Andhra Pradesh State Skill Development Corporation

Skill AP
APSSDC

eSkill AP
Learn Anytime Anywhere
Andhra Pradesh State Skill Development Corporation

Machine Learning

K-Nearest Neighbors

CHAPTER 12

K-Nearest Neighbors

Introduction and Understanding K-Nearest Neighbors (KNN)

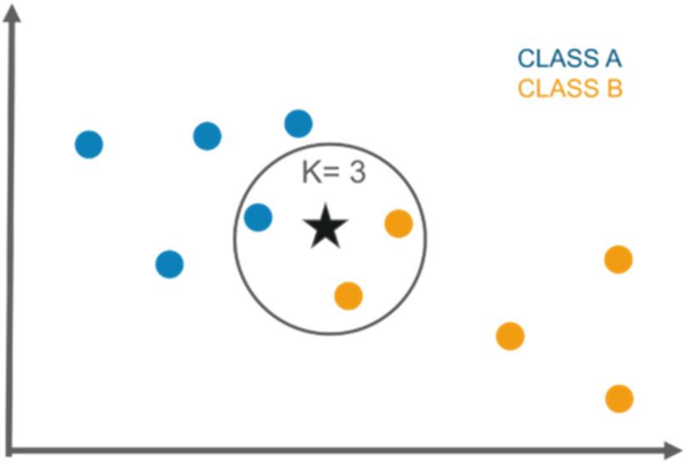
K-Nearest Neighbors (kNN) is a supervised machine learning algorithm that can be used to solve both classification and regression tasks. We can see kNN as an algorithm that comes from real life. People tend to be effected by the people around them. Our behaviour is guided by the friends we grew up with. Our parents also shape our personality in some ways. If you grow up with people who love sports, it is highly likely that you will end up loving sports. There are of course exceptions. kNN works similarly.

In Pattern Recognition, the K-Nearest Neighbors algorithm (k-NN) is a non-parametric method used for classification.In both cases, the input consists of the k closest training examples in the feature space. In k-NN classification, the output is a class membership.
An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small).

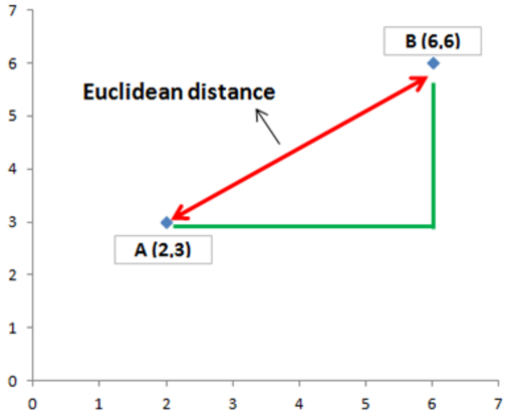
If k = 1, then the object is simply assigned to the class of that single nearest neighbor.As a part of kNN algorithm,the unknown and unlabelled data which comes for a prediction problem is judged on the basis of the training data set elements which are similar to the unknown element.So,the class label of the unknown element is assigned on the basis of the class labels of the similar training data set of elements(metaphorically can be considered as neighbors of the unknown element).

k-NN is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The neighbors are taken from a set of objects for which the class (for k-NN classification).This can be thought of as the training set for the algorithm, though no explicit training step is required.A peculiarity of the k-NN algorithm is that it is sensitive to the local structure of the data.K nearest neighbors or KNN Algorithm is a simple algorithm which uses the entire dataset in its training phase. Whenever a prediction is required for an unseen data instance, it searches through the entire training dataset for k-most similar instances and the data with the most similar instance is finally returned as the prediction

- The value of a data point is determined by the data points around it. If you have one very close friend and spend most of your time with him/her, you will end up sharing similar interests and enjoying the same things. That is kNN with k=1.
- If you always hang out with a group of 5, each one in the group has an effect on your behavior and you will end up being the average of 5. That is kNN with k=5.kNN classifier determines the class of a data point by majority voting principle. If k is set to 5, the classes of 5 closest points are checked. Prediction is done according to the majority class.



