# Mall Customers Project

EC410

Group members: Yi Ding, Zhenyu Wang, Jiajie Duan, Liming Pang

University of West Alabama

May 2, 2021

Contribution of each member:

Yi Ding: Literature Review, Empirical strategy(K-Mean), Interpret the results, Discussion.

Zhenyu Wang: Literature Review, Empirical strategy(K-Mean), Data summary, preprocessing

Jiajie Duan: Introduction, Empirical strategy (DBSCAN), Analysis & Results.

Liming Pang: Motivation, Empirical strategy (DBSCAN), Analysis & Results, Comparison, Conclusion

# Introduction

In this project, we will build a model that will use a mall customers data to segment the customers into different groups based on their behavior patterns. Such customer segmentation is a highly useful marketing tactic used by brands and marketers to boost sales and revenue while also increasing customer satisfaction. We do this based on the mall customers dataset, that includes the records of people who visited the mall, such as gender, age, customer ID, annual income, spending score, etc. We will consider the basis ideas of recommending algorithm that main companies in online shopping platform do, such as Amazon, Taobao and JD, but make several adjustments and keep our focus on the part of segmenting. The pool of the algorithms and models that may be used in this project includes k-Means Clustering, kernelized support vector machines, DBSCAN Clustering, unsupervised algorithms, etc. The basic logic behind that is unsupervised learning, so what we have to do is to preprocess and scale the data, make dimensionality reduction, feature extraction or maybe some manifold learning, and do the proper clustering as well as summarize them.

# Motivation

The business model has evolved so far, not just to sell what can be made, but to make what can be sold. The business of an enterprise must be built on customers with certain needs, so the market expansion will be carried out according to the analysis of customers' behavior. Only a better understanding of customer habits, customer preferences, customer profiles can we better innovate, improve, and iterate on products. Hence, we want to use data analysis to help us do that.

With the improvement of science and technology, products and exist technologies will

eventually become obsolete, but the basic market needs have always been there and will always be, only as consumers' demands become higher or change. Today's market is in the environment of high cost and high competition. If the enterprise cannot make use of data analysis to do fine operation, it will produce a huge waste of resources, which is bound to make the enterprise's operation cost rise and lack of competitiveness.

There are three main aspects that motivate us to build the analysis model:

From the perspective of products, analyzing consumers' behaviors helps verify the feasibility of products, study product decisions, clearly understand consumers' behavior habits, and find out the defects of products, so as to facilitate the iteration and optimization of requirements. Consumer-centered design has always been the primary design principle of products. In order to create value, every product we have ever seen or used in our daily life needs to be connected with consumers. It can be said that the research on consumers is the first step in product design.

In terms of design, acquiring customer data can help increase the friendliness of experience, match customer emotion, finely fit customer personalized service, and find out the deficiencies of interaction for the perfection and improvement of design. From the perspective of operation, helping consumers' behavior data can realize precise marketing, comprehensively mining customers' usage scenarios, and analyzing operational problems, so as to facilitate the transformation and adjustment of decision-making. Reality, through the analysis of monitoring data obtained from the consumer behavior, can let enterprise more detailed and clear understanding of consumer behavior, so as to find out website, marketing channels, such as the enterprise marketing environment problems, help enterprises explore

the high conversion page, make enterprise marketing more precise and effective, improve conversion rate, thus improve enterprise's advertising revenue. Therefore, we want to create a model that classifies consumers according to their behavior to understand how to improve customer satisfaction and profit of the enterprise.

## Literature Review

According to Iakovou, Kanavos and Tsakalidis (2016) they used data mining techniques to propose a prediction model based on individual customer behavior. The model makes use of a supermarket database and an additional database from Amazon, both of which contain information about customer purchases. This data is then analyzed by the authors' model to classify customers and products, and then trained and validated with real data. The model aims to classify consumers according to their consumption behavior, thus proposing new products that are more likely to be purchased by them. The corresponding predictive model is intended as a tool for marketers to provide analysis of targeted and specified consumer behavior. Katrodia, Naude and Soni (2018) mentions that to explore gender differences in consumer buying behavior in selected shopping centers in Durban. This is an observational cross-sectional study of 700 randomly selected respondents. The study was conducted to investigate the purchasing power, purchasing behavior and shopping experience of male and female consumers in shopping centers in Durban, South Africa. Data were collected through a semi-structured predictive questionnaire with closed questions. The study showed significant gender differences in men's and women's shopping behavior. Female consumers spend significantly more time and money in stores than male consumers. The results showed

that personal traits and mall attraction factors played an important role in influencing shopping behavior. The study concluded that in shopping centers in Durban, South Africa, gender differences were prevalent in the buying behavior of customers. Women spend more time on average than men, which affects their average spending at the mall. Psychological, social and cultural factors have a great impact on consumers' purchasing behavior in shopping malls. This paper includes the gender variable of our research topic, so by analyzing the conclusions of this paper, we can measure how much the gender variable affects consumer behavior.

 Sabbeh (2018) states that machine learning techniques could be used for customer retention. This strategy lists multiple algorithms to deal with the problem about customer retention. Random Forest and ADA boost show up the maximum accuracy of prediction with above 96%, even consider the minimum accuracy of prediction, 93% accuracy in predicting brought by these two algorithms are also most attractive compared to other algorithms in the testing group. However, LDA and logistic algorithms struggle with explaining customer data since both only bring up 87% accuracy predicting the retention in ‘maximum’ and below 85% accuracy in ‘minimum’. Ezenkwu, Ozuomba and Kalu (2015) state that K-means algorithm could be appied to customer segmentation and make it a more efficient process. This strategy shows the application of K-Mean – an unsupervised learning algorithm in offering services to customers with specific behavior codes. 95% of accuracy in predicting customer segmentation is an attractive result since K-means algorithm is a simple one compared to other machine learning methodology. Also, this result could not only be predicted, but it could also be explained by real-world logic.

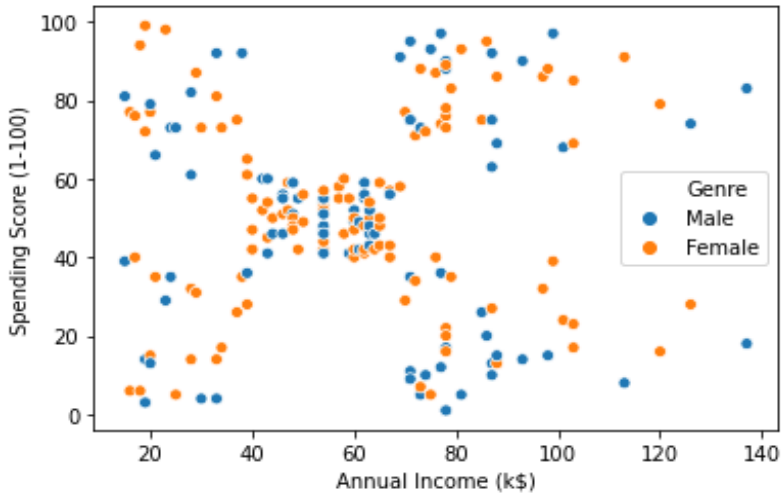
## Empirical strategy

The source of our data use is a database uploaded by Kaggle users. We tried to find the ultimate source of the data, but since there was only one version of the data, the user did not indicate the ultimate source of the data, and it has been a long time since uploaded the data, we did not find the terminal source. Therefore, we used the direct data on Kaggle as the data source.

