Final Report

Data Visualization

Mall Customer Segmentation Dataset

Student Name : Manuel Salgueiro Matilla
Student ID : 110021200
Instructor : Thi-Van Nguyen

2024-06

Chapter 1: Introductions

This semester we've been introduced to new tools for assisting with the visualization of relevant data and patterns within any given dataset we might come across, from Python libraries to complete Desktop applications with powerful tools for different reports and displays.

The aim for the final report of our Data Visualization course was to find a dataset that could be worth analyzing, and use the aforementioned tools to be able to successfully display the data in a comprehensive way that allows us to preprocess the data, compare it after training and testing it, in order to draw conclusions about it.

There was a variety of datasets and tools from which to choose from, as the internet is filled with repositories with databases, tables and spreadsheets to choose from, and we have several of the tools necessary to work on them at our disposal.

In my case, I chose to work on the Mall Customers Segmentation dataset and Python Jupyter Notebooks as the tool for display.

To achieve the goal, we will try to find what features within the data are worth exploring, if they tell us much information about the nature of the domain, if there's correlation between its features, if there's a way to cluster the data points to find useful patterns, then make before/after comparisons and reflect on the gains of doing proper preprocessing before testing our models against data.

Chapter 2: The Project

The Dataset

The dataset is Mall_Customer.csv, a light-weight csv file with information on 200 mall customers, providing the following fields:

- Gender: Male or Female values.
- Age: numerical, from 18 to 70 years old.
- Annual Income: numerical, from 15 to 137 (x1000 in USD).
- Spending Score: numerical, from 0 to 100.

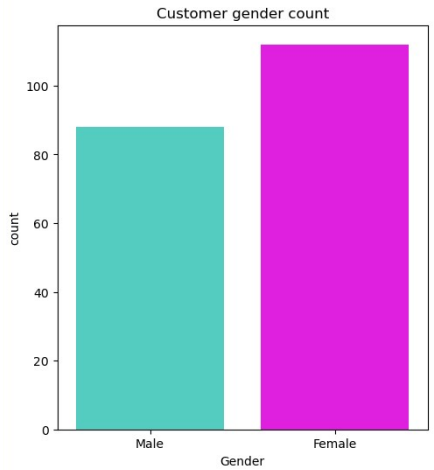
CustomerID	Gender	Age	Annual Income	Spending Score
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40
6	Female	22	17	76
7	Female	35	18	6
8	Female	23	18	94
9	Male	64	19	3

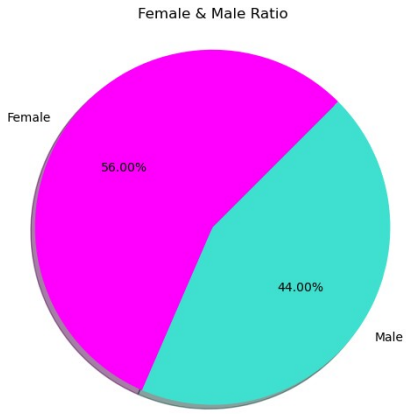
Extract from dataset

Features

Gender feature

First we try to look for interesting variances that set the data apart. We deploy the data in different charts to understand these patterns:

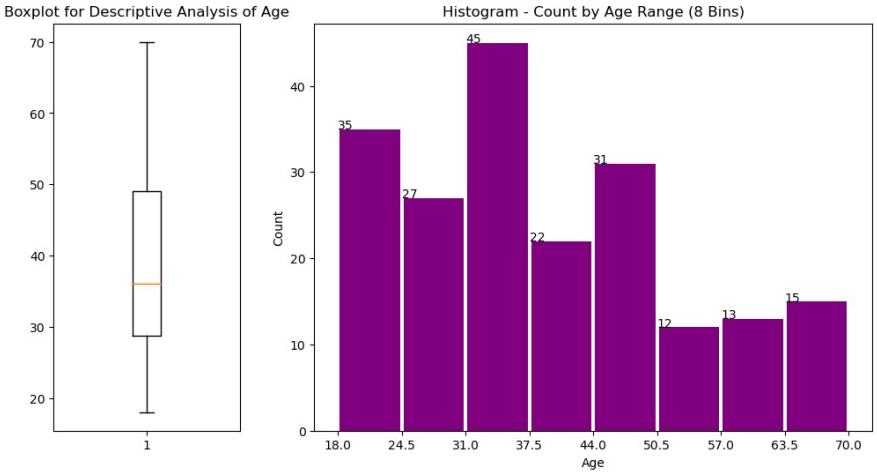




For starters, we see there's more representation of female customers than male. This could have implications but it's binary data and we don't get much insight though the graphs.

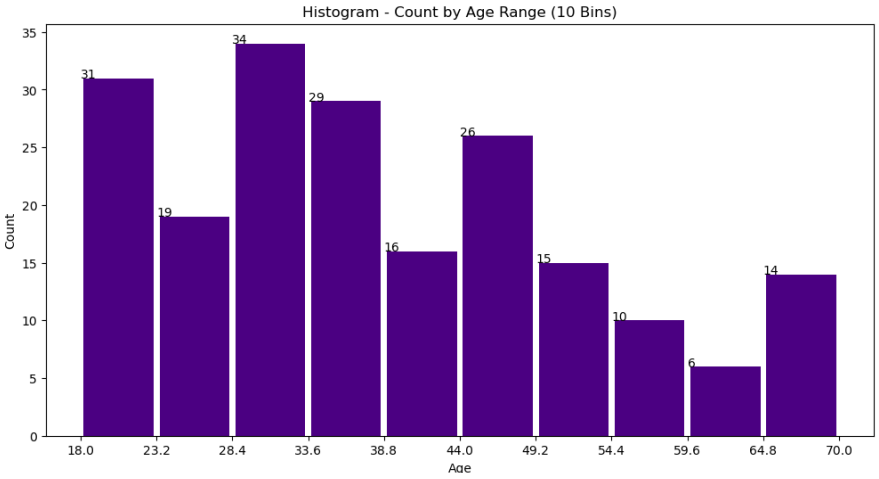
Age feature

What about other features like age? Let's put in context with a boxplot to check relevant data about the distribution of this feature, and let's display it in a histogram separated in 8 bins (I thought it's adecuate considering the nature of human life-cycles).



We can see in the boxplot Q1 and Q3 are noticeably skewed towards the younger ages, having Q2 (the mean) at 38.85 years old, which is consistent with the skewdness. The histogram shows clearly there's over-representation of customers under the age of 50, with peaks between 31.0 to 37.5 years old, whereas there's underrepresentation of elder customers.

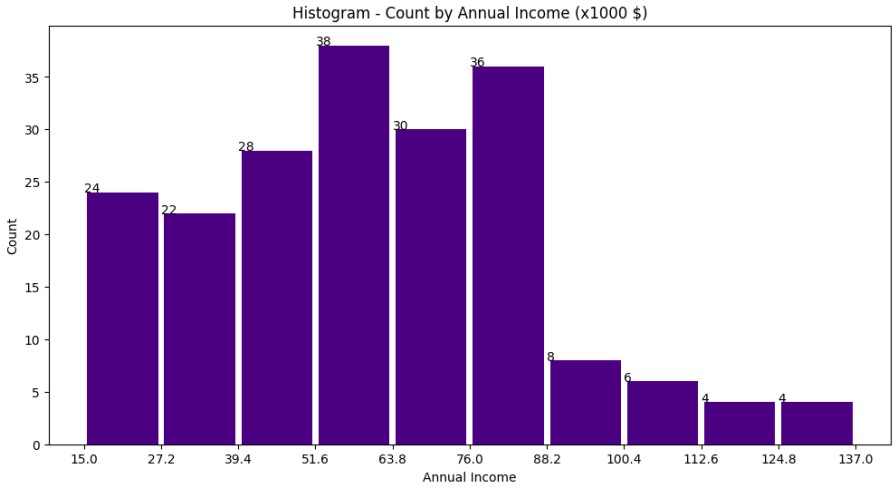
I decided to split the histogram into more bins to see if I could get more detail on each range while not compromising the visual comprehensiveness:



The result with 10 bins is the distribution maintains it's skewness to the younger ages, while spreading the count relieving some ranges from over-representation. At the same time we get more detail about the under-representation of some ranges among the elder customers. I will keep the 10-bin split for the next histograms.

