```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
from sklearn.datasets import make_blobs
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score
```

Main Objective

The main focus of this project is to identify distinct customer segments through clustering in order to understanding different customer group better and provide more useful plan for business

Dataset Description

This is a dataset that I have taken from Kaggle. The dataset have some data about their customer: Customer ID, Gender, Age, Income and Spending Score. The Spending score is something that the owner of this dataset created base on certain characteristic like purchase history and behavior. The Dataset have 5 columns: CustomerID, Gender, Age, Annual Income (k$), and Spending Score (1-100). Objective: To segment the customers base on Annual Income and Spending Score.

```
data = pd.read_csv('Mall_Customers.csv')
data.head()
```

	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
4	5	Female	31	17	40

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
```

```
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

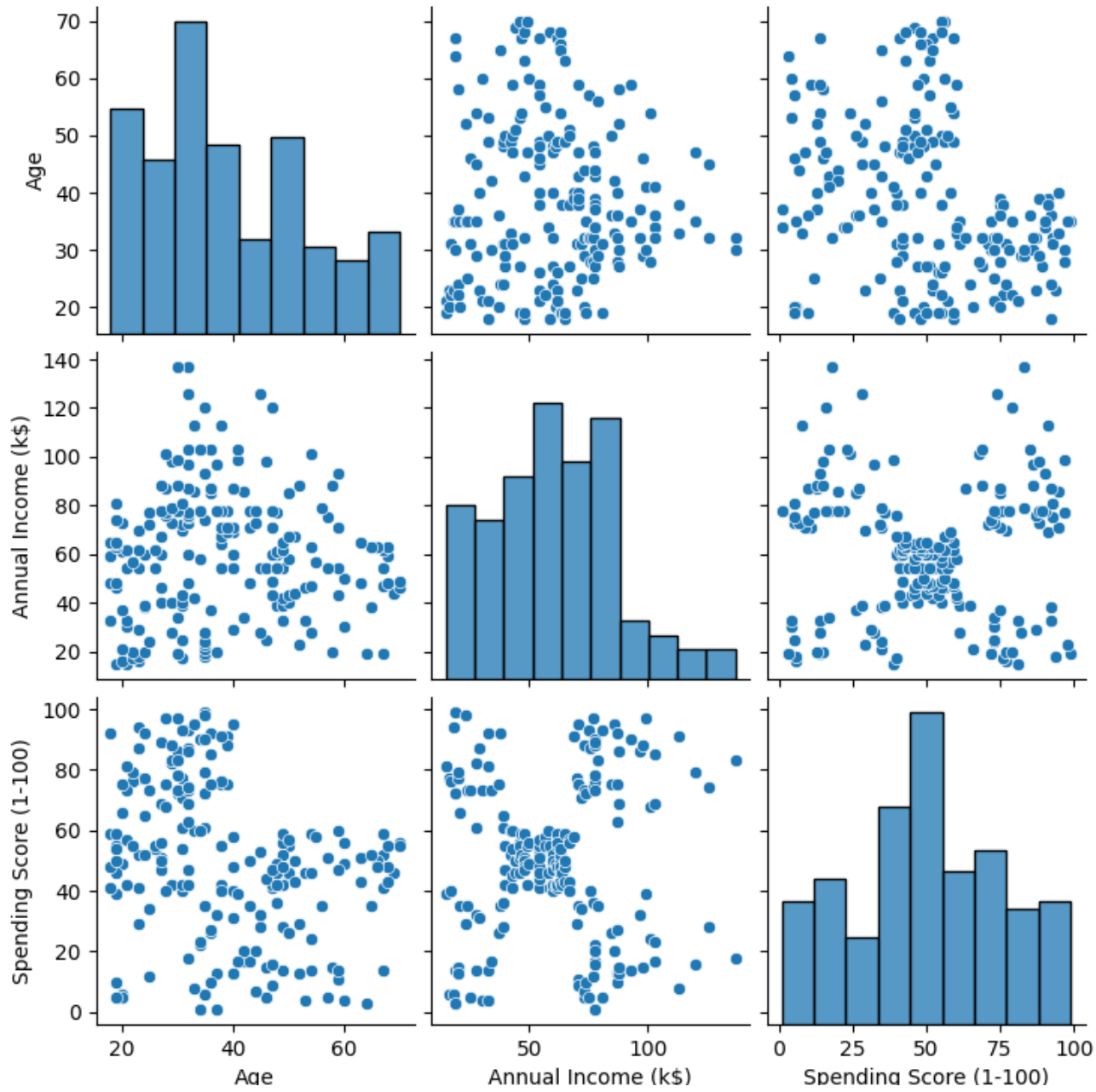
As we can see, the data have no null value

```
#Drop the unnecessary columns
data.drop(['CustomerID', 'Gender'], axis = 1, inplace = True)
data
```

	Age	Annual Income (k$)	Spending Score (1-100)
0	19	15	39
1	21	15	81
2	20	16	6
3	23	16	77
4	31	17	40
...	...	...	...
195	35	120	79
196	45	126	28
197	32	126	74
198	32	137	18
199	30	137	83

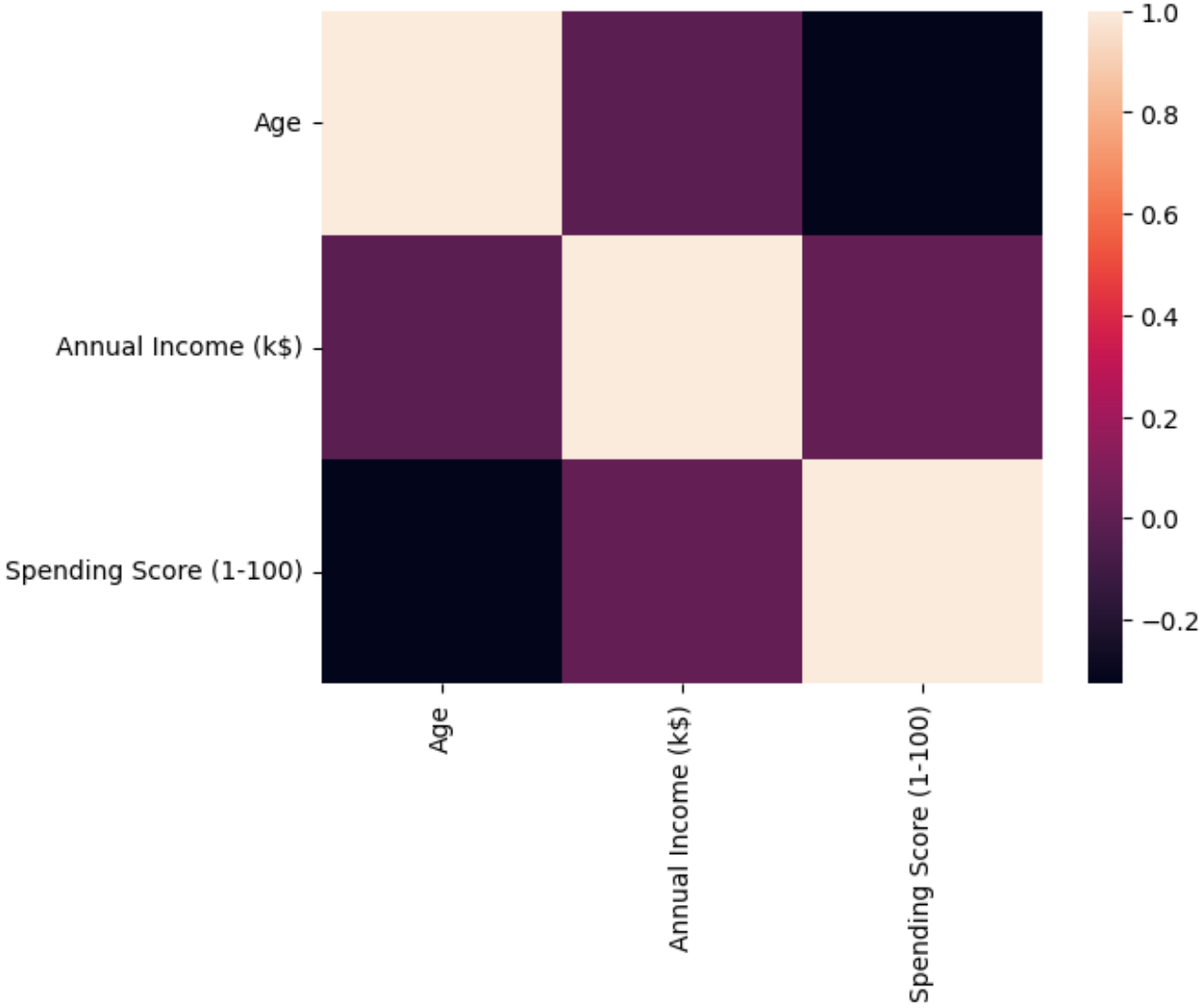
200 rows × 3 columns

```
#Relationship between Age, Annual Income and Spending Score
sns.pairplot(data[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
plt.show()
```




```
sns.heatmap(data.corr())
```

```
<Axes: >
```




```
#Choosing the feature
X = data[['Annual Income (k$)', 'Spending Score (1-100)']]


scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)


#KMeans model
kmeans = KMeans(n_clusters = 5, init = 'k-means++', random_state = 42)
data['cluster-kmeans']= kmeans.fit_predict(X_scaled)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/cluster/_kmeans.py:870: Futur
  warnings.warn(
```