**1. Data collection and preprocessing process**

This dataset includes 200 observations, with four features (gender, age, annual income, and spending score). According to the Principal Component Analysis (PCA), we can consider spending score as the first principal component and annual income as the second principal component. So, we can now plot the first two principal components by taking gender as legend. (See Table 1)

**Table 1:**




Another application of PCA is feature extraction. The idea behind feature extraction is that it is possible to find a representation of your data that is better suited to analysis than the raw representation you were given. In that case, we only have four features, so it is

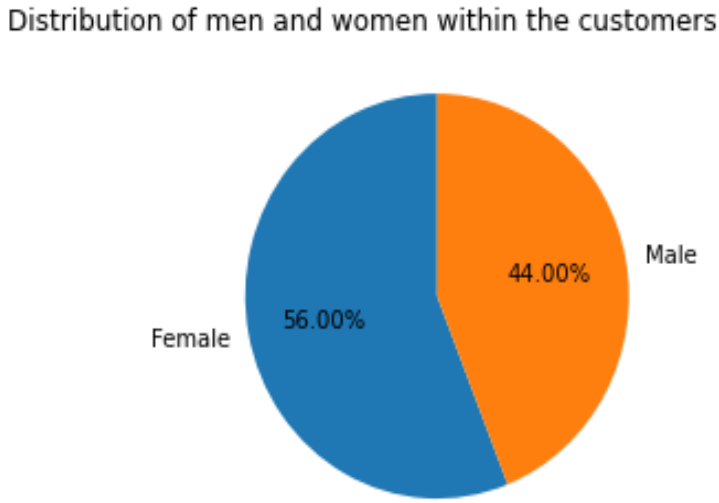
unnecessary to do any work of feature extraction. There also is a class of algorithms for visualization called manifold learning algorithms that allow for much more complex mappings, and often provide better visualizations. Because this case is only number, no information of any images or mappings, this algorithm will not be considered either.

The given dataset already shows up a pattern which does not need too much prepossessing, since it is not complicated, and algorithms could easily find a pattern. Raw data is enough in this case because trimming is used when the number of observations is large yet the number of observations from raw data is small enough. Dimensionality could not be applied to our raw data since raw data only have four features which may not cause the unhelpful signal. The all-four-dimensional features contribute the same to the success of a machine learning algorithm.

**2. Features and sample descriptions**

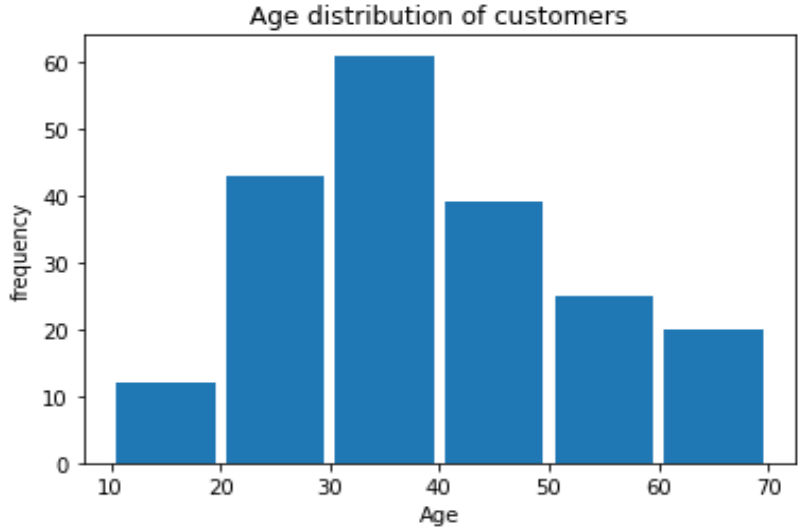
The mall customer data set contains information about the people visiting the mall. The data set includes gender, customer ID, age, annual income, and spending scores. It gathers insights from data and groups customers based on their behavior. Segmentation of customers by age, gender, and interests. The customer IDs are from 1 to 200, so our sample size is 200 and 56 percent of them are women and 44 percent are men. Let us look at the men and women via histogram. (See Table 2)

**Table 2:**



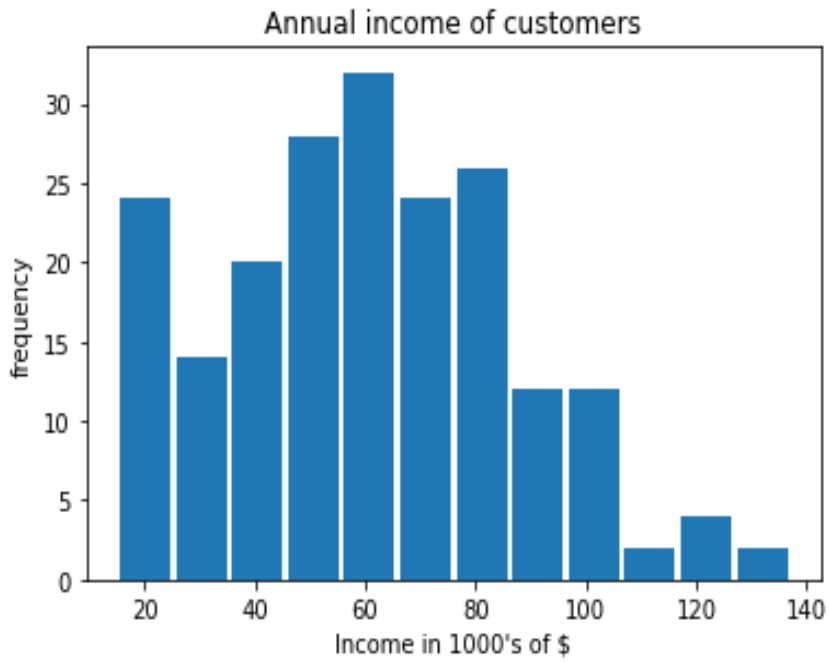

The minimum age is 18 and the maximum is 70. And the average age of customers is 36. We can see from the chart (See Table 3). Each bin could represent 10 years.

