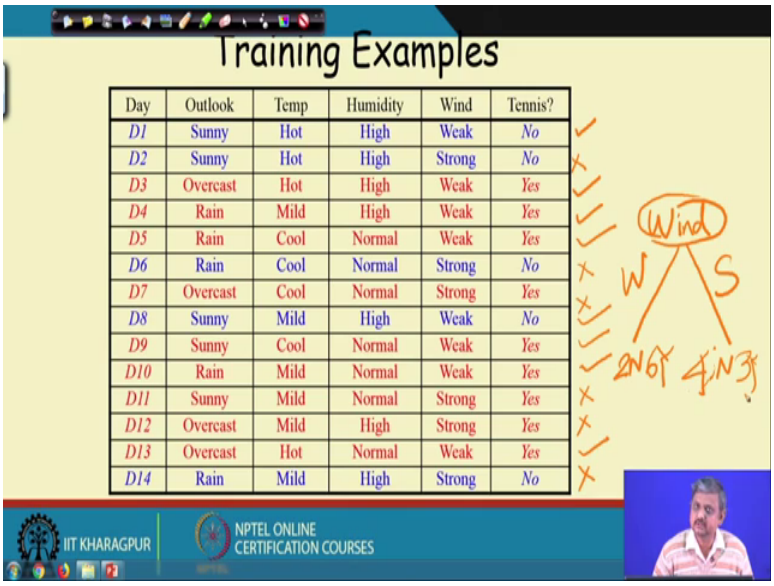
Data Mining
Prof. Pabitra Mitra
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

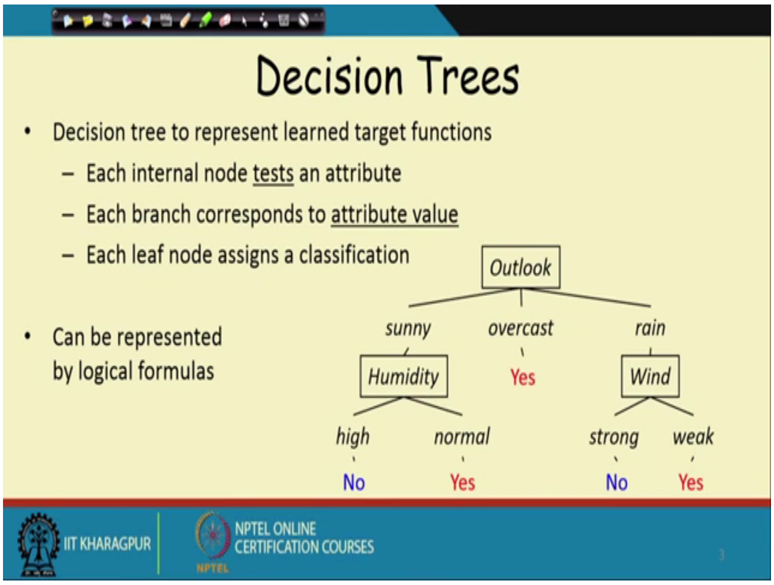
Lecture – 09
Decision Tree – II

(Refer Slide Time: 00:27)



Welcome to the second lecture on decision trees. So, as we mentioned before that we have a training set, which is there before you a table like this. And what I basically want is to draw an decision tree which best fits this table; in other words a tree which correctly predicts the final column yes or no as accurately as possible on these examples.

(Refer Slide Time: 00:58)



So, as we have told that a decision tree is nothing but some attributes and then branch as nodes and branching on them and finally, till you get to the leaf. So, the question that is to be answered is which attribute should I check first and then which attribute then which attribute and so on to decide. So, there are four attributes, which one should I check first. Let me come back to the table. If you have actually I told done what I told you to do that is think on what intuitively is a good thing, you have probably done something like this. So, let me check among these four attributes, which one is the best attribute for root.

So, what I actually do is that let me say temperature or start from this side wind suppose my root is my wind. So, let us see the root breaks down this table, the value of the wind, it breaks down into two groups the one for which wind is weak and one for which will be strong. So, let us mark them. So, I will write w 1 yes, so or maybe tick and cross. So, here wind is weak, here wind is weak, here wind is weak, here wind is weak, here wind is weak, weak here, weak here, weak here and weak here. So, among the rows which I have picked as weak how many had yes tennis and no as tennis. So, if I see so one no, three yes, two no, five yes, six yes; 6 - yes and 2 - no.

So, if I split my training examples into two groups, they extend the value of wind over wind they extend the value of wind weaker strong, if I split my examples into two group they actually each of the group actually from mixed. So, the wind equal to weak group has two no your tennis and six yes tennis mix both are there both classes are there.

Similarly, if I take wind as strong wind as strong the cross I am putting. So, how many yes I have one no, two no, three no and four no and two yes, I have 4 - no and 2 - yes.

