Unit IV: Clustering and Classification

Parameter	CLASSIFICATION	CLUSTERING
Type	used for supervised learning	used for unsupervised learning
Basic	process of classifying the input instances based on their corresponding class labels	grouping the instances based on their similarity without the help of class labels
Need	it has labels so there is need of training and testing dataset for verifying the model created	there is no need of training and testing dataset
Complexity	more complex as compared to clustering	less complex as compared to classification
Example Algorithms	Logistic regression, Naive Bayes classifier, Support vector machines, etc.	k-means clustering algorithm, Fuzzy c-means clustering algorithm, Gaussian (EM) clustering algorithm, etc.

Differences between Classification and Clustering

1. Classification is used for supervised learning whereas clustering is used for unsupervised learning.
2. The process of classifying the input instances based on their corresponding class labels is known as classification whereas grouping the instances based on their similarity without the help of class labels is known as clustering.
3. As Classification have labels so there is need of training and testing dataset for verifying the model created but there is no need for training and testing dataset in clustering.
4. Classification is more complex as compared to clustering as there are many levels in the classification phase whereas only grouping is done in clustering.
5. Classification examples are Logistic regression, Naive Bayes classifier, Support vector machines, etc. Whereas clustering examples are k-means clustering algorithm, Fuzzy c-means clustering algorithm, Gaussian (EM) clustering algorithm, etc.

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.

It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties.

It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training.

It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters.

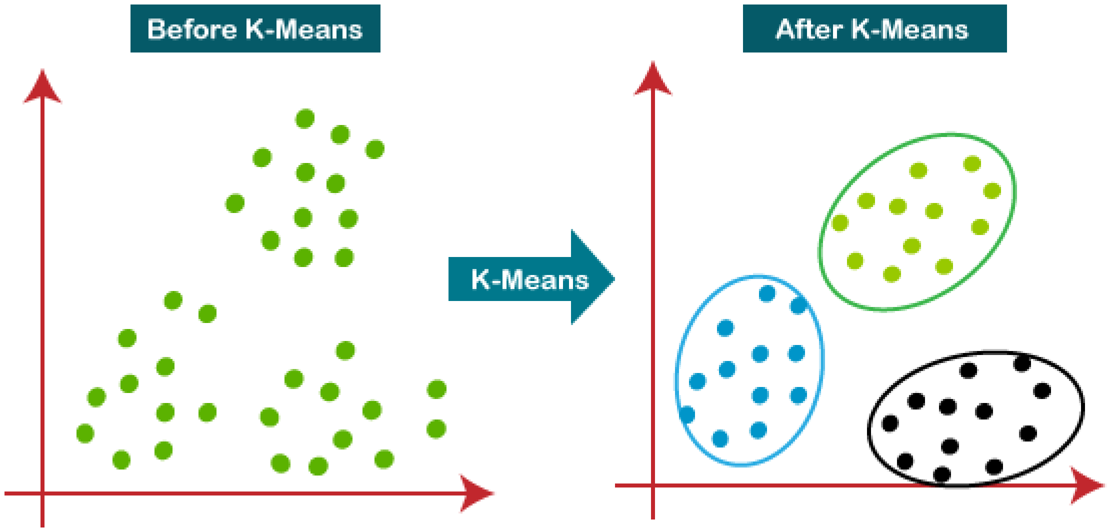
The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.

The k-means clustering algorithm mainly performs two tasks:

- Determines the best value for K center points or centroids by an iterative process.
- Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

Hence each cluster has datapoints with some commonalities, and it is away from other clusters.

The below diagram explains the working of the K-means Clustering Algorithm:



How does the K-Means Algorithm Work?

The working of the K-Means algorithm is explained in the below steps:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

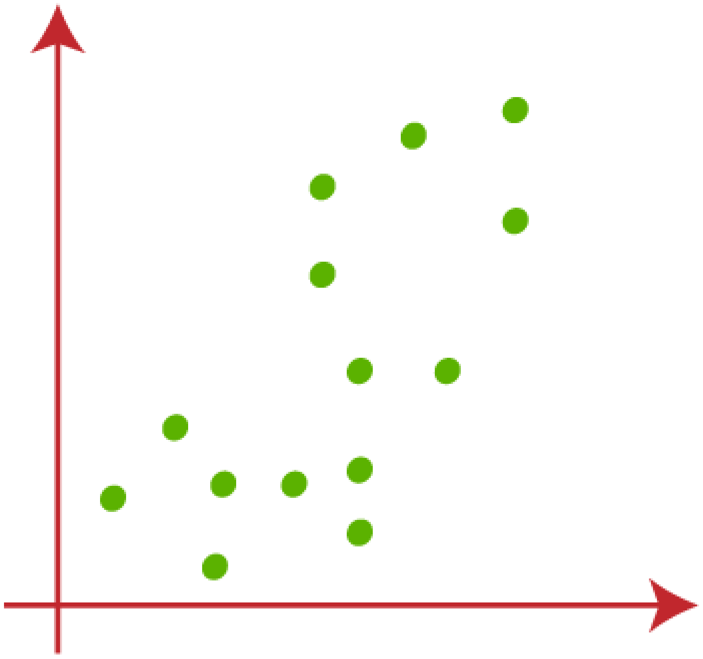
Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

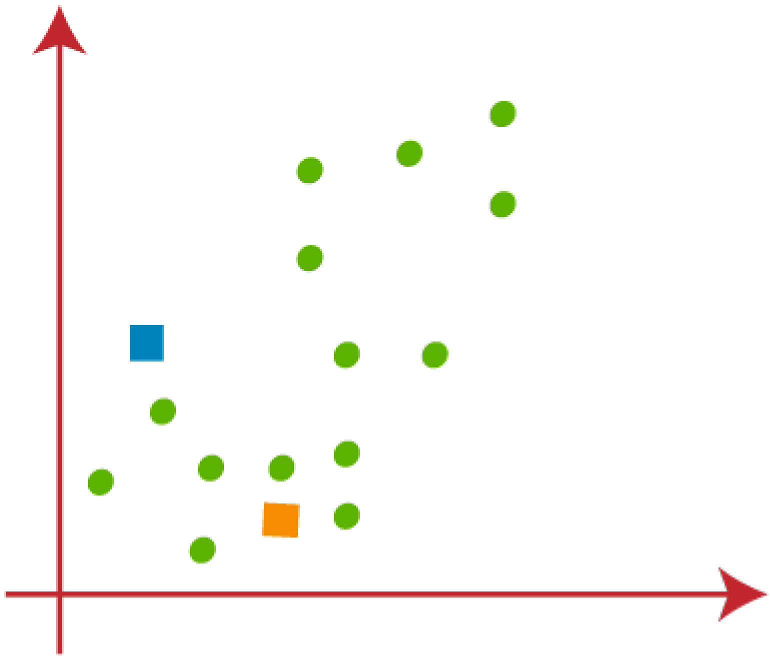
Step-7: The model is ready.

Let's understand the above steps by considering the visual plots:

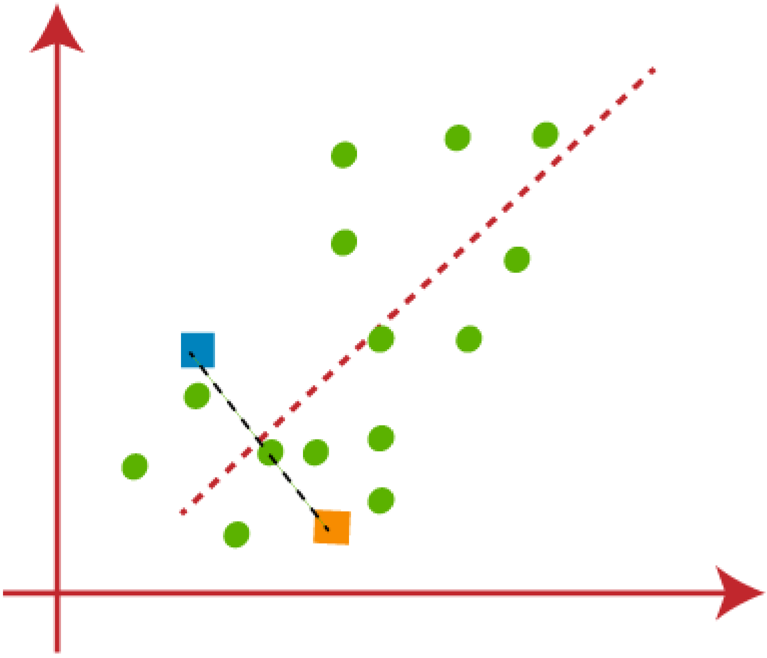
Suppose we have two variables M1 and M2. The x-y axis scatter plot of these two variables is given below:



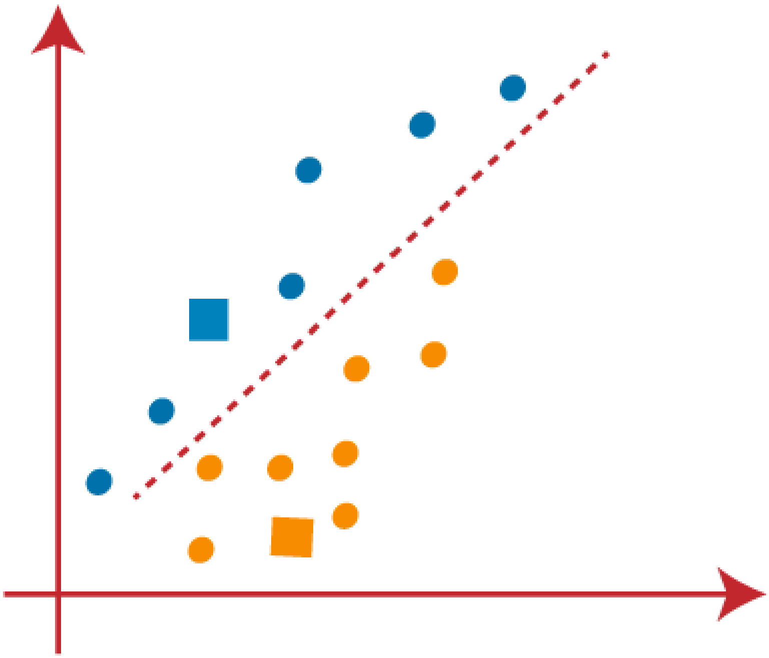
- Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters.
- We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset. Consider the below image:



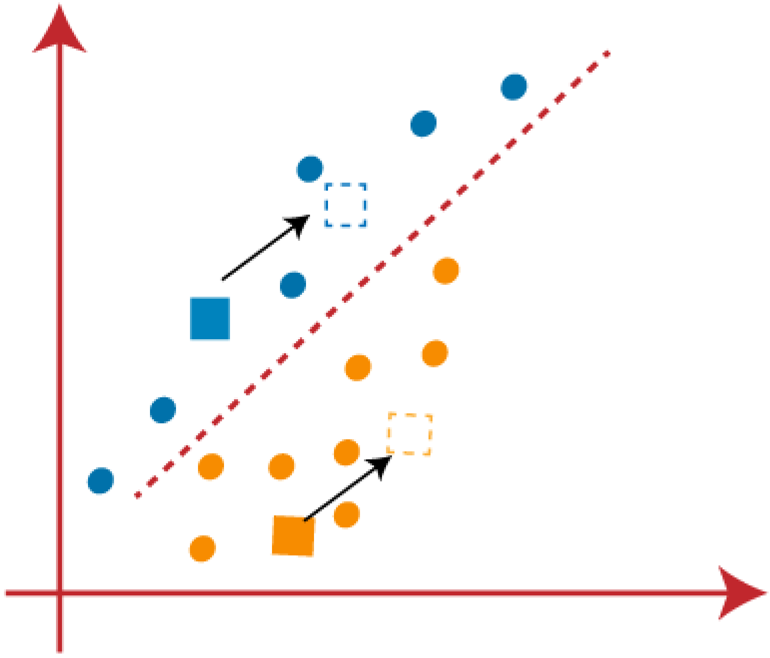
- Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids. Consider the below image:



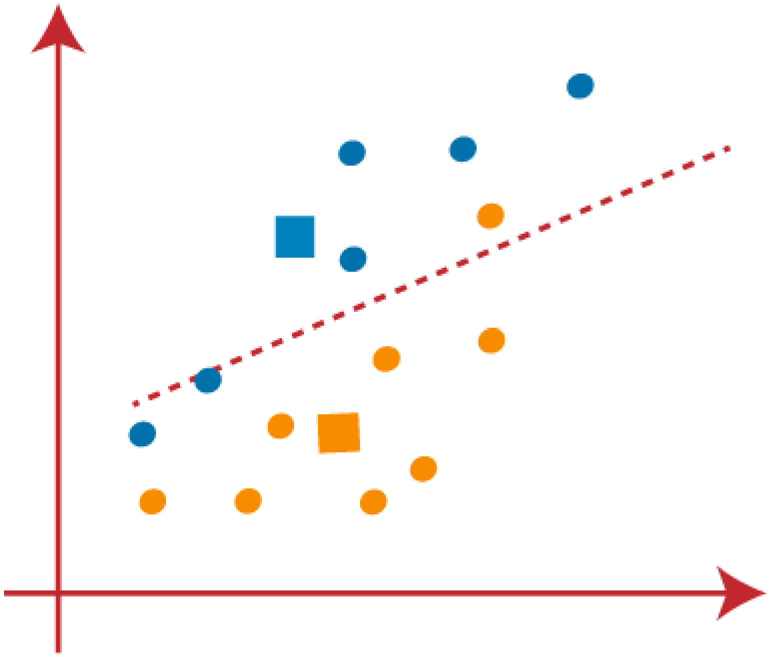
From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



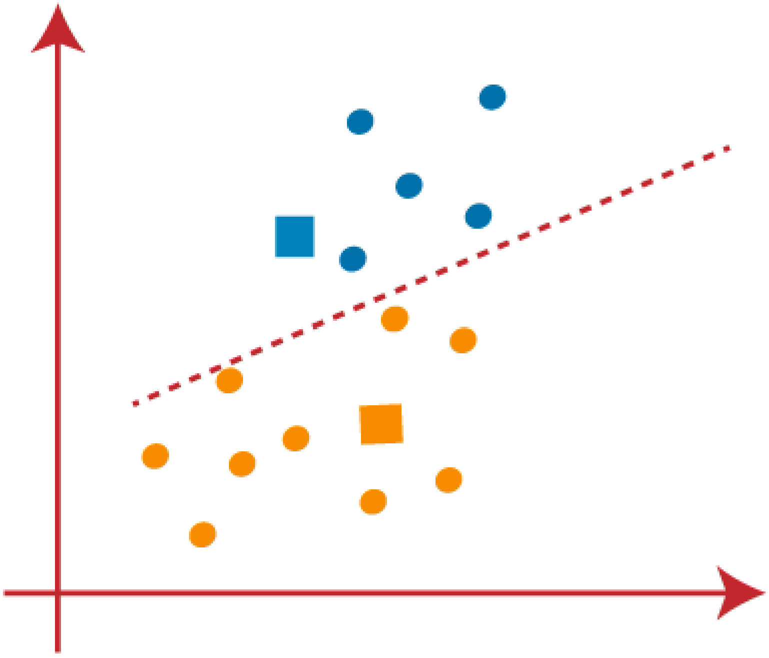
- As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids as below:



