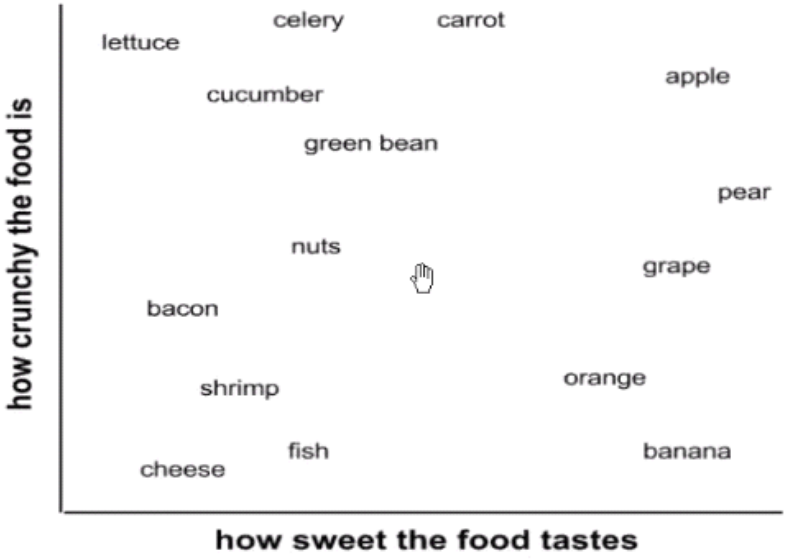
# K-NEAREST NEIGHBOURHOOD CLASSIFIER (KNN)

It is also a Non-Parametric method. It means not assuming of any distribution of that particular classifier. Most of the machine learning technique are Non-Parametric only.

We are not writing any equation either in most of the machine learning algorithms. KNN is one of them. It's a supervised Learning.

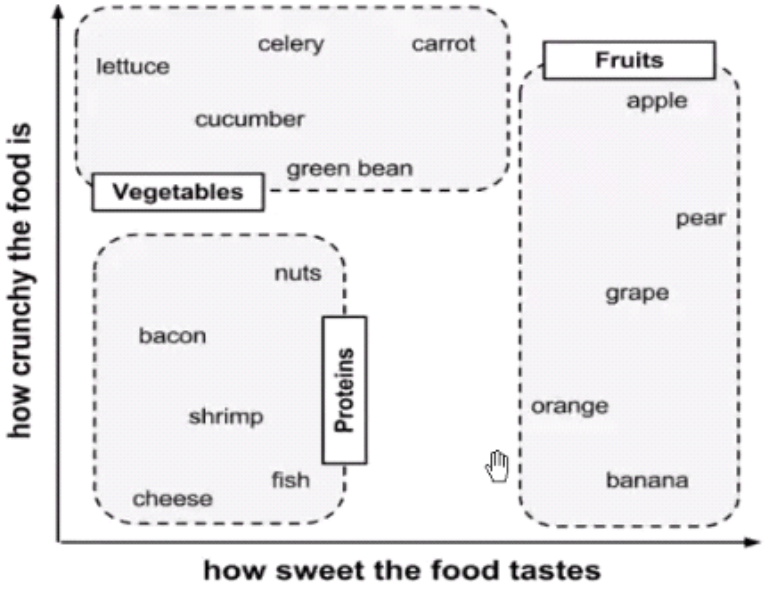
| ingredient | sweetness | crunchiness | food type |
|---|---|---|---|
| apple | 10 | 9 | fruit |
| bacon | 1 | 4 | protein |
| banana | 10 | 1 | fruit |
| carrot | 7 | 10 | vegetable |
| celery | 3 | 10 | vegetable |
| cheese | 1 | 1 | protein |