**Table 3:**




Annual income represents the income of each customer who visits the store. The lowest earners made $15,000 a year, while the highest earners made $137,000. The average annual income of the 200 people in the sample was $61,500. The variance of annual income is $26.26, which shows that the difference of annual income is relatively large. (See Table 4)

**Table 4:**




The spending score represents the customer's ability to spend. Rank the sample by how much they spend, and then rescale them on a scale of 1 to 99, from the lowest to the highest. On a scale from low to high, low scores indicate low spending power. On the contrary, people with high scores have more spending power. (See Table 5)

**Table 5:**

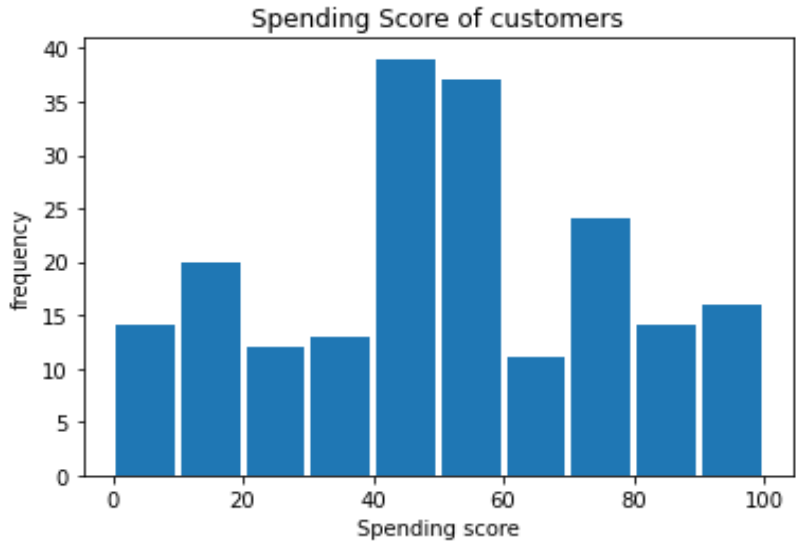


**Table 6：Summary Statistic**

| **Variables** | | Number | Percentage share |
|---|---|---|---|
| Gender | | | |
| | Female | 112 | 56% |
| | Male | 88 | 44% |
| **Total** | | 200 | 100% |

| | Mean | S.D | Min | Max |
|---|---|---|---|---|
| Age | 36 | 13.97 | 18 | 70 |
| Annual Income | 61.50 | 26.26 | 15 | 137 |
| Spending Score | 50 | 25.82 | 1 | 99 |

The first algorithms we choose is k-Means Clustering. It is one of the simplest and most commonly used clustering algorithms. We can use it to find cluster centers that are representative of certain regions of the data. Also, DBSCAN is our choice which stands for “density based spatial clustering of applications with noise”. DBSCAN does not require us to set the number of clusters a priori, it can capture clusters of complex shapes, and it can identify points that are not part of any cluster. DBSCAN allows for the detection of “noise points” that are not assigned any cluster, and it can help automatically determine the number of clusters. Moreover, it allows for complex cluster shapes.

# Analysis & Results

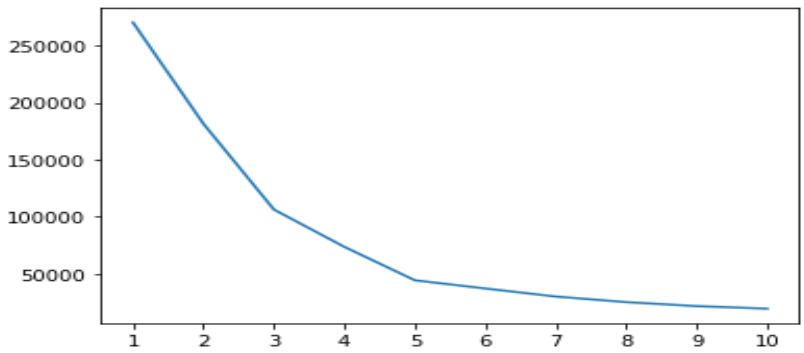
### 1. K-means algorithm

In this research, first method we used is K-means algorithm which is one of the simple data methodologies belonged to unsupervised learning. The mechanism of this algorithm could be described following: firstly, the number of cluster of dataset would be defined whether by intuition or by computational methods. Two computational methods would be introduced in this case before applying K-means algorithm to classify the data. One is called the Elbow method and the other one is Silhouette test.

### 2. **Elbow Method**

The first one method, Elbow method, is one of computational methods to seek for sweet spot between 'within cluster sum of squares' and the number of K. The mechanism of elbow method is that this algorithm would calculate the distance within cluster sum of squares each time the number of K is defined to be one, two, three to ten. In figure 5-1, point (5, 5000) is the sweet spot. Therefore, based on result of elbow method, the number of K would be defined as five since this is the most efficient point.(See Table 7)

**Table 7: Elbow method**




**Determine number of clusters**

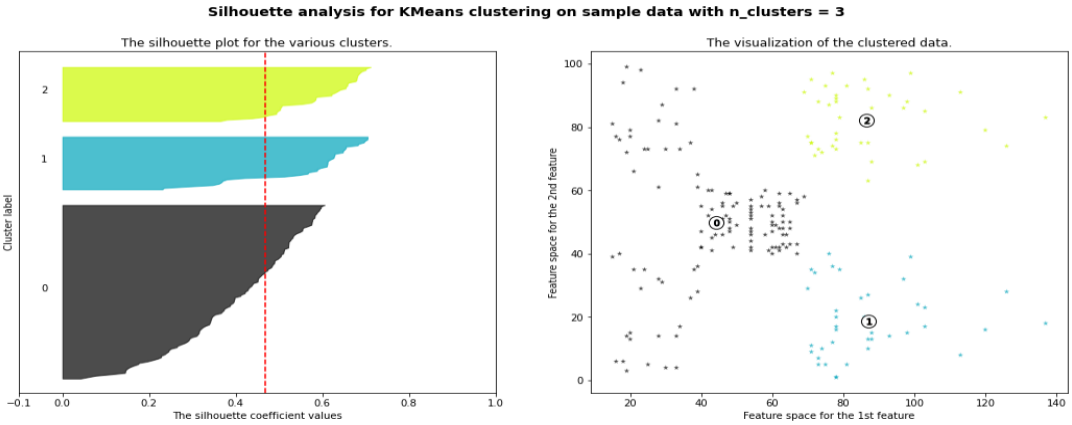
---

After analyzing the dataset, the silhouette test is introduced to see which number of cluster is available.

---

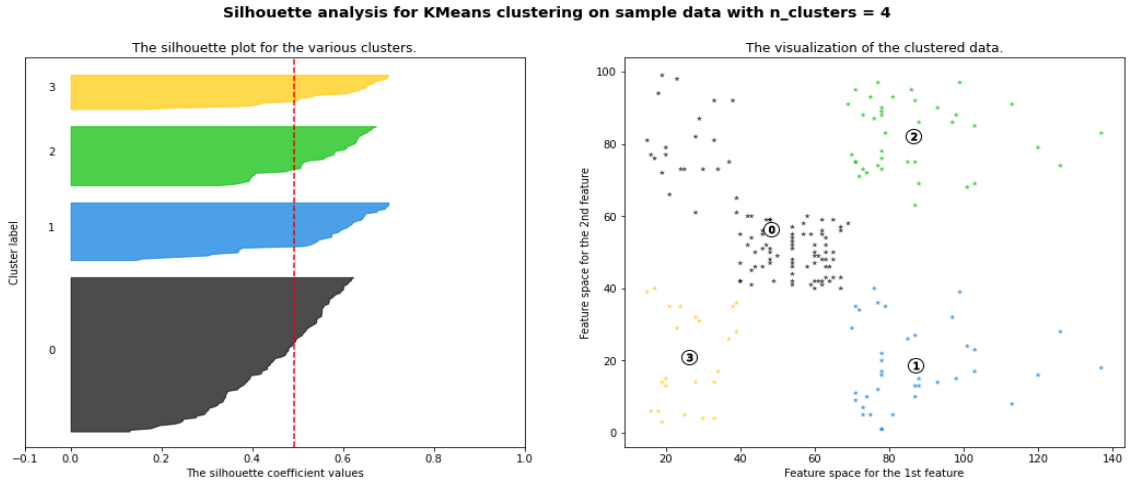
When it comes to the second method-the silhouette test, the silhouette score computes the compactness of a cluster, where higher is better, with a perfect score of 1. While compact clusters are good, compactness does not allow for complex shapes. Four numbers [3,4,5,6] represent the situation of K-means cluster respectively. When K equals to 3, which means dataset is divided into 3 clusters, the silhouette for '1' group and the '2' groups both are around 0.7 yet the '3' group is only around 0.6. Therefore, when the number of cluster is 3, silhouette is around 0.6 to 0.7. (See Table 8)

