ABSTRACT

At its most basic, customer segmentation (also known as market segmentation) is the division of potential customers in a given market into discrete groups. That division is based on customers having similar enough:

1. Needs, i.e., so that a single whole product can satisfy them.
2. Buying characteristics, i.e., responses to messaging, marketing channels, and sales channels, that a single go-to-market approach can be used to sell to them competitively and economically.

There are three main approaches to market segmentation:

- ***A priori*** segmentation, the simplest approach, uses a classification scheme based on publicly available characteristics — such as industry and company size — to create distinct groups of customers within a market. However, a priori market segmentation may not always be valid, since companies in the same industry and of the same size may have very different needs.

- ***Needs-based*** segmentation is based on differentiated, validated drivers (needs) that customers express for a specific product or service being offered. The needs are discovered and verified through primary market research, and segments are demarcated based on those different needs rather than characteristics such as industry or company size.

- ***Value-based*** segmentation differentiates customers by their economic value, grouping customers with the same value level into individual segments that can be distinctly targeted.

TABLE OF CONTENTS

INTRODUCTION

A very common example of an unsupervised machine learning, clustering is the process of grouping similar data points into a cluster. Given a finite set of data points, clustering aims to find homogeneous subgroups of data points with similar characteristics.

In this project, we perform Customer segmentation using simple clustering algorithm called K-Means and use Elbow Method. We will also implement it with the popular Scikit-learn library.

Customer Segmentation is the subdivision of a market into discrete customer groups that share similar characteristics. Customer Segmentation can be a powerful means to identify unsatisfied customer needs. Using the above data companies can then outperform the competition by developing uniquely appealing products and services.

The most common ways in which businesses segment their customer base are mentioned in the abstract.

Advantages of Customer Segmentation

1. Determine appropriate product pricing.
2. Develop customized marketing campaigns.
3. Design an optimal distribution strategy.
4. Choose specific product features for deployment.
5. Prioritize new product development efforts.

UNSUPERVISED MACHINE LEARNING

A task involving machine learning may not be linear, but it has a number of well known steps:

- Problem definition.
- Preparation of Data.
- Learn an underlying model.
- Improve the underlying model by quantitative and qualitative evaluations.
- Present the model.

One good way to come to terms with a new problem is to work through identifying and defining the problem in the best possible way and learn a model that captures meaningful information from the data. While problems in Pattern Recognition and Machine Learning can be of various types, they can be broadly classified into three categories:

- **Supervised Learning**: The system is presented with example inputs and their desired outputs, given by a “teacher”, and the goal is to learn a general rule that maps inputs to outputs.
- **Unsupervised Learning:** No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- **Reinforcement Learning**: A system interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The system is provided feedback in terms of rewards and punishments as it navigates its problem space.

Between supervised and unsupervised learning is semi-supervised learning, where the teacher gives an incomplete training signal: a training set with some (often many) of the target outputs missing. We will focus on unsupervised learning and data clustering in this blog post.

Unsupervised Learning

In some pattern recognition problems, the training data consists of a set of input vectors x without any corresponding target values. The goal in such unsupervised learning problems may be to discover groups of similar examples within the data, where it is called *clustering*, or to determine how the data is distributed in the space, known as *density estimation*. To put forward in simpler terms, for a n-sampled space x1 to xn, true class labels are not provided for each sample, hence known as *learning without teacher*.

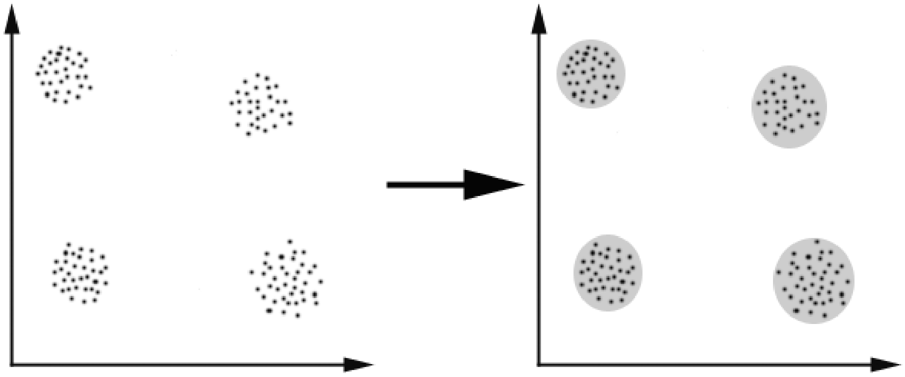
Issues with Unsupervised Learning:

- Unsupervised Learning is harder as compared to Supervised Learning tasks..
- How do we know if results are meaningful since no answer labels are available?
- Let the expert look at the results (external evaluation)

Why Unsupervised Learning is needed despite of these issues?

