Programming for AI Lab 11

|  |  |
| --- | --- |
| Course: Programming for AI (AI2001) | Semester: Fall 2023 |
| Instructor: Sameer Faisal | T.A: N/A |
| Note:   * Maintain discipline during the lab. * Listen and follow the instructions as they are given. * Just raise hand if you have any problem. * Completing all tasks of each lab is compulsory. * Get your lab checked at the end of the session. |  |

**Unsupervised Algorithms**

You have seen supervised algorithms in the previous labs.

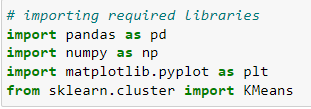
Now let’s see two an unsupervised classification algorithm.

Algorithms which are applied when labels/classes/targets are not given. So these algorithms make groups (clusters) of similar data points

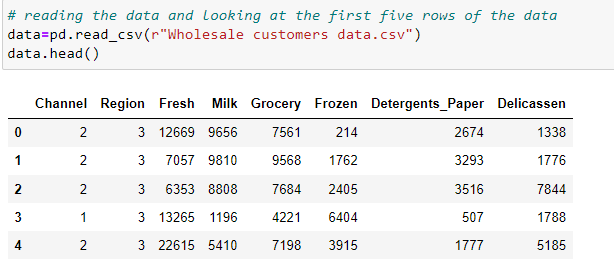
For Example: K-means clustering, Hierarchical clustering, Agglomerative Clustering, etc.

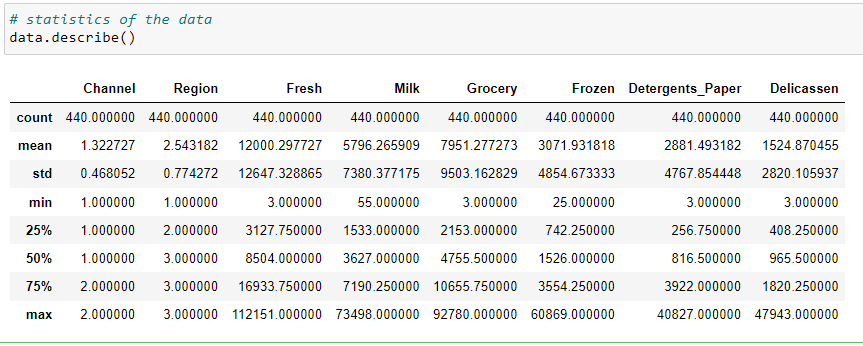
**K-Means Clustering**

K-means clustering is a simple and efficient algorithm used for grouping similar data points into a specified number of clusters. The "k" in K-means represents the number of clusters desired. The algorithm works by iteratively assigning data points to clusters based on their similarity and adjusting cluster centroids until a stable configuration is achieved. It's commonly used for segmentation and pattern recognition in data analysis.



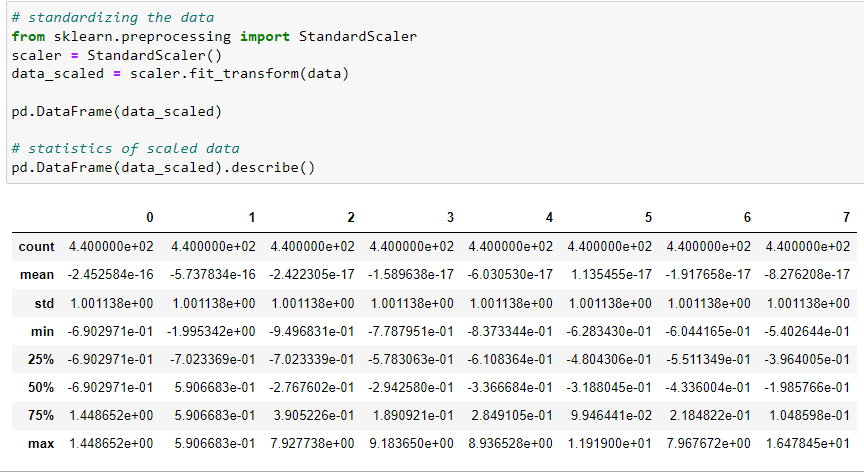
Next, let’s read the data and look at the first five rows of wholesale dataset:



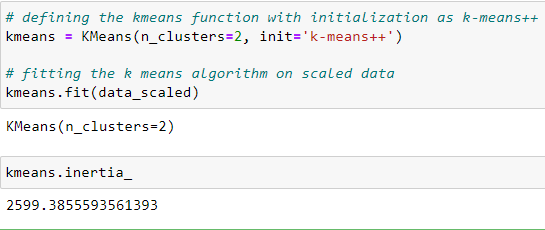


Here, we see that there is a lot of variation in the magnitude of the data. Variables like Channel and Region have low magnitude whereas variables like Fresh, Milk, Grocery, etc. have a higher magnitude.