**Table 8: Number of Clusters = 3**



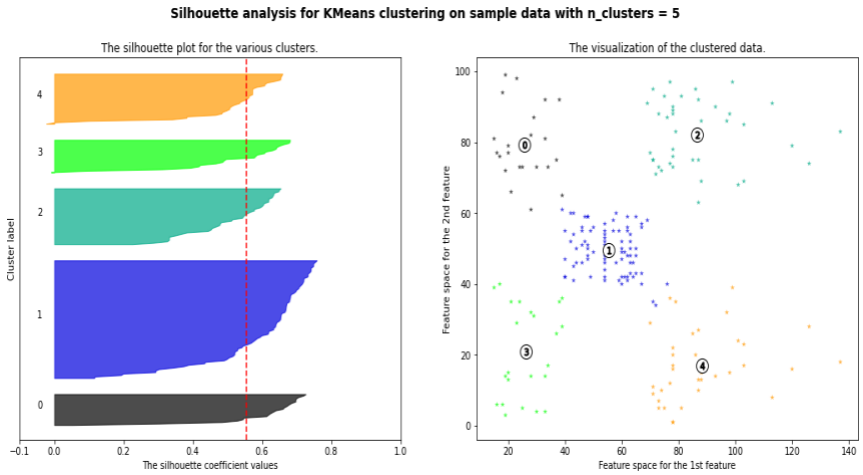

When K equals to 4, which represents dataset is divided into 4 clusters, group '1','2' and '3' have a comparatively high silhouette which is around 0.7, but the problem is that group '0' still have bad compactness. (See Table 9)

**Table 9: Number of Clusters = 4**



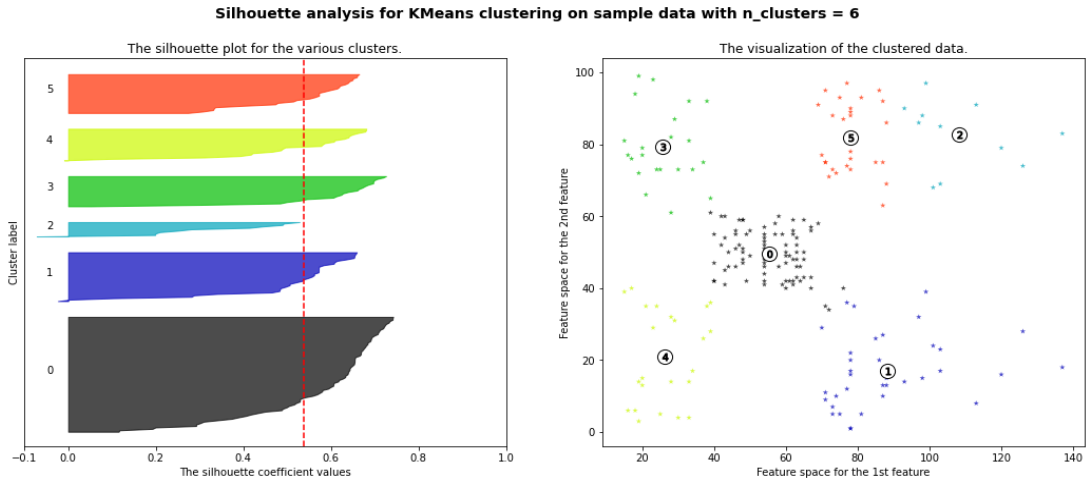

In Table 10, the compactness of five groups is overall better than the others, since four groups are around 0.7 and one group '1' reach almost reach 0.8 yet the compactness of group '1' has a bad compactness in other groups. (See Table 10)

**Table 10: Number of Clusters = 5**



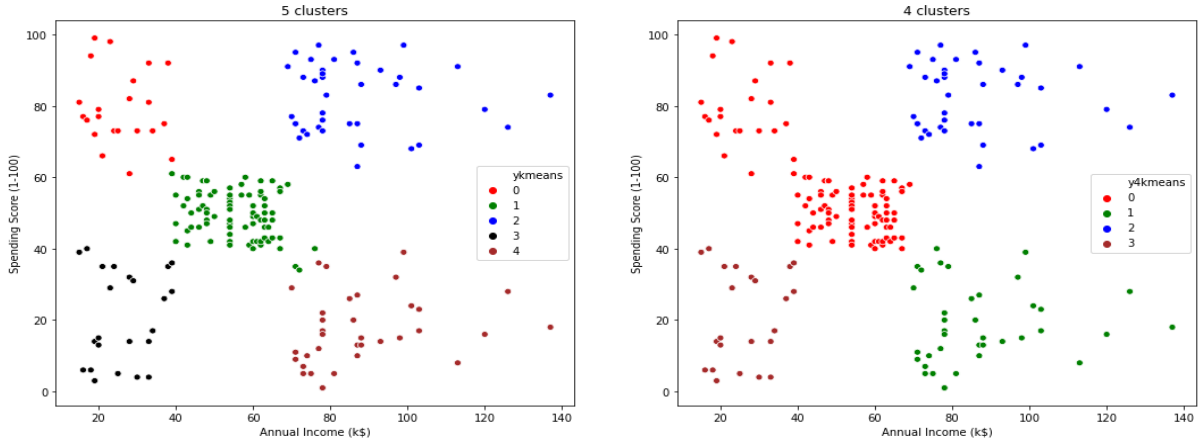

In Table 11, group '2' does not reach the 0.5 line so 6 as the number of clusters is not considered. (See Table 11)

**Table 11: Number of Clusters = 6**




Eventually, four clusters are compared to five clusters since they show almost the same compactness. Intuitively, five is better than four since the 'red' part really shows a difference in compactness. (See Table 14)

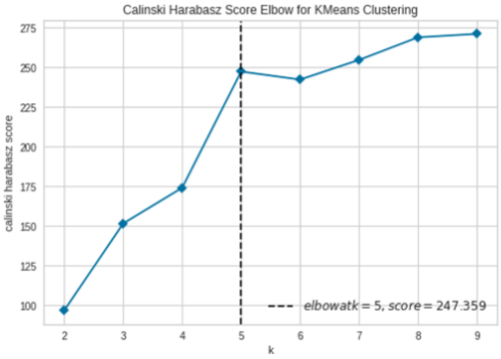
**Table 12: Comparison of Clusters 4 & 5**

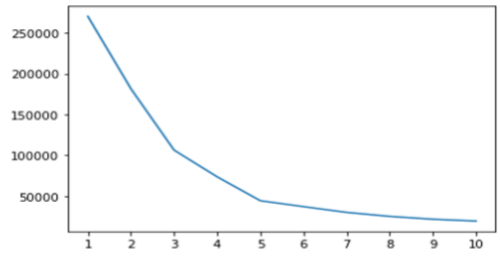



**Comparison between Elbow method and the silhouette test**

It is tempted to see that elbow method and silhouette test get the same result.(See Table 15)

**Table 13: Comparison of silhouette test and elbow method**




Intriguingly, the silhouette test shows that when number of clusters reaches five, the score is already the efficient point which is 247.359, namely, the sweet spot for the silhouette test is (5,247.3599) . For elbow method, the sweet spot is (5,5000), which means both computational methods find five an optimal number of clusters. Therefore, five is chosen to be the number of K.

