of customers:

(score = spending score)

label 0 : customers with high income and low score

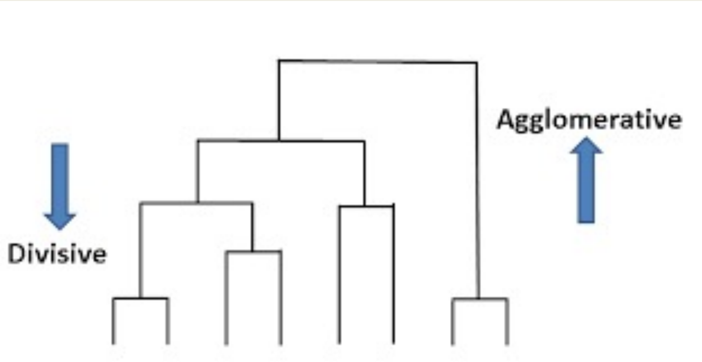
label 1 : customers with high income and high score

label 2 : customers with medium income and medium score

label 3 : customers with low income and low score

label 4 : customers with low income and high score

Clustering models - Hierarchy agglomerative clustering



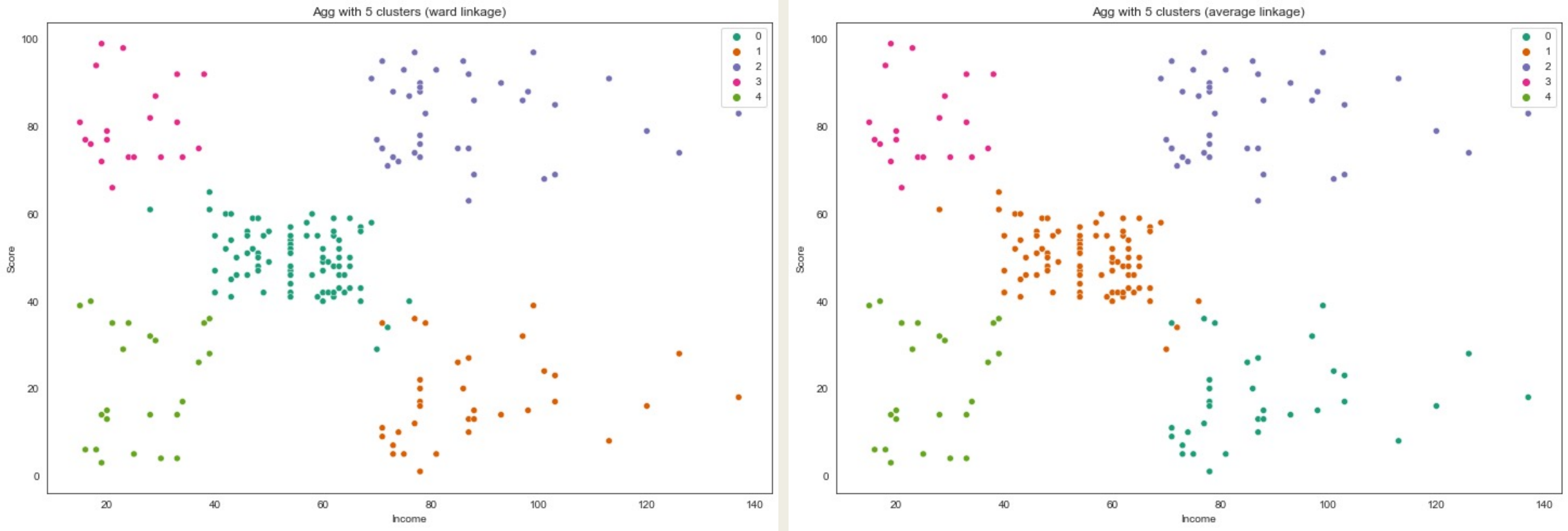
Since K=5,
We will compare the different linkage. - ‘ward’ vs. ‘average’