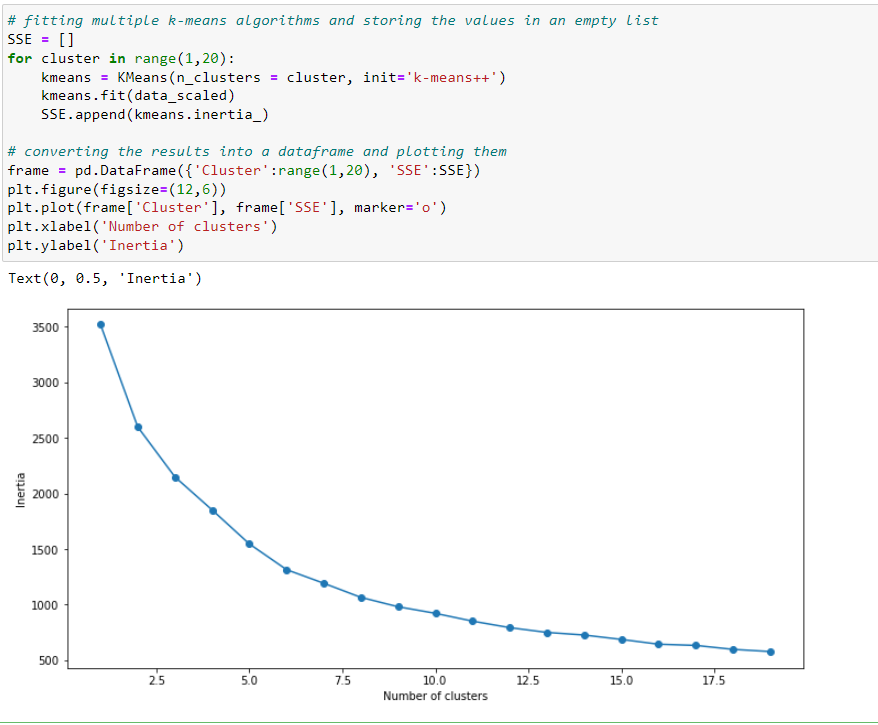
Since K-Means is a distance-based algorithm, this difference of magnitude can create a problem. So let’s first bring all the variables to the same magnitude:



The magnitude looks similar now. Next, let’s create a k-means function and fit it on the data:



Inertia (SSE) measures how well a dataset was clustered by K-Means. It is calculated by measuring the distance between each data point and its centroid, squaring this distance, and summing these squares across one cluster. However, this is a tradeoff because as K increases, inertia decreases.



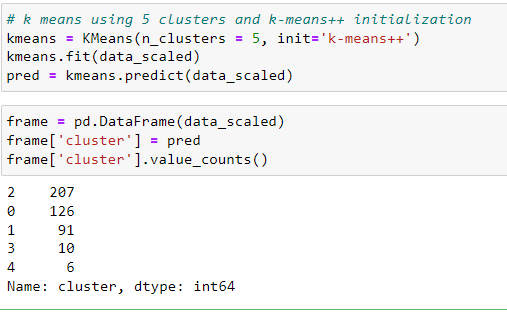
We draw elbow curve just to find out how many clusters should be formed

The above elbow curve shows how inertia (distance between data points and centroids in cluster) increases or decreases as number of clusters increases from 1 to 20

Can you tell the optimum number of clusters value from this plot? We should make clusters having lesser value of inertia as clusters should be compact (distance between points and centroids minimal). Lowest inertia will be at highest number of clusters. But we cannot make high number of clusters. Because if we do then whats the point of making clusters! We want to categorize data points in few clusters not many.

Looking at the above elbow curve, we can choose any number of clusters between 5 to 8. As these are reasonable number of clusters with reasonable inertia

Let’s set the number of clusters as 5 and fit the model:



The left column above shows cluster number and right column shows number of data points in that cluster.

**Lab Tasks**

1. Download any clustering dataset from internet and after finding how many optimum number of clusters should be formed using elbow curve, apply k-means on it.