**K-means algorithm application**

Move from the determination of number of clusters to the K-means algorithm application,

---

then several points from test set depending on the number of clusters defined would fall on the training set and the closet points to the center point of a single cluster would be assigned to that cluster.

---

---

the mechanism of K-means algorithm is that it would assign new points to a specific cluster by calculating out the sum of squares of distance between new points to center points of this cluster. The center point of a cluster would be re-chosen to achieve the least sum of square

between center to the other points. In applying K-means algorithm, two indicators are introduced to separate customers into five types. The first one is spending score ranging from 0 to 100 and the higher this indicator is, the more active an individual consumer is. The second one is annual income ranging from 20 to 140, the higher this number is, the wealthier a client is, as showing in Table 14, where x axe stands for annual income and y axe stands for spending score. (See Table 14)

Table 14: Clusters of Clients


Previous result from elbow method has already showed that ‘five’ is the optimal number of K, Based on the idea from one of the Kaggle user Hbao (2021) concludes that low annual income clients with low desire in shopping is assigned to be ‘careful’, low annual income clients with strong intend to consume products are assigned to be ‘careless’, ‘standard’ group includes consumers with average annual income and average intend to spend money, wealthy people with low intend to consume products are assigned to be ‘sensible’ and wealthy people

with strong tendency to spend score is assigned to be ‘target’, namely, the targeted customers.

Previous result from elbow method has already showed that ‘five’ is the optimal number of K. Based on the idea from one of the Kaggle user Hbao (2021) concludes that low annual income clients with low desire in shopping is assigned to be ‘careful’, low annual income clients with strong intend to consume products are assigned to be ‘careless’, ‘standard’ group includes consumers with average annual income and average intend to spend money, wealthy people with low intend to consume products are assigned to be ‘sensible’ and wealthy people with strong tendency to spend score is assigned to be ‘target’, namely, the targeted customers.

**2. DBSCAN Clustering**

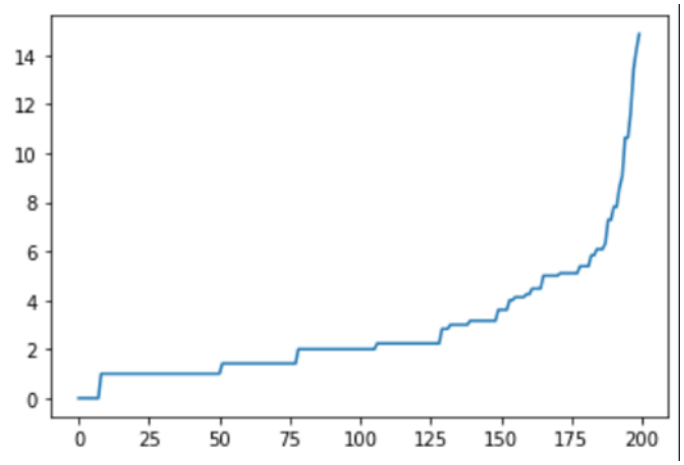
DBSCAN Clustering (where DBSCAN is short for Density-Based Spatial Clustering of Applications with Noise) involves finding high-density areas in the domain and expanding those areas of the feature space around them as clusters (Müller & Guido, 2016). We believed DBSCAN is very useful clustering algorithm. DBSCAN can detect “noise points” that are not assigned to any cluster, and it can help automatically determine the number of clusters (Müller & Guido, 2016). So, we think it can help us to classification. However, for this data, DBSCAN is not a good choice.

First thing we need to do is to find optimal value of Epsilon (eps) and min sample, which are two parameters of DBSCAN. Since eps is more important cause it determines what it means for points to be “close” (Müller & Guido, 2016). In order to get best model, we tested multiple models with different parameters.

First, we make an elbow plot (Figure 15). The ideal value of eps will be equal to the

distance value at the “crook of the elbow”, or the point of maximum curvature. This point represents the optimization point where diminishing returns are no longer worth the additional cost. (Mullin. T, 2020). Intuitively, “6” located in the “crook of the elbow”, hence, we chose it as the optimal number for eps.

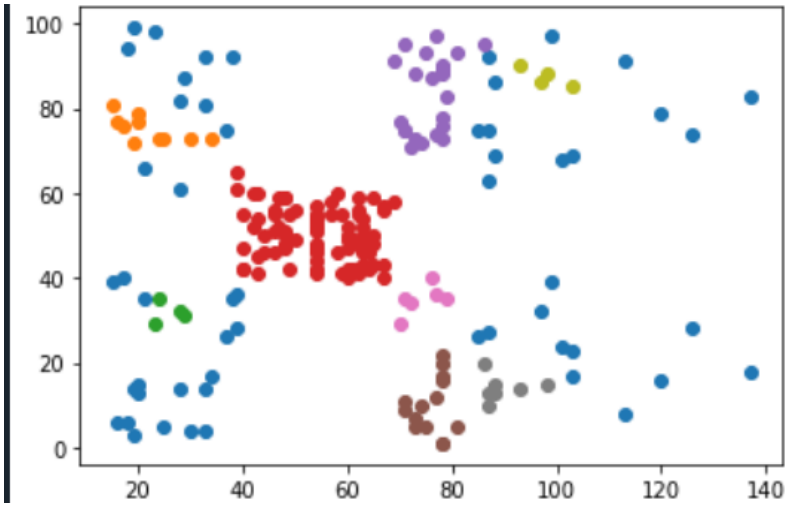
**Figure 15:**




(Where Y-axe for number of clusters, X-axe for number of points in one cluster.)

And for the min sample, because this is 2-dimensional data, based on research from Ester, Kriegel, Sander, and Xiaowei in 1996, we used 4 first which is the DBSCAN’s default value. The result shows in Figure 16.