Annual Income

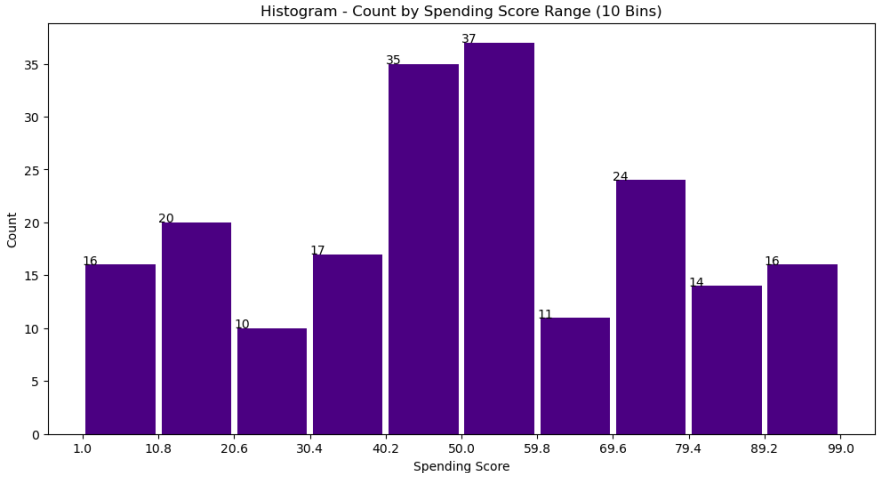
The next feature to analyze will be the Annual Income of Customers. This feature is really informative on the customer's purchasing power, although it doesn't necessarily inform about the customers spending habits:



We can observe the distribution is sensibly skewed towards the lower and apecially middle income, which gives us insight on how the population that visit malls in this region consist mostly in middle and working class people. The mena sits at 60.56 and the median at 61.5.

Spending Score

Lastly, let's analyze the spending score of the customers.



In this histogram, we don't see any significant pattern, the distribution is irregular. This is any way the target feature of our experiment, as it provides relevant information regarding the domain knowledge when put against other features.

Correlation and Clustering

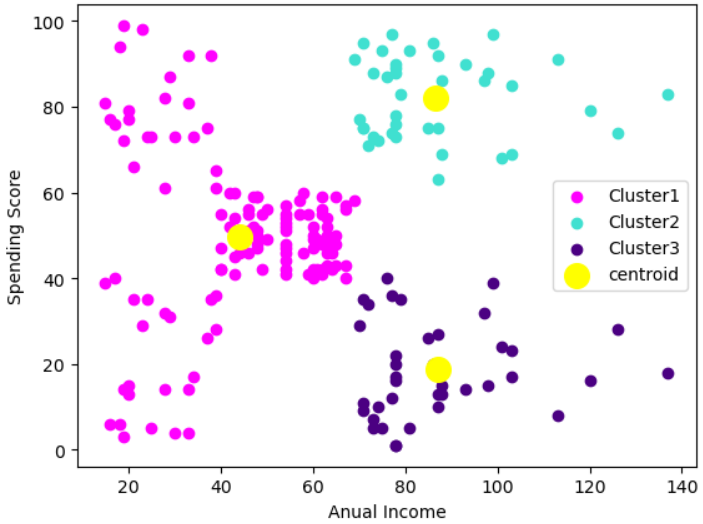
Annual Income vs Spending Score

First let's see the correlation Annual Income and Spending Score through a correlation matrix:

	Annual Income	Spending Score
Annual Income	1.00000	0.009903
Spending Score	0.009903	1.00000

It seems like there's no correlation between both features, which is a surprising finding, as one would expect income to be a driver for spending in many cultures.

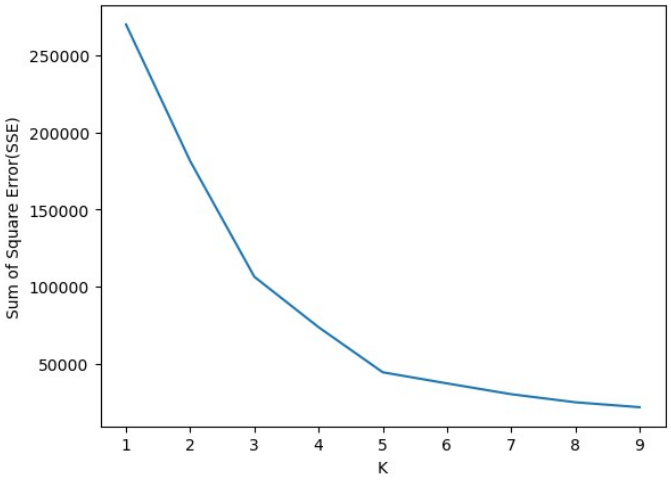
Let's try to view this correlation in a different shade. Considering in economy populations are usually split into three socioeconomic status (working, middle and upper class), let's put the Annual Income feature against the Spending Score feature together to find Spending patterns among the different socioeconomic segments through the use of K-Means, with K= 3.



