Report: Customer segmentation ML

Table of contents

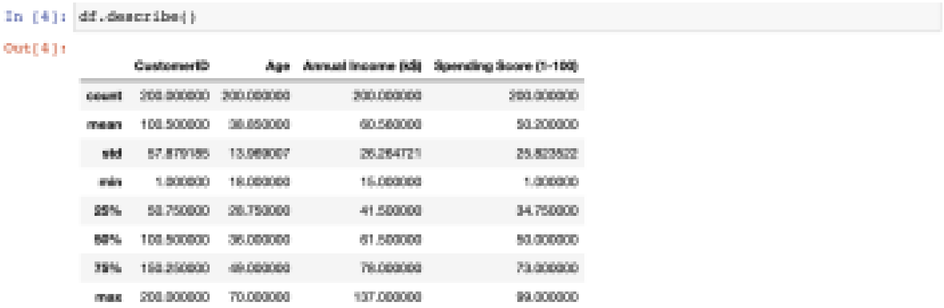
1.0 Data Exploratory analysis **1**

2.0 Clustering algorithm design **7**

1.0 Data Exploratory analysis

Different methods of descriptive statistics

As retrieved by df. describe(Fig1), the data frame measures the central tendency via mean values of all columns. It also measures data dispersion through standard deviation, minimum and maximum value of column data. The variation of Spending score versus Income is also shown in the boxplot. The range of spending score is higher than the annual income range(Fig2).



```
In [4]: df.describe()
Out[4]:
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Figure 1. *Mean, std, min and max of data*

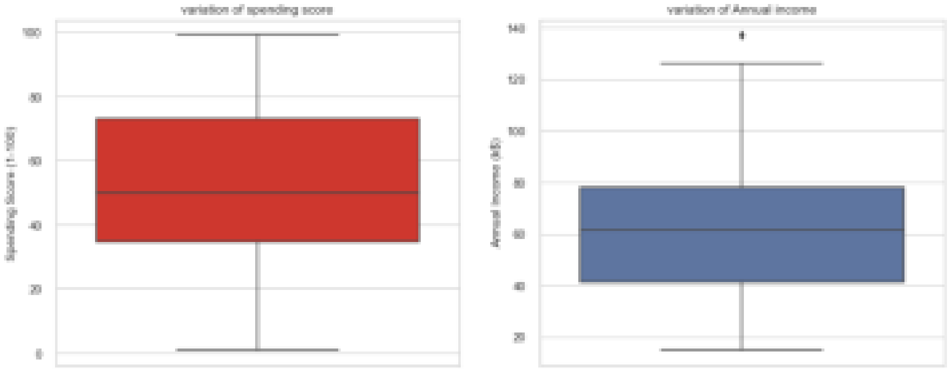
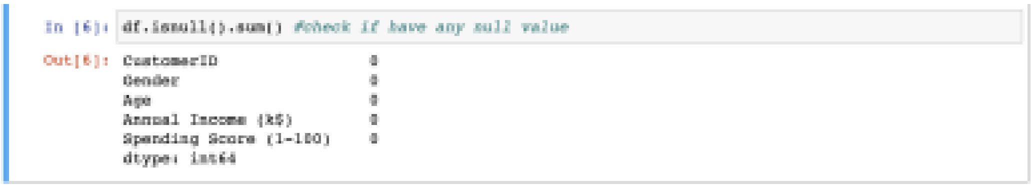


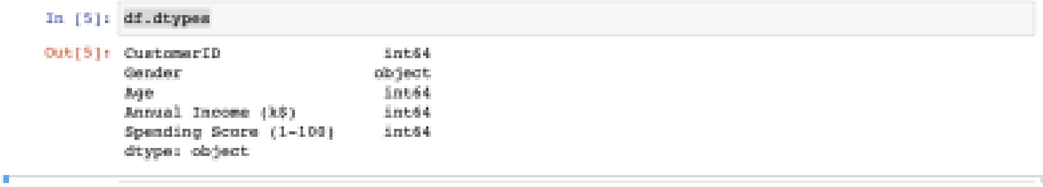
Figure 2. *Variation of Spending score VS Income*

To avoid bias, I made sure there is no null value in the raw dataset(Fig3). All fields are whole numbers except Gender are categorical values(Fig4). This means Gender should be dropped to support PCA analysis. Including CustomerID doesn't help for further exploratory. I also dropped it. The shape of the trimmed data for further data exploratory is 200 rows and 4 columns(See output on Jupyter file).