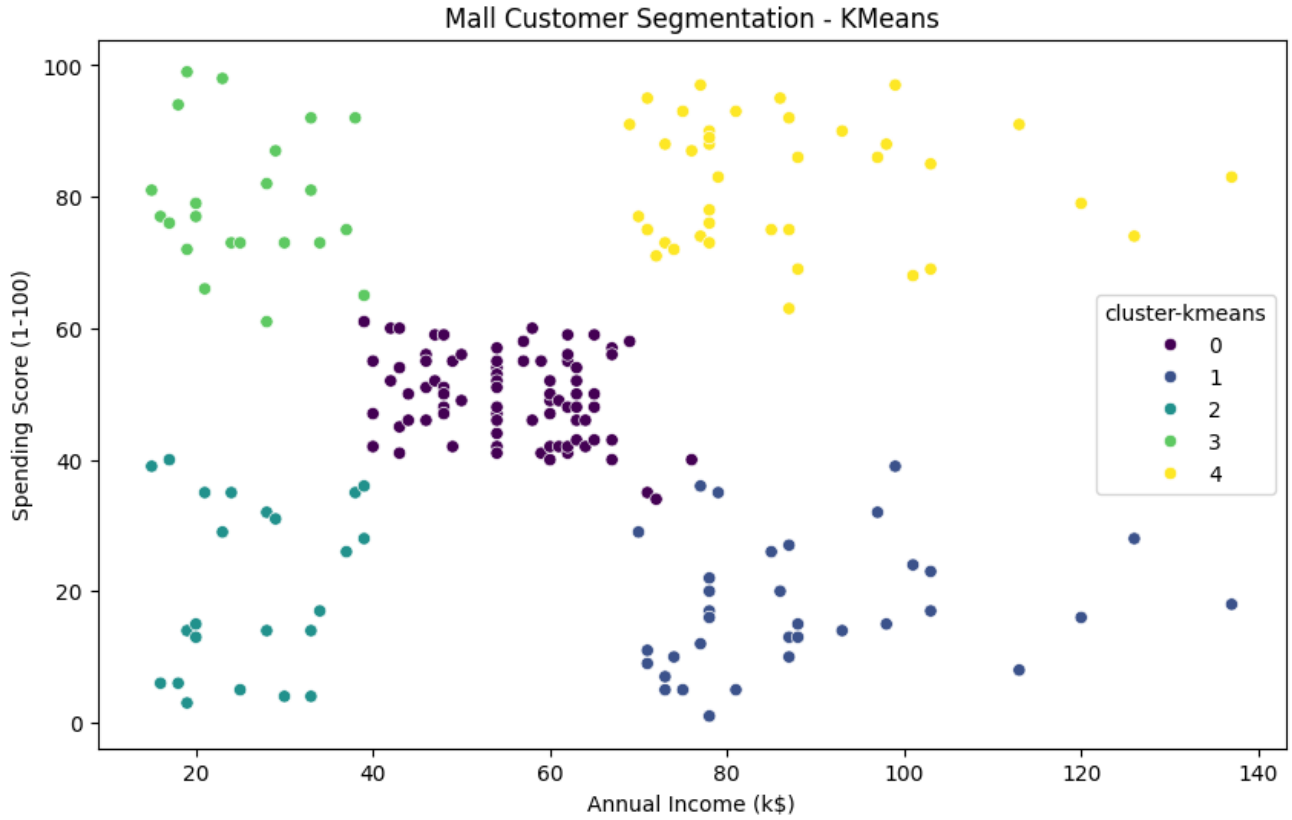
```
# Hierarchical Clustering (Agglomerative)
agglomerative = AgglomerativeClustering(n_clusters=5)
data['cluster-agglomerative'] = agglomerative.fit_predict(X_scaled)


# DBSCAN
dbscan = DBSCAN(eps=0.5, min_samples=5)
data['cluster-dbscan'] = dbscan.fit_predict(X_scaled)
```

```
#Evaluation
silhouette_avg_kmean = silhouette_score(X_scaled, data['cluster-kmeans'])
silhouette_avg_hierarchical = silhouette_score(X_scaled, data['cluster-agglomerat
silhouette_avg_dbscan= silhouette_score(X_scaled, data['cluster-dbscan'])
silhouette_avg_kmean, silhouette_avg_hierarchical, silhouette_avg_dbscan
```

```
(0.5546571631111091, 0.5538089226688662, 0.35044619989666004)
```