The distance between data points is measured. There are many methods to measure the distance. Euclidean distance (Minkowski distance with p=2) is one of most commonly used distance measurements. The figure below shows how to calculate euclidean distance between two points in a 2-dimensional space. It is calculated using the square of the difference between x and y coordinates of the points.



$$\boldsymbol{Euclidean\ distance}\ (\boldsymbol{a}, \boldsymbol{b}) = \sqrt{(\boldsymbol{a_1} - \boldsymbol{b_1})^2 + (\boldsymbol{a_2} - \boldsymbol{b_2})^2}$$

In the case above, Euclidean Distance is the square root of (16 + 9) which is 5. Euclidean distance in two dimensions reminds us of the famous Pythagorean theorem.It seems very simple for two points in 2-dimensional space. Each dimension represents a feature in the dataset. We typically have many samples with many features.

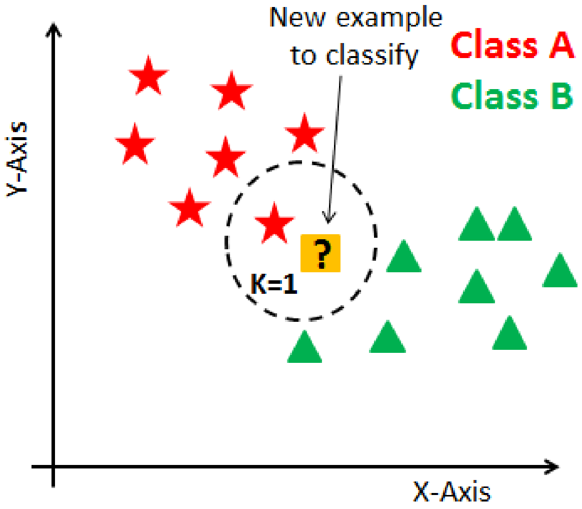
Simple Example for KNN Implementation

KNN is a non-parametric and lazy learning algorithm. Non-parametric means there is no assumption for underlying data distribution. In other words, the model structure determined from the dataset. This will be very helpful in practice where most of the real world datasets do not follow mathematical theoretical assumptions. Lazy algorithm means it does not need any training data points for model generation. All training data used in the testing phase. This makes training faster and the testing phase slower and costlier. Costly testing phase means time and memory.

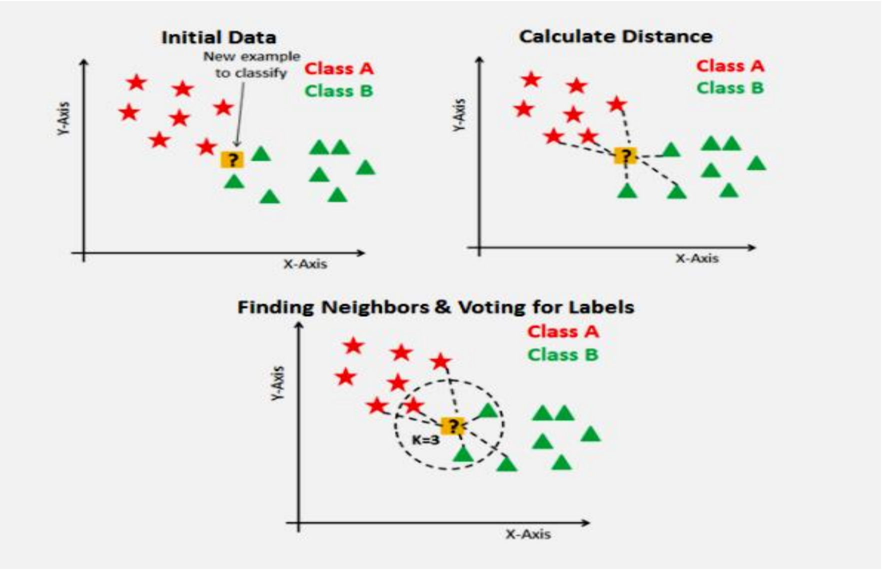
In the worst case, KNN needs more time to scan all data points and scanning all data points will require more memory for storing training data.