```
In [6]: df.isnull().sum() #check if have any null value
Out[6]: CustomerID                0
        Gender                    0
        Age                       0
        Annual Income (k$)        0
        Spending Score (1-100)    0
        dtype: int64
```

Figure 3. *Check on null values*



```
In [5]: df.dtypes
Out[5]: CustomerID                 int64
        Gender                    object
        Age                        int64
        Annual Income (k$)         int64
        Spending Score (1-100)     int64
        dtype: object
```

Figure 4. *Demonstration on data types*

Data characteristics

I began with the data density. According to the graphs(Fig5), the 30 age group has the highest density. Most customers have annual income liaises between 50k to 70k. Large numbers of customers have a spending score of 50. Gender-wise, there are more females in comparison to males. From there, I explored the distribution of Age, Income, and Spending scores based on Gender. There is a very high density of females at 30ishes, as compared to males. The density of Income is a bit high at somewhere around 65k. A similar density of incomes between females and males. Similarly, the density of spending between females and male are similar which is at 45-50(Fig6).

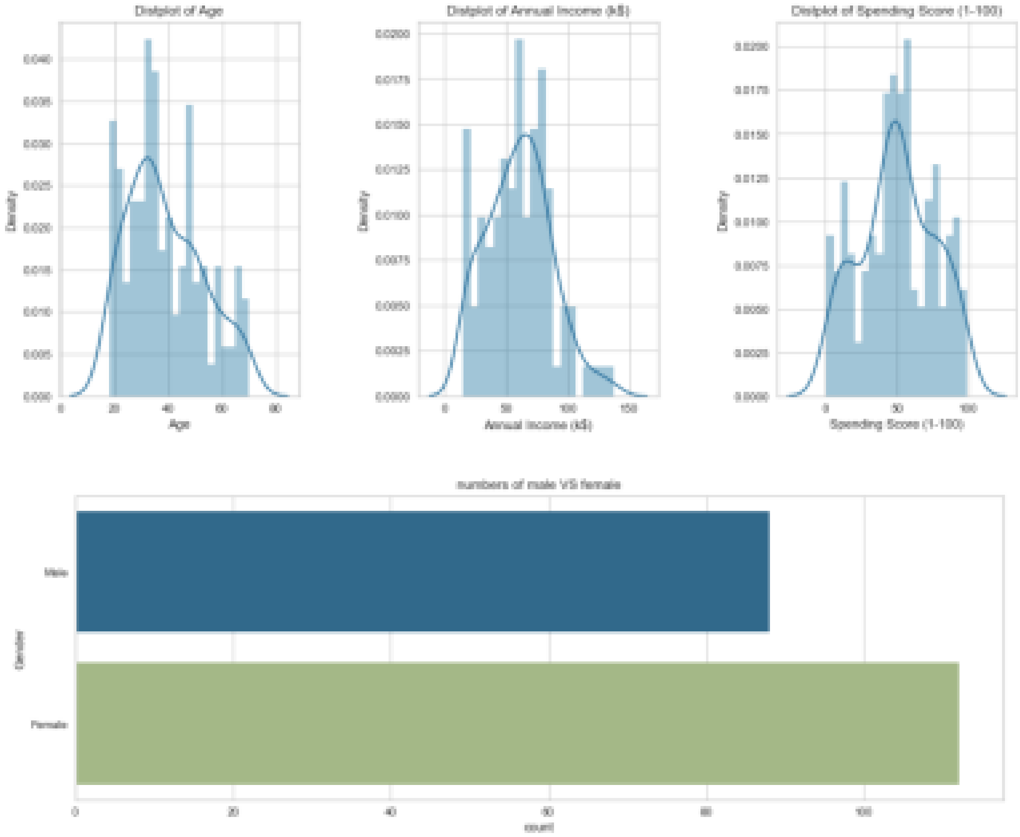


***Figure 5**. Density of Age, Income, Spending and Gender*

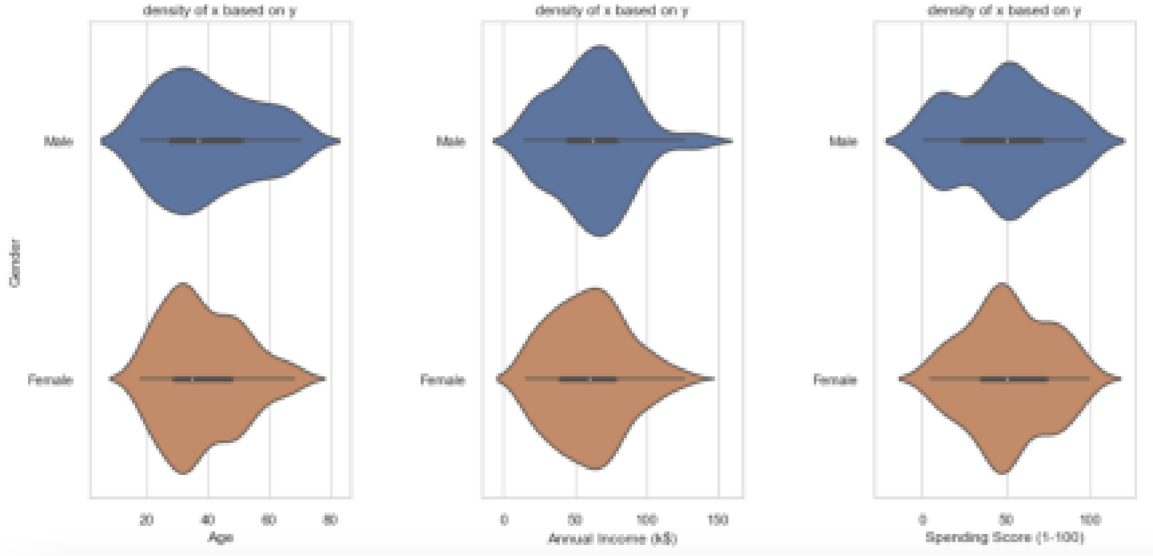
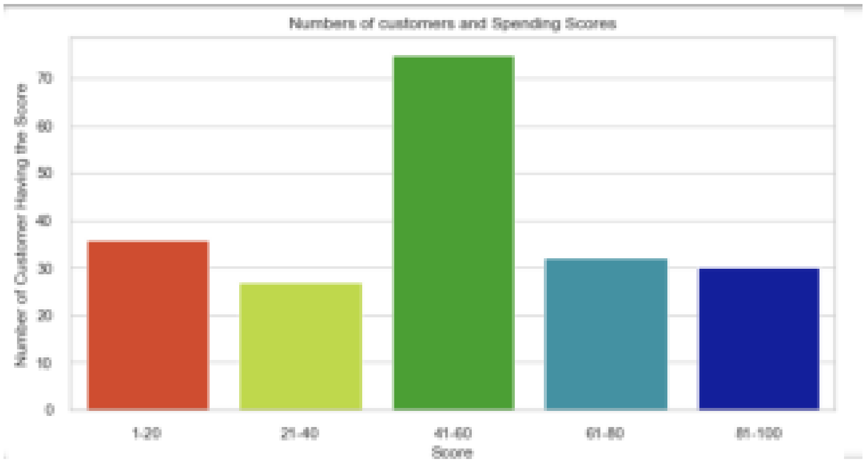


Figure 6. *Density Age, Income and Spending by Gender*

To understand the rationale behind variation between Spending scores. I plotted graphs to investigate the characteristics that the data with high Spending scores have. First, most of the customers have a 41-60 Spending score. The majority of customers are aged 26-35 and have

Incomes of 60k-90k(Fig7). However, customers with the highest spending score are those the age 36-45(Fig8).



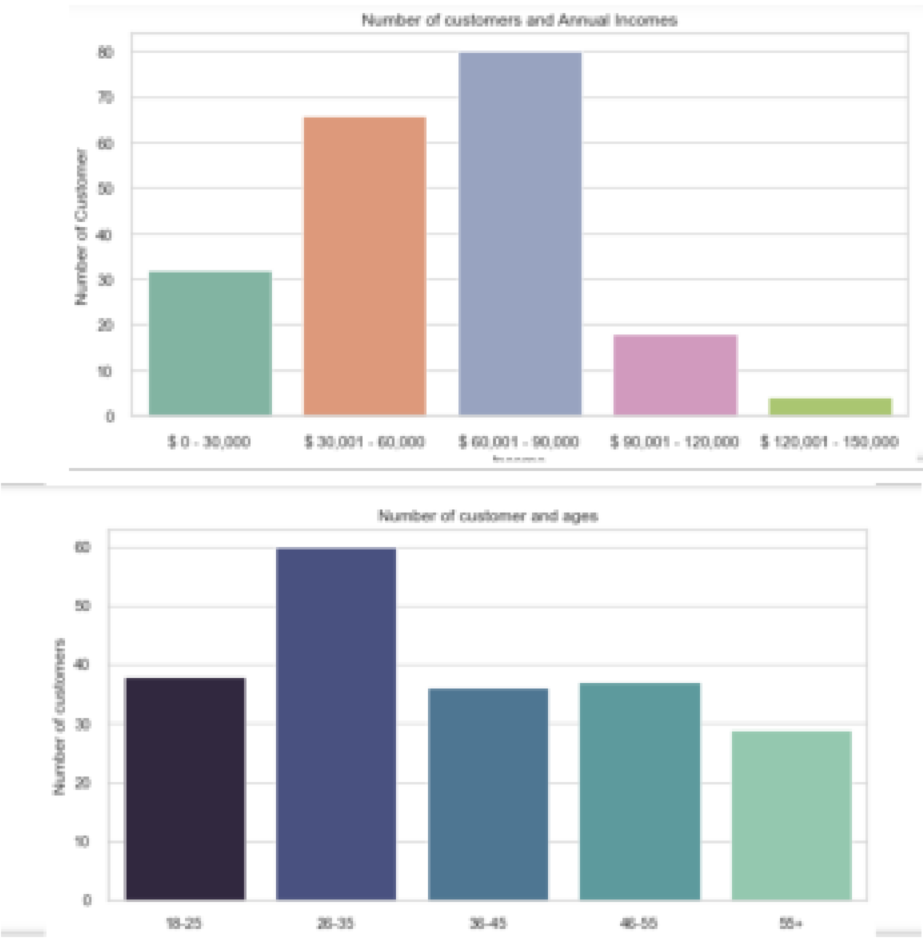


Figure 7. *Distribution of overall customers on spending score, Income and Age*

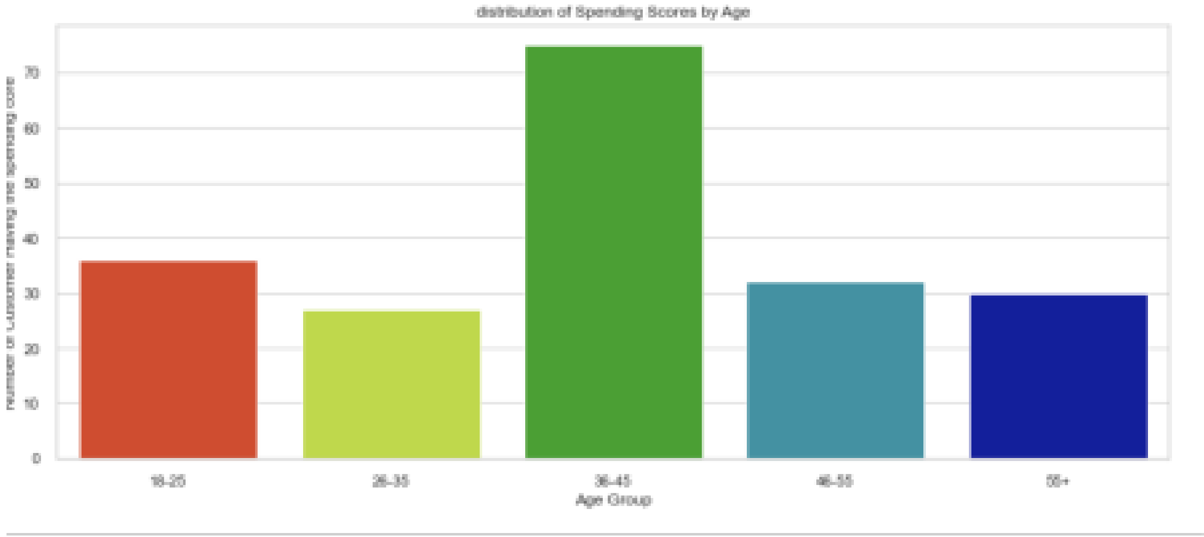


Figure 8. *Distribution of spending score by Age*

Correlations

The above leads to the data correlations between Spending score, Age, and Income. According to the Seaborn plot, the younger these customers are, the more Spending score they make. In particular, customers between 18-40 have a higher spending score, which reaches 60-100(Fig9). There isn't much of a relationship between Spending score and income of 20k-40k and 80k-140k. The correlation only appears on those customers earning 40k-60k, who have a typical Spending score sitting between 40-60(Fig10).

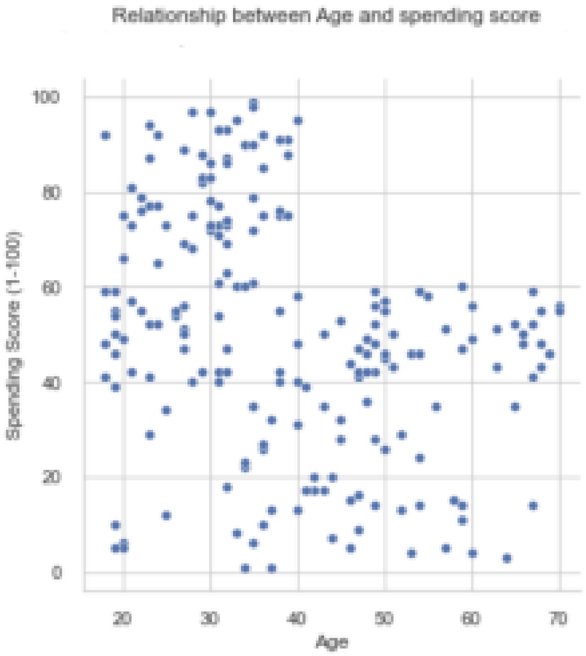


Figure 9. *Relationship between Age and Spending score*

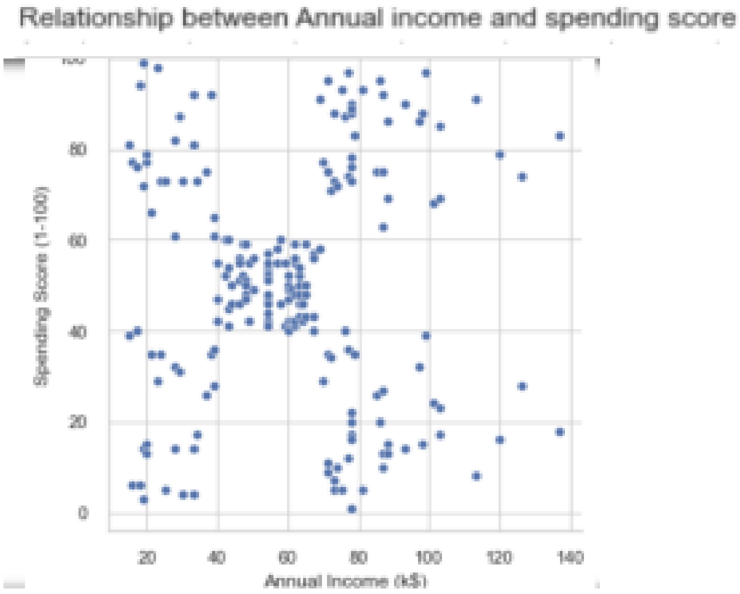


Figure 10. *Relationship between Income and Spending score*

2.0 Clustering algorithm design

Why is clustering a right task to do for this goal?

The clustering algorithm is good at understanding the detailed variation among similar categories. When it comes to customer segmentation, given a loose dataset, data preprocessing techniques of clustering can analyze the fixed demographic and dynamic behavioral information given in the dataset, and summarize them into neat features to be easily learned by the machine. Then the clustering algorithm efficiently captures these features and learns from them. In that way, the segmentation forms on account of hidden correlations and patterns in given customer data. Details on how clustering helps are in the following sections.