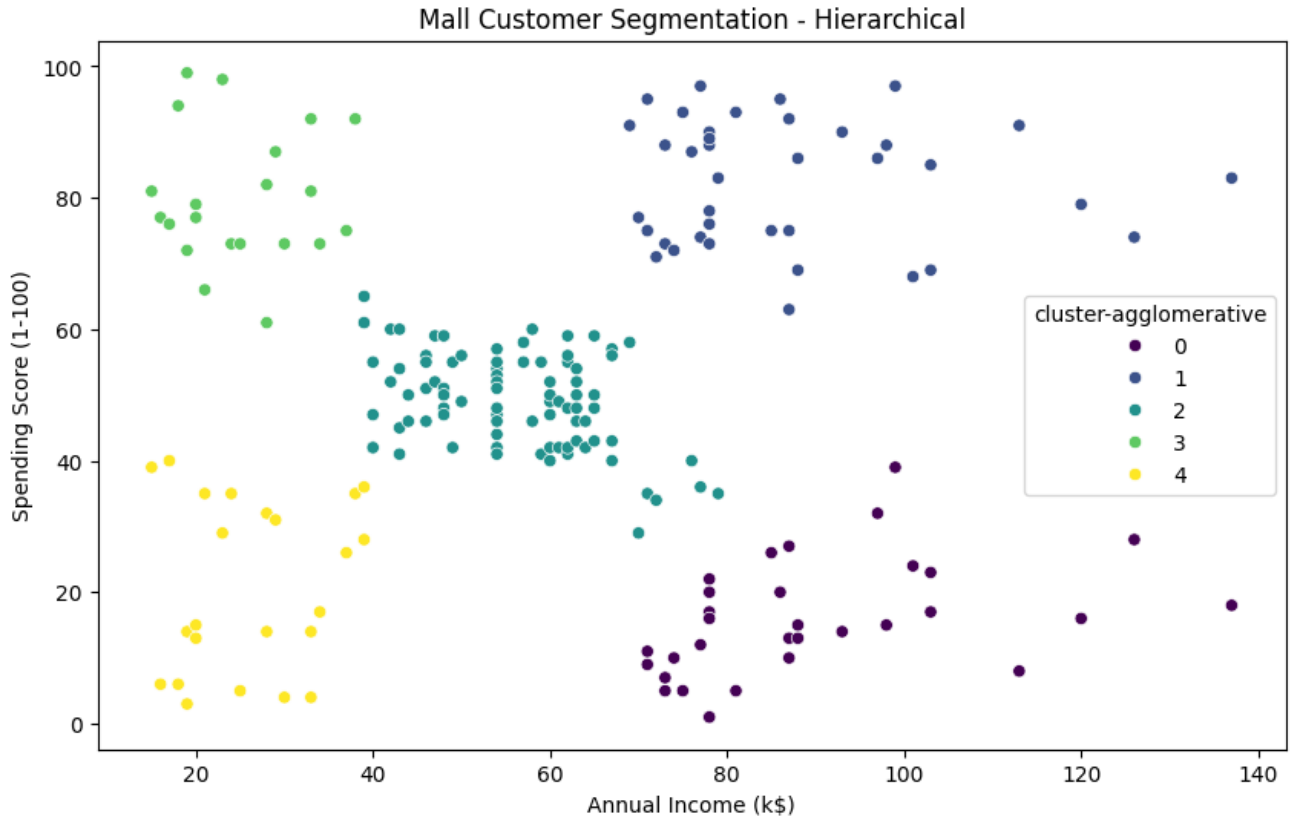
```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='cluster-
plt.title('Mall Customer Segmentation - KMeans')
plt.show()
```