- Annotating large datasets is very costly and hence we can label only a few examples manually. Example: Speech Recognition
- There may be cases where we don’t know how many/what classes is the data divided into. Example: Data Mining
- We may want to use clustering to gain some insight into the structure of the data before designing a classifier.

CLUSTRING AND CLASSIFICATION



Clustering can be considered the most important *unsupervised learning* problem; so, as every other problem of this kind, it deals with finding a *structure* in a collection of unlabeled data. A loose definition of clustering could be “the process of organizing objects into groups whose members are similar in some way”. A *cluster* is therefore a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

***Distance-based clustering*:**

Given a set of points, with a notion of distance between points, grouping the points into some number of *clusters*, such that

- Internal (within the cluster) distances should be small i.e members of clusters are close/similar to each other.
- External (intra-cluster) distances should be large i.e. members of different clusters are dissimilar.

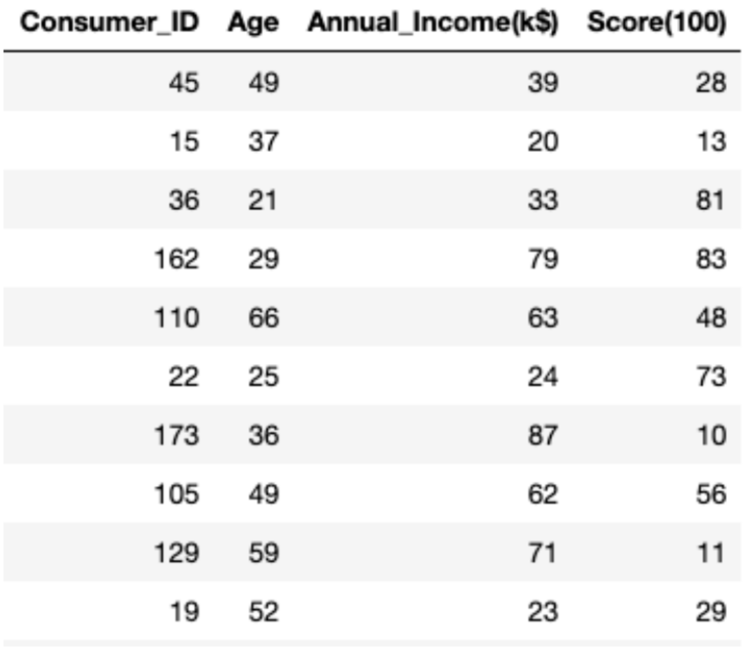
The Goals of Clustering

The goal of clustering is to determine the internal grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user who should supply this criterion, in such a way that the result of the clustering will suit their needs.

Clustering may sound similar to the popular classification type of problems, but unlike classification wherein a labeled set of classes are provided at the time of training, the idea of clustering is to form the classes or categories from the data which is not pre-classified into any set of categories, which is why clustering is an unsupervised learning algorithm.

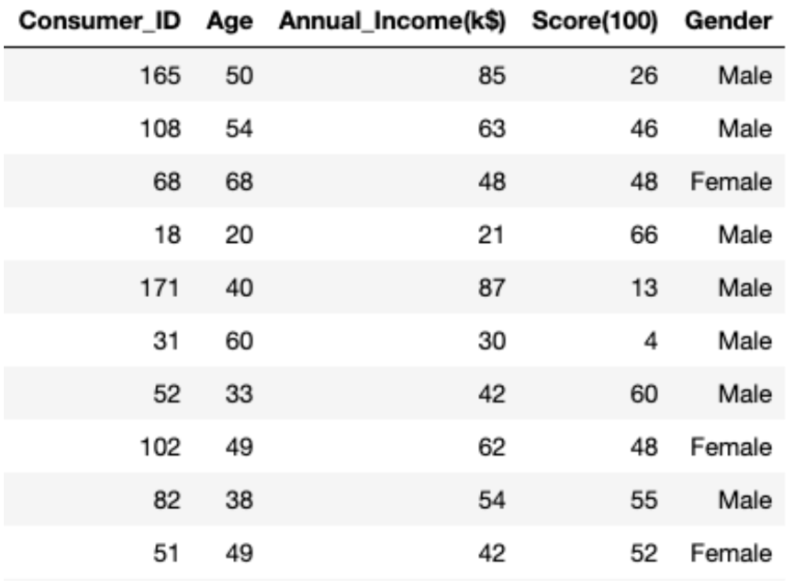
Let’s look at a basic example to distinguish a clustering and a classification problem.