How does the Task1 help you to decide about clustering algorithms?
What clustering algorithm is appropriate to do the task and why?

Because of the distribution and correlation of the column data found in part 1, I considered Spending score, Age, and Income in clustering. In that case, I trained the clustering model with multiple features. As part of dimension reduction, PCA combines Age, Income, and spending scores into more obvious, meaningful features in a 2d array. Clustering with data obtained from PCA analysis helps to reduce training costs and make clustering more compact. That is why I did PCA analysis in part 1 and fed data after PCA into the model.

Besides, the data characteristics suggest a suitable clustering algorithm to adopt. As shown in data exploratory, the data provided have varying densities. Varying densities of the data points don’t affect the K-means algorithm. Same feature size for the dataset, which satisfied the requirement of K-Means. Hence, I chose K-means. The data doesn’t have many dimensions after PCA, although it retains its shape as 200 rows. Potential outliers occur in the dataset. DBS will work efficiently given these data conditions and can handle outliers. Thereby, the second clustering algorithm chosen is DBScan.

Which algorithm works better and justifies why is that the case?

Model performance of Kmeans and DBScan were justified through cluster validation metrics(). The silhouette coefficient is 0.553 for the current 5 clusters model. This score is close to 1 which denotes the best, meaning each cluster here is far away from the other. I also checked the relative size and distribution of the clusters, using an inter-cluster distance map. Only one cluster is quite big in comparison to the others. A decent separation stands between all clusters. No overlapping of the original features was detected(See output on Jupyter file). Although the DBScan is built with optimal minPts and epochs(calculated the distance from each point to its closest neighbor & found at the point of maximum curvature using the knee), Kmeans still performs better than DBScan, whose silhouette coefficient is 0.060. Kalinski Harabasz Score for K-means is 249.136 versus 10.604 in DBScan. This means K-means clustering brings high variance between clusters and low variation among data points inside the cluster, which means it’s well-defined. Davies Bouldin Score 0.584 for K-means, whereas DBScan has 1.982. K-means score here is closer to the best score 0. This highlights each cluster’s centroid is not very far away from the other with K-means. Given these justifications, K-means work better in this case of customer segmentation.