We have 4 different variables in the given data

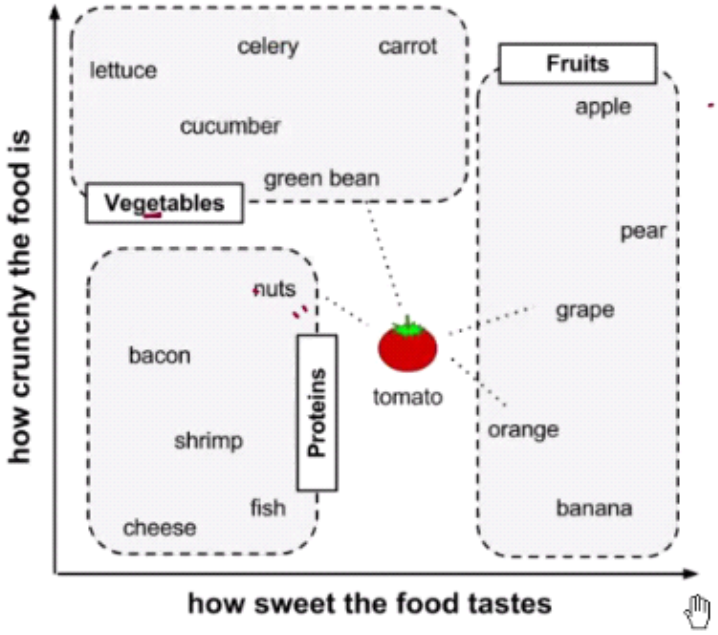



Did you notice any Patterns or Class from the above distribution of points?

I have made separate groups as below based on the nearest possible points




So, our job is to find out the new data point belongs which group from the above classified groups?




The New data point is belongs to which group?

So, I need to check the nearest distance from the given data point to all other groups. After calculation of distance whichever group is nearest to that particular point I will move this point in to that particular group.
So I will try with K=1, K=2,…..

**How to calculate Distance:**

Locating the tomato's nearest neighbour requires a distance function which measures the similarities between the two instances. There are many types of distance functions are available in mathematics but we are considering **Euclidian Distance method**.

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

So, the new observation Tomato contains sweetness = 6 and crunchiness = 4.

| ingredient | sweetness | crunchiness | food type | distance to the tomato |
|---|---|---|---|---|
| grape | 8 | 5 | fruit | sqrt((6 - 8)^2 + (4 - 5)^2) = 2.2 |
| green bean | 3 | 7 | vegetable | sqrt((6 - 3)^2 + (4 - 7)^2) = 4.2 |
| nuts | 3 | 6 | protein | sqrt((6 - 3)^2 + (4 - 6)^2) = 3.6 |
| orange | 7 | 3 | fruit | sqrt((6 - 7)^2 + (4 - 3)^2) = 1.4 |

If k =1, that means 1-NN classifier so among all the above distances which is having minimum distance i.e. 1.4 so it is belongs to orange that means fruit.

If k =2, that means 2-NN classifier so among all the above distances which is having minimum distance i.e. 2.2 so it is belongs to grape that means fruit.

If k =3, that means 3-NN classifier so among all the above distances which is having minimum distance i.e. 3.6 so it is belongs to nuts that means protein.

If k =4, that means 4-NN classifier so among all the above distances which is having minimum distance i.e. 4.2 so it is belongs to green bean that means vegetable.

So, in the above fruit are getting two votes among all four. Whichever is getting more number of votes the new observation will be moved to that particular group.

**So, How you will choose the K values here**

If we choose k= Higher value it may be give wrong prediction as above. It is like increasing in Bias and variance is coming down which is not correct. Similiarly If k=1, your decision is only one data point variance will be very high.

So, choose always k neither too high nor too low such it should not impact on variance and bias as well.

Using trail and Error method we can known readings and gives us the best results.

**NOTE:**

- KNN is useful when the relationship between variables and response variables are complicated , numerous , difficult to understand.

- It is approachable when our regression methods is not working or any other classifier is not working.

- If K is large variance will go down and Bias will go up and If K is small variance will go up and Bias go down.

- In practice, we can choose the K values based on number of records or complexity of the problem.

- Few recommendations:
  K value will be suggestible between 3 to 10.
  Root(n) = K , n is number of training records.

- So, whenever if you try to find the distance better you need to standardize or Normalize the data first.

If the variables are normal

$$\text{Standardize: } Z = \frac{X - Mu}{Sigma}$$

If the variables are not normal

$$\text{Normalization: } N = \frac{X - Min(x)}{Max(x) - Min(x)}$$

Strengths:
- It is Simple and Effective.
- No Assumptions on distribution of data
- Fast training phase
- When classification is non-linear boundaries it is effective.

Weakness:
- Does not produce any model, hence difficult to understand any relationship.

- Suppose if we have categorical variables , how you will find the distance between those categorical variables?
  There is a method called **Hamming Distance**

Ex: Find the distance between the words "Ramesh" and "Ram"
Either from any end you need to make 3 letters change either Ramesh to Ram or Ram to Ramesh.