- Next, we will reassign each datapoint to the new centroid. For this, we will repeat the same process of finding a median line. The median will be like below image:

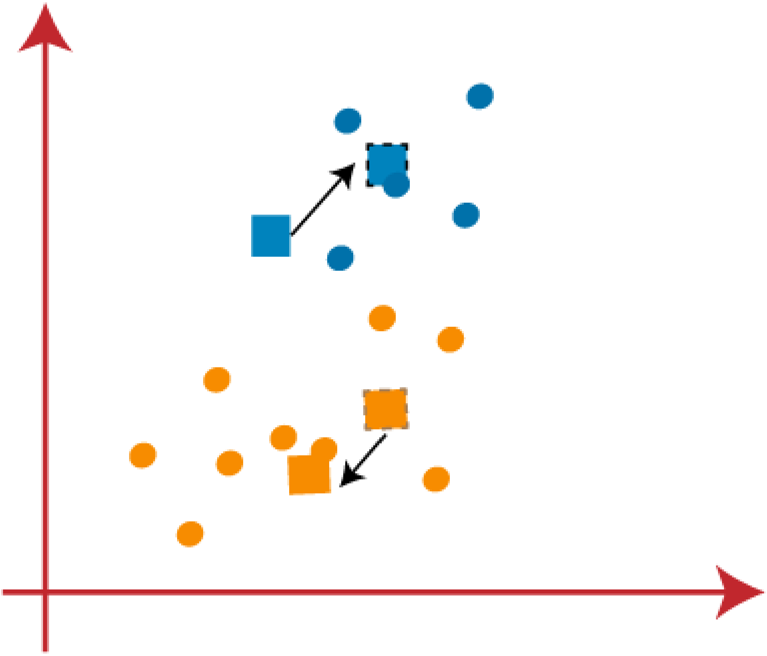


From the above image, we can see, one yellow point is on the left side of the line, and two blue points are right to the line. So, these three points will be assigned to new centroids.

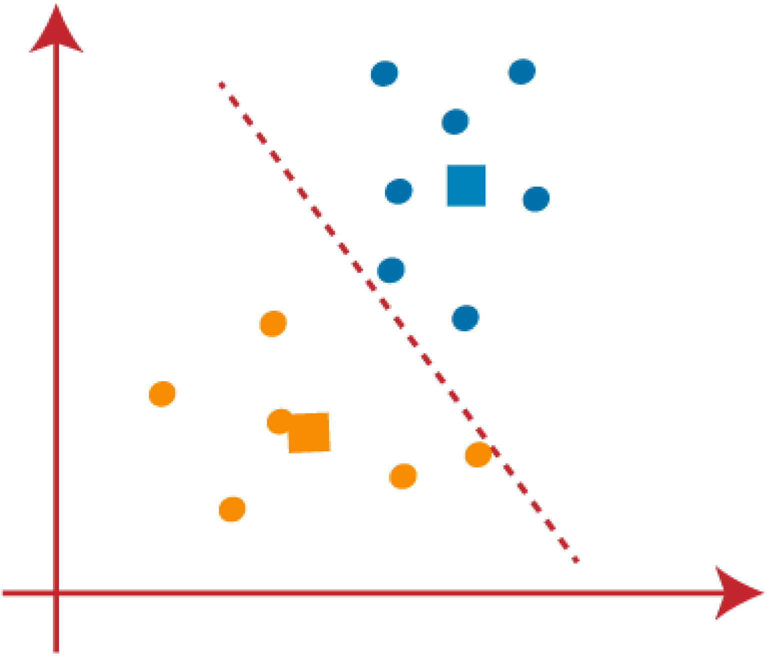


As reassignment has taken place, so we will again go to the step-4, which is finding new centroids or K-points.

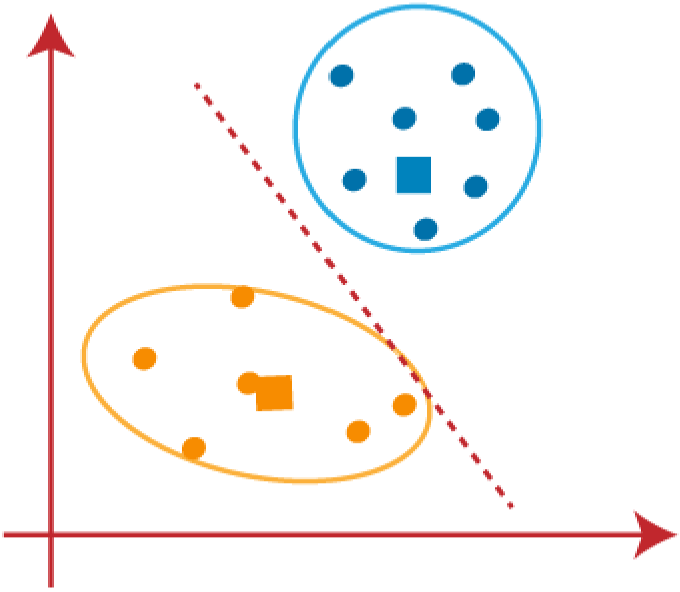
- We will repeat the process by finding the center of gravity of centroids, so the new centroids will be as shown in the below image:



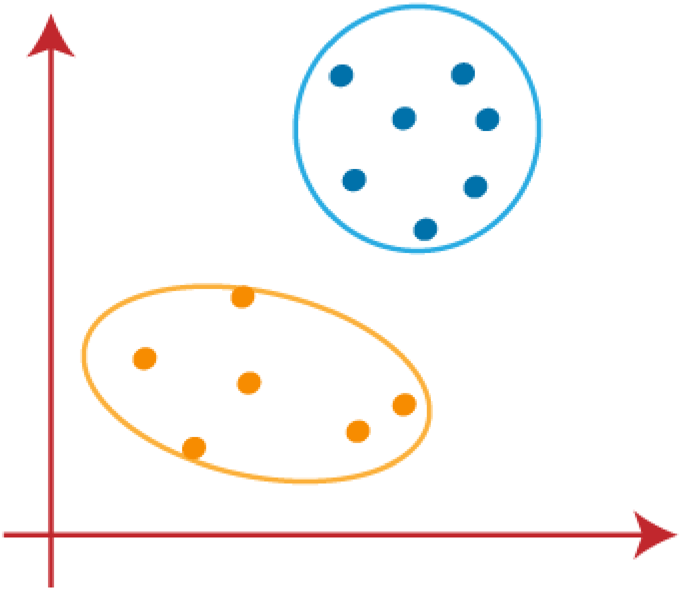
- As we got the new centroids so again will draw the median line and reassign the data points. So, the image will be:



- We can see in the above image; there are no dissimilar data points on either side of the line, which means our model is formed. Consider the below image:



As our model is ready, so we can now remove the assumed centroids, and the two final clusters will be as shown in the below image:



How to choose the value of "K number of clusters" in K-means Clustering?