How does the KNN algorithm work?

In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm. This is the simplest case. Suppose P1 is the point, for which the label needs to predict. First, you find the one closest point to P1 and then the label of the nearest point assigned to P1.



Suppose P1 is the point, for which the label needs to predict. First, you find the k closest point to P1 and then classify points by majority vote of its k neighbors. Each object votes for their class and the class with the most votes is taken as the prediction. For finding closest similar points, you find the distance between points using distance measures such as Euclidean distance, Hamming distance, Manhattan distance and Minkowski distance. KNN has the following basic steps:**Calculate distance,Find closest neighborsVote for labels.**



How do you decide the number of neighbors in KNN?

At this point, the question arises that How to choose the optimal number of neighbors? And what are its effects on the classifier? The number of neighbors(K) in KNN is a hyperparameter that you need to choose at the time of model building. You can think of K as a controlling variable for the prediction model.

Research has shown that no optimal number of neighbors suits all kind of data sets. Each dataset has its own requirements. In the case of a small number of neighbors, the noise will have a higher influence on the result, and a large number of neighbors make it computationally expensive. Research has also shown that a small number of neighbors are the most flexible fit which will have low bias but high variance and a large number of neighbors will have a smoother decision boundary which means lower variance but higher bias.

Generally, Data scientists choose as an odd number if the number of classes is even. You can also check by generating the model on different values of k and check their performance.

So we will create our own dataset. Here you need two kinds of attributes or columns in your data: Feature and label. The reason for two types of column is "supervised nature of KNN algorithm".As we have **two features (weather and temperature) and one label(play).**

Encoding data columns

Various machine learning algorithms require numerical input data, so you need to represent categorical columns in a numerical column.In order to encode this data, you could map each value to a number. e.g. Overcast:0, Rainy:1, and Sunny:2.This process is known as Label Encoding, and sklearn conveniently will do this for you using Label Encoder.Then we combine the features and generate the model using the KNeighborsClassifier module and create KNN classifier object by passing argument number of neighbors in KNeighborsClassifier() function.

KNN Implementation

KNN stands for K Nearest Neighbour is the easiest, versatile and popular supervised machine learning algorithm. This algorithm is used in various applications such as finance, healthcare, image, and video recognition.

KNN is used for both regression and classification problems and is a non-parametric algorithm which means it doesn’t make any assumption about the underlying data, it makes its selection based on the proximity to other data points regardless of what feature the numerical values represent.

Working of KNN

When we have several data points that belong to some specific class or category and a new data point gets introduced, the KNN algorithm decides which class this new datapoint would belong to on the basis of some factor.

The K, in KNN, is the number of nearest neighbors that surrounds the new data point and is the core deciding factor. We pick a value for K and will take K nearest neighbors of the new data point according to their Euclidean distance.

Suppose that the value of K = 5, we will choose 5 nearest neighbors to the new data point whose euclidean distance will be less. Among these neighbors(K), we will count the number of data points in each category and the new data point will be assigned to that category to which the majority of 5 nearest points belong.

So we will work on Wine dataset which is a popular dataset which is famous for multi-class classification problems. This data is the result of a chemical analysis of wines grown in the same region in Italy using three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.

The dataset comprises 13 features and a target variable(a type of cultivars).This data has three types of cultivar classes: **‘class_0’, ‘class_1’, and ‘class_2’.** Here, you can build a model to classify the type of cultivar. The dataset has been imported from the Sklearn library.So we compare with different neighbors values and understand how “K” value plays a role in Performance of the model.