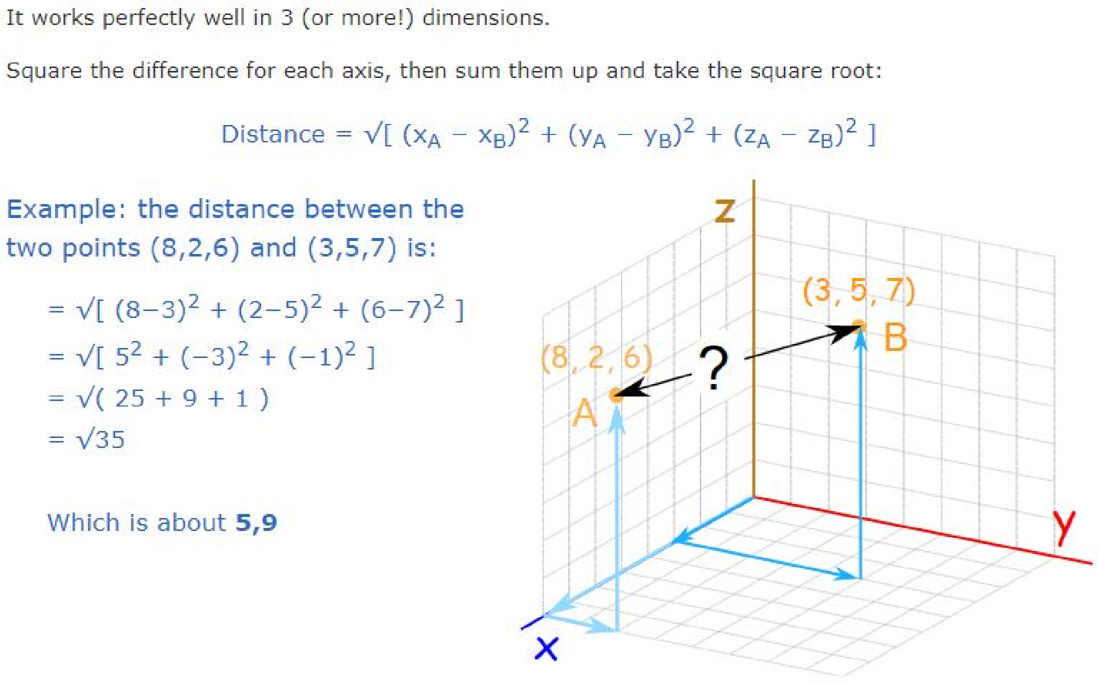
1. **computing distances between points (vectors) -- e.g using Euclidian distance, computing dot product**

Euclidian



It works perfectly well in 3 (or more!) dimensions.

Square the difference for each axis, then sum them up and take the square root:

$$\text{Distance} = \sqrt{[(x_A - x_B)^2 + (y_A - y_B)^2 + (z_A - z_B)^2]}$$

Example: the distance between the two points (8,2,6) and (3,5,7) is:

$$= \sqrt{[(8-3)^2 + (2-5)^2 + (6-7)^2]}$$
$$= \sqrt{[5^2 + (-3)^2 + (-1)^2]}$$
$$= \sqrt{(25 + 9 + 1)}$$
$$= \sqrt{35}$$

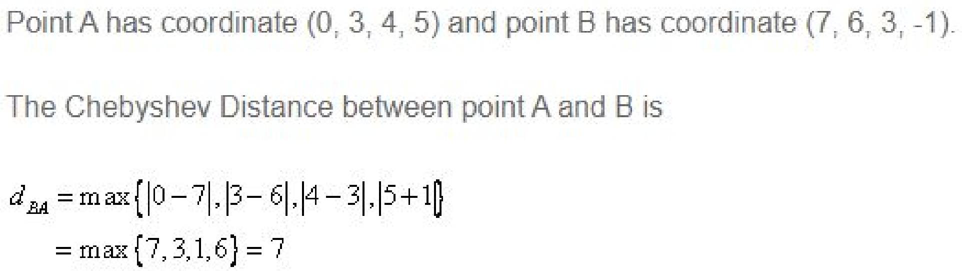
Which is about **5,9**

Manhattan Distance

between two points (x1, y1) and (x2, y2) is:

|x1 – x2| + |y1 – y2|

Cherbyshev distance

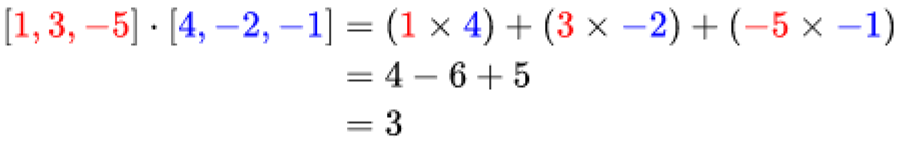


Point A has coordinate (0, 3, 4, 5) and point B has coordinate (7, 6, 3, -1).

The Chebyshev Distance between point A and B is

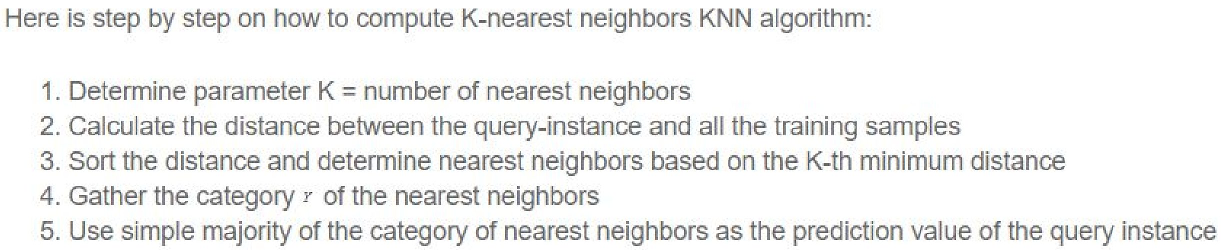
$$d_{BA} = \max\{|0-7|, |3-6|, |4-3|, |5+1|\}$$
$$= \max\{7, 3, 1, 6\} = 7$$