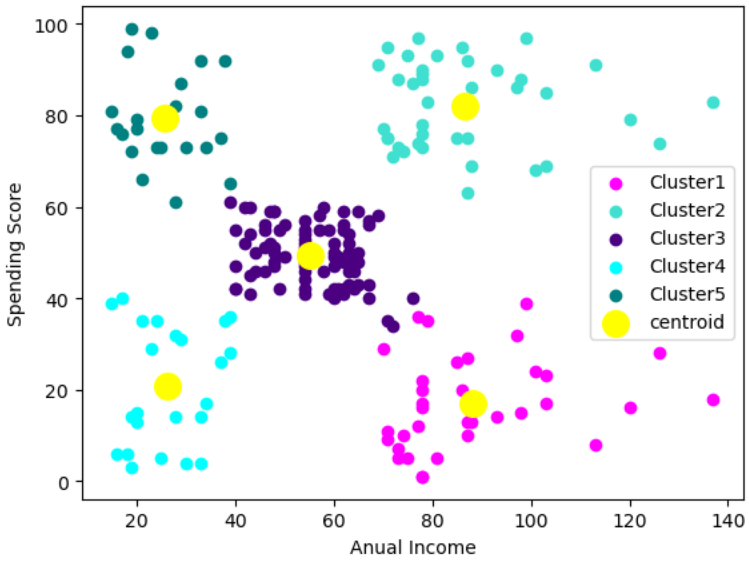
The first insight when crossing both features is the surprising disconnection of the spending for middle to upper incomes: Their spending is either too low or too high. Something similar is captured for the lower income data points, they group in the same fashion. Meanwhile the incomes around the mean (60.56) have an average spending score, and are all clustered in that area. We find K-Means is correctly identifying two obvious clusters Cluster 2 and 3, but at K=3 there are groups of data points in the lower income ranges that get drawn into the Cluster 1 when they should be their own cluster.

We keep the results of this clustering for each customer in the dataset under cluster_AI_KM3.

Using the measurement for the Sum of Square Error SSE in a graph for different values of K, via the rule of the Elbow we determine the best value of K for this prediction:



The turning point sits at K = 5. Let's do the same clustering experiment with that value for K:



We can observe K-Means can correctly identify the 2 clusters on the lower income ranges at K = 5. We keep the results of this clustering for each customer in the dataset under cluster_AI_KM5.

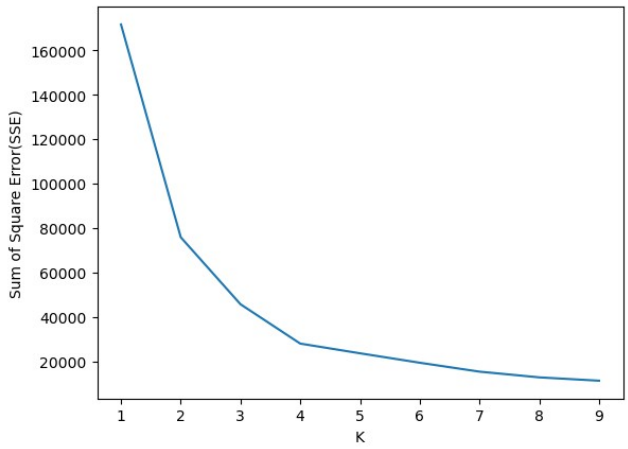
Age vs Spending Score

Now let's analyze the Spending Score against the Age of the Costumers. The correlation matrix throws the next results:

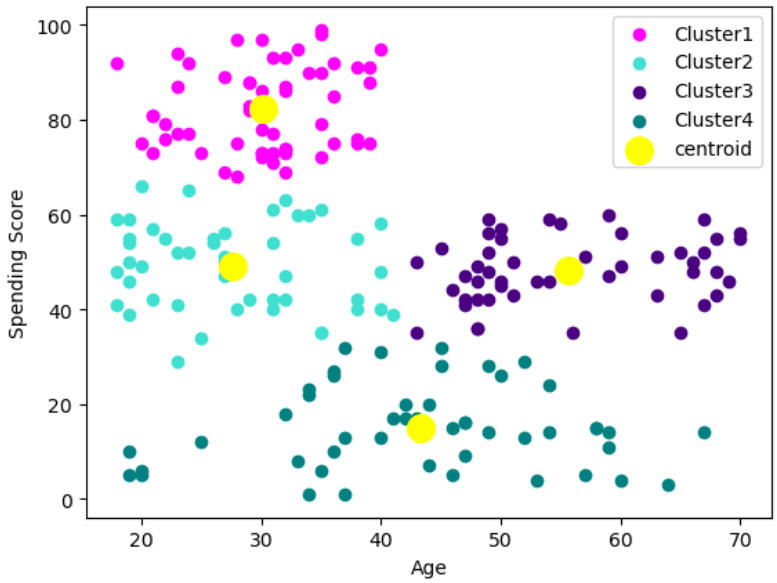
	Age	Spending Score
Age	1.00000	-0.327227
Spending Score	-0.327227	1.00000

Actually what we see is there might be a negative correlation between Age and Spending Score, which is something we can work with. We are expecting younger people to have better spending score.

Let's find the best value of K before-hand:



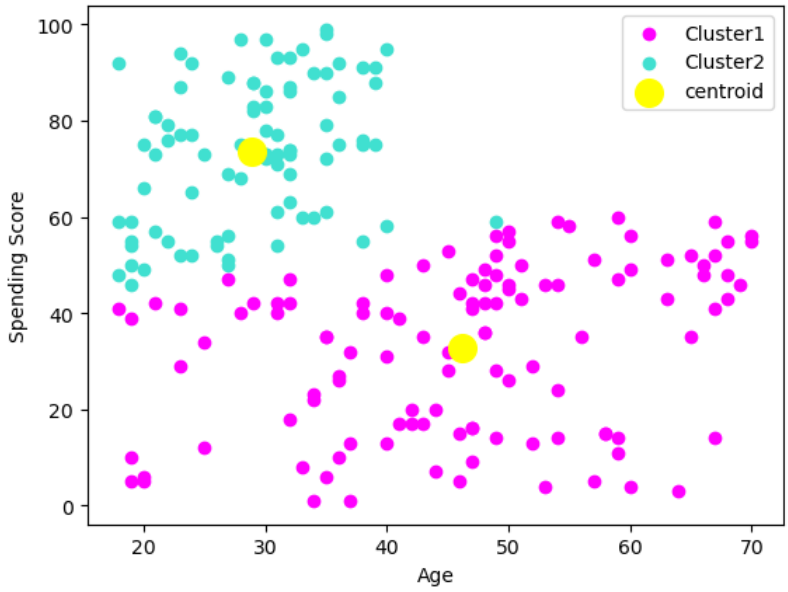
Following the Elbow Rule, the turning point seems to be K = 4. These are the results of K-Means clustering:



As we short of predicted with the correlation matrix, we can observe that the negative correlation comes from young people being the highest spending customers, that group being 100% of the customers with a Spending Score higher than 66. The rest of the sample seems normally distributed, with the lowest Spending Scores slighlty skewed on to the older ages, considering the mean for Age sits at 38.85 years.

We keep the results of this clustering for each customer in the dataset under cluster_Age_KM4.

K = 4 does identify well several clusters, although I want to see if it can identify well the data points between two clusters: elder people with a lower Spending Score and younger customers with a higher one. We'll use K = 2 for this purpose:



The algorithm does mostly a good job, although I think it gives Spending Score a lot of weight on the clustering calculation. In this sense, a data point being under or over 50 % of Spending Score, has more weight at deciding the cluster it belongs to than Age and proximity to another centroid.

We keep the results of this clustering for each customer in the dataset under cluster_Age_KM2.

Model Performance

I chose Linear Regression and Lasso Regression within sk-learn, the simplest Regression model to perform tests before and after preprocessing the data (clustering). These are the results of the training and testing before clustering, after clustering, and after clustering and normalization of numerical values, expressed in MSE (Mean Squared Error).

