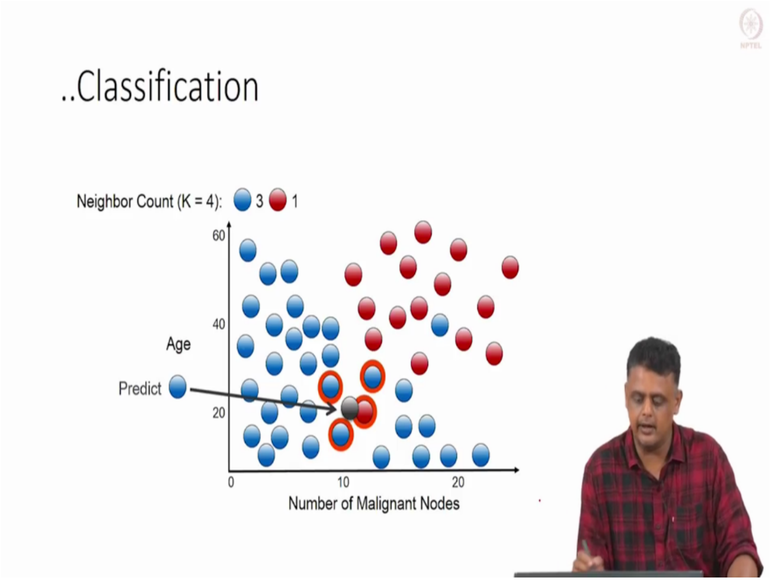
Machine Learning for Engineering and Science Applications
Doctor Ganapathy Krishnamurthy
Department of Engineering Design
Indian Institute of Technology Madras

K Nearest Neighbours (KNN)

Hello and welcome back, in this video we will look at K Nearest Neighbours, one of the more simple classification algorithms. All the figures or graphs illustration in this in the slides that we are going to see a provided by Intel software.

(Refer Slide Time: 0:30)



..Classification

So let's look at this dataset, it has two features, it's the cancer dataset. One feature is a number of Malignant nodes or these are the cancerous lumps or lesions that are seen in the patient, maybe in the images. The other feature is the age of the patient and what we are trying to look at is the survival of the patient, so did that patient survival or not, okay. So this is one piece of the dataset.

So all the red points correspond to patients who did not survive and all the blue points correspond to the [inaudible 1:03] correspond to the patients who survived, okay. So, what we want to do is to predict when a new patient comes in, new patient data comes in. So we have the age of the patient and the number of Malignant modules for that patient and let say it falls here in the dataset. And we want to predict whether the patient will survive or not, okay.

So, how does K-Nearest Neighbour go algorithm go about doing that, okay. So, it has one hyper parameter if you can call it, it's the neighbourhood count called K, that's why it's called K-Nearest Neighbour. So what we do is we look at the neighbourhood of this point that where we are where this test data is located and let’s say we can say only one neighbour, okay. One neighbour in the sense I consider the nearest neighbour, right, the nearest.

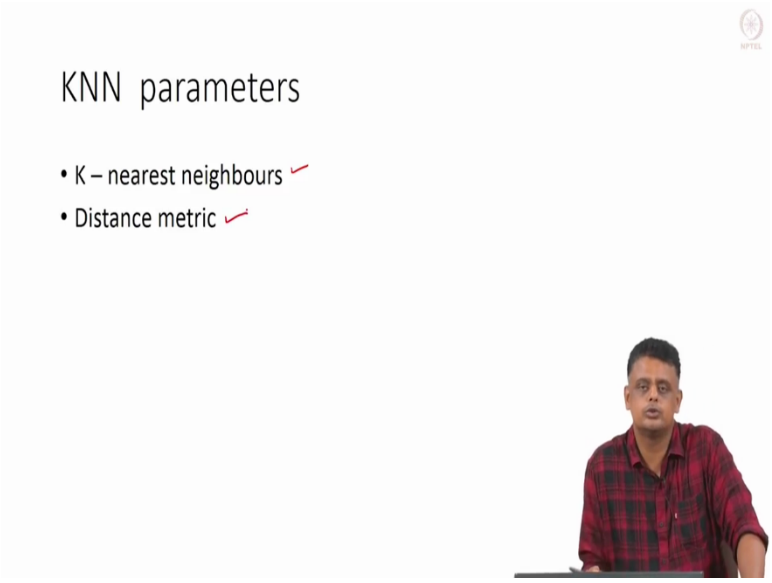
We will later on see what the, what do you mean by nearest neighbour but we assume that we have some way of figuring out the nearest neighbour and so in this case the nearest neighbour is a red circle which a red spear which a which represents patients who did not survive. So hence we classify this patient as not was not survived, okay. So this is when K is equal to 1.