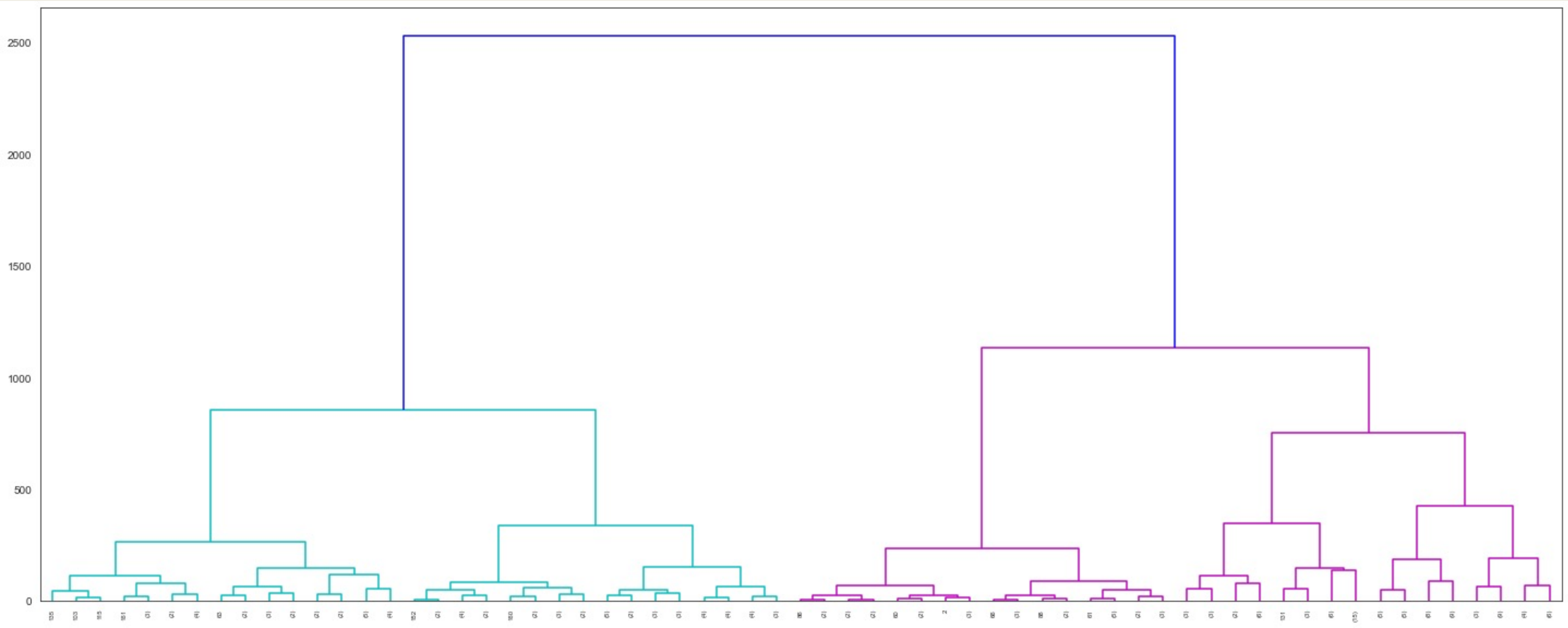
5.2.1 cluster=5 ward linkage

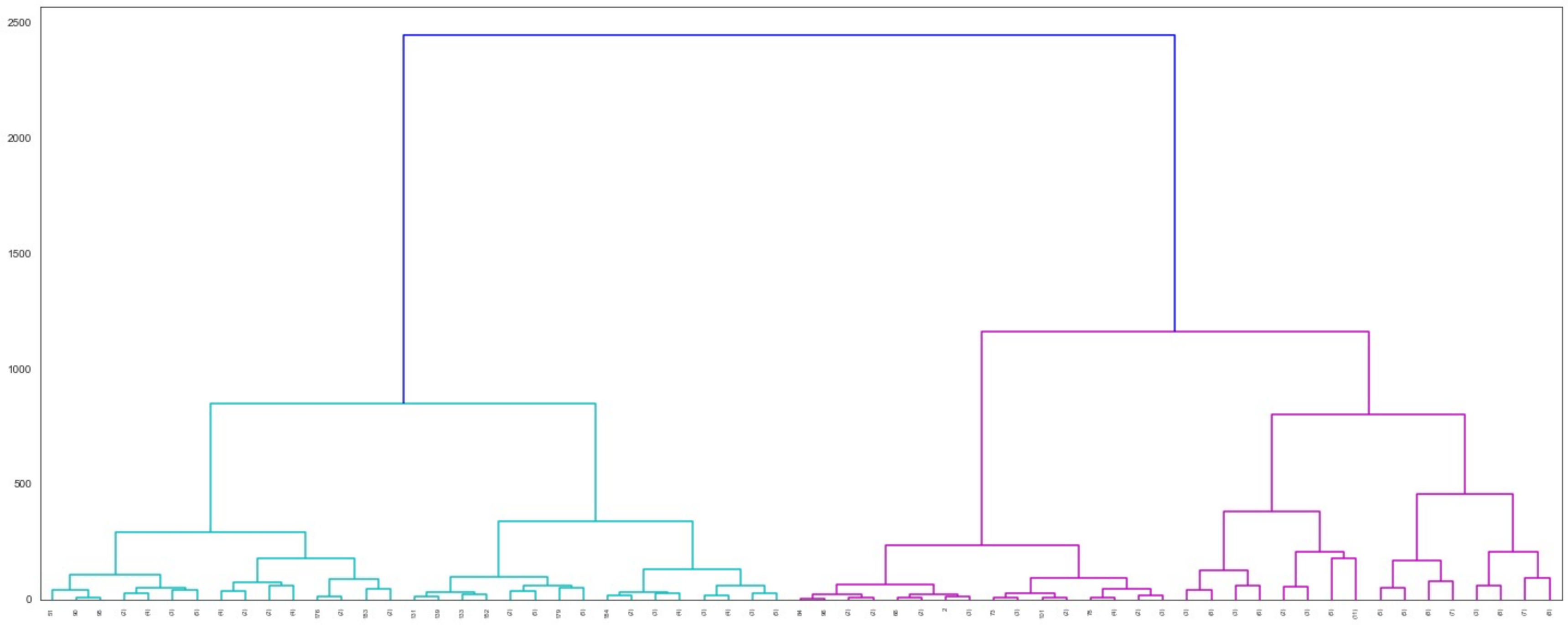
```
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster import hierarchy

ag = AgglomerativeClustering(n_clusters=5, linkage='ward', compute_full_tree=True)
ag = ag.fit(df)
df['Labels'] = ag.labels_
```

‘ward’ vs ‘average’

Difference is minor!







‘ward’

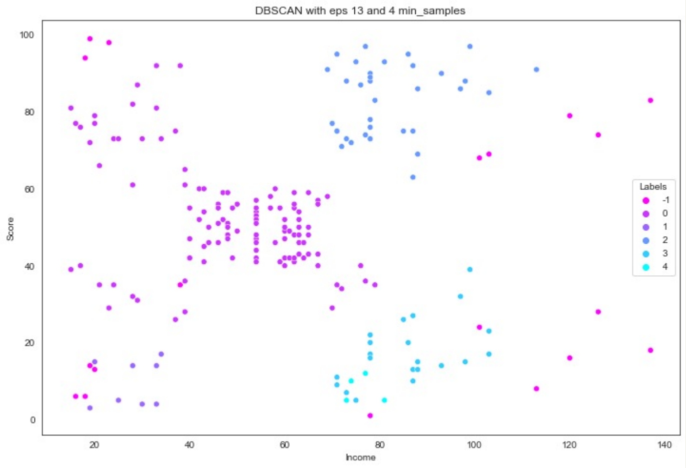
vs

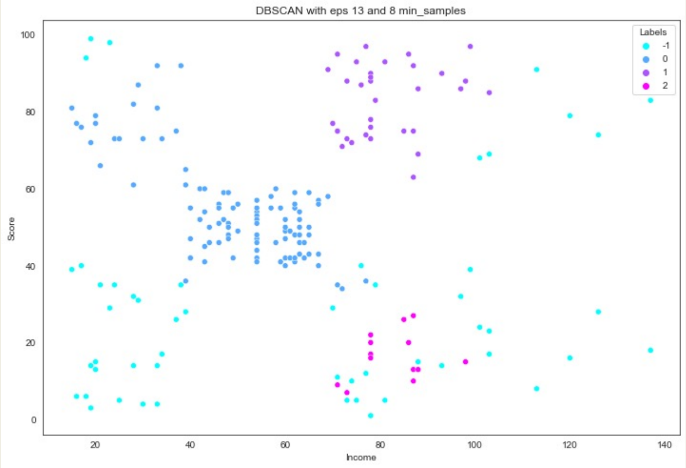
‘average’