Before Clustering

The dataset:

Gender	Age	Annual Income	Spending Score
0	19	15	39
0	21	15	81
1	20	16	6
1	23	16	77
1	31	17	40
...	...	...	...
1	35	120	79
1	45	126	28
0	32	126	74
0	32	137	18
0	30	137	83

Linear Regression:

```
X = data[['Age', 'Annual Income', 'Gender']]
y = data['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.4s

```
Mean Squared Error: 480.673141707248
```

Lasso Regression Model:

```
X = data[['Age', 'Annual Income', 'Gender']]
y = data['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = Lasso()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.4s

```
Mean Squared Error: 483.29564616568894
```

The MSE is high before clustering and normalization of data for both models.

After Clustering

We fit and predict the model using the clustered data we obtained when clustering for Age and Income

The dataset:

Gender	Age	Annual Income	Spending Score	cluster_AI_KM3	cluster_AI_KM5	cluster_Age_KM4
0	19	15	39	0	2	1
0	21	15	81	0	0	0
1	20	16	6	0	2	3
1	23	16	77	0	0	0
1	31	17	40	0	2	1
...	...	...	...	...	...	...
1	35	120	79	1	1	0
1	45	126	28	2	3	3
0	32	126	74	1	1	0
0	32	137	18	2	3	3
0	30	137	83	1	1	0

Linear Regression:

```
X = data_pred[['cluster_AI_KM5', 'cluster_Age_KM4', 'Gender']]
y = data_pred['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.3s

```
Mean Squared Error: 134.79983739090875
```

Lasso Regression Model:

```
X = data_pred[['cluster_AI_KM5', 'cluster_Age_KM4', 'Gender']]
y = data_pred['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = Lasso()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.4s

```
Mean Squared Error: 131.9806690846081
```

The MSE is significantly lower than in the trial with no clustering, for both models. Lasso Model has an edge over the Linear Regression Model..

After Clustering and Normalization

Following from the previous prediction, we now normalize all numerical data into values ranging from 0 to 1.

The dataset:

Gender	Age	Annual Income	Spending Score	cluster_AI_KM3	cluster_AI_KM5	cluster_Age_KM4
0.0	0.019231	0.000000	0.387755	0.0	0.50	0.333333
0.0	0.057692	0.000000	0.816327	0.0	0.00	0.000000
1.0	0.038462	0.008197	0.051020	0.0	0.50	1.000000
1.0	0.096154	0.008197	0.775510	0.0	0.00	0.000000
1.0	0.250000	0.016393	0.397959	0.0	0.50	0.333333
...	...	...	...	...	...	...
1.0	0.326923	0.860656	0.795918	0.5	0.25	0.000000
1.0	0.519231	0.909836	0.275510	1.0	0.75	1.000000
0.0	0.269231	0.909836	0.744898	0.5	0.25	0.000000
0.0	0.269231	1.000000	0.173469	1.0	0.75	1.000000
0.0	0.230769	1.000000	0.836735	0.5	0.25	0.000000

Linear Regression:

```
X = df[['cluster_AI_KM5', 'cluster_Age_KM4', 'Gender']]
y = df['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.6s

```
Mean Squared Error: 0.014035801477603999
```

Lasso Regression Model:

```
X = df[['cluster_AI_KM5', 'cluster_Age_KM4', 'Gender']]
y = df['Spending Score']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = Lasso()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
```

✓ 0.2s

```
Mean Squared Error: 0.05781503800499791
```

We get an impressively (and suspiciously) low MSE for both Models when normalizing the data. It's the most accurate prediction, with the Linear Regression Model having an edge over the Lasso Model.

Chapter 3: Conclusion

This dataset's data being a small sample of a population seems to be providing conclusions that are not definitive towards analyzing given population: There are important gaps when correlating features, like the Income against the Spending Score, that support this dataset is manipulated or really insufficient.

Regarding the clustering, the dataset does have recognizable patterns between features that can be easily identified via clustering algorithms like K-Means. It's easy to observe how clustering segments the sample depending on the correlation taken, allowing to draw further conclusions, always in the dataset's doubtful terms.

After clustering and putting a Linear Regression model to test, we observed that preprocessing had a great impact on the results of the prediction: through clustering several features against the target class, we get correlated data that apparently helps with predicting future observations of given dataset.