K-Nearest Neighbors Algorithm Solved Example

K-Nearest Neighbors Algorithm Solved Example in Machine Learning

K-Nearest Neighbors Algorithm is an instance-based supervised machine learning algorithm. It is also known as the Lazy Learner algorithm as it delays the learning process till the arrival of a new example.

In this tutorial, we will understand how to apply k nearest neighbors algorithm to classify the new example.

Problem Deninition:

“Restaurant A” sells burgers with optional flavors: Pepper, Ginger, and Chilly.

See also Concept Learning in Machine Learning
Every day this week you have tried a burger (A to E) and kept a record of which you liked.

Using Hamming distance, show how the 3NN classifier with majority voting would classify **{ pepper: false, ginger: true, chilly: true}**

Training Examples:

	Pepper	Ginger	Chilly	Liked
A	True	True	True	False
B	True	False	False	True
C	False	True	True	False
D	False	True	False	True
E	True	False	False	True

Solution:

The training examples contain three attributes, Pepper, Ginger, and Chilly. Each of these attributes takes either True or False as the attribute values. Liked is the target that takes either True or False as the value.

In the k-nearest neighbor’s algorithm, first, we calculate the distance between the new example and the training examples. using this distance we find k-nearest neighbors from the training examples.

To calculate the distance the attribute values must be real numbers. But in our case, the dataset set contains the categorical values. Hence we use hamming distance measure to find the distance between the new example and training examples.

See also Decision Tree using CART algorithm Solved Example 3
Let x1 and x2 be the attribute values of two instances.

Then, in the hamming distance, if the categorical values are the same or matching that is x1 is the same as x2 then the distance is 0, otherwise 1.

For example,

If the value of **x1 is blue** and **x2 is also blue** then the distance between x1 and x2 is **0**.

If the value of **x1 is blue** and **x2 is red** then the distance between x1 and x2 is **1**.

The following table shows the distance between the new example and the training example, calculated using hamming distance.

	Pepper	**Ginger**	**Chilly**	**Liked**	**Distance**
A	True	True	True	False	1 + 0 + 0 = 1
B	True	False	False	True	1 + 1 + 1 = 3
C	False	True	True	False	0 + 0 + 0 = 0
D	False	True	False	True	0 + 0 + 1 = 1
E	True	False	False	True	1 + 1 + 1 = 3

Next, Based on the distance we find 3 nearest neighbors (3NN), which are marked in the last column.

	Pepper	**Ginger**	**Chilly**	**Liked**	**Distance**	**3NN**
A	True	True	True	False	1 + 0 + 0 = 1	2
B	True	False	False	True	1 + 1 + 1 = 3	
C	False	True	True	False	0 + 0 + 0 = 0	1
D	False	True	False	True	0 + 0 + 1 = 1	2
E	True	False	False	True	1 + 1 + 1 = 3	

Finally, majority voting is used to assign the classification label to the new example. In this case, we have, **two False** and **one True** nearest examples. Hence the new example is classified as **FLASE**.