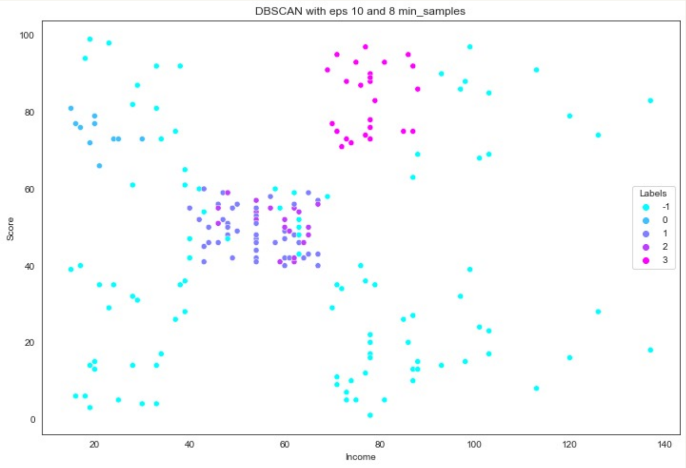
Difference is minor!

Clustering models - DBSCAN

- In DBSCAN, two parameters will be determined. Epsilon & min_samples.
- After trying:
  - *Eps=13, min_samples=4, it will be 6 clusters.*
  - *Eps=13, min_samples=8, it will be 4 clusters.*
  - *Eps=10, min_samples=8, it will be 5 clusters.*







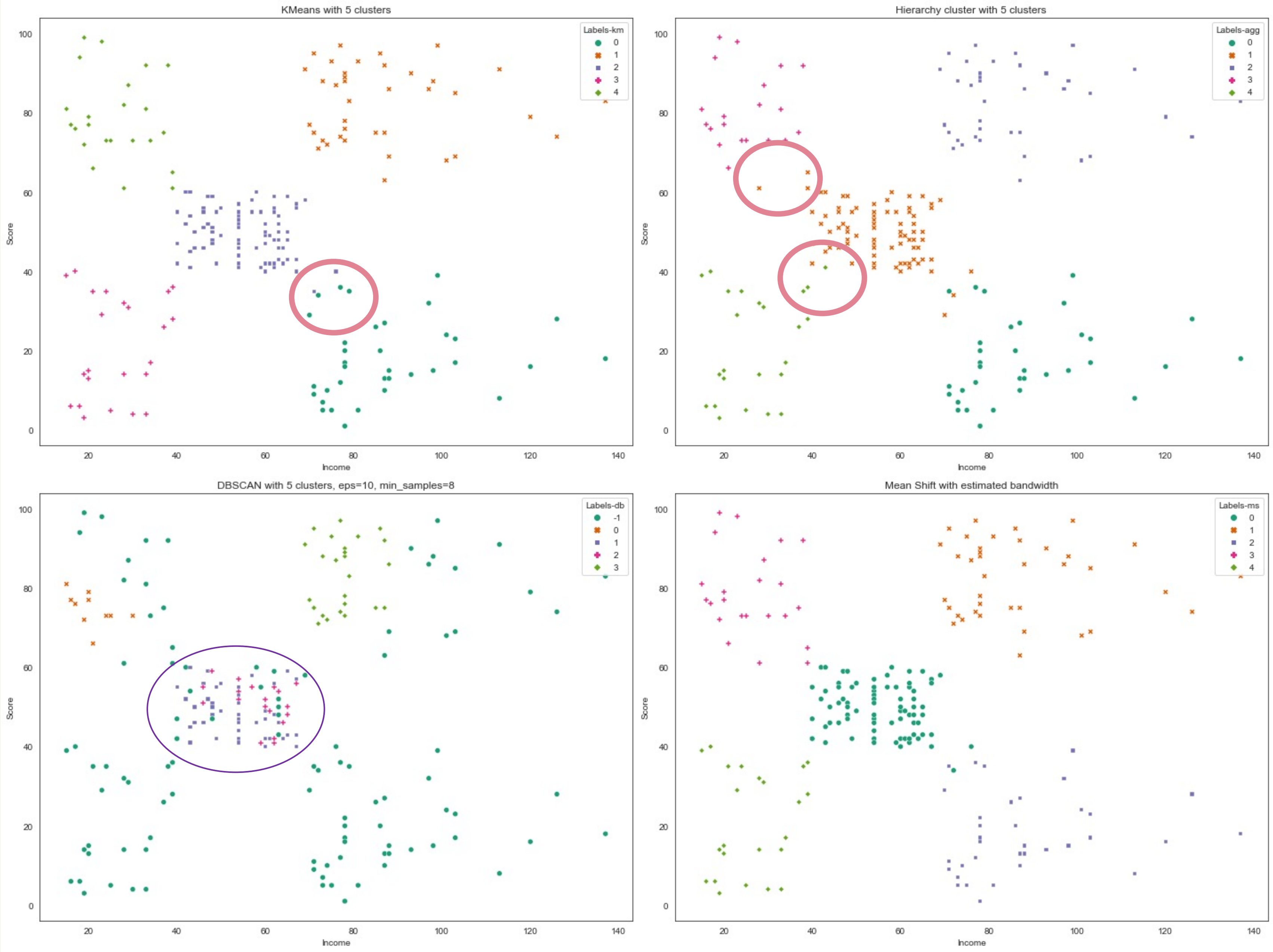
- There are difficulties to find the optimal parameters, as clusters mixed up and many outliers exist. In a business prospective view of concentrating in several valuable customers, this model may provide valuable help for outliers.

