Practical Machine Learning with R

Lecture 3

Objectives:

- Introduction to k Nearest Neighbor a lazy learner.
- The method to measure similarity in order to classify the observations.
- Showcasing the usage of KNN for complicated relationship amongst features.

Characteristics of kNN Algorithm :

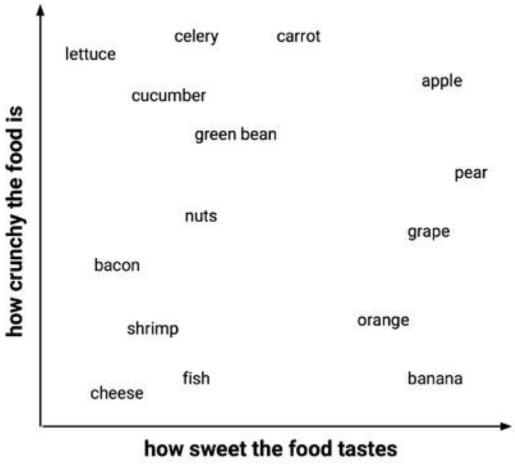
- This is an algorithm which classifies well if the inter relationships between features are complex but nevertheless can be differentiated by the Knn Algorithm by the similarity between observations.
- The Algorithm does not work well if the data has too much noise which causes errorneous classification of the data since there is no distinct boundary separation between groups.
- This algorithm takes unlabeled observations/examples and assigns them class labels as per the similar labeled examples
- This algorithm is prevalently used for detecting genetic diseases, facial recognition, music recommendation etc.
- The algorithm requires choosing k the number of nearest neighbors that will be assigned to ascertain the class label.
- The next step requires a training data set whose examples are grouped in categories that have been differentiated by a nominal measure.
- The final step of the algorithm involves evaluating each unlabeled example/observation of the test data and checking k nearest labeled training examples on the basis of similarity to make a determination of the class label for that specific example.
- kNN is simple and has a fast training phase and does not make any assumption related to underlying data distribution(non parametric) but the caveat is that kNN does not generate a model. Since the model is not generated the classification phase is slow.
- If the features/attributes are nominal then additional processing is used.

Example: For illustration purposes if we wish to use the following data to create the Food type classification

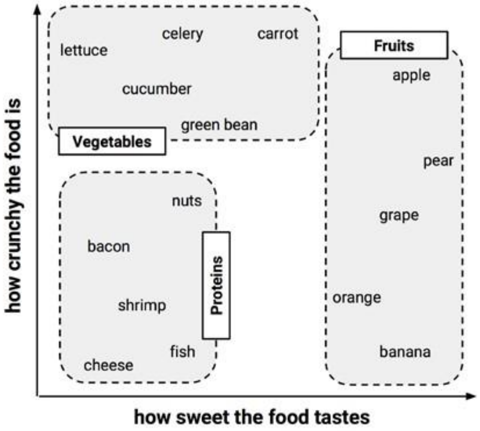
we would follow the provided steps:

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

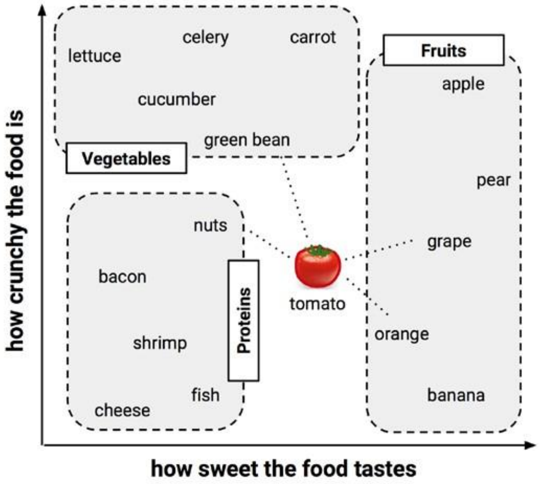
- Please notice in this data we have two features that will determine whether the data is a fruit, protein or vegetable.
- Usually the data will have multiple features and therefore the data can be represented into multidimensional space.
- The current data set can be represented as follows:



- The naturally occurring pattern is clearly visible in context of the two features in conjunction with each other.



- We now decide whether tomato is a vegetable, protein or fruit.