**Figure 16**



when we are increasing min_samples, fewer points be core points, and more points will be labeled as noise. As long as the number is higher than 22, all the points will be the same color. And when we decrease the value, it also cannot show us a clearly result, the data of different colors are randomly distributed, so we do not get good clustering distributions

(min sample = 1, 10 and 22, as showing in Figure 17, 18 and 19)

**Figure 17**

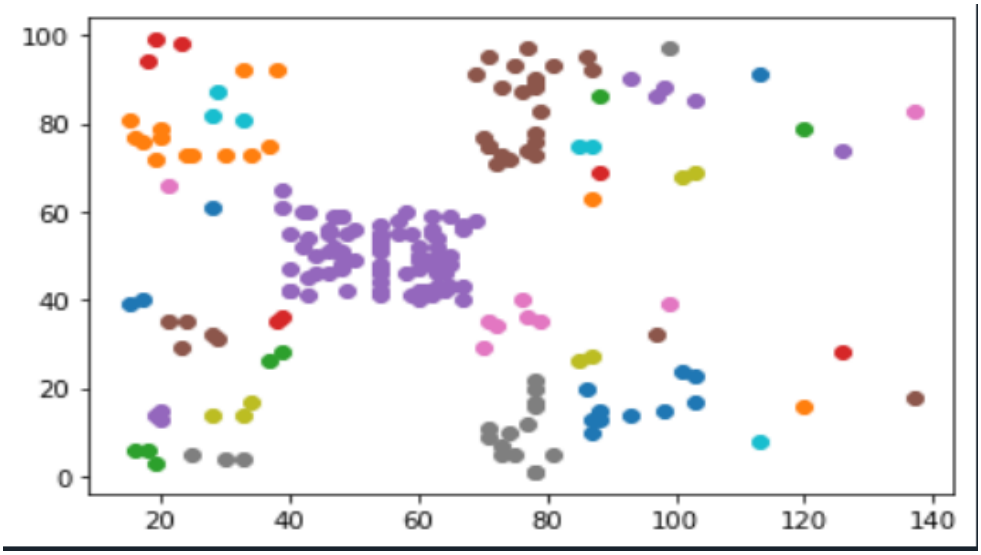



**Figure 18**

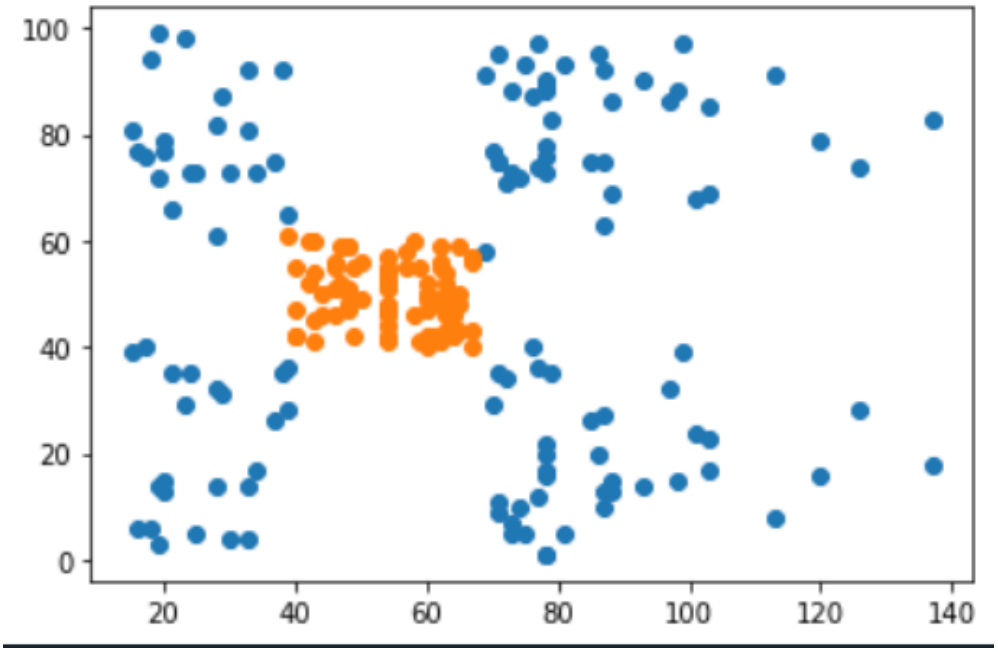
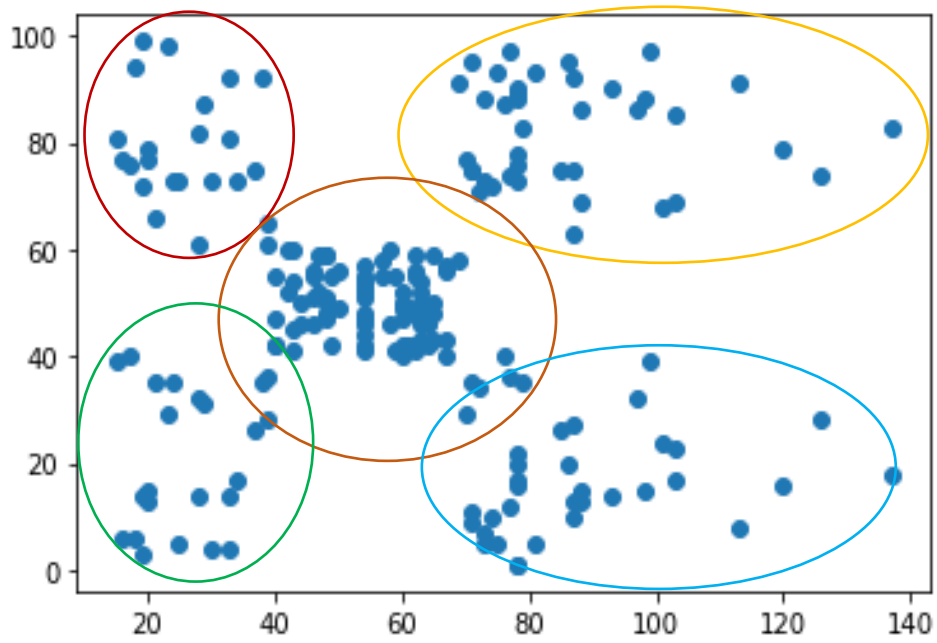


**Figure 19**




(X-axis is Feature 0 and Y-axis is Feature 1)

After several tests, we define model which set two parameters, eps equal 6 and min-sample equal 22. As we can see from figure above, the clustering obtained using DBSCAN are not very good. Although you can see that there approximately are five clusters which are circled within different colors, the direct connection between the groups is more complicated. At the same time, we cannot distinguish the five groups intuitively by using DBSCAN. As the result showing in Figure 17 to Figure 19, the clustering is not clear enough to tell what exactly these group are because of the unclear color, so that we cannot identify the data represented by each cluster.

**3. K-Means vs DBSCAN**

The results obtained by k-means algorithm and DBSCAN are actually very similar. We can see from the figure that there are about five clusters. However, it is obvious that k-means algorithm is more accurate than DBSCAN. In terms of K-Means, we can easily identify the

data represented by each cluster. But, for DBSCAN's perspective, first of all, we could not find EPS and Min Sample that could make it present recognizable clusters. The two coefficients we found are already the optimal values, but it is still difficult for us to accurately explain how many clusters there are, nor to accurately group them. We also use the method of silhouette coefficient to compare the two algorithms. We set the number of DBSCAN clusters as 5, which is the same as the value obtained by K means algorithm using the silhouette method. When we are evaluating this clustering, the Silhouette Score is around 0.55393, which is not better than K-means. Therefore, we drop this method. K-means algorithm clustering allows us to specify the number of desired clusters, and we can get a characterization of the clusters using the cluster means. Also, it can be viewed as a decomposition method, where each data point is represented by its cluster center.