So, we can also see when K equal to 2 we have two points which are closest to it in some sense and one of them is blue rather this red so then it’s a tie so the difficult to classify that way. Then we consider let’s say three nearest neighbours wherein three other nearest neighbours out of this 2 says that patient will survive and one says that patient will not. So then if we take the maximum vote we take the majority vote and then we can say that, okay, the new data patient that came in indicates at the patient will survive, okay.

So we can of course keep increasing the number of neighbours in this fashion and in this case again when we consider 4 nearest neighbours will see that three out of the four points correspond to patient who survive so hence we can safely say that the patient will survive or I mean at least predict that the patient will survive, okay. So this is the basic algorithm, so what we have are the two parameters, one is the K value which tells you how many points to consider, here when we say points to consider we are only looking at the training data points, right.

So we are given a dataset which we considered as a training dataset and we will only look at new data point comes in, test data point comes in, when we say distance we are looking at only distance to the points which are present in the training data. So, it’s not like continue I means only depends on how much training points you have, okay.

(Refer Slide Time: 3:48)

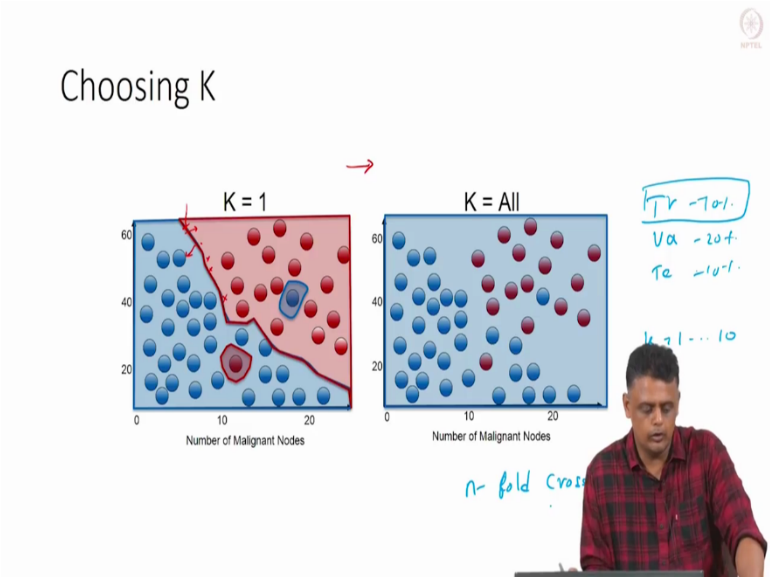


KNN parameters

- K – nearest neighbours
- Distance metric

So, in that sense there are only two parameters here one is the K, okay and the other one is the distance metric, okay. So we will see how to choose K and what kind of distant metrics are typically used, okay.

(Refer Slide Time: 4:03)



Choosing K

So we will consider the two extreme cases here, when we consider only one neighbour at a time and the other where we consider all the neighbours, when we say all it means the entire training dataset and not the continue as the graph implies, okay. So now we consider only one data point at a time, then what we do is we can see that you know if you look at this particular plot there is a bunch of red points which indicates the patient did corresponding to patients who did not survive and bunch of blue points here which correspond to patients that survive.

There is an odd blue and red data point here. So what do this curve implies? So this is the decision boundary, so this red curve or we can look at as the blue curve whichever way you look at it, it's a decision boundary, which means that anything to the one side of the boundary means that all test points fall on that on this side of the boundary, means the patient will not survive and all test points that form, fall on the side of the boundary means the patient will survive.

So, it depends on, so we draw this boundary by looking at the distance to the first nearest neighbour, okay. So, which all these says that if there is a data point somewhere here very close to the boundary, red boundary, that's the and if we put any all the data points very close to that boundary, all these data points would correspond to patients who will not survive and [inaudible 5:20] we move on to the other side it give corresponding data points where the patient will not survive, okay.