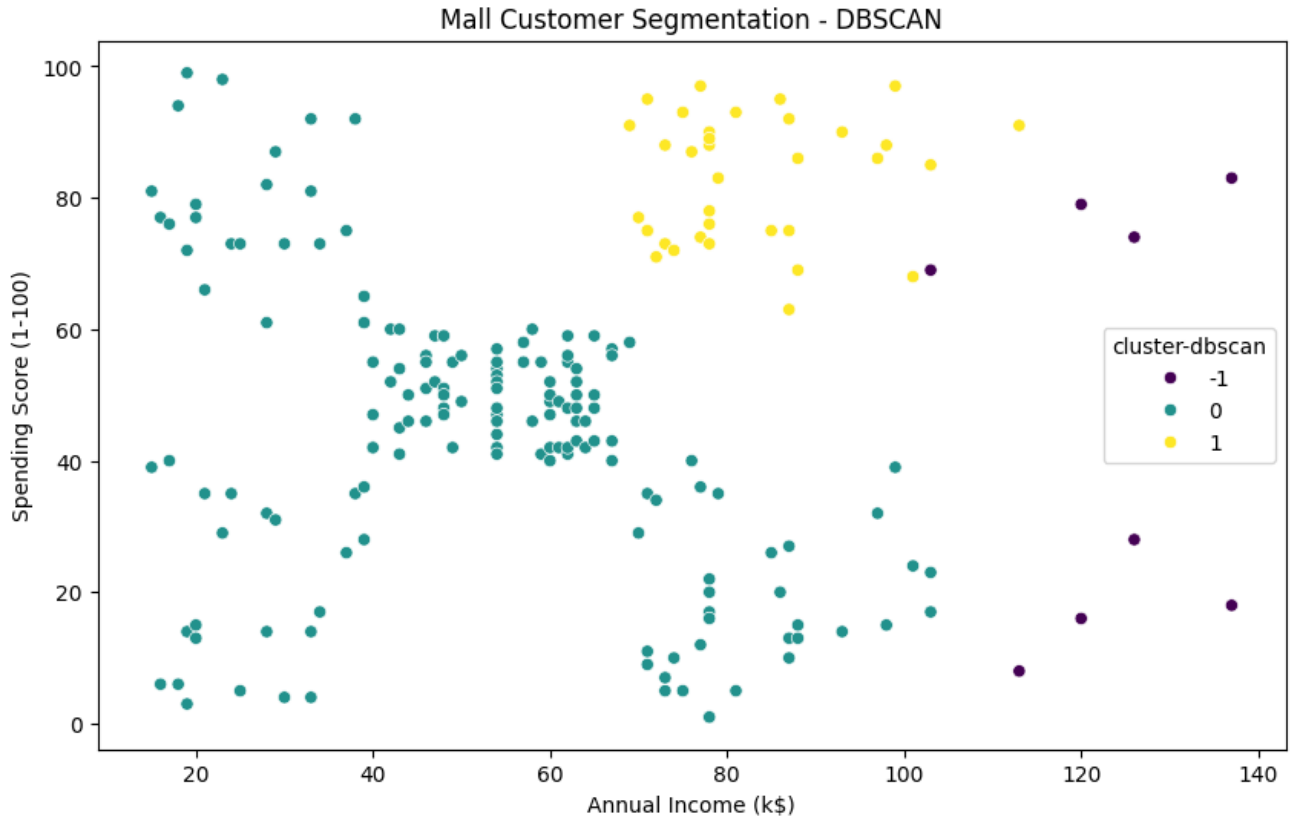

```
cluster_counts = data['cluster-kmeans'].value_counts()
cluster_counts
```

```
0    81
4    39
1    35
2    23
3    22
Name: cluster-kmeans, dtype: int64
```

```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='cluster-
plt.title('Mall Customer Segmentation - Hierarchical')
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Annual Income (k$)', y='Spending Score (1-100)', hue='cluster-
plt.title('Mall Customer Segmentation - DBSCAN')
plt.show()
```



Keys finding

- The customer in the data is divided in to 5 pretty distinct group
- The group in the middle seem to be average in both annual income and their spending score, and they are also the largest group by a large margin (85)
- The other group are relatively the same size, around 20-40
- Overall, the customer segmentation is pretty well defined in Kmeans and Hierarchical