## Conclusion

All in all, two unsupervised learning algorithms are prepared to deal with this mall project, one is K-means algorithm and the other one is DBSCAN. K-means algorithm is used to separate the data points into several groups by seeking the minimal distance between each new data point to center of a single cluster. When determining which number of cluster should be used in this mall customer case, elbow method is introduced to look for the sweet spot since there are hundreds of thousands of results for different number of cluster. After using elbow methods, when number of cluster equal to three, four, five or six, the compactness is acceptable. To look into the compactness of these four results more specifically, the silhouette test method is conducted to each situation. Finally, five is the

optimal number among these four numbers. K-mean clustering has been performed over a mall customer dataset to classify customers into different segments.

There are Five customer segments were found having different age, income, and spending trends. Based on our analysis, in order to make it better for the mall management to retain customers and increase sales, it is recommended that management focuses on retaining the following segments: 1. Rich and high spending people between their 20 and 40 years old. 2.Relatively poor and high spending people between their 15 and 30 years old.

Specifically, for segment - low annual incomes and low spending scores and high annual income and low spending scores, customers tend to be over the age of 35 and mostly male or elderly female, mall management need to find a way to attract regular customers, such as special offers on products they often buy, and investigate the reasons for the low spending score.

For segment - low annual income and high spending scores, mall management should design a loyalty program that offers discounts to members to keep these customers. People who have high annual incomes and high spending scores mostly are middle-aged customers, they are the most valuable customers, mall should keep these customers by offering reasonable prices or discounts. And last group - medium annual incomes and medium spending scores, mall management can attract them with a marketing campaign.

# References


Iakovou, S.A., Kanavos, A., Tsakalidis, A. (2016). Customer Behavior Analysis for Recommendation of Supermarket Ware. *Springer International Publishing Switzerland*, 471-480. DOI: 10.1007/978-3-319-44944-941.

Katrodia, A., Naude, M.J., Soni, S. (2018). Consumer Buying Behavior at Shopping Malls: Does Gender Matter? *Journal of Economics and Behavioral Studie,* 10(1), 125-134. ISSN: 2220-6140

Sabbeh,S.F. (2018). Machine-Learning Techniques for Customer Retention: A Comparative Study. *International Journal of Advanced Computer Science and Applications.* 9(2).

Ezenkwu,C.P., Ozuomba,S., Kalu,C. (2015). Application of K-Means Algorithm for Efficient Customer Segmentation: A Strategy for Targeted Customer Services. *International Journal of Advanced Research in Artificial Intelligence*. 4(10).

Kaggle user of "Hbao12" (2021). Machine Learning - Hierarchical Clustering. *Kaggle database.* Retrieved from: https://www.kaggle.com/hbao12/machine-learning-hierarchical-clustering

Müller, A. C., Guido, S. (2016). Introduction to Machine Learning with Python: A Guide for Data Scientists. United States: O'Reilly Media.

Tara Mullin. (2020), DBSCAN Parameter Estimation Using Python, *MEDIUM,* Retrieved from: https://medium.com/@tarammullin/dbscan-parameter-estimation-ff8330e3a3bd

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining.

# Appendix

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.metrics import silhouette_samples, silhouette_score

from sklearn.cluster import DBSCAN

import matplotlib.cm as cm


df = pd.read_csv('C:/Users/duanj/Desktop/Mall_Customers.csv')

df.head(20)

df.shape


#data summary

df.describe()

df.isna().sum()

len(df)


#visualization (gender spending score and income)

sns.scatterplot(x='Annual Income (k$)' , y = 'Spending Score (1-100)',data=df , hue='Genre')

x = df.iloc[:,[3,4]].values

x[:10]
```

```
#Let's fist do basic visualisation to understand the data and the distribution of the different
variables

#let's look at the men and women via histogram
sns.countplot(x='Genre', data=df)
plt.title('Customer Gender')

#to make a piechart:
gender=df.Genre.value_counts()
gender_label=['Female','Male']
plt.pie(gender, labels=gender_label, autopct='%0.2f%%',startangle=90)
plt.title('Distribution of men and women within the customers of the Mall')
plt.show()

#let's see the max and min of ages
df.describe()

#The minimum age is 18 and the maximum is 70. We can create 6 bins to group people by
age group. Each bin could represent 10 years
bin_list=[10,20,30,40,50,60,70]
plt.hist(df['Age'], bins=bin_list, rwidth=0.9)
```

```
plt.xlabel('Age')

plt.ylabel('frequency')

plt.title('Age distribution of customers')


#Annual income of customers

plt.hist(df['Annual Income (k$)'], bins=12, rwidth=0.9)

plt.xlabel("Income in 1000's of $")

plt.ylabel("frequency")

plt.title('Annual income of customers')


#Spending score of the customers

plt.hist(df['Spending Score (1-100)'], bins=[0,10,20,30,40,50,60,70,80,90,100], rwidth=0.9)

plt.xlabel("Spending score")

plt.ylabel("frequency")

plt.title('Spending Score of customers')


#The elbow method shows that the optimal number of clusters for this data set is 5

from sklearn.cluster import KMeans

wcss = []

for i in range(1,11):

    k_means = KMeans(n_clusters=i,init='k-means++',random_state=42)

    k_means.fit(x)
```

```
        wcss.append(k_means.inertia_)


plt.plot(range(1,11),wcss)
plt.xticks(range(1,11));


#For this exercise, we will just look at 2 variables,
#income and spending score
X = df.iloc[:,[3,4]].values


#Using the dendogram to find optimal number of clusters
import scipy.cluster.hierarchy as sch
dendrogram = sch.dendrogram(sch.linkage(X, method='ward'))
#ward - minimize variance within each cluster
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Euclidean Distance")
plt.tick_params(axis='x',labelbottom=False)
plt.show()


#From the graph, we can identify that the ideal number of clusters
#is 5, so we will proceed with applying the Hierarchical Clustering
#algorithm for 5 clusters
```

```
from sklearn.cluster import AgglomerativeClustering

hc = AgglomerativeClustering(n_clusters=5, affinity='euclidean',linkage='ward')

y_hc = hc.fit_predict(X)


#Visualizing the clusters

plt.scatter(X[y_hc == 0, 0], X[y_hc == 0, 1], s=50, c='red', label='Sensible')

plt.scatter(X[y_hc == 1, 0], X[y_hc == 1, 1], s=50, c='blue', label='Standard')

plt.scatter(X[y_hc == 2, 0], X[y_hc == 2, 1], s=50, c='green', label='Target')

plt.scatter(X[y_hc == 3, 0], X[y_hc == 3, 1], s=50, c='cyan', label='Careless')

plt.scatter(X[y_hc == 4, 0], X[y_hc == 4, 1], s=50, c='magenta', label='Careful')

plt.title("Clusters of Clients")

plt.xlabel("Annual Income ($K)")

plt.ylabel("Spending Score (0-100)")

plt.legend()

plt.show()


#(1)K-means using 5 clusters

kmeans = KMeans(n_clusters=5,init='k-means++',n_init=10,max_iter=300,random_state=42)

ykmeans = kmeans.fit_predict(x)

ykmeans


#Visualix=ziing the clusters
```

```
df2 = df

df2['ykmeans'] = ykmeans

df2.head(10)

sns.scatterplot(x='Annual Income (k$)' , y = 'Spending Score (1-100)',data=df2 ,

hue='ykmeans',palette=['red','green','blue','black','brown'])


#let's say possible clusters are [3,4,5,6]

range_n_clusters = [3, 4, 5, 6]


for n_clusters in range_n_clusters:

# Create a subplot with 1 row and 2 columns

fig, (ax1, ax2) = plt.subplots(1, 2)

fig.set_size_inches(18, 7)


# The 1st subplot is the silhouette plot

# The silhouette coefficient can range from -1, 1 but in this example all

# lie within [-0.1, 1]

ax1.set_xlim([-0.1, 1])

# The (n_clusters+1)*10 is for inserting blank space between silhouette

# plots of individual clusters, to demarcate them clearly.

ax1.set_ylim([0, len(x) + (n_clusters + 1) * 10])
```

```
# Initialize the clusterer with n_clusters value and a random generator

# seed of 10 for reproducibility.

clusterer = KMeans(n_clusters=n_clusters, random_state=42)

cluster_labels = clusterer.fit_predict(x)


# The silhouette_score gives the average value for all the samples.

# This gives a perspective into the density and separation of the formed

# clusters

silhouette_avg = silhouette_score(x, cluster_labels)

print("For n_clusters =", n_clusters,"The average silhouette_score is :", silhouette_avg)


# Compute the silhouette scores for each sample

sample_silhouette_values = silhouette_samples(x, cluster_labels)


y_lower = 10

    for i in range(n_clusters):

        # Aggregate the silhouette scores for samples belonging to

        # cluster i, and sort them

        ith_cluster_silhouette_values = \

            sample_silhouette_values[cluster_labels == i]

            ith_cluster_silhouette_values.sort()
```

```
        size_cluster_i = ith_cluster_silhouette_values.shape[0]
        y_upper = y_lower + size_cluster_i

        color = cm.nipy_spectral(float(i) / n_clusters)
        ax1.fill_betweenx(np.arange(y_lower, y_upper),
                          0, ith_cluster_silhouette_values,
                          facecolor=color, edgecolor=color, alpha=0.7)

        # Label the silhouette plots with their cluster numbers at the middle
        ax1.text(-0.05, y_lower + 0.5 * size_cluster_i, str(i))

        # Compute the new y_lower for next plot
        y_lower = y_upper + 10    # 10 for the 0 samples

ax1.set_title("The silhouette plot for the various clusters.")
ax1.set_xlabel("The silhouette coefficient values")
ax1.set_ylabel("Cluster label")

# The vertical line for average silhouette score of all the values
ax1.axvline(x=silhouette_avg, color="red", linestyle="--")

ax1.set_yticks([])    # Clear the yaxis labels / ticks
```

```
ax1.set_xticks([-0.1, 0, 0.2, 0.4, 0.6, 0.8, 1])


# 2nd Plot showing the actual clusters formed

colors = cm.nipy_spectral(cluster_labels.astype(float) / n_clusters)

ax2.scatter(x[:, 0], x[:, 1], marker='*', s=30, lw=0, alpha=0.7,

                    c=colors, edgecolor='k')


# Labeling the clusters

centers = clusterer.cluster_centers_

# Draw white circles at cluster centers

ax2.scatter(centers[:, 0], centers[:, 1], marker='o',

                    c="white", alpha=1, s=200, edgecolor='k')


for i, c in enumerate(centers):

      ax2.scatter(c[0], c[1], marker='$%d$' % i, alpha=1,

                          s=50, edgecolor='k')


ax2.set_title("The visualization of the clustered data.")

ax2.set_xlabel("Feature space for the 1st feature")

ax2.set_ylabel("Feature space for the 2nd feature")


plt.suptitle(("Silhouette analysis for KMeans clustering on sample data "
```

```
                                "with n_clusters = %d" % n_clusters),

                               fontsize=14, fontweight='bold');


plt.show()

#As our intuation 5 would be the optimal number of clusters

kmeans = KMeans(n_clusters=4,init='k-means++',n_init=10,max_iter=300,random_state=42)

y4kmeans = kmeans.fit_predict(x)


df2['y4kmeans'] = y4kmeans


fig , (ax1,ax2) = plt.subplots(1,2)

fig.set_size_inches(18, 7)

sns.scatterplot(ax=ax1,x='Annual Income (k$)' , y = 'Spending Score (1-100)',data=df2 ,

hue='ykmeans',palette=['red','green','blue','black','brown'])

ax1.set_title("5 clusters");

## Now do the same thing with 4 clusters and compare

sns.scatterplot(ax=ax2,x='Annual Income (k$)' , y = 'Spending Score (1-100)',data=df2 ,

hue='y4kmeans',palette=['red','green','blue','brown'])

ax2.set_title("4 clusters");



#(2)Means Shift Algorithm
```

```
#Mean shift clustering involves finding and adapting centroids based on the density of
examples in the feature space.
#drop three title, only two main factors left
df.drop(['CustomerID'], axis=1, inplace=True)
df.drop(['Genre'], axis=1, inplace=True)
df.drop(['Age'], axis=1, inplace=True)
import seaborn as sns
sns.pairplot(df)
plt.figure(figsize=(10, 8))
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.scatter(df['Annual Income (k$)'],
                    df['Spending Score (1-100)'],)
plt.show()
from sklearn.cluster import MeanShift
from numpy import unique
model_ms = MeanShift(bandwidth=25)
model_ms.fit(df)
#
yhat_ms = model_ms.predict(df)
clusters_ms = unique(yhat_ms)
print("Clusters of Mean Shift.",clusters_ms)
```

```
#
labels_ms = model_ms.labels_
centroids_ms = model_ms.cluster_centers_
#
plt.figure(figsize=(10, 8))
plt.scatter(df['Annual Income (k$)'],
                    df['Spending Score (1-100)'],
                    c=labels_ms, s=100)
plt.scatter(centroids_ms[:,0], centroids_ms[:,1], color='red', marker='*', s=200)
plt.xlabel('Annual Income')
plt.ylabel('Spending Score')
plt.title('Mean Shift')
plt.grid()
plt.show()

from sklearn import metrics
score_ms = metrics.silhouette_score(df,labels_ms)
print("Score of Mean Shift = ", score_ms)



#(3)DBSCAN:DBSCAN Clustering (where DBSCAN is short for Density-Based Spatial
Clustering of Applications with Noise)
```

```
#involves finding high-density areas in the domain and expanding those areas of the feature
space around them as clusters.
# dbscan clustering
from numpy import unique
from numpy import where
data_X = df.iloc[:,[0,1]].values
# define the model
model = DBSCAN(eps=0.7, min_samples=90)
# fit model and predict clusters
yhat = model.fit_predict(data_X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
    # get row indexes for samples with this cluster
    row_ix = where(yhat == cluster)
    # create scatter of these samples
    plt.scatter(data_X[row_ix, 0], data_X[row_ix, 1])
# show the plot
plt.show()
#For this data, could not get a good result.
```