The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method

The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$\text{WCSS} = \sum_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2 + \sum_{P_i \text{ in Cluster2}} \text{distance}(P_i\ C_2)^2 + \sum_{P_i \text{ in CLuster3}} \text{distance}(P_i\ C_3)^2$$

In the above formula of WCSS,

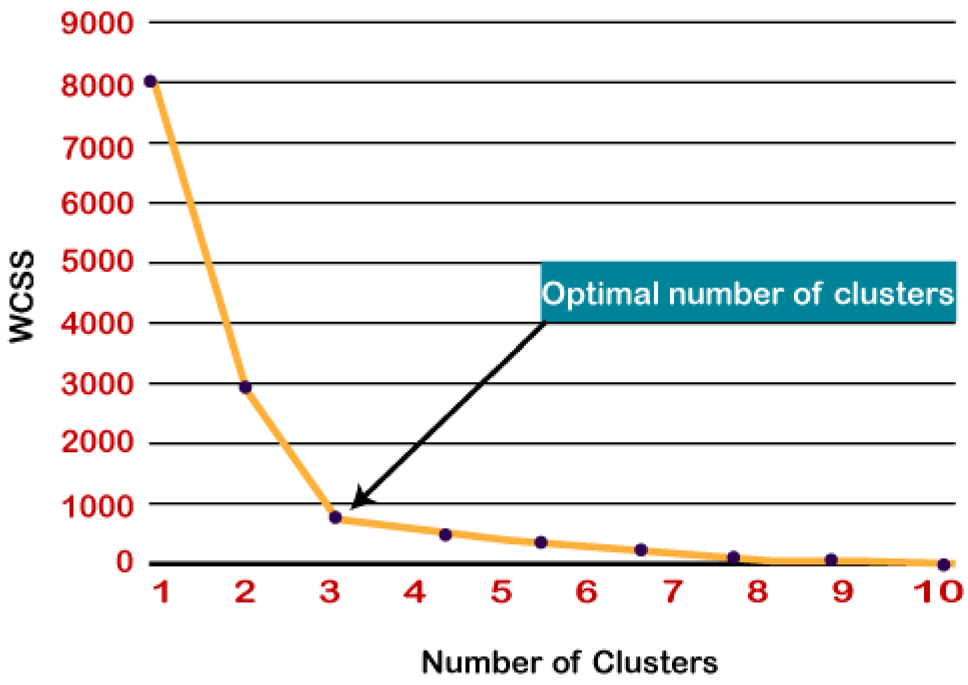
$\sum_{P_i \text{ in Cluster1}} \text{distance}(P_i\ C_1)^2$: It is the sum of the square of the distances between each data point and its centroid within a cluster1 and the same for the other two terms.

To measure the distance between data points and centroid, we can use any method such as Euclidean distance or Manhattan distance.

To find the optimal value of clusters, the elbow method follows the below steps:

- It executes the K-means clustering on a given dataset for different K values (ranges from 1-10).
- For each value of K, calculates the WCSS value.
- Plots a curve between calculated WCSS values and the number of clusters K.
- The sharp point of bend or a point of the plot looks like an arm, then that point is considered as the best value of K.

Since the graph shows the sharp bend, which looks like an elbow, hence it is known as the elbow method. The graph for the elbow method looks like the below image:



Pros

1. It is simple, highly flexible, and efficient. The simplicity of k-means makes it easy to explain the results in contrast to Neural Networks.
2. The flexibility of k-means allows for easy adjustment if there are problems.
3. The efficiency of k-means implies that the algorithm is good at segmenting a dataset.
4. An instance can change cluster (move to another cluster) when the centroids are recomputed
5. Easy to interpret the clustering results.

Cons

1. It does not allow to develop the most optimal set of clusters and the number of clusters must be decided before the analysis. How many clusters to include is left at the discretion of the researcher. This involves a combination of common sense, domain knowledge, and statistical tools. Too many clusters tell you nothing because of the groups becoming very small and there are too many of them. There are statistical tools that measure within-group homogeneity and group heterogeneity. There are methods like the elbow method to decide the value of k. Additionally, there is a technique

called a dendrogram. The results of a dendrogram analysis provide a recommendation of how many clusters to use. However, calculating a dendrogram for a large dataset could potentially crash a computer due to the computational load and the limits of RAM.

2. When doing the analysis, the k-means algorithm will randomly select several different places from which to develop clusters. This can be good or bad depending on where the algorithm chooses to begin at. From there, the centre of the clusters is recalculated until an adequate "centre'' is found for the number of clusters requested.
3. The order of the data has an impact on the final results.

Relationship between Clustering and Regression

Clustering

Definition:

- (p. pr. & vb. n.) of Cluster

Example Sentences:

- (1) The patterns observed were: clusters of granules related to the cell membrane; positive staining localized to portions of the cell membrane, and, less commonly, the whole cell circumference.

- (2) We used the polymerase chain reaction (PCR) to amplify the breakpoint area of alpha-thalassemia-1 of Southeast Asia type and several parts of the alpha-globin gene cluster to make a differential diagnosis between alpha-thalassemia-1 and Hb Bart's hydrops fetalis.

- (3) In some cervical nodes, a few follicles, lymphocyte clusters, and a well-developed plasmocyte population were also present.

- (4) The 68C intermolt puff of Drosophila melanogaster contains a cluster of three glue protein genes, Sgs-3, Sgs-7, and Sgs-8.

- (5) This analysis demonstrated that more than 75% of cosmids containing a rare restriction site also contained a second rare restriction site, suggesting a high degree of CpG-rich restriction site clustering.

- (6) Typically the iron-iron axis (gz) of the binuclear iron-sulfur clusters is in the membrane plane.
- (7) The fourth cluster included the type strains of Actinobacillus lignieresii, A. equuli, A. pleuropneumoniae, A. suis, A. ureae, H. parahaemolyticus, H. parainfluenzae, H. paraphrohaemolyticus, H. ducreyi, and P. haemolytica.
- (8) Each species has approximately 500 core histones cluster repeats per haploid genome.
- (9) mycoides cluster' at a similarity level (S) of 66% and which remained undivided at up to 78% S. At higher similarity levels, these strains fell heterogeneously into mixed sub-phenons containing strains of both subspecies.
- (10) Thus, succinate dehydrogenase is the first enzyme which has been shown to contain all 3 of these Fe-S clusters.
- (11) We examined 10 life areas clustered around the general categories of "substance use," "social functioning," and "emotional and interpersonal functioning."
- (12) Genetic regulation of the ilvGMEDA cluster involves attenuation, internal promoters, internal Rho-dependent termination sites, a site of polarity in the ilvG pseudogene of the wild-type organism, and autoregulation by the ilvA gene product, the biosynthetic L-threonine deaminase.
- (13) Neutral sucrose density sedimentation patterns indicate that neutron-induced double strand-breaks sometimes occur in clusters of more than 100 in the same phage and that the effeciency with which double strand-breaks form is about 50 times that of gamma-induced double strand-breaks.
- (14) The difference in Brazil will be the huge distances involved, with the crazy decision not to host the group stages in geographical clusters leading to logistical and planning nightmares.
- (15) Fifty-four cases were analysed, and a two-fold excess of clustering within one year was observed, both within single districts and between adjacent districts.
- (16) All of the multivariate data were treated with mathematic method of cluster analysis.

- (17) The perinatal development of the levator ani (LA) muscle in male and female rats was investigated by measuring the total number of muscle units (MU) (i.e., mononucleate cells, clustered or independent myotubes, and muscle fibers) in transverse semithin sections of the entire muscle and the MU cross-sectional area in 22-day-old fetuses (F22), 1-day-old (D1 = day of birth), 3-day-old (D3), and 6-day-old (D6) newborns.
- (18) Since only a few of these medium sized terminals in any one cluster degenerate after tectal lesions, and none degenerate after cortical lesions, it is suggested that the morphological arrangement of these clusters may permit the convergence of axons from several sources, some of which are unidentified, onto the same dendritic segment.
- (19) A transurethral prostatic resection for prostatism in a 73 year old man showed a cluster of richly capillarised clear cells originally thought to be indicative of invasive carcinoma.
- (20) Moderately differentiated tumor revealed a wider range of nucleus size, less clustering (coefficient--3.59) and more hyperchromatic (70.1%) and "bare" (49.4%) nuclei and large nucleoli (22.2%).