Dot product



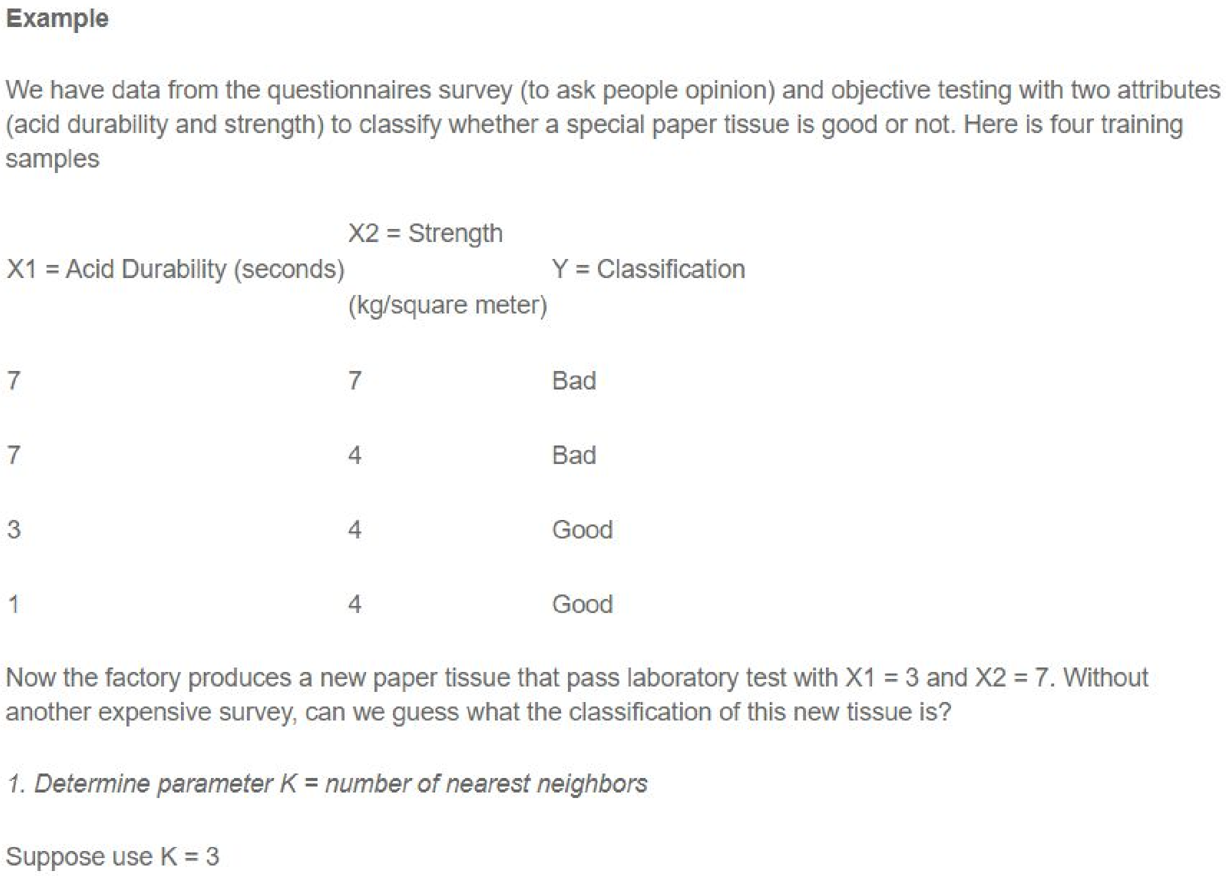
$$\begin{aligned}[1, 3, -5] \cdot [4, -2, -1] &= (1 \times 4) + (3 \times -2) + (-5 \times -1) \\ &= 4 - 6 + 5 \\ &= 3\end{aligned}$$

2. K-nn algorithm



Here is step by step on how to compute K-nearest neighbors KNN algorithm:

1. Determine parameter K = number of nearest neighbors
2. Calculate the distance between the query-instance and all the training samples
3. Sort the distance and determine nearest neighbors based on the K-th minimum distance
4. Gather the category r of the nearest neighbors
5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance



Example

We have data from the questionnaires survey (to ask people opinion) and objective testing with two attributes (acid durability and strength) to classify whether a special paper tissue is good or not. Here is four training samples

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Y = Classification
7	7	Bad
7	4	Bad
3	4	Good
1	4	Good

Now the factory produces a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7. Without another expensive survey, can we guess what the classification of this new tissue is?

1. Determine parameter K = number of nearest neighbors

Suppose use K = 3

2. Calculate the distance between the query-instance and all the training samples

Coordinate of query instance is (3, 7), instead of calculating the distance we compute square distance which is faster to calculate (without square root)

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	**Square Distance to query instance (3, 7)**
7	7	$(7-3)^2+(7-7)^2=16$
7	4	$(7-3)^2+(4-7)^2=25$
3	4	$(3-3)^2+(4-7)^2=9$
1	4	$(1-3)^2+(4-7)^2=13$

3. *Sort the distance and determine nearest neighbors based on the K-th minimum distance*

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	**Rank minimum distance**	**Is it included in 3-Nearest neighbors?**
7	7	$(7-3)^2+(7-7)^2=16$	**3**	**Yes**
7	4	$(7-3)^2+(4-7)^2=25$	**4**	**No**
3	4	$(3-3)^2+(4-7)^2=9$	**1**	**Yes**
1	4	$(1-3)^2+(4-7)^2=13$	**2**	**Yes**

4. Gather the category Y of the nearest neighbors. Notice in the second row last column that the category of nearest neighbor (Y) is not included because the rank of this data is more than 3 (=K).

X1 = Acid Durability (seconds)	X2 = Strength (kg/square meter)	Square Distance to query instance (3, 7)	Rank minimum distance	Is it included in 3-Nearest neighbors?	**Y = Category of nearest Neighbor**
7	7	$(7-3)^2+(7-7)^2=16$	3	Yes	**Bad**
7	4	$(7-3)^2+(4-7)^2=25$	4	No	**-**
3	4	$(3-3)^2+(4-7)^2=9$	1	Yes	**Good**
1	4	$(1-3)^2+(4-7)^2=13$	2	Yes	**Good**

5. *Use simple majority of the category of nearest neighbors as the prediction value of the query instance*

We have 2 good and 1 bad, since 2>1 then we conclude that a new paper tissue that pass laboratory test with X1 = 3 and X2 = 7 is included in **Good** category.