Clustering:



Consumer_ID	Age	Annual_Income(k$)	Score(100)
45	49	39	28
15	37	20	13
36	21	33	81
162	29	79	83
110	66	63	48
22	25	24	73
173	36	87	10
105	49	62	56
129	59	71	11
19	52	23	29

Classification:



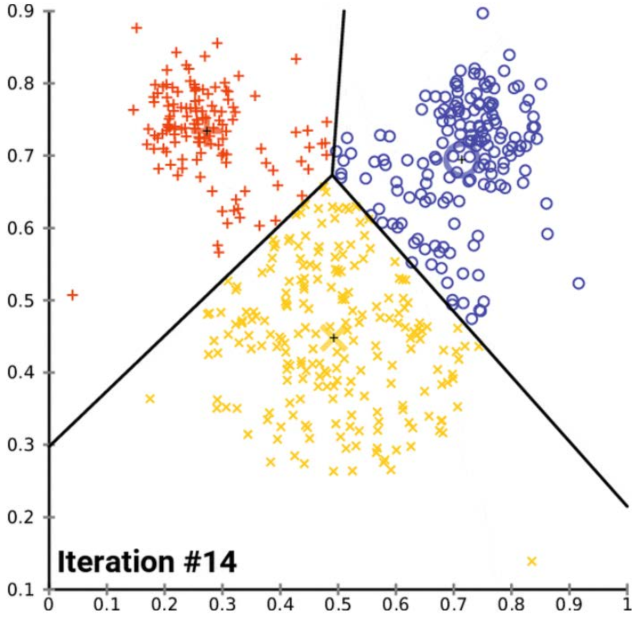
Consumer_ID	Age	Annual_Income(k$)	Score(100)	Gender
165	50	85	26	Male
108	54	63	46	Male
68	68	48	48	Female
18	20	21	66	Male
171	40	87	13	Male
31	60	30	4	Male
52	33	42	60	Male
102	49	62	48	Female
82	38	54	55	Male
51	49	42	52	Female

In the first dataset, we do not see any labelled features that we can use to classify the data points based on any characteristics. But what clustering can do is it can provide us with classes that can categorise the given data points based on the features. For example, we can use the first dataset to group consumers into different categories based on their annual income and their spending score. In the second data set, we can use the features Age, Annual_Income and Score to predict or classify whether a consumer is male or female.

K-MEANS CLUSTRING

K-means is one of the simplest unsupervised learning algorithms that solves the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centres, one for each cluster. These centroids should be placed in a smart way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more.

K-means clustering is a clustering method that subdivides a single cluster or a collection of data points into K different clusters or groups.



The algorithm analyzes the data to find organically similar data points and assigns each point to a cluster that consists of points with similar characteristics. Each cluster can then be used to label the data into different classes based on the characteristics of the data.K-Means clustering works by constantly trying to find a centroid with closely held data points. This means that each cluster will have a centroid and the data points in each cluster will be closer to its centroid compared to the other centroids.

Choosing The Right Number Of Clusters:

The number of clusters that we choose for a given dataset cannot be random. Each cluster is formed by calculating and comparing the distances of data points within a cluster to its centroid. An ideal way to figure out the right number of clusters would be to calculate the Within-Cluster-Sum-of-Squares (WCSS).

WCSS is the sum of squares of the distances of each data point in all clusters to their respective centroids.

$$\text{WCSS} = \sum_{C_k}^{C_n} \left(\sum_{d_i \, in \, C_i}^{d_m} distance(d_i, C_k)^2 \right)$$

Where,

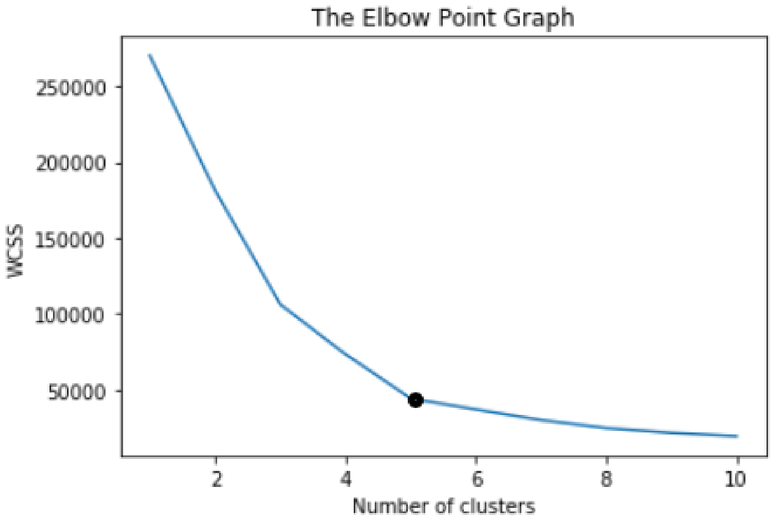
C is the cluster centroids and d is the data point in each Cluster.