Regression

Definition:

- (n.) The act of passing back or returning; retrogression; retrogradation.

Example Sentences:

- (1) Multiple stored energy levels were randomly tested and the percent successful defibrillation was plotted against the stored energy, and the raw data were fit by logistic regression.
- (2) It was found that linear extrapolations of log k' versus ET(30) plots to the polarity of unmodified aqueous mobile phase gave a more reliable value of log k'w than linear regressions of log k' versus volume percent.
- (3) Using multiple regression, a linear correlation was established between the cardiac index and the arterial-venous pH and PCO2 differences throughout shock and resuscitation (r2 = .91).

- (4) The ED50 and ED95 of mivacurium in each group were estimated from linear regression plots of log dose vs probit of maximum percentage depression of neuromuscular function.

- (5) Time-series analysis and multiple-regression modeling procedures were used to characterize changes in the overall incidence rate over the study period and to describe the contribution of additional measures to the dynamics of the incidence rates.

- (6) Regression curves indicate that although all three types of pulmonary edema can be characterized by slightly different slopes, the differences are statistically insignificant.

- (7) Excretion of inactive kallikrein again correlated with urine flow rate but the regression relationship between the two variables was different for water-load-induced and frusemide-induced diuresis.

- (8) Definite tumor regression, improvement of some clinical symptoms, and continuous remission over 6 mo or more were observed in six, nine, and three patients, respectively.

- (9) Regression analysis on the 21 clinical or laboratory parameters studied showed that the only variable independently associated with CSF-FN was the total protein concentration in the CSF; this, however, explained only 14% of the observed variation in the CSF-FN concentration and did not show any correlation with CNS involvement.

- (10) An experimental model was established in the ewe allowing one to predict with accuracy an antral follicle that coincidentally would either undergo ovulation (6-8 mm diameter) or atresia (3-4 mm diameter) following synchronization of luteal regression and the onset of the gonadotropin surge.

- (11) (2) A close correlation between the obesity index and serum GPT was recognized by elevation of the standard partial regression coefficient of serum GPT to obesity index and that of obesity index to serum GPT when the data from all 617 students was analysed in one group.

- (12) Regression of the tumor occurred during an episode of mechanical small bowel obstruction.

- (13) Odds ratios were computed by multiple logistic regression analysis and revealed no additional relationships; however, there were suggested dose-response gradients for height, weight at age 20, and body surface area in the Japanese women and for breast size in the Caucasian women.

- (14) The summary statistics examined are (a) the slope of the least-squares regression of the marker, (b) the average of the last r measurements, and (c) the difference between the averages of the last r and the first s measurements.
- (15) The authors used a linear multivariate regression to evaluate the effects of distance from the highway, age and sex of the child, and housing condition.
- (16) A multiple regression analysis between maxBIL and the significantly correlated parameters showed that only gestational age and birth weight remained significantly correlated with maxBIL.
- (17) Data from 579 medical students from the classes of 1979-80 through 1983-84 attending a midwestern medical college were analyzed via moderated multiple regression.
- (18) Comparing the regression lines of HR-QT and HR-QS2 separately for both groups, we found that both intervals decreased in parallel and the mean QT remained shorter than QS2 in both groups during exercise.
- (19) Early in the regression process, cholesterol esters are reduced at least partly by hydrolysis to yield cholesterol, some of which may crystallize and inhibit rapid regression.
- (20) Technically speaking, this modality of brief psychotherapy is based on the nonuse of transferential interpretations, on impeding the regression od the patient, on facilitating a cognitice-affective development of his conflicts and thus obtain an internal object mutation which allows the transformation of the "past" into true history, and the "present" into vital perspectives.

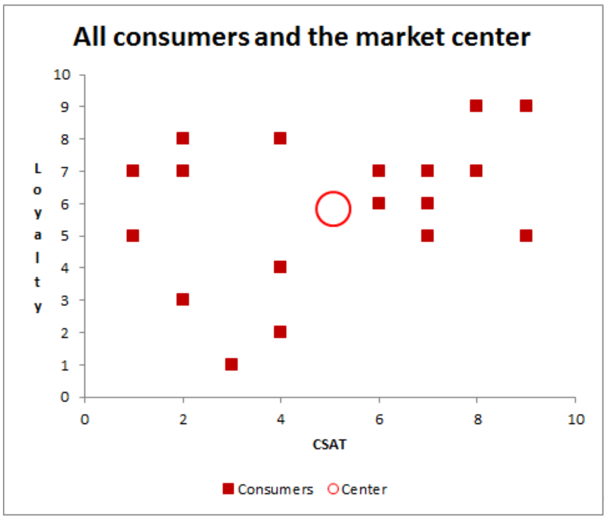
Market Segmentation with Cluster Analysis

There are multiple ways to segment a market, but one of the more precise and statistically valid approaches is to use a technique called cluster analysis. Cluster analysis is a tool that is used in lots of disciplines – not just marketing – basically anywhere there is lots of data to condense into clusters (or groups) – what we call market segments in marketing.

Let’s not be too concerned if the technique sounds too challenging – it’s actually quite straightforward and easy to understand – and you should get a lot of value from using the technique. So let’s start at the beginning – a cluster is a related set of data, things or objects. You might have heard people refer to a group of stars in the sky as a “cluster of stars” – just a group of stars that appear to sit together.

The same concept applies to the market segmentation process – in that we are trying to group consumer data (their behaviors, needs, attitudes, and so on) into related sets. And to help to undertake this grouping (clustering) process, we use cluster analysis to review and create market segments.

A simple example of how cluster analysis works



To get a quick understanding of how cluster analysis works for market segmentation purposes, let's use the two variables of "customer satisfaction" scores and a "loyalty" metric to help segment the customers on a database. Let's assume that we have customer satisfaction (CSAT) scores of 1 to 9 (where 1 = very dissatisfied and 9 = very satisfied). And we have similar scores for the customer's level of loyalty (1 = high switcher-low loyalty and 9 = non-switcher-high loyalty).

This graph shows this customer database information mapped onto a scatter-plot graph. The red squares represent the scores of the individual customers and the large red circle is the average score of all the customers for CSAT (average = 5.05) and loyalty (average = 5.85).

But if we look closely at the plot points – for the purpose of identifying clusters (market segments), there is a suggestion of three possible inherent market segments – this is done using a rough visual basis as shown in the next chart

– which the same as above, except for the addition of a top-level segmentation approach (using the extra large circles).

You should be able to see that there are three clusters (segments) of consumers suggested by the data as presented. The black circle (top-right) appears to be loyal customers, with a high level of customer satisfaction. The blue circle (bottom-left) appears to be less loyal customers, with a lower level of CSAT. This relationship is probably obvious and to be expected – and our existing marketing programs (to existing customers) are probably built around this CSAT-loyalty correlation.

However – take a look at the red circle (top-right) – this segment consists of largely unsatisfied, yet quite loyal customers. This is an interesting finding and perhaps unexpected (somewhat of a marketing insight). This is one reason why looking at different approaches to market segments is often worthwhile.

Running cluster analysis on Excel

Note: In addition to this Market Segmentation Study Guide, I have also developed Cluster Analysis for Marketing – where you can download a free Excel template for quickly and easily running cluster analysis.

With the above example – because we are only considering two variables (CSAT and loyalty) – we can attempt to segment the customer data on a visual basis as we have done above. But if we have lots more customers to graph, or if we want to consider more than two variables to create the segmentation, then we can easily use the cluster analysis Excel template (see link above) to construct the segments and produce some helpful statistical measures.

This new graph has been automatically produced by the Excel template – you don't need to perform any calculations or even have a good knowledge of Excel.

As you can see, the Excel spreadsheet has classified each customer data point into a market segment (e.g. blue diamonds = segment 1), and it has also calculated the center (average) for each segment. In this case, the center (average of customers) in segment 1 is 7.40 for CSAT and 6.80 for loyalty.

What is really interesting is that segment 2 (black dots/circles) actually has a higher loyalty score (7.0) average than segment 1.

This template for automatically running cluster analysis will also calculate the relative segment sizes for you and produce this graph as well.

As you can see, the segment positions are the same as the previous chart, but without the individual customer data (which is helpful if you have lots of customer data to simplify).

This graph – known as a segmentation map – also includes the size of each market segment, allowing you to measure and forecast the segment and its potential.

Introduction to Classification:

Classification may be defined as the process of predicting class or category from observed values or given data points. The categorized output can have the form such as “Black” or “White” or “spam” or “no spam”.

Mathematically, classification is the task of approximating a mapping function (f) from input variables (X) to output variables (Y). It is basically belongs to the supervised machine learning in which targets are also provided along with the input data set.

An example of classification problem can be the spam detection in emails. There can be only two categories of output, “spam” and “no spam”; hence this is a binary type classification.

To implement this classification, we first need to train the classifier. For this example, “spam” and “no spam” emails would be used as the training data. After successfully train the classifier, it can be used to detect an unknown email.

Types of Learners in Classification

We have two types of learners in respective to classification problems −

Lazy Learners

As the name suggests, such kind of learners waits for the testing data to be appeared after storing the training data. Classification is done only after getting the testing data. They spend less time on training but more time on predicting. Examples of lazy learners are K-nearest neighbor and case-based reasoning.

Eager Learners

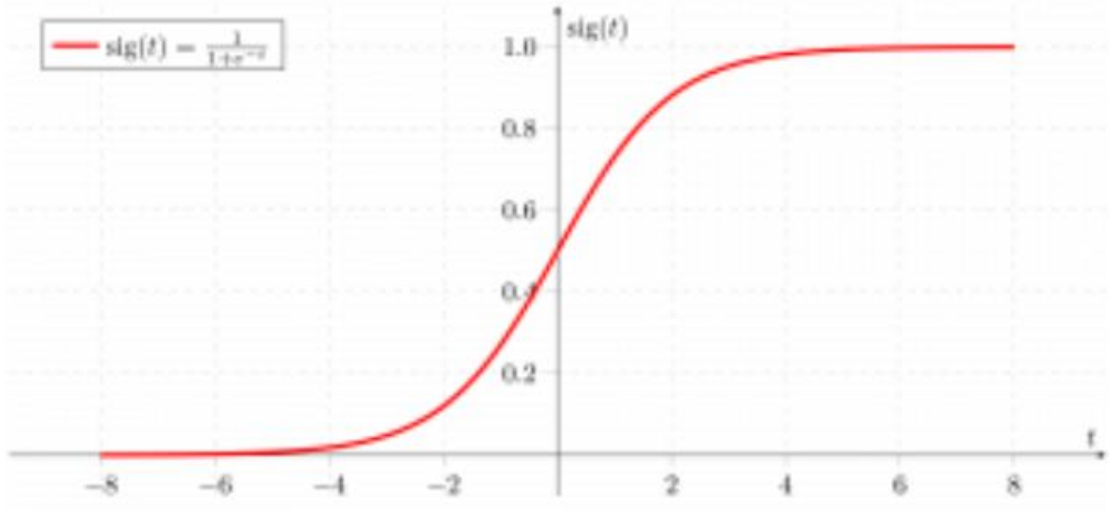
As opposite to lazy learners, eager learners construct classification model without waiting for the testing data to be appeared after storing the training data. They spend more time on training but less time on predicting. Examples of eager learners are Decision Trees, Naïve Bayes and Artificial Neural Networks (ANN).

4 Applications of Classification Algorithms

- Sentiment Analysis Sentiment analysis is a machine learning text analysis technique that assigns sentiment (opinion, feeling, or emotion) to words within a text, or an entire text, on a polarity scale of Positive, Negative, or Neutral. ...
- Email Spam Classification ...
- Document Classification

- # Logistic Regression

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for a given set of features(or inputs), X. Contrary to popular belief, logistic regression IS a regression model. The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the sigmoid function.



Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

The decision for the value of the threshold value is majorly affected by the values of precision and recall. Ideally, we want both precision and recall to be 1, but this seldom is the case.

In the case of a Precision-Recall tradeoff, we use the following arguments to decide upon the threshold:-

1. Low Precision/High Recall: In applications where we want to reduce the number of false negatives without necessarily reducing the number of false positives, we choose a decision value that has a low value of Precision or a high value of Recall. For example, in a cancer diagnosis application, we do not want any affected patient to be classified as not affected without giving much heed to if the patient is being wrongfully diagnosed with cancer. This is because the absence of cancer can be detected by further medical diseases but the presence of the disease cannot be detected in an already rejected candidate.

2. High Precision/Low Recall: In applications where we want to reduce the number of false positives without necessarily reducing the number of false negatives, we choose a decision value that has a high value of Precision or a low value of Recall. For example, if we are classifying customers whether they will react positively or negatively to a personalized advertisement, we want to be absolutely sure that the customer will react positively to the advertisement because otherwise, a negative reaction can cause a loss of potential sales from the customer.

Based on the number of categories, Logistic regression can be classified as:

1. **binomial:** target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.
2. **multinomial:** target variable can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.
3. **ordinal:** it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

SVM:

Introduction to SVMs: In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

What is Support Vector Machine?

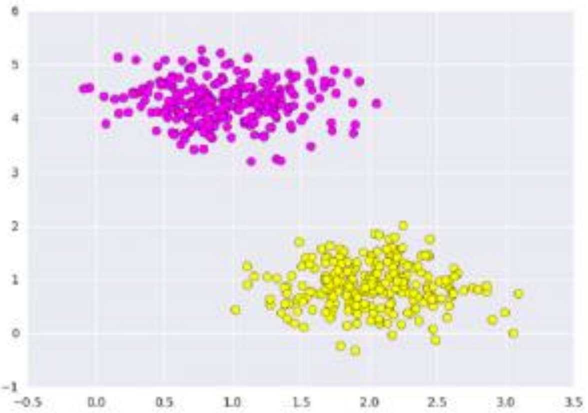
An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification, implicitly mapping their inputs into high-dimensional feature spaces.

What does SVM do?

Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. Let you have basic understandings from this article before you proceed further. Here I'll discuss an example about SVM classification of cancer UCI datasets using machine learning tools i.e. scikit-learn compatible with Python. **Pre-requisites:** Numpy, Pandas, matplot-lib, scikit-learn Let's have a quick example of support vector classification. First we need to create a dataset:

- python3

```
# importing scikit learn with make_blobs
from sklearn.datasets.samples_generator import make_blobs


# creating datasets X containing n_samples
# Y containing two classes
X, Y = make_blobs(n_samples=500, centers=2,
                  random_state=0, cluster_std=0.40)
import matplotlib.pyplot as plt
# plotting scatters
plt.scatter(X[:, 0], X[:, 1], c=Y, s=50, cmap='spring');
plt.show()
```

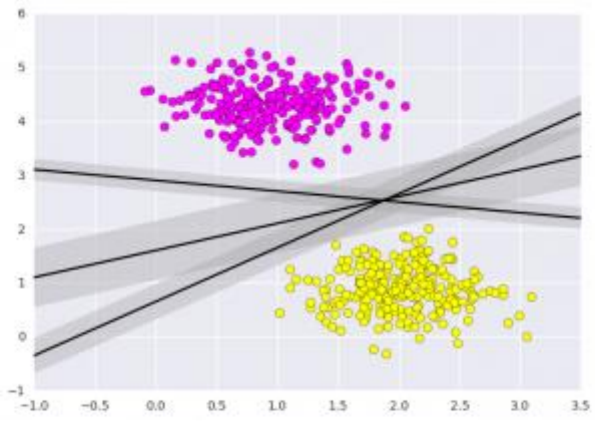
Output: 

What Support vector machines do, is to not only draw a line between two classes here, but consider a region about the line of some given width. Here's an example of what it can look like:

- python3

```
# creating linspace between -1 to 3.5

xfit = np.linspace(-1, 3.5)



# plotting scatter

plt.scatter(X[:, 0], X[:, 1], c=Y, s=50, cmap='spring')



# plot a line between the different sets of data

for m, b, d in [(1, 0.65, 0.33), (0.5, 1.6, 0.55), (-0.2, 2.9, 0.2)]:

    yfit = m * xfit + b

    plt.plot(xfit, yfit, '-k')

    plt.fill_between(xfit, yfit - d, yfit + d, edgecolor='none',

    color='#AAAAAA', alpha=0.4)
```

```
plt.xlim(-1, 3.5);

plt.show()
```



Importing datasets

This is the intuition of support vector machines, which optimize a linear discriminant model representing the perpendicular distance between the datasets. Now let’s train the classifier using our training data. Before training, we need to import cancer datasets as csv file where we will train two features out of all features.

- python3

```
# importing required libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt




# reading csv file and extracting class column to y.

x = pd.read_csv("C:\...\cancer.csv")

a = np.array(x)

y  = a[:,30] # classes having 0 and 1
```

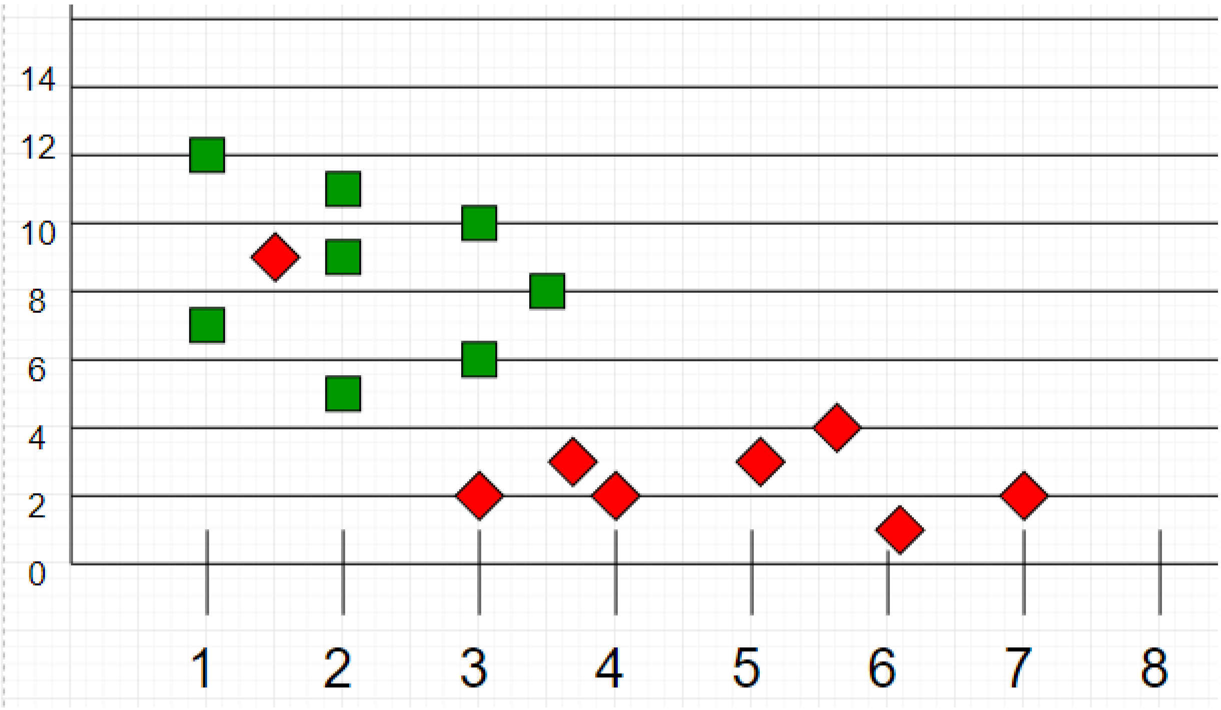
```
# extracting two features

x = np.column_stack((x.malignant,x.benign))



# 569 samples and 2 features

x.shape



print (x),(y)
```

```
[[  122.8   1001.  ]
 [  132.9   1326.  ]
 [  130.    1203.  ]
 ...,
 [  108.3    858.1 ]
 [  140.1   1265.  ]
 [   47.92   181.  ]]

array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
0.,
        0.,  0.,  0.,  0.,  0.,  0.,  1.,  1.,  1.,  0.,  0.,  0.,
0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.,
0.,
        0.,  0.,  0.,  0.,  0.,  0.,  0.,  1.,  0.,  1.,  1.,  1.,
1.,
        1.,  0.,  0.,  1.,  0.,  0.,  1.,  1.,  1.,  1.,  0.,  1.,
....,
        1.])
```

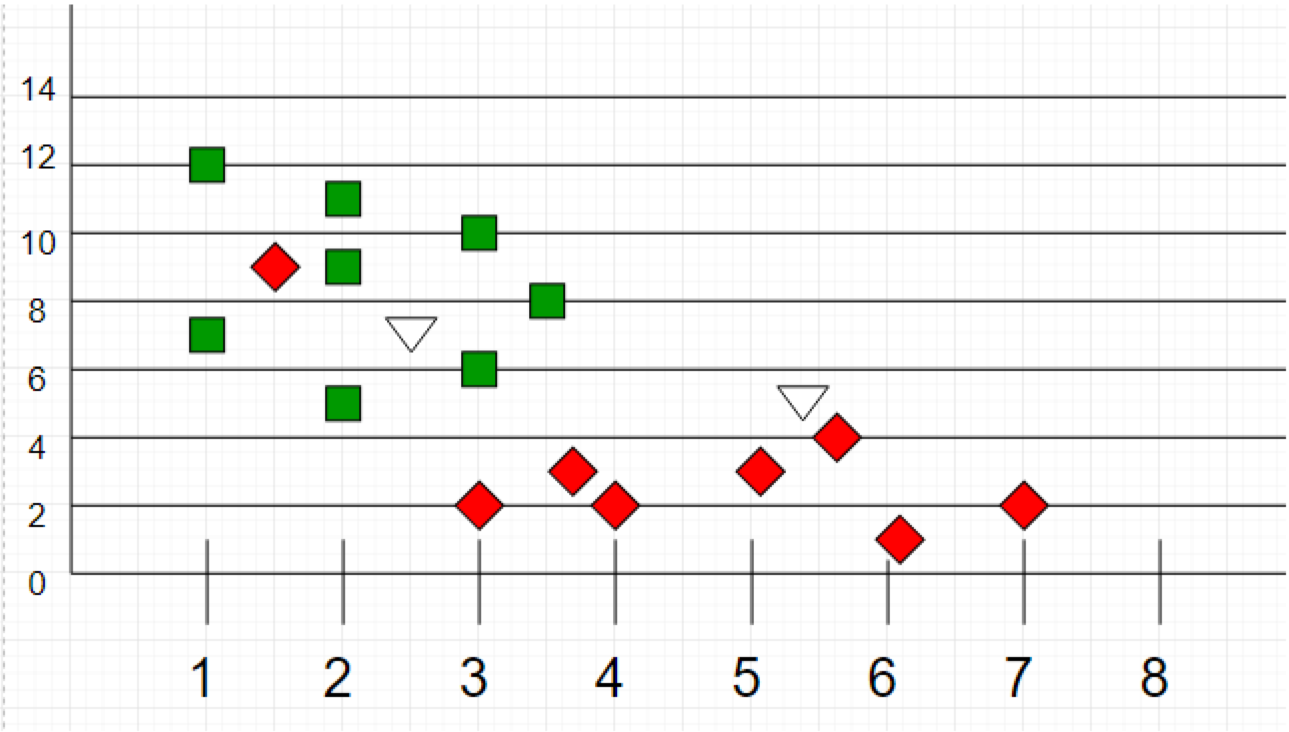
Fitting a Support Vector Machine

Now we’ll fit a Support Vector Machine Classifier to these points. While the mathematical details of the likelihood model are interesting, we’ll let read about those elsewhere. Instead, we'll just treat the scikit-learn algorithm as a black box which accomplishes the above task.

K-Nearest Neighbours is one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.
It is widely disposable in real-life scenarios since it is non-parametric, meaning, it does not make any underlying assumptions about the distribution of data (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).
We are given some prior data (also called training data), which classifies coordinates into groups identified by an attribute.
As an example, consider the following table of data points containing two features:



Now, given another set of data points (also called testing data), allocate these points a group by analyzing the training set. Note that the unclassified points are marked as 'White'.



Intuition
If we plot these points on a graph, we may be able to locate some clusters or groups. Now, given an unclassified point, we can assign it to a group by observing what group its nearest neighbours belong to. This means a point close to a cluster of points classified as 'Red' has a higher probability of getting classified as 'Red'.
Intuitively, we can see that the first point (2.5, 7) should be classified as 'Green' and the second point (5.5, 4.5) should be classified as 'Red'.
Algorithm
Let m be the number of training data samples. Let p be an unknown point.

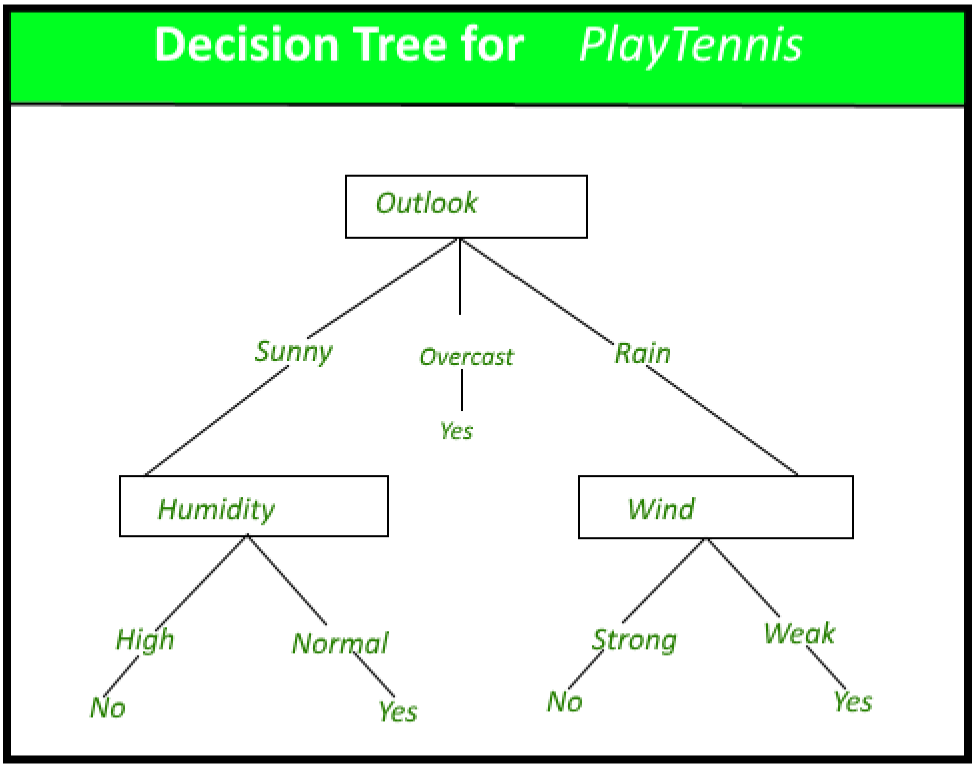
1. Store the training samples in an array of data points arr[]. This means each element of this array represents a tuple (x, y).

```
for i=0 to m:

  Calculate Euclidean distance d(arr[i], p).
```

1. Make set S of K smallest distances obtained. Each of these distances corresponds to an already classified data point.
2. Return the majority label among S.

Decision Trees:

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.



A decision tree for the concept PlayTennis.

Construction of Decision Tree: A tree can be *"learned"* by splitting the source set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called *recursive partitioning*. The recursion is completed when the subset at a node all has the same value of the target variable, or when splitting no longer adds value to the predictions. The construction of a decision tree classifier does not require any domain knowledge or parameter setting, and therefore is appropriate for exploratory knowledge discovery. Decision trees can handle high-dimensional data. In general decision tree classifier has good accuracy. Decision tree induction is a typical inductive approach to learn knowledge on classification.

Decision Tree Representation:

Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the above figure. This process is then repeated for the subtree rooted at the new node.

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No).
For example, the instance

(Outlook = Sunny, Temperature = Hot, Humidity = High, Wind = Strong)

would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.
In other words, we can say that the decision tree represents a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ^ Humidity = Normal) v (Outlook = Overcast) v (Outlook = Rain ^ Wind = Weak)

Gini Index:

Gini Index is a score that evaluates how accurate a split is among the classified groups. Gini index evaluates a score in the range between 0 and 1, where 0 is when all observations belong to one class, and 1 is a random distribution of the elements within classes. In this case, we want to have a Gini index score as low as possible. Gini Index is the evaluation metrics we shall use to evaluate our Decision Tree Model.

Strengths and Weaknesses of the Decision Tree approach
The strengths of decision tree methods are:

- Decision trees are able to generate understandable rules.
- Decision trees perform classification without requiring much computation.
- Decision trees are able to handle both continuous and categorical variables.
- Decision trees provide a clear indication of which fields are most important for prediction or classification.

The weaknesses of decision tree methods :

- Decision trees are less appropriate for estimation tasks where the goal is to predict the value of a continuous attribute.
- Decision trees are prone to errors in classification problems with many classes and a relatively small number of training examples.
- Decision tree can be computationally expensive to train. The process of growing a decision tree is computationally expensive. At each node, each candidate splitting field must be sorted before its best split can be found. In some algorithms, combinations of fields are used and a search must be made for optimal combining weights. Pruning algorithms can also be expensive since many candidate sub-trees must be formed and compared.