So, we can actually construct this particular decision boundary by considering by changing the number of nearest neighbours. So, what will happen if we consider all the data points in the sense instead of considering one neighbour we consider let's say in this case this has about I think close 30 odd points, okay. And we consider all 30 points as neighbours and then see.

If we do that, then by virtue of their being more patients in the dataset who survived than those who do not survive give the result that the patient will survive, okay. So, this is the parameter that we will have to learn to tune, okay. So, this is the extreme cases, K equal to 1 we are considering only one neighbour at a time, again remember when I say neighbour we are only talking about the training data point that are made available to you.

So, now the question is how do we choose this K, right. So then that solved by splitting your data into training, validation and testing so maybe in the ratio 70%, 20% and 10% of your data, okay. So based on the performance in the validation data for a particular choice of K you can then decide on that choose that K. So, for instance you can vary K given your split of your training dataset became vary the value of K maybe you can go from 1 to let it be 10 or 100 whichever depending on the size of your dataset and find out for the K value for which you get very good performance on your validation dataset and of course you can go at into the testing on it.

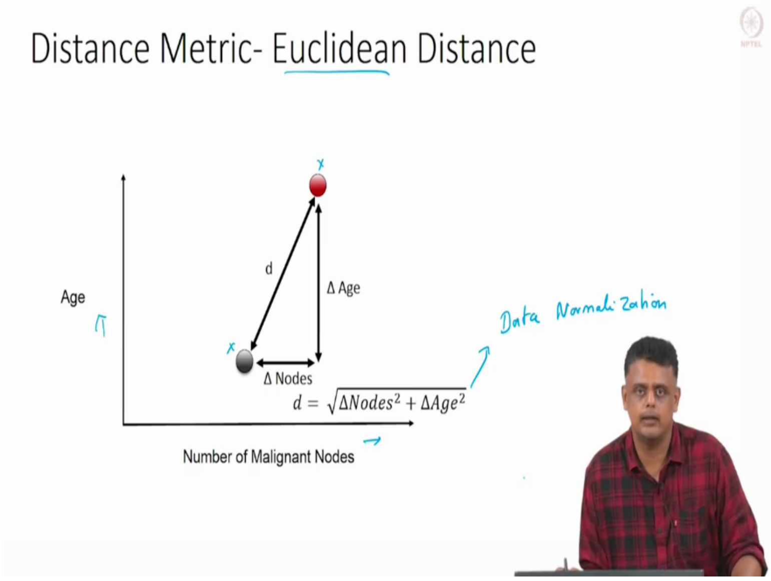
The other techniques would be to try out n-fold cross validation, right. Here you will choose different splits of your data into training, testing and validation or you can just to training and testing. And then of course vary K for each of these splits and see for the K value for which you get a low variance and reasonably good accuracy. So, this is your standard way of making sure that you don't over fit, okay. So because it is very easy to over fit with K nearest neighbours like we saw here we just do K equal to 1 you can get a very sharp boundary but of course that means that you will be over fitting.

So, by splitting your data into training, validation and testing new can figure out the value of K using your validation dataset or by doing n-fold cross validation and for every fold you can try out different values of K and find out the value of K for which you get low variance and reasonably high accuracy.

So this is of course very simple method in the sense that you don't actually do any training I'm in there is

You just load all the data into memory that's of course a problem and then the dataset become very large so you would consider all the data points the same time that is of course you after you splitted into training, testing and validation and then you just find out the nearest neighbour, okay.

(Refer Slide Time: 8:39)