3. Perceptron (e.g know learning procedure and how to train perceptron)

Training rule

Updating Weight

$$W' = W + (d - y) \alpha X$$

W'- new weight

w - current weight

d - desired outcome

y - actual outcome

a - learning rule, between 0...1 (mostly random)

X -Input

Activation of perceptron

$$W1*I1 + W2I2 + \cdots + Wn*In >= \ominus$$

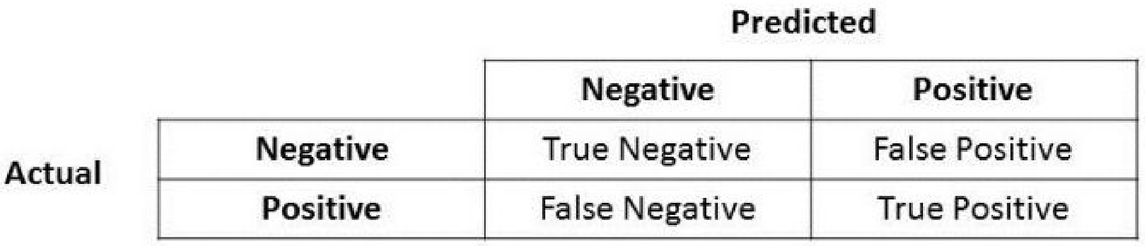
Update thershold

$$\ominus' = \ominus +(\text{activation state} - <W1*I1 + W2I2 + \cdots + Wn*In>) * \alpha *-1$$

X - activation state (1 or 0)

4. Calculate accuracy, precision, recall, and F-measure and know what is "true positive", "false positive" etc in a confusion matrix

True positive - Correctly predicted/classified as true
True negative - Correctly predicted/classified as false
False positive - Uncorrectly predicted/classified as true
False negative - Uncorrectly predicted/classified as false



		Predicted	
		Negative	**Positive**
Actual	**Negative**	True Negative	False Positive
	Positive	False Negative	True Positive

$$Accuracy \;=\; \frac{True\ Positive + True\ Negative}{All}$$

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

$$\text{F1} = 2 \text{ x } \frac{Precision \ast Recall}{Precision + Recall}$$

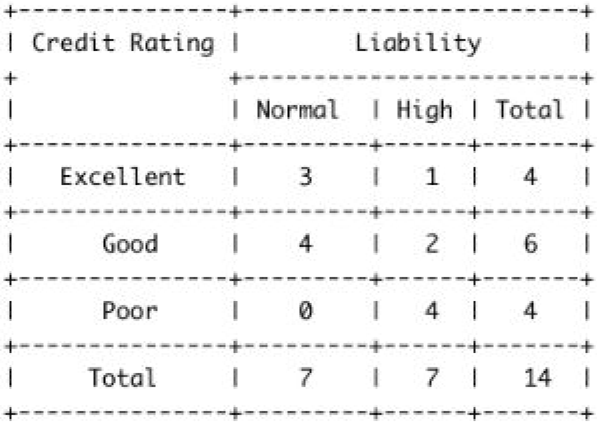
5. How to calculate Entropy

$$E(S)=\sum_{i=1}^{c}-p_i\log_2 p_i$$

6 białych i 4 czarne kulki:

$$-\frac{6}{10}\log_2(0.6)+-\frac{4}{10}\log_2(0.4)$$

Entropy = 0.970951



Credit Rating	Liability		
	Normal	High	Total
Excellent	3	1	4
Good	4	2	6
Poor	0	4	4
Total	7	7	14

Information gain from x on y

$$IG(\,Y,X)\;=E(\,Y)\;-\;E(\,Y|X)$$

$$E(\,Liability)=-\frac{7}{14}\log_2\left(\frac{7}{14}\right)-\frac{7}{14}\log_2\left(\frac{7}{14}\right)$$

$$=-\frac{1}{2}\log_2\left(\frac{1}{2}\right)-\frac{1}{2}\log_2\left(\frac{1}{2}\right)$$

$$=1$$

$$E(\mathit{Liability} \mid CR = \mathit{Excellent}) = -\ \frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) \approx 0.811$$

$$E(\mathit{Liability} \mid CR = \mathit{Good}) = -\ \frac{4}{6}\log_2\left(\frac{4}{6}\right) - \frac{2}{6}\log_2\left(\frac{2}{6}\right) \approx 0.918$$

$$E(\mathit{Liability} \mid CR = \mathit{Poor}) = -\ 0\log_2(0) - \frac{4}{4}\log_2\left(\frac{4}{4}\right) = 0$$

Weighted Average:

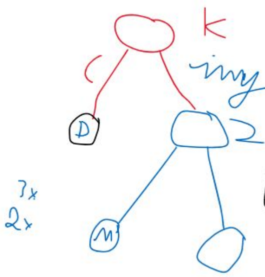
$$E(\mathit{Liability} \mid CR) = \frac{4}{14} \times 0.811 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0$$

$$= 0.625$$

IG = 1-0.625

6. Build an optimal decision tree using the entropy criterion

$$IG(Y, X) = E(Y) - E(Y|X)$$



iny kala

6 4

k	C	B
M	3	2
D	1	4

$E(R) = -\frac{1}{2} \cdot \log_2 \frac{1}{2} - \frac{1}{2} \log_2 \frac{1}{2}$

$IG = 0.97 - (0,97 \cdot \frac{5}{10} + 0,72 \cdot \frac{5}{10} = 0,845)$

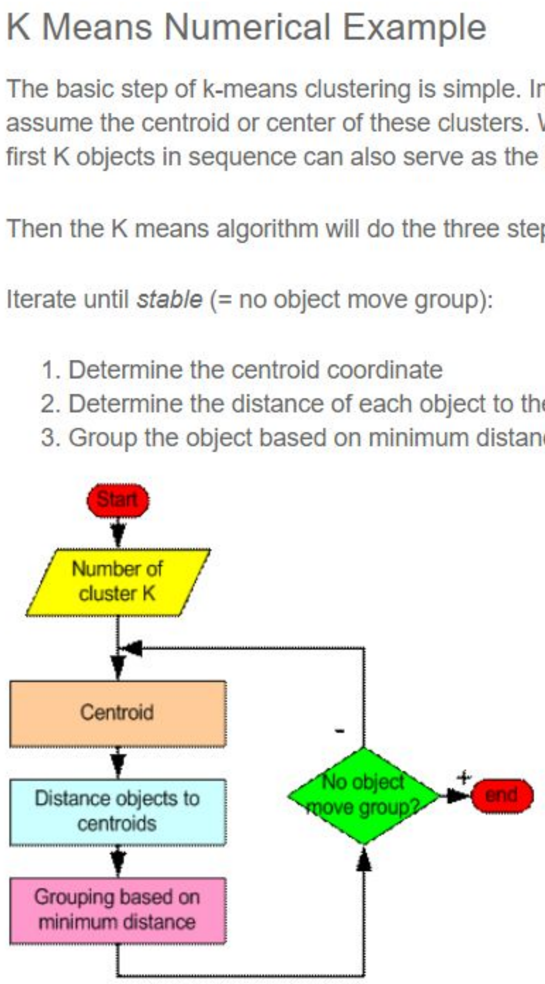
$IG = 0.97 - 0,845 = 0,125$

$E(Color|M) = 0,97$

$E(Color|D) = 0,72$

$IG = 0.97 - ((-\frac{3}{5}) \log_2 \frac{3}{5} - \frac{2}{5} \log_2 \frac{2}{5}$

7. Perform the k-means algorithm for the given points.



K Means Numerical Example

The basic step of k-means clustering is simple. In the beginning we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects in sequence can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence

Iterate until *stable* (= no object move group):

1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance

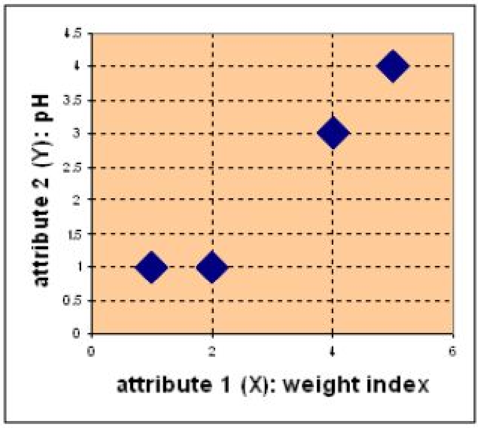
Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

The numerical example below is given to understand this simple iteration. You may download the implementation of this numerical example as Matlab code here . Another example of interactive k- means clustering using Visual Basic (VB) is also available here . MS excel file for this numerical example can be downloaded at the bottom of this page.

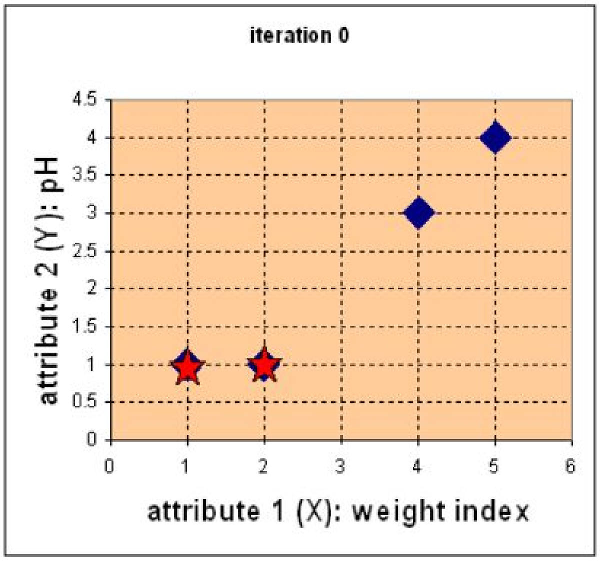
Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. Our goal is to group these objects into K=2 group of medicine based on the two features (pH and weight index).

Object	attribute 1 (X): weight index	attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Each medicine represents one point with two attributes (X, Y) that we can represent it as coordinate in an attribute space as shown in the figure below.



1. *Initial value of centroids* : Suppose we use medicine A and medicine B as the first centroids. Let $\mathbf{c}_1$ and $\mathbf{c}_2$ denote the coordinate of the centroids, then $\mathbf{c}_1 = (1,1)$ and $\mathbf{c}_2 = (2,1)$



2. *Objects-Centroids distance* : we calculate the distance between cluster centroid to each object. Let us use Euclidean distance , then we have distance matrix at iteration 0 is

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{c} X \\ Y \end{array}$$

Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from medicine C = (4, 3) to the first centroid $\mathbf{c}_1 = (1,1)$ is $\sqrt{(4-1)^2 + (3-1)^2} = 3.61$, and its distance to the second centroid $\mathbf{c}_2 = (2,1)$ is $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$, etc.

3. *Objects clustering* : We assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{array}{cc} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{l} group-1 \\ group-2 \end{array} \\ \begin{array}{cccc} A & B & C & D \end{array} & \end{array}$$

4. *Iteration-1, determine centroids* : Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in $\mathbf{c}_1 = (1,1)$. Group 2 now has three members, thus the centroid is the average coordinate among the three members: $\mathbf{c}_2 = (\frac{2+4+5}{3}, \frac{1+3+4}{3}) = (\frac{11}{3}, \frac{8}{3})$.

5. *Iteration-1, Objects-Centroids distances* : The next step is to compute the distance of all objects to the new centroids. Similar to step 2, we have distance matrix at iteration 1 is

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{array}{ll} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3},\frac{8}{3}) & group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{l} X \\ Y \end{array}$$

6. *Iteration-1, Objects clustering:* Similar to step 3, we assign each object based on the minimum distance. Based on the new distance matrix, we move the medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{array}{l} group-1 \\ group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

7. *Iteration 2, determine centroids:* Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are $\mathbf{c}_1 = (\frac{1+2}{2},\frac{1+1}{2}) = (1\frac{1}{2},1)$ and $\mathbf{c}_2 = (\frac{4+5}{2},\frac{3+4}{2}) = (4\frac{1}{2},3\frac{1}{2})$

8. *Iteration-2, Objects-Centroids distances* : Repeat step 2 again, we have new distance matrix at iteration 2 as

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{array}{ll} \mathbf{c}_1 = (1\frac{1}{2},1) & group-1 \\ \mathbf{c}_2 = (4\frac{1}{2},3\frac{1}{2}) & group-2 \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{l} X \\ Y \end{array}$$

9. *Iteration-2, Objects clustering:* Again, we assign each object based on the minimum distance.

$$\mathbf{G}^2 = \begin{array}{c} \left[\begin{array}{cccc} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array}\right] \\ \begin{array}{cccc} A & B & C & D \end{array} \end{array} \begin{array}{l} group-1 \\ group-2 \\ \end{array}$$