K-means algorithm is sensitive to the number of clusters. I found the optimal numbers of clusters with WCSS(See output on Jupyter file). WCSS value becomes constant than before since the k value is 5. So I took 5 as the number of clusters. The number of clusters was appropriately maximized and in the limiting case, each data point becomes its cluster centroid. Taking 5 clusters helps improve clustering performance as validated in Silhouette analysis(See output on Jupyter file). When cluster number is 2, a negative Silhouette value occurs which is a bad sign. When the numbers of clusters are set as 4, 5, and 6, all Silhouette values are positives(See output on Jupyter file). Models with over 5 clusters seem to be overly clustered, as their average silhouette score starts decreasing(See output on Jupyter file). I also found K-means for this dataset perform better without data normalization(Fig11). This can be a reason resulting in good clustering in K-means.

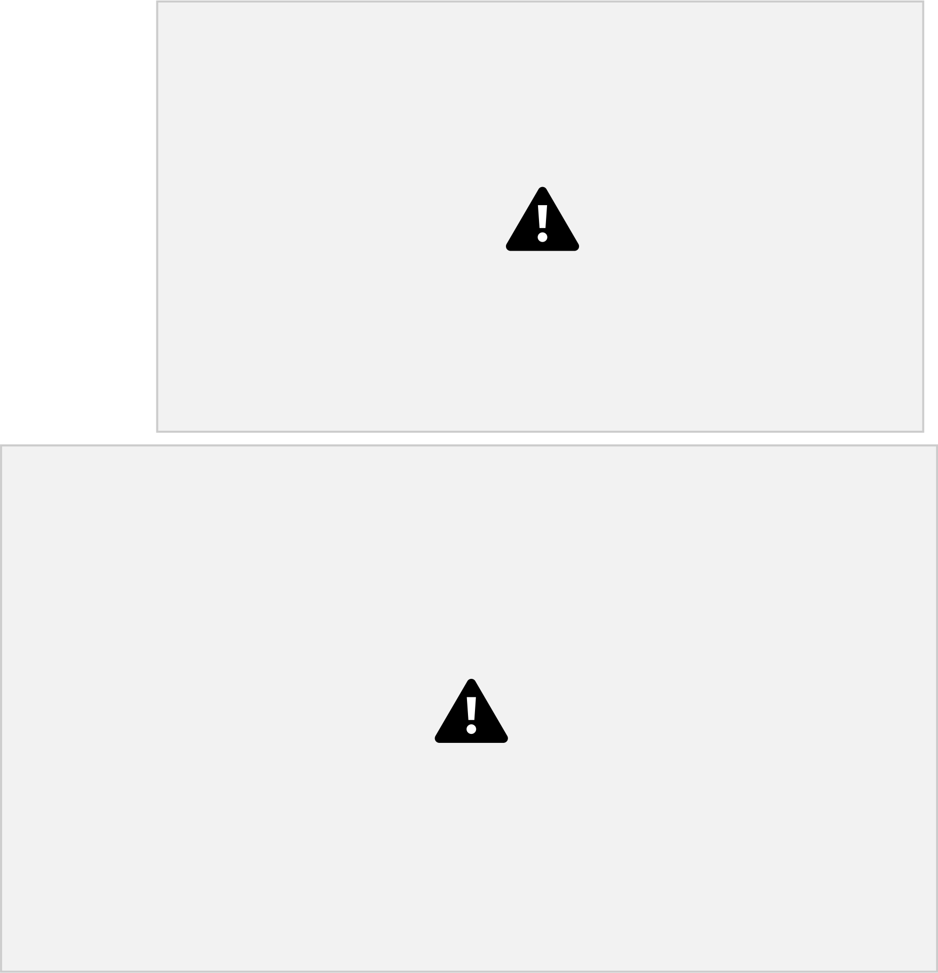


Figure 11.

K-means with data normalization(up) VS K-means without normalization(down)