The idea is to minimise the sum. Suppose there are n observation in a given dataset and we specify n number of clusters (k = n) then WCSS will become zero since data points themselves will act as centroids and the distance will be zero and ideally this forms a perfect cluster, however this doesn’t make any sense as we have as many clusters as the observations. Thus there exists a threshold value for K which we can find using the Elbow point graph.

ELBOW METHOD

We can find the optimum value for K using an Elbow point graph. We randomly initialise the K-Means algorithm for a range of K values and will plot it against the WCSS for each K value.

The resulting graph would look something like what's shown below:



For the above-given graph, the optimum value for K would be 5. As we can see that with an increase in the number of clusters the WCSS value decreases. We select the value for K on the basis of the rate of decrease in WCSS. For example, from cluster 1 to 2 to 3 in the above graph we see a sudden and huge drop in WCSS. After 5 the drop is minimal and hence we chose 5 to be the optimal value for K.

The Random Initialisation Trap

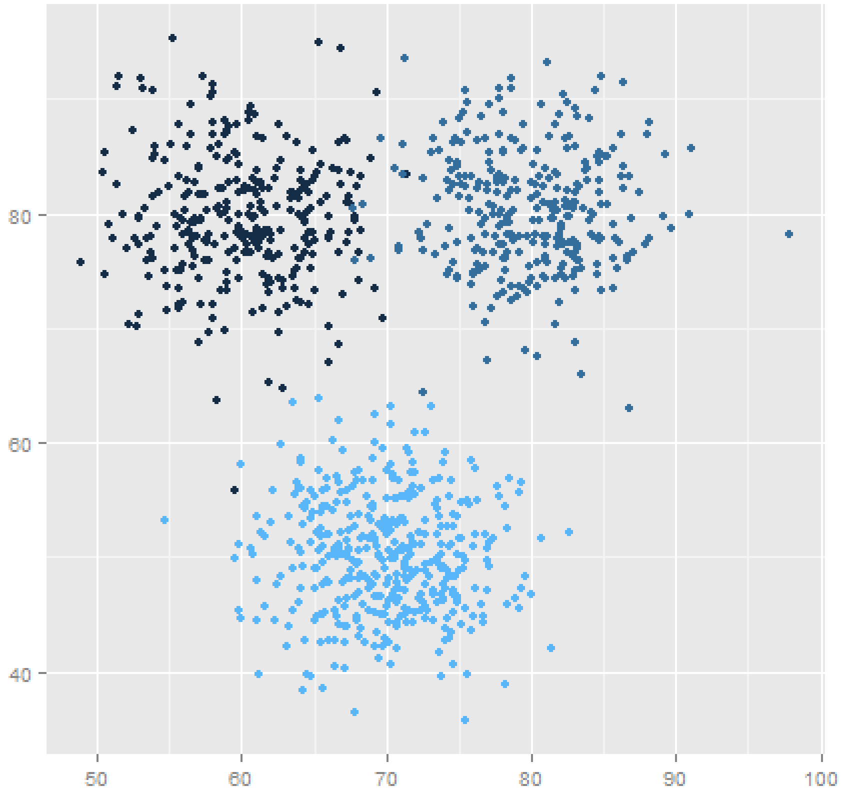
One major drawback of K-Means clustering is the random initialisation of centroids. The formation of clusters is closely bound by the initial position of a centroid. The random positioning of the centroids can completely alter clusters and can result in a random formation.

The solution is K-means++. K-Means++ is an algorithm that is used to initialise the K-Means algorithm.

ALGORITHM AND IMPLIMENTATION

K-Means Algorithm

1. Selecting an appropriate value for K which is the number of clusters or centroids
2. Selecting random centroids for each cluster
3. Assigning each data point to its closest centroid
4. Adjusting the centroid for the newly formed cluster in step 4
5. Repeating step 4 and 5 till all the data points are perfectly organised within a cluster space.



K Means Clustering where K=3

CONCLUSION

K means clustering is one of the most popular clustering algorithms and usually the first thing practitioners apply when solving clustering tasks to get an idea of the structure of the dataset. The goal of K means is to group data points into distinct non-overlapping subgroups. One of the major application of K means clustering is segmentation of customers to get a better understanding of them which in turn could be used to increase the revenue of the company.