- Similarity metric is primarily measured by Euclidean distance ie the straight line path distance.
- Mathematically:

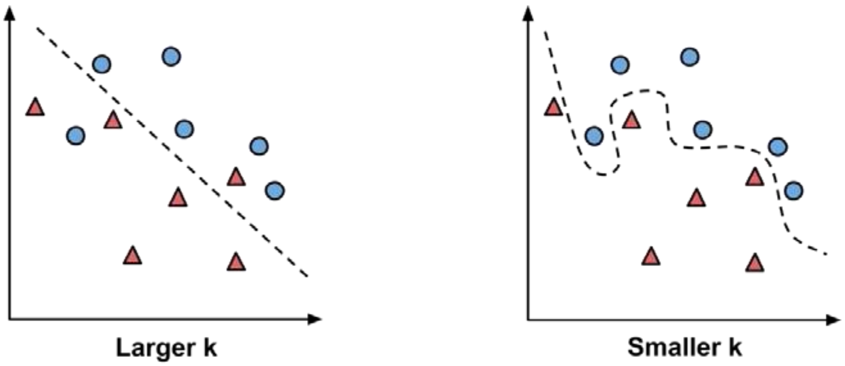
$$\text{dist}(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2}$$

Tomatoes: Sweetness: 3(p_1) Crunchiness: 7(q_1)

$$\text{dist}(\text{tomato, green bean}) = \sqrt{(6-3)^2 + (4-7)^2} = 4.2$$

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 =3 then tomato will be categorized as per the majority class which is a fruit.
- The decision regarding the value of k is important because a very large k might underfit the data ie the variance in the data is minimized which in certain case might cause not taking cognizance of a minute though significant pattern. This will cause bias in the learner since the classification is determined not by very close nearest neighbors but by the majority of the k examples.
- In contrast if excessive details are considered a very small value of k might overfit the data. Too much cognizance is given to the variance or noise will cause an unlabeled test example to be misclassified as per the nearest neighbors/neighbor which itself might be misclassified.



- The value of k has to balance the bias variance tradeoff. A common heuristic for the value of k is square root of the number of training examples.
- Another approach that can be taken is testing different k values with test data and determine which one gives the best result.
- If the data is very large though not very noise then the value of k is not very important because the subtle nuances will be extracted from the data and effective classification will occur.
- Data set usually has numerous features and their numeric values have disparate ranges. If the range of a feature/attribute is wide then this feature will dominate the nearest neighbor determination in context of the distance measurement.
- To solve this problem the features are typically rescaled by min max normalization algorithm or by the zscore standardization.
- Min Max Normalization implements the following algorithm:

$X_{new} = \frac{X-min(X)}{max(X)-min(X)}$ This captures how far the original value is on a scale of 0 to 100%

- Z-score standardization can be implemented by the formula
  $X_{new} = \frac{X-\mu}{\sigma} = \frac{X-mean}{St-Dev}$

  Z score values determine the number of standard deviation the value is above or below the mean value. This does not take the lowest and highest value into consideration.
- The kNN algorithm is a lazy learner since the abstraction phase and the generalization phase is skipped. The training data is stored as it is and thought the training gets executed quickly the testing process is slow since the comparison of the distance measure has to be conducted for every test data with the training examples/instances. For this reason it is called instance based learning.
- kNN does not build a model therefore it is defined as a non parametric algorithm as it is does not create any generalization pertaining the data.
- kNN does capture the inherent pattern but some times it is not apparent how the classifier is using the data.
- kNN has its weakness but despite these all things considered it is still a powerful classification algorithm.

Advantages:

Non Parametric: No assumptions regarding the underlying distribution.

Interpretable.

Can be used for Classification and Regression(Prediction).

The algorithm tends to overfit therefore optimizing the value of k is important.

Is a discriminative algorithm that creates a very non linear seperability.

Good performance requires to optimize the two hyperparameters k and distance function like Caberra, Manhattan, Minkowski etc.

Disadvantages

High memory requirement because model is not created and all the training data is used for classification.

Sensitive to noisy data.

Sensitive to outliers.

As the complexity increases(number of features) the performance comes down therefore dimension reduction techniques like PCA can be used to achieve better accuracy.