Clustering models - Mean Shift

- In Mean Shift, a parameter (bandwidth) will be determined.
- After several try, W=3,7,10,12, their results seemed mixed up.
- From sklearn.cluster, the section named ‘estimate_bandwidth’ helps.
- It turns out w=5 is better!

```
from sklearn.cluster import estimate_bandwidth
df = df.drop(['Labels'],axis=1)
bandwidth = estimate_bandwidth(df, quantile=0.1)
ms = MeanShift(bandwidth).fit(df)
df['Labels'] = ms.labels_

plt.figure(figsize=(12,8))
sns.scatterplot(df['Income'], df['Score'], hue=df['Labels'], palette='cool_r')
plt.title("Mean Shift with bandwidth=5 ")
plt.show()
```

Summary of models comparison



- Kmeans is my preferable model if considering the data size, time consuming and elbom (inertia) point to locate the cluster k.
- Mean Shift is second preferable model when using the 'estimation_bandwidth' to estimate automatically w, and quickly find out the clusters.
- When using the elbow point (3 or 5), it is easy to use hierarchy clustering to find out the details of hierarchy structure of data.
- DBSCAN is helpful for outliers. However, to find out the optimal epsilon and min_samples point is difficult. If not using inertia elbom way to roughly find k-clusters, it is very hard to tune the parameters fine and easy to confused by lots of outliers.

	Age	Income	Score	Labels-km	Labels-agg	Labels-db	Labels-ms
0	19	15	39	3	4	-1	4
1	21	15	81	4	3	0	3
2	20	16	6	3	4	-1	4
3	23	16	77	4	3	0	3
4	31	17	40	3	4	-1	4
5	22	17	76	4	3	0	3
6	35	18	6	3	4	-1	4
7	23	18	94	4	3	-1	3
8	64	19	3	3	4	-1	4
9	30	19	72	4	3	0	3

Key findings and insights

```
df['Labels-km'].value_counts(ascending=False)
```

```
2    79
1    39
0    36
3    23
4    23
Name: Labels-km, dtype: int64
```

- Using the Kmeans with cluster=5, we can locate the customers as follow:

label 0 : 36 customers with high income and low score
label 1 : 39 customers with high income and high score
label 2 : 79 customers with medium income and medium score
label 3 : 23 customers with low income and low score
label 4 : 23 customers with low income and high score

- Furthermore, the segmentation is very useful in business marketing strategy and customer retention in future.

Suggestions of next move

- If data size becomes larger, and more features information of customers gatherer, it has more space to explore on these 4 clustering algorithms to try.
- If in business aspect, outliers can be also valuable. DBSCAN has advantages of finding outliers which helps to locate some potential customers.
- Also, in marketing aspect, market fractionize will be important to build marketing strategies while the hierarchical agglomerative clustering will be useful to locate the customers in different market fractionizations/segments in good-visual way.
- If there is target data, such as churn-or-not/membership pay-or-not data in real-world, will be helpful to evaluate clustering good or not, and becomes supervised problems to validate results.

References:

Ipynb code in my gist:
https://gist.github.com/apple9855/dcd1e79266f1617cce989d22f8de7a71