We obtain result that $\mathbf{G}^2 = \mathbf{G}^1$. Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore. Thus, the computation of the k-mean clustering has reached its stability and no more iteration is needed. We get the final grouping as the results

Object	Feature 1 (X): weight index	Feature 2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

Ck- is the number of observations in the K-th cluster

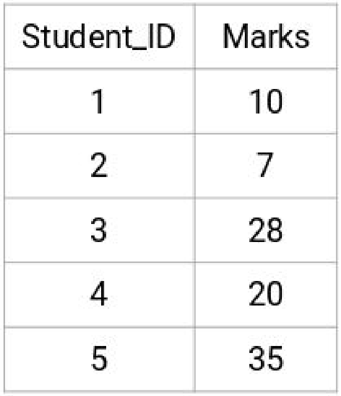
$$\mathrm{WCV}(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2,$$

8. Perform a hierarchical clustering.

Setting up the Example

Suppose a teacher wants to divide her students into different groups. She has the marks scored by each student in an assignment and based on these marks, she wants to segment them into groups. There's no fixed target here as to how many groups to have. Since the teacher does not know what type of students should be assigned to which group, it cannot be solved as a supervised learning problem. So, we will try to apply hierarchical clustering here and segment the students into different groups.

Let's take a sample of 5 students:

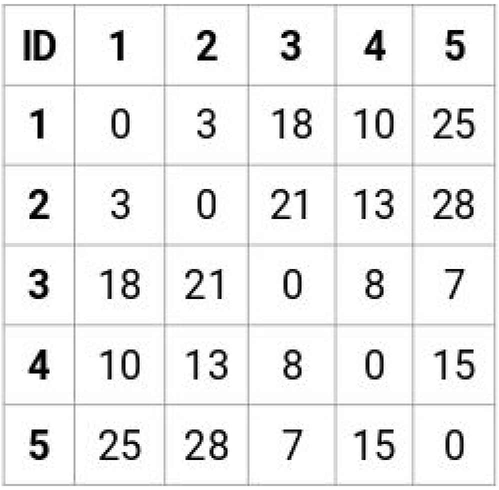


Student_ID	Marks
1	10
2	7
3	28
4	20
5	35

Creating a Proximity Matrix

First, we will create a proximity matrix which will tell us the distance between each of these points. Since we are calculating the distance of each point from each of the other points, we will get a square matrix of shape n X n (where n is the number of observations).

Let’s make the 5 x 5 proximity matrix for our example:



ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

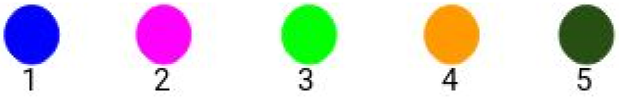
The diagonal elements of this matrix will always be 0 as the distance of a point with itself is always 0. We will use the Euclidean distance formula to calculate the rest of the distances. So, let’s say we want to calculate the distance between point 1 and 2:

$$\sqrt{(10-7)^2} = \sqrt{9} = 3$$

Similarly, we can calculate all the distances and fill the proximity matrix.

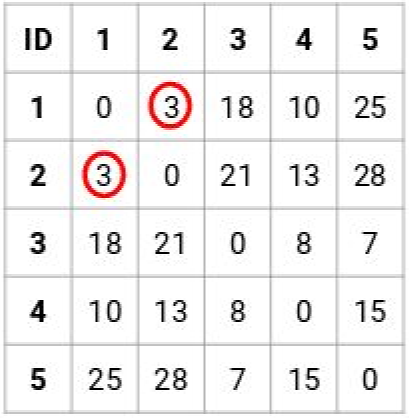
Steps to Perform Hierarchical Clustering

Step 1: First, we assign all the points to an individual cluster:



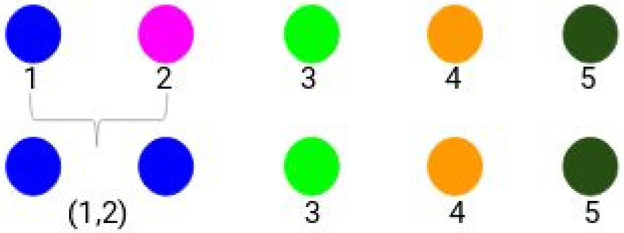
Different colors here represent different clusters. You can see that we have 5 different clusters for the 5 points in our data.

Step 2: Next, we will look at the smallest distance in the proximity matrix and merge the points with the smallest distance. We then update the proximity matrix:

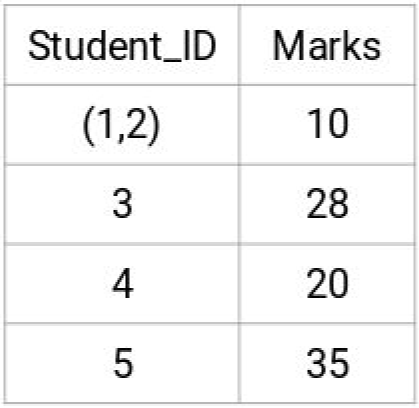


ID	1	2	3	4	5
1	0	3	18	10	25
2	3	0	21	13	28
3	18	21	0	8	7
4	10	13	8	0	15
5	25	28	7	15	0

Here, the smallest distance is 3 and hence we will merge point 1 and 2:

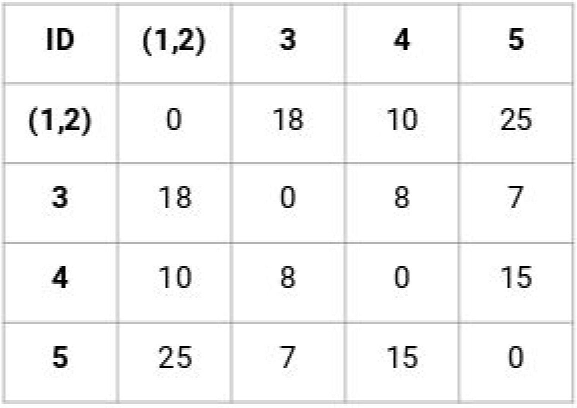


Let’s look at the updated clusters and accordingly update the proximity matrix:



Student_ID	Marks
(1,2)	10
3	28
4	20
5	35

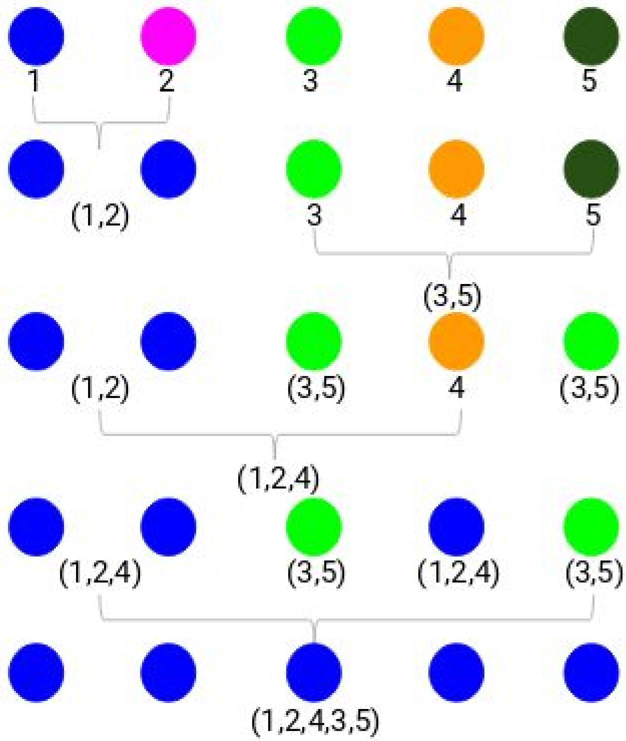
Here, we have taken the maximum of the two marks (7, 10) to replace the marks for this cluster. Instead of the maximum, we can also take the minimum value or the average values as well. Now, we will again calculate the proximity matrix for these clusters:



ID	(1,2)	3	4	5
(1,2)	0	18	10	25
3	18	0	8	7
4	10	8	0	15
5	25	7	15	0

Step 3: We will repeat step 2 until only a single cluster is left.

So, we will first look at the minimum distance in the proximity matrix and then merge the closest pair of clusters. We will get the merged clusters as shown below after repeating these steps:



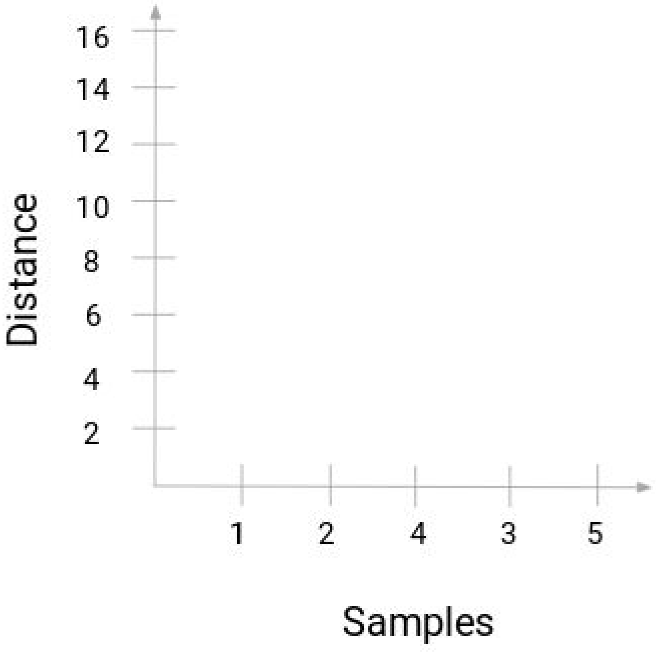
We started with 5 clusters and finally have a single cluster. **This is how agglomerative hierarchical clustering works.** But the burning question still remains – how do we decide the number of clusters? Let’s understand that in the next section.

How should we Choose the Number of Clusters in Hierarchical Clustering?

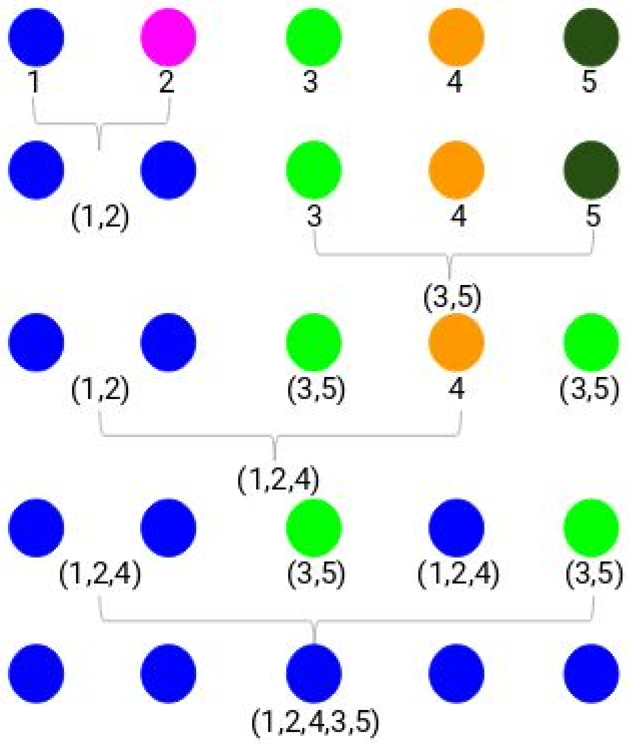
Ready to finally answer this question that's been hanging around since we started learning? To get the number of clusters for hierarchical clustering, we make use of an awesome concept called a **Dendrogram**.

A dendrogram is a tree-like diagram that records the sequences of merges or splits.

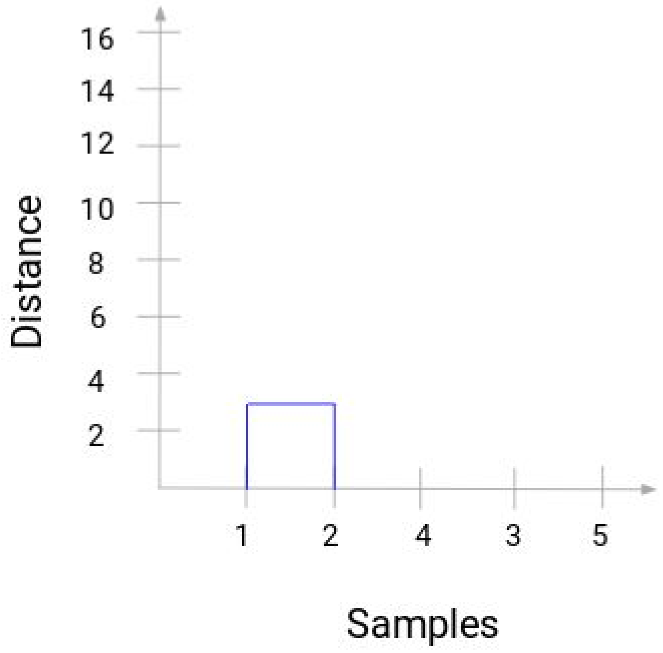
Let's get back to our teacher-student example. Whenever we merge two clusters, a dendrogram will record the distance between these clusters and represent it in graph form. Let's see how a dendrogram looks like:



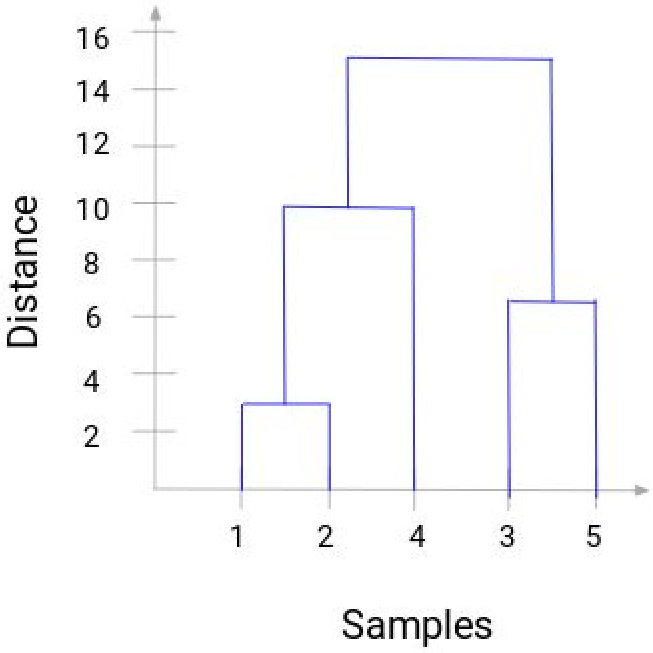
We have the samples of the dataset on the x-axis and the distance on the y-axis. **Whenever two clusters are merged, we will join them in this dendrogram and the height of the join will be the distance between these points.** Let's build the dendrogram for our example:



Take a moment to process the above image. We started by merging sample 1 and 2 and the distance between these two samples was 3 (refer to the first proximity matrix in the previous section). Let's plot this in the dendrogram:

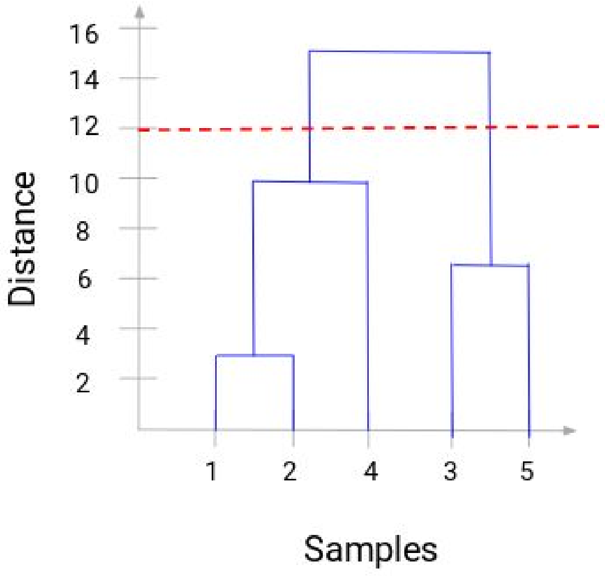


Here, we can see that we have merged sample 1 and 2. The vertical line represents the distance between these samples. Similarly, we plot all the steps where we merged the clusters and finally, we get a dendrogram like this:



We can clearly visualize the steps of hierarchical clustering. More the distance of the vertical lines in the dendrogram, more the distance between those clusters.

Now, we can set a threshold distance and draw a horizontal line (*Generally, we try to set the threshold in such a way that it cuts the tallest vertical line*). Let's set this threshold as 12 and draw a horizontal line:



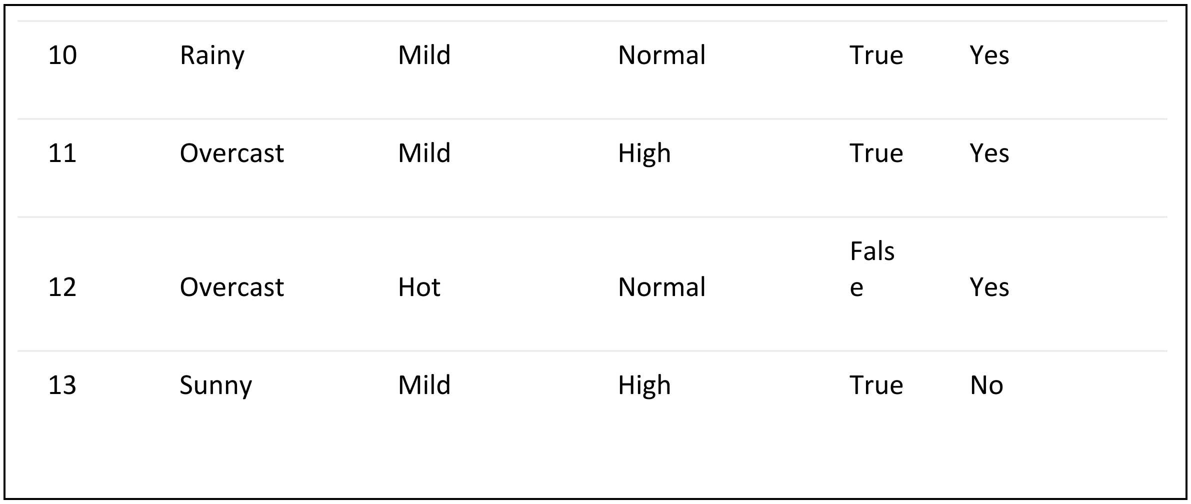
The number of clusters will be the number of vertical lines which are being intersected by the line drawn using the threshold. In the above example, since the red line intersects 2 vertical lines, we will have 2 clusters. One cluster will have a sample (1,2,4) and the other will have a sample (3,5). Pretty straightforward, right?

This is how we can decide the number of clusters using a dendrogram in Hierarchical Clustering. In the next section, we will implement hierarchical clustering which will help you to understand all the concepts that we have learned in this article.

9. **For the given observations compute Bayes classifier (and remember about smoothing when needed).**

Data:

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Rainy	Hot	High	False	No
1	Rainy	Hot	High	True	No
2	Overcast	Hot	High	False	Yes
3	Sunny	Mild	High	False	Yes
4	Sunny	Cool	Normal	False	Yes
5	Sunny	Cool	Normal	True	No
6	Overcast	Cool	Normal	True	Yes
7	Rainy	Mild	High	False	No
8	Rainy	Cool	Normal	False	Yes
9	Sunny	Mild	Normal	False	Yes



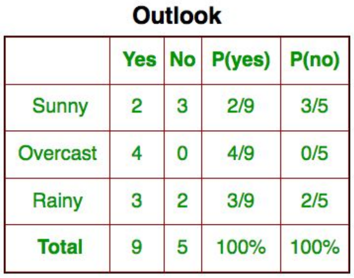
10	Rainy	Mild	Normal	True	Yes
11	Overcast	Mild	High	True	Yes
12	Overcast	Hot	Normal	False	Yes
13	Sunny	Mild	High	True	No

So, finally, we are left with the task of calculating $P(y)$ and $P(xi \mid y)$.

Please note that $P(y)$ is also called **class probability** and $P(xi \mid y)$ is called **conditional probability**.

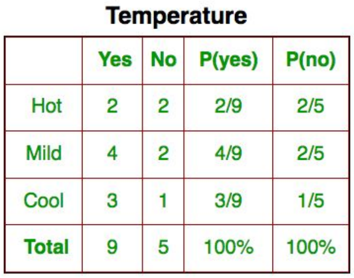
The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of $P(xi \mid y)$.

Let us try to apply the above formula manually on our weather dataset. For this, we need to do some precomputations on our dataset. We need to find $P(xi \mid yj)$ for each xi in X and yj in y. All these calculations have been demonstrated in the tables below:



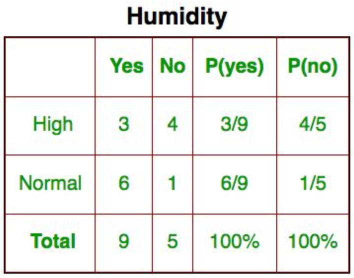
Outlook

	Yes	No	P(yes)	P(no)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rainy	3	2	3/9	2/5
Total	9	5	100%	100%



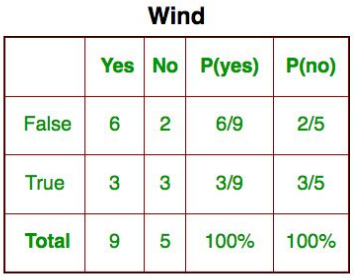
Temperature

	Yes	No	P(yes)	P(no)
Hot	2	2	2/9	2/5
Mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5
Total	9	5	100%	100%



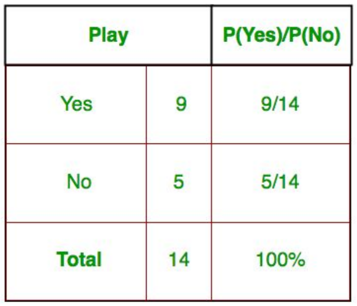
Humidity

	Yes	No	P(yes)	P(no)
High	3	4	3/9	4/5
Normal	6	1	6/9	1/5
Total	9	5	100%	100%



Wind

	Yes	No	P(yes)	P(no)
False	6	2	6/9	2/5
True	3	3	3/9	3/5
Total	9	5	100%	100%



Play		**P(Yes)/P(No)**
Yes	9	9/14
No	5	5/14
Total	14	100%

Smoothing +1 to nominator and + number of possible answers to denominator.

In that case Outlook P(NO)= 1/7

So, in the figure above, we have calculated *P(xi | yj)* for each *xi* in *X* and *yj* in y manually in the tables 1-4. For example, probability of playing golf given that the temperature is cool, i.e

P(temp. = cool | play golf = Yes) = 3/9.

Also, we need to find class probabilities *(P(y))* which has been calculated in the table 5. For example, *P(play golf = Yes) = 9/14*.

So now, we are done with our pre-computations and the classifier is ready!

Let us test it on a new set of features (let us call it today):

```
today = (Sunny, Hot, Normal, False)
```

So, probability of playing golf is given by:

$$P(Yes|today) = \frac{P(SunnyOutlook|Yes)P(HotTemperature|Yes)P(NormalHumidity|Yes)P(NoWind|Yes)P(Yes)}{P(today)}$$

and probability to not play golf is given by:

$$P(No|today) = \frac{P(SunnyOutlook|No)P(HotTemperature|No)P(NormalHumidity|No)P(NoWind|No)P(No)}{P(today)}$$

Since, *P(today)* is common in both probabilities, we can ignore *P(today)* and find proportional probabilities as:

$$P(Yes|today) \propto \frac{2}{9}\cdot\frac{2}{9}\cdot\frac{6}{9}\cdot\frac{6}{9}\cdot\frac{9}{14} \approx 0.0141$$

$$P(Yes|today) \propto \frac{2}{9}\cdot\frac{2}{9}\cdot\frac{6}{9}\cdot\frac{6}{9}\cdot\frac{9}{14} \approx 0.0141$$

and

$$P(No|today) \propto \frac{3}{5}\cdot\frac{2}{5}\cdot\frac{1}{5}\cdot\frac{2}{5}\cdot\frac{5}{14} \approx 0.0068$$

Now, since

$$P(Yes|today) + P(No|today) = 1$$

These numbers can be converted into a probability by making the sum equal to 1 (normalization):

$$P(Yes|today) = \frac{0.0141}{0.0141+0.0068} = 0.67$$

and

$$P(No|today) = \frac{0.0068}{0.0141+0.0068} = 0.33$$

Since

$$P(Yes|today) > P(No|today)$$

So, the prediction that golf would be played is ‘Yes’.

The method that we discussed above is applicable for discrete data. In the case of continuous data, we need to make some assumptions regarding the distribution of values of each feature. The different naive Bayes classifiers differ mainly by the assumptions they make regarding the distribution of *P(xi | y)*.