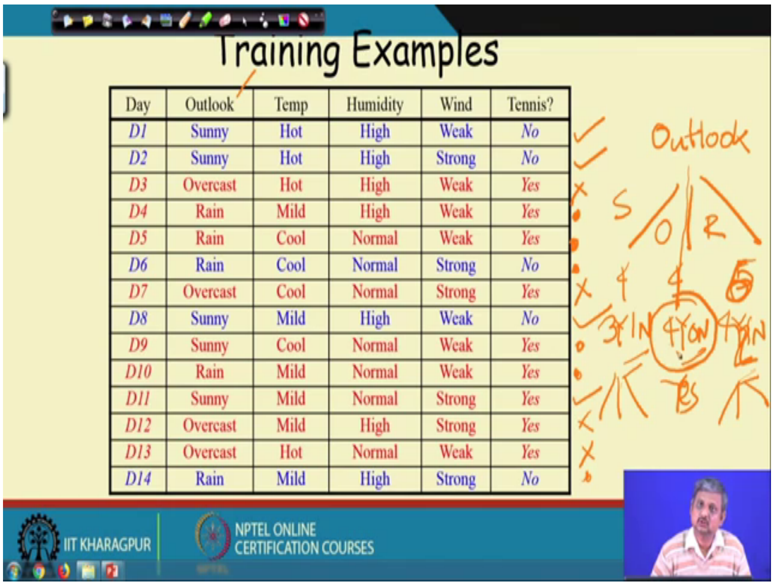
So, again if I use say something like this if I check on wind. and if I split into two group weak and strong. In the weak group I have 2-no and 6- yes; 2-blue and 6-red full rows and here I have 6 no, 6 no. Let me check 5 no? 5 no, I may be wrong I am an old man I may be wrong you do it properly. 5 no, 1, 2, 3, 4 – 4, 4 no actually four no actually and how many yes in the process it is wind equal to strong I have I have one yes, two yes, three yes, yes 3 yes, 3 yes, I have 4 no and 3 yes. So, the attribute wind is not. So, good at distinguishing between playtennis and playtennis no it is not so good among there are some people who have wind equal to weak and they are both no and no, and some are yes and so on. It is like the following example.

So, suppose I want to I describe a student by say three attributes, how tall is she, how tall is she, what is her mother tongue, how many hours she studies and how many hours she sleeps. So, maybe by this four attributes I describe a student. And I see the height of the student and similarly the mother tongue of the student. And I want to put the students into two categories one who does good in the class one who does not do good in the class. Well, everybody is good suppose if I go by mark somebody get good marks somebody does not get good marks. And I want to predict looking at these four values what type of student it is. So, that she will get a good mark or she gets a bad mark, suppose I am a recruitment company or something I want to do it.

And I find that this attribute height and maybe it has two values tall and short or maybe three values tall short and medium, this does not really help me in predicting whether she will get good marks or bad marks. People who are tall get good and bad marks people who are short also get good and bad marks. Similarly, if I look at mother tongue say Bengali or Telugu or Hindi or Tamil that also there are Bengali people who are good and bad, there are Tamil people who are good and bad. So, that also does not really help me distinguish.

But maybe if I look at hour of study maybe again for the sake of simplicity two categories long and short. And then I find that people who study long hours do good most of the time, most of the time, so that is a better attribute that is a better attribute, because it help me segregate the training set into two groups where each of the group have most of the people from one of the classes. So, for example, let us look into these attribute let me erase this let me erase this and let me. So, wind I have already checked and wind I have already checked. Let me check my temperature now. Let me directly go to that outlook. So, outlook has three values. So, it will actually segregate this fourteen days D 1 to D 14 into three groups.

(Refer Slide Time: 09:07)



Day	Outlook	Temp	Humidity	Wind	Tennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

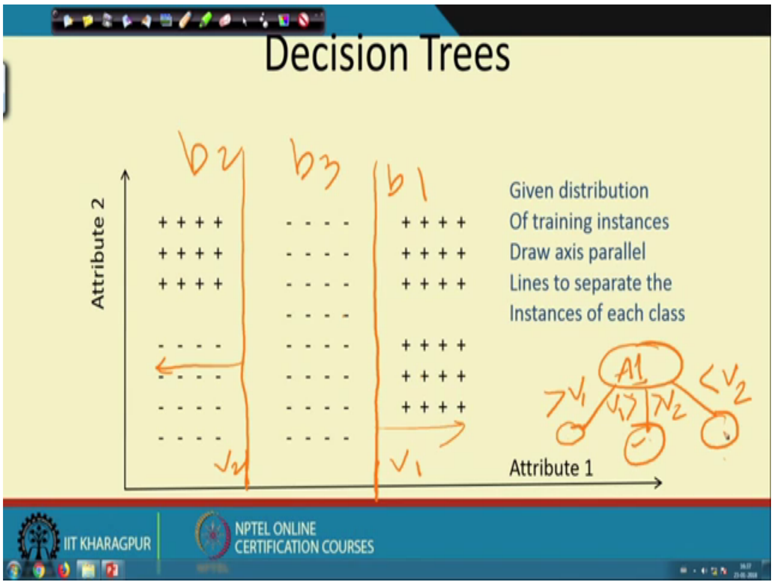
So, let me check the one with outlook sunny, I will put a tick on them. So, this one, this one well sunny, this one well sunny, this one; and overcast maybe I put a cross over cast is this, overcast is this, overcast is this, and overcast is this; and rain I put a dot this and this and this and this and this. So, I have four ticks meaning sunny; four crosses meaning overcast; and five dots meaning rain. If I write down these 14 examples, if I write down outlook, outlook as an attribute it will split the 14 examples into two groups. So, the sunny group has four examples, the overcast group has four examples and so this is sunny, this is overcast, the rain group has five examples is it or am I making a mistