Springboard Data Science Intensive Program
Capstone Project #2

# Exploring Marketing Strategies for Mall Business Owners

By Remzi Kizilboga
July, 2020

## Problem Statement

Consider that you want to or already own a business in a mall. You probably have or will have customers every single day. You want to make changes or know more about your potential customers to better serve them and make more benefit by developing marketing strategies. However, you have no idea about the characteristics of your target customers. In an article, Ian Blair came up with 16 marketing strategies that business owners use. These strategies are:

1. Advertise on Facebook.
2. Rank your Google my business listing.
3. Use Google Adwords.
4. Invest in content marketing.
5. Grow your organic social reach.
6. Run a coupon deal.
7. Build an email marketing funnel.
8. Host a webinar.
9. Promote a free consultation
10. Offer staff incentives
11. Advertise in niche print media
12. Write a column
13. Join local business groups
14. Partner with other businesses
15. Direct mail marketing
16. Speak at events (https://buildfire.com/marketing-strategies-for-small-businesses/)

These are the strategies obtained from a single resource. When searched online, there are many more strategies because the marketing strategies can vary based on the type of business, work culture, the type of customers, systems, location, etc. (https://blog.simplemachinesmarketing.com/what-to-consider-when-changing-marketing-strategies). This study aims at bringing a perspective on marketing strategies for mall business owners based on different features, such as age, gender, annual income, and spending scores.

## Potential Stakeholders

The primary stakeholders of this study are the current or future business owners, who can use the model developed in this study to make decisions on marketing strategies. The secondary stakeholders are the customers who shop in malls. They are secondary because they will not be directly affected until business owners make actions and apply marketing strategies.

## Data and Deliverables

The was obtained from a Kaggle competition (https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python). There are five features in total with 200 data points. However, CustomerID will not be considered as a feature for the analysis because it is a label feature for customers. Figure 1 summarizes the first five data points in the dataset. The features are: gender, age, annual income in dollars, and spending score between 1 and 100. According to the Kaggle definition, 'Spending Score' is something you assign to the customer based on your defined parameters like customer behavior and purchasing data.

| | 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 |

Figure 1. Summary of the first five data points in the dataset

### Data Wrangling

For the data wrangling process, I first explored each feature separately. In terms of customers' genders, I found out that females are 56% while males were 44% of 200 customers in total. Similarly, while the maximum customer age was 70, the youngest age customer(s) was 18 years old. I wanted to categorize customers based on their ages and create labels for different ranges as seen in Figure 2.

```
In [9]: # Customer ages range between 18 and 70. I want to categorize customers based on their ages.
        #I will create labels for different ranges
        dfmallcustomer['Age'] = pd.cut(dfmallcustomer.Age, bins=[0, 25, 35, 60, 70],labels=['Young Adult','Adult',
                                                                                             'Middle Age','Older Adult'])
```

Figure 2. Python code for age categorization

After categorizing the age groups, I found out that 42.5% of customers are middle ages, which was between 35-60 years old. The other age groups and their percentages are: adults (25-35 years old), 30%; younger adults (0-25 years old), 19%; and older adults (60 years old and olders), 8.5%.

In terms of the annual income, the maximum income was $137k while the minimum income was $15k. Similar to what I did for age groups, I categorized the customers based on their annual income using lower, lower middle, middle, upper middle, and upper as the labels. As seen in Figure 3, the majority of the customers were in the middle or upper middle (64%).

```
In [15]: #minimum annual income is 15k while the maximum one is 137k.I want to group them as well based on their income
         #The categories I will use are: lower, lower middle, middle, upper middle, and upper
         dfmallcustomer['Annual Income (k$)'] = pd.cut(dfmallcustomer['Annual Income (k$)'],
                                                       bins=[0, 20, 35, 65, 85, 137],labels=['Lower','Lower Middle', 'Middle',
                                                                                             'Upper Middle','Upper'])

In [16]: dfmallcustomer['Annual Income (k$)'].value_counts(normalize=True, dropna=False)

Out[16]: Middle          0.40
         Upper Middle    0.24
         Upper           0.17
         Lower Middle    0.11
         Lower           0.08
         Name: Annual Income (k$), dtype: float64
```

Figure 3. Categorization and percentage of customers based on annual incomes

I applied the last categorization for the spending scores using the labels “low score, good score, very good score, and excellent score” (see Figure 4). The majority of the customers have either good scores or very good scores (25-75, 61.5%). The customers who have low score was 19.5% of the customers while 19% of the customers have excellent score.

```
In [18]: #I want to create 4 groups for the spending scores: 0-24 Low Score, 25-49 Good Score, 50-74 Very Good Score
         # 75-100 Excellent Score
         dfmallcustomer['Spending Score (1-100)'] = pd.cut(dfmallcustomer['Spending Score (1-100)'],
                                                           bins=[0, 25, 50, 75, 100],labels=['Low Score','Good Score',
                                                                                             'Very Good Score','Excellent Score'])

In [19]: #recheck the spencing scores
         dfmallcustomer['Spending Score (1-100)'].value_counts(dropna=False, normalize=True)

Out[19]: Good Score         0.320
         Very Good Score    0.295
         Low Score          0.195
         Excellent Score    0.190
         Name: Spending Score (1-100), dtype: float64
```

Figure 4. Categorization and percentage of customers based on spending scores

Finally, I checked the number of the null values before saving the dataset in CSV format and found out that there were no null values.

**Data Storytelling**

In order to understand data using visual graphics and plots, I first visualized what I did in data wrangling as seen in Appendix A. After those univariate analyses, I applied bivariate analyses. This section summarizes those bivariate analyses with their graphical representations.

The first question that I visualized was the age groups based on their genders. As seen in Figure 5, I found out that female customers were the majority for each age group except older adults.

```
In [14]: #age groups based on gender
sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)
sns.countplot(x='Age', hue='Gender',data=stmallcustomer)
plt.xlabel('Age group based on gender')
plt.ylabel('Number of users')
sns.despine()
```

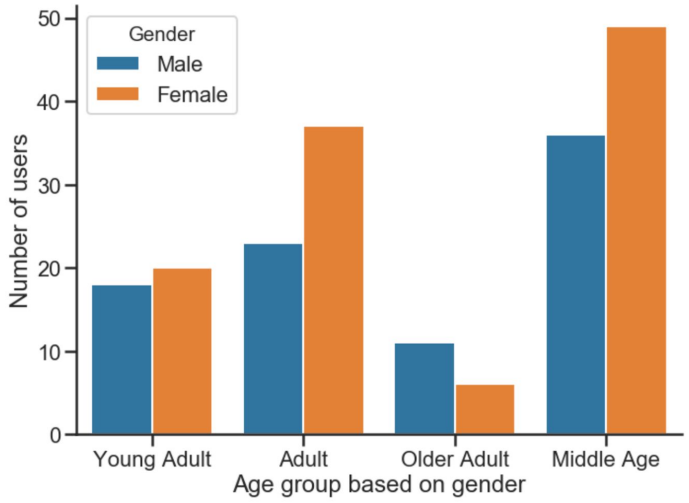


Figure 5. Age groups based on customer genders

Secondly, I checked the income groups based on customer genders as seen in Figure 6. Females were again the majority group in all income groups.

```
In [15]: #income groups based on gender
sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)
sns.countplot(x='Annual Income (k$)', hue='Gender',data=stmallcustomer)
plt.xlabel('Income group based on gender')
plt.ylabel('Number of users')
sns.despine()
```

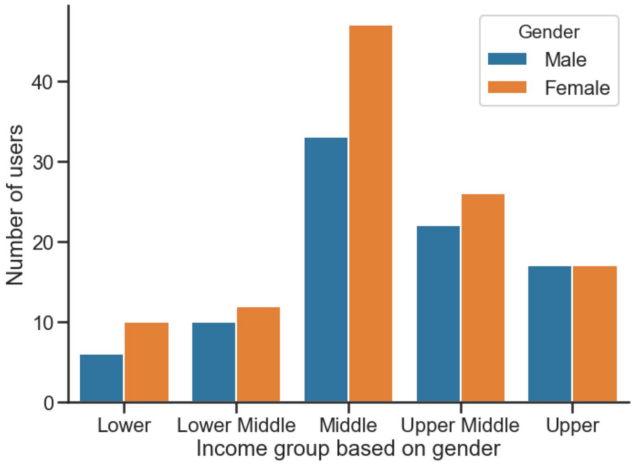


Figure 6. Income groups based on customer genders

Thirdly, I looked at the spending score groups based on genders. As seen in Figure 7, the majority of customers were female in all spending score categories except low score. These results was completely making sense because females were spending more than males so that in the low spending score, males were supposed to be the majority group.

```
In [16]: #spending score groups based on gender
         sns.set_style('ticks')
         fig, ax = plt.subplots()
         fig.set_size_inches(11, 8)
         sns.countplot(x='Spending Score (1-100)', hue='Gender',data=stmallcustomer)
         plt.xlabel('Spending scores based on gender')
         plt.ylabel('Number of users')
         sns.despine()
```

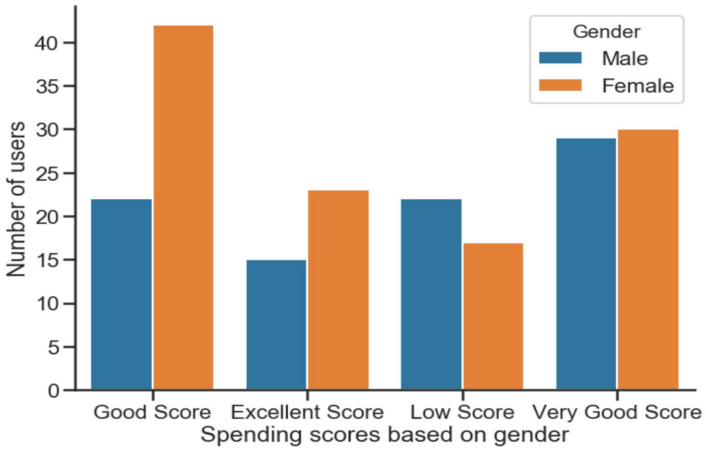


Figure 7. Spending score groups based on customer genders

Then, I checked annual income and spending score groups based on age groups. As seen in Figure 8, middle age customers had middle annual income. In the upper group, there were young adults and older adults. That totally makes sense because older adults might probably get retired while young adults either did not have a job or were in their early career.

```
In [18]: #Annual income based on age groups
sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)
sns.countplot(x='Annual Income (k$)', hue='Age',data=stmallcustomer)
plt.xlabel('Annual income based on age groups')
plt.ylabel('Number of users')
sns.despine()
```

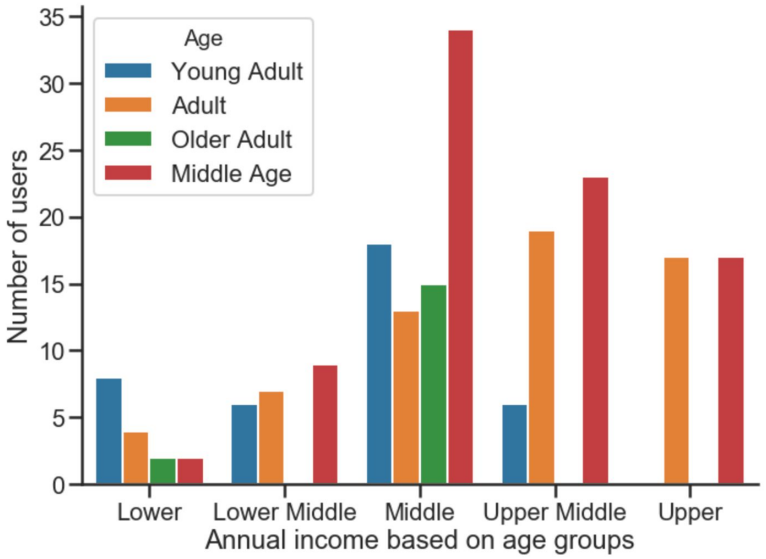


Figure 8. Annual income groups based on customer age groups

On the other hand, as seen in Figure 9, when I checked the spending score groups based on the age groups, I found out that older adults did not have excellent scores while adults had the best scores (very good and excellent score).

```
In [20]: #Spending scores based on age groups
sns.set_style('ticks')
fig, ax = plt.subplots()
fig.set_size_inches(11, 8)
sns.countplot(x='Spending Score (1-100)', hue='Age',data=stmallcustomer)
plt.legend(loc='upper right')
plt.xlabel('Spending Score based on age groups')
plt.ylabel('Number of users')
sns.despine()
```

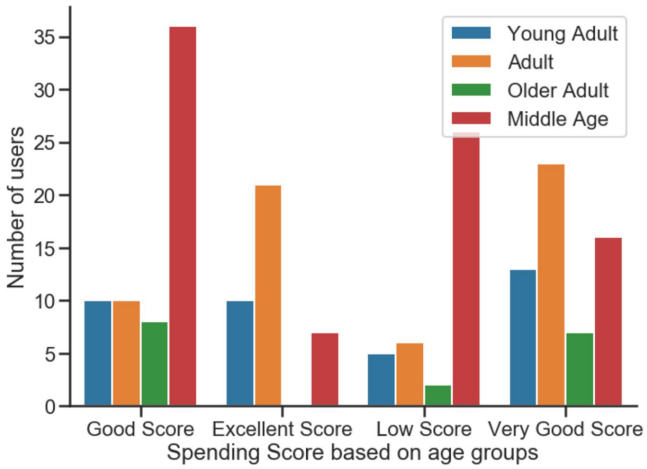


Figure 9. Spending score groups based on customer age groups

As the final step in data storytelling, I checked the correlations between the features using a correlation heatmap (see Figure 10). I found out that there was no significant correlation between the features so that I could include all features in my analyses.

```
In [28]: #check the correlation
         fig.set_size_inches(11, 8)
         sns.heatmap(dfmallcustomer.corr(), cmap='Accent', annot = True)
         plt.title('Heatmap for the Variables', fontsize = 28)
         plt.show()
```

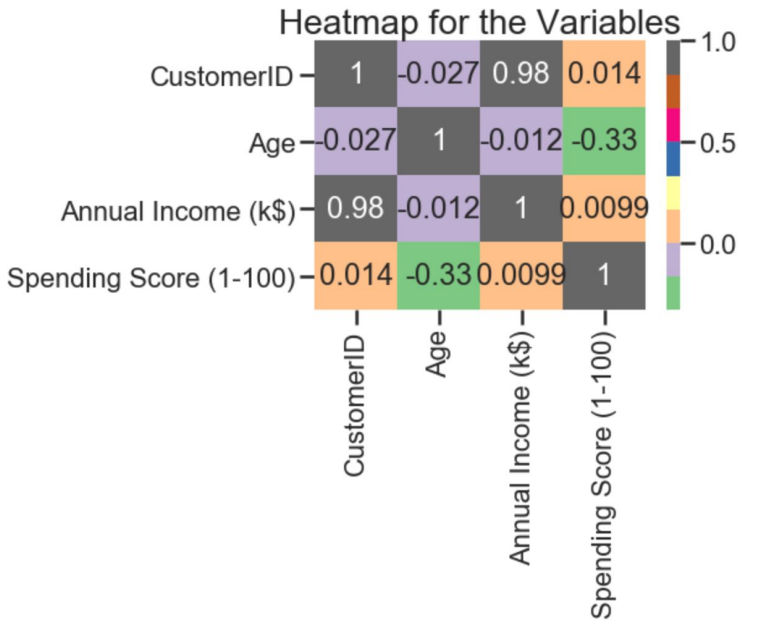


Figure 10. Correlation heatmap between features

**Inferential Statistics**

In terms of the inferential statistics, I applied hypothesis tests using Student-T test and Bootstrap methods. For the student t-test, the null hypothesis was there was no significant difference in spending scores between males and females while the alternative hypothesis was there was a significant difference in the spending scores between males and females. I first checked the mean and standard deviation scores for females and males. As seen in Figure 11, the mean and standard deviation scores for males were about 48.5 and 27.9 respectively. On the other hand, females mean and standard deviations scores were 54.5 and 24.1 respectively. And, the difference between the means of the spending scores of males and females was about 3.02. The standard error value was about 0.0565.

```
In [6]: #calculate the mean and standard deviations of males
male_ss_mean = male['Spending Score (1-100)'].mean()
print(male_ss_mean)

male_ss_std = male['Spending Score (1-100)'].std()
print(male_ss_std)
```

```
48.51136363636363
27.896769605833597
```

```
In [7]: #calculate the mean and standard deviations of females
female_ss_mean = female['Spending Score (1-100)'].mean()
print(female_ss_mean)

female_ss_std = female['Spending Score (1-100)'].std()
print(female_ss_std)
```

```
51.526785714285715
24.114949877478647
```

```
In [8]: #difference between the mean of the spending scores of males and females
mean_difference= female_ss_mean - male_ss_mean
print("The difference between the means of the spending scores of males and females is ", mean_difference)
```

```
The difference between the means of the spending scores of males and females is  3.015422077922082
```

Figure 11. Means and standard deviations of females and males based on spending scores

After calculating the t and p values manually and using scipy (see Figure 12), I found out that the p-value was higher than 0.05 so that I decided not to reject the null hypothesis and say that there was not a significant difference between the means of males and females ages. This result aligned with the results that I found in the data storytelling process because even though there seemed to be a difference between females and males in terms of their spending score, the difference was not significant.

```
In [12]: #calculate t value manually
n0 = len(male_ss)
n1= len(female_ss)
std0 = male_ss.std()
std1= female_ss.std()
mean0 = mean_male
mean1= mean_female
sp = np.sqrt( ((n0-1)*(std0)**2 + (n1-1)*(std1)**2)/ (n0+n1-2) )
t_ = (mean1 - mean0)/(sp * np.sqrt(1/n0 + 1/n1))
print(t_)
```

```
0.8190464150660334
```

```
In [13]: # Use 0.05 Significance level in two sample t-test
t_val=((male_ss_mean - female_ss_mean)-0)/SE
print(t_val)
```

```
-53.34416477675928
```

```
In [14]: #calculate p value manually
p_value = (1 - t(n0 + n1 - 1).cdf(t_)) * 2
p_value
```

```
Out[14]: 0.4137397159674374
```

```
In [15]: #calculate t and p values using scipy
ttest_ind(male_ss, female_ss)
```

```
Out[15]: Ttest_indResult(statistic=-0.8190464150660333, pvalue=0.4137446589852176)
```

Figure 12. Calculating t and p-values manually and using scipy

For the bootstrap and confidence interval implementation, I listed the Null and Alternative Hypothesis as follows:

Ho: there is no difference in standard deviations between males and females.
Ha : there is a difference in standard deviations between males and females.

I found the 95% confidence interval of the difference in standard deviation between two groups as [-0.16439370371649292 , 7.500105521493774], which I interpreted as 10000 Bootstrap replicates with a 95% confidence interval indicated that the difference in standard deviations between the two groups had a 95% chance of lying within [-0.16439370371649292 , 7.500105521493774].

Then, I shifted the dataset so that two groups had equal means by first calculating the combined mean and then generating the shifted dataset. Drawing the bootstrap replicates from the shifted dataset, I got the observed difference from the actual dataset, which had a value of -3.015422077922082. Finally, I calculated the p-value by comparing the bootstrap replicates against the observed difference of the means. I found the p-value as 0.7956 which I interpreted that it was sufficiently likely that the null hypothesis was true and thus I retained the null hypothesis. There was no significant difference in spending scores between females and males (see Figure 13).

```
In [25]: # # Shifting the Dataset so that the two groups have equal means

         # First calculating the combined mean
         combined_mean = np.mean(np.concatenate((male_bts, female_bts)))

         # Generate the shifted dataset
         male_shifted = male_bts - np.mean(male_bts) + combined_mean
         female_shifted =  female_bts - np.mean(female_bts) + combined_mean

In [27]: # Draw the bootstrap replicates from the shifted dataset
         bs_replicates_male = draw_bs_reps(male_shifted, np.mean, size=10000)
         bs_replicates_female = draw_bs_reps(female_shifted, np.mean, size=10000)

In [28]: # Get the differences for the bootstrap simulated sample
         bs_differences = bs_replicates_male - bs_replicates_female


         # Get the observed difference from the actual dataset
         obs_diff = np.mean(male_bts) - np.mean(female_bts)
         obs_diff

Out[28]: -3.015422077922082

In [29]: # Calculate the p-value by comparing the bootstrap replicates against the observed difference of the means
         # The fraction of values WITHIN bootstrap replicates array that meet a certain criteria against the obs_diff

         p = np.sum(bs_differences >= obs_diff)/ len(bs_differences)
         print('p-value =', p)

         p-value = 0.7956
```

Figure 13. Calculating p-value using bootstrap and confidence interval method

**Model Application**

In this study, I used an unsupervised learning model because I drew inferences from the provided dataset consisting of input data without labeled responses. I first clustered the data using the K-Means method to explore the number of clusters. Finally, I created stories based on each cluster to offer marketing strategies for business owners.

In order to explore the number of clusters, I applied an elbow method. As seen in Figure 14, the analysis showed that five clusters could be a reasonable number. The elbow method was applied on the numerical variables which are age, annual income, and spending scores.

```
In [6]: #elbow method. resources: https://medium.com/@ODSC/unsupervised-learning-evaluating-clusters-bd47eed175ce
        #https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

        wcss = [] #Within Cluster Sum of Squared Errors
        for k in range(1,10):
            kmeans = KMeans(n_clusters = k, init = 'k-means++', max_iter = 300, tol=0.0001, verbose=0, n_init = 10, random_state = 0)
            kmeans.fit(x)
            wcss.append(kmeans.inertia_)
        plt.figure(figsize=(12,8))
        plt.plot(range(1,10), wcss, linewidth=2, color="blue", marker ="8")
        plt.title('The Elbow Method', fontsize = 20)
        plt.xlabel('Number of Clusters')
        plt.ylabel('wcss')
        plt.xticks(np.arange(1,10,1))
        plt.show()
```

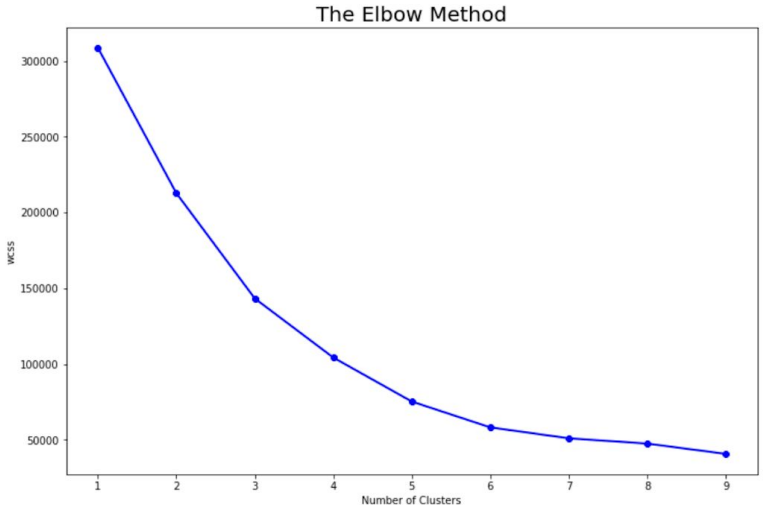


Figure 14. The elbow method with its python code

After deciding on five clusters, I used k-Means clustering method to see the distribution of each cluster on a 3 dimensional plot (see Figure 15).

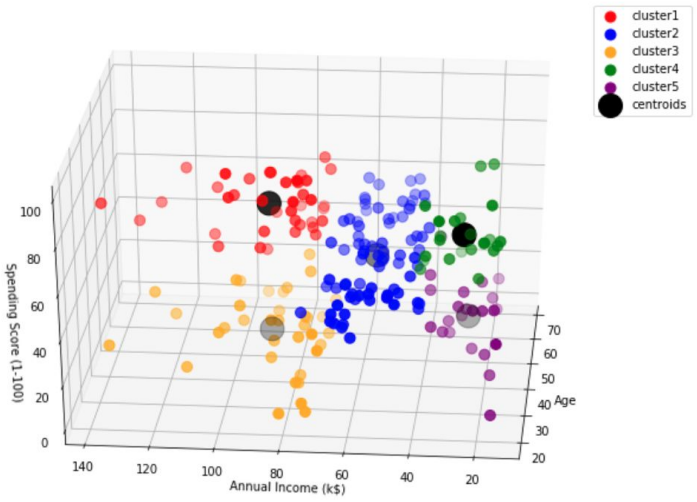


Figure 15. Distribution of each of five clusters based on KMeans clustering
I interpreted each cluster as follows:

- **Cluster1 (red cluster):** Generally people older than 40 years old with lower annual income and lower spending score.
- **Cluster2 (blue cluster):** Generally people between 20-50 years old with more than 80k annual income and moderate spending score.
- **Cluster3 (orange cluster):** These are the people who are spreaded in terms of age. It seems that their centroids are about 60 years old. But they have higher annual income (80k and more) but don't spend too much.
- **Cluster4 (green cluster):** These customers do not have either very high or low annual income. They are spreaded to all ages. They also do not have either very high or low spending scores.
- **Cluster5 (purple cluster):** These customers have the lowest population. They are younger with lower annual income. Their spending score is moderate.

However, I realized that some of the data points in some clusters overlapped with the data points in another cluster. After rerunning the model with 4 and 6 clusters, I decided to apply another clustering algorithm. I utilized a hierarchical clustering algorithm and found out that the model could give me better results with 3 clusters and as seen in Figure 16, the results with three clusters were better than the results with five clusters.

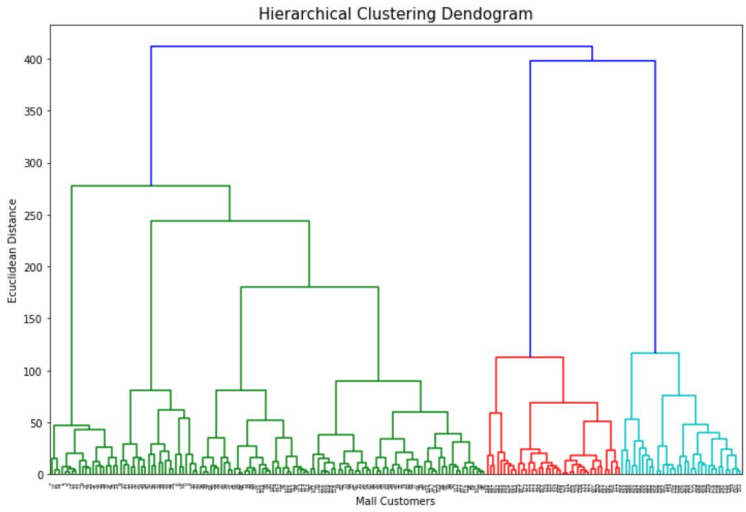


Figure 16. Hierarchical clustering dendogram

Rerunning a k-Means clustering with three clusters(see Figure 17), even though some blue and orange data overlap, three clusters were also as clear as five clusters model. Based on the plot in Figure 17, I would say that the mall business owners should target the blue cluster who represent younger people with higher annual income. This was quite similar to what I found with five clusters but more obvious than that.

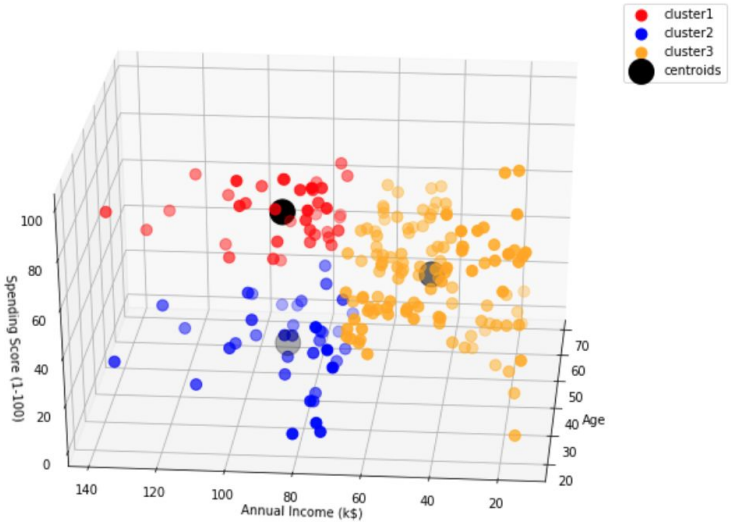


Figure 17. KMeans clustering with three clusters

Finally, I explored each cluster separately. Here are the boxplots for each cluster and their interpretations:

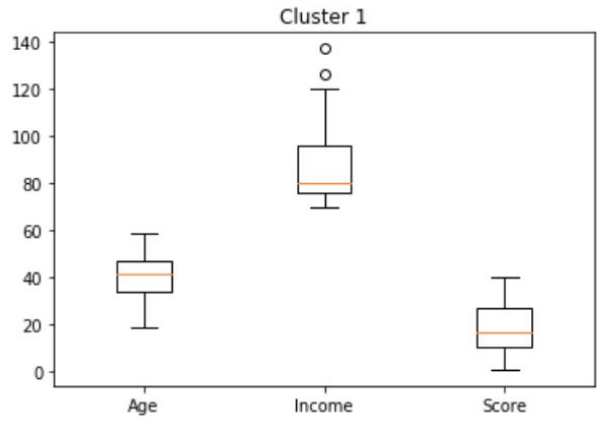


Figure 18. Boxplot for cluster 1

As seen in Figure 18, there were only two outliers in the income group. This group's ages range between 20 and 40 while the majority is between 20 and 30 years old so that this group represented the younger people. They mostly had a higher income of 70k and more. The median of income is about 80k and the majority of customers have a higher income than 80k. Also, their spending scores were higher compared to other groups and the median of the spending score was over 80.

On the other hand, for the second cluster as seen in Figure 19, the ages ranged between 18 and 70 years old. The median and mean were very close to each other. So, the median age was about 40 years old. Also, the majority of customers spreaded between 30 and 50 years old. These customers had moderate income between about 20k and 70k. The majority of them was earning about 30k and 60k per year. On the other hand, their spending scores varied and there were many outliers in this group. However, the median and mean in this group were also identical and about 50.

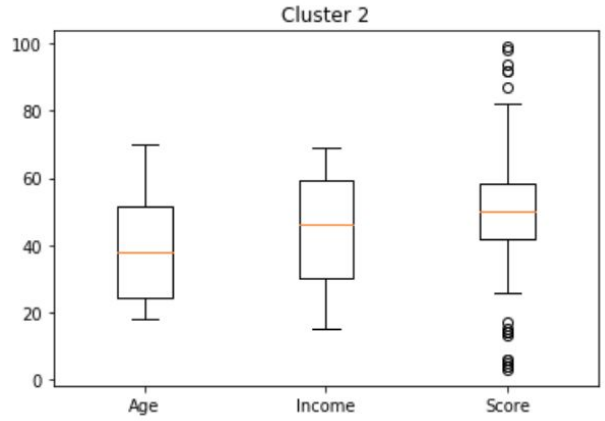


Figure 19. Boxplot for cluster 2

Finally, the customers in cluster 3 were very young. They were younger than 40 years old and the majority of them were even between 20 and 35 years old. In terms of income, they had 70k and more. There were only two outliers who earned more than 120k. On the other hand, the people in this cluster spent a lot. The majority of the customers in this group had a spending score between 70 and 85 (see Figure 20).

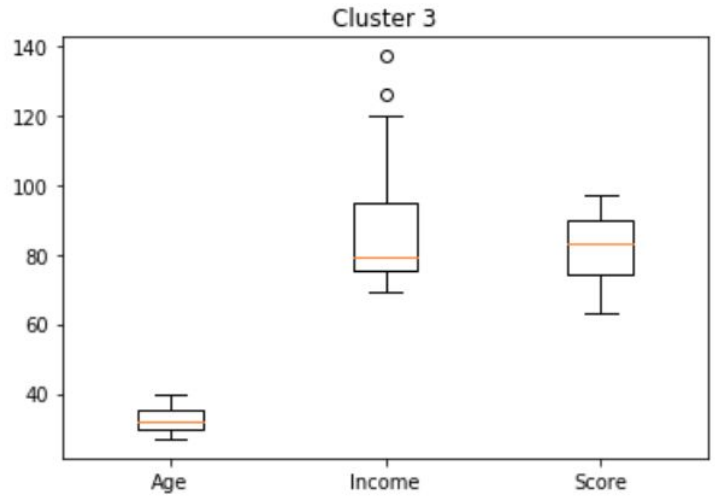


Figure 20. Boxplot for cluster 3

## Conclusion and Future Work

Looking at the initial elbow method, I thought that 5 clusters should be the optimal number of the clusters. However, I still saw some overlap in some of the clusters. Applying a hierarchical clustering algorithm, I was able to better understand the number of the clusters. The hierarchical clustering dendogram showed me that 3 clusters could be a better choice. When I reapplied k-means clustering with 3 clusters, I got better results. However, interestingly, one of the clusters did not change at all. This cluster was the cluster 2 in the second algorithm while cluster 5 in the first algorithm. The number of data points didn't change at all. Finally, I explored each cluster separately using box plots where the results showed that the second group of people (younger people with the ages of 20-40 years old, and with moderate to high annual income) could be the target customers for the mall business owners. In this clustering analysis, hierarchical clustering method was a better solution to find out the optimal number of clusters. To access the entire analysis process, you can visit https://github.com/remzikboga/springboard_datascience/tree/master/Capstone_Project2.

Future work might focus on an alternative clustering strategy, such as DBSCAN because DBSCAN is a lovely clustering algorithm which doesn’t need a parameter, k. It would be possible

to develop better strategies using a stronger model when the analysts should be provided with a larger dataset.

## Recommendation for Clients

1. People in their 30s should be the main target customer group because both they earn and spend more than other people.
2. Even though female customers are the majority group, there is no significant difference between their spending scores and male customers' spending scores. It could be a waste of time to invest based on gender.
3. In order to provide more insight and create better marketing strategies, mall business owners should provide more data.

## Resources Used

- **Matplotlib:** https://matplotlib.org/
- **Pandas:** https://pandas.pydata.org/
- **Pickle:** https://docs.python.org/3/library/pickle.html
- **Seaborn:** https://seaborn.pydata.org/
- **Sci-kit Learn:** https://scikit-learn.org/stable/
- **k-Means:** https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

**Appendix A. Plots for customers' age, annual incomes, and spending scores groups**