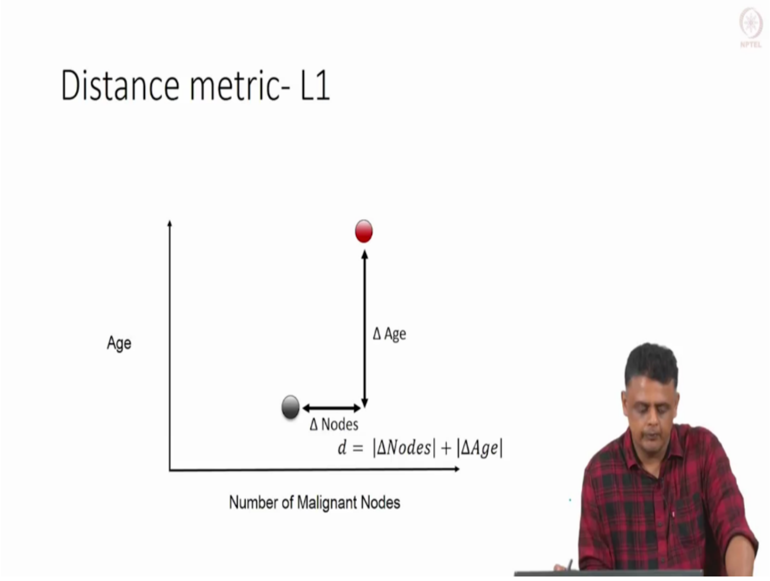
Distance Metric- Euclidean Distance

$$d = \sqrt{\Delta Nodes^2 + \Delta Age^2}$$

So, one more thing to clarify is what you mean by nearest neighbour so typically we will use the Euclidean distance. So, in this case let's say we have these two features number of Malignant Nodes and age. So, if we want to figure out a distance between your test data point and one of the training data points, we just have to calculate the difference between the feature values, square them and add them take the square root of course, now of course remember that when we do this we have to make sure that we do data normalization. This you should understand and [inaudible 9:13] because if you look at the number of Malignant Nodes it is going to be different in the sense the range of these axis is going to be different from the range of the age axis, okay.

So then it makes sense to normalize them to make them more meaningful. So, you do the you can do the equivalent distance provided that you have than the data normalization and in which case the equivalent distance would make sense to some extent.

(Refer Slide Time: 9:37)

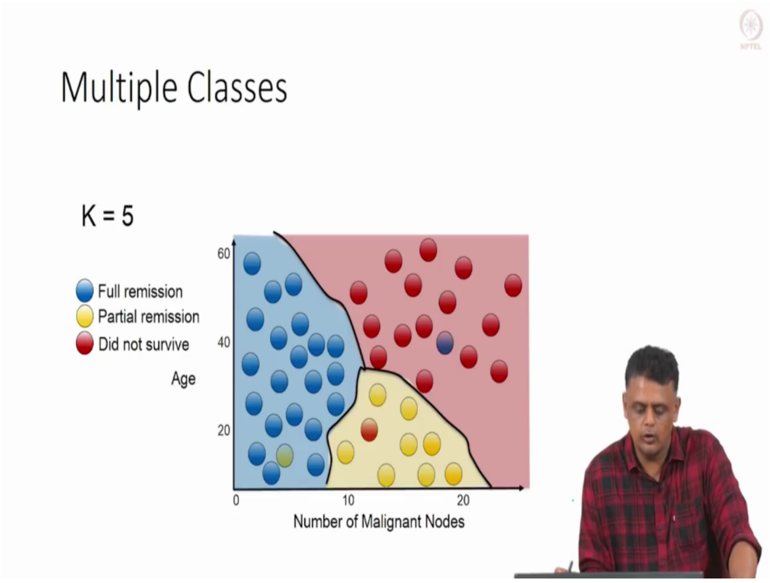


Distance metric- L1

$d = |\Delta Nodes| + |\Delta Age|$

Of course, we can also, so that is the L2 distance, this is the L1 distance we can consider wherein which is just the sum of the absolute values of the difference between the feature values. Once again here in this case again we expect that data normalization has to be done before we can compute these metrics, okay.

(Refer Slide Time: 9:56)



Multiple Classes

K = 5

We can also do multi-classification using K-Nearest Neighbours. Of course it’s the same procedure and there is again the possibility of tie that can happen, okay. So, then you have to vary K to the point where this ties don’t happen that often. So to summarise, we have look that the K-Nearest Neighbours algorithms one of the simpler algorithms and in many cases it will work very well depending on your data, the basic principle is that you load the entire dataset along with these features of course after splitting it into training, validation and testing.

And for a new incoming data point you find out, of course this is a supervised technique so you actually know the ground root. So, for a new incoming data point you find out you decide the nearest neighbours to consider so you find out the K-Nearest Neighbours and do a majority voting among them to find out the class to which your test data belongs to, okay. So, we will look at other classification algorithms in machine learning in the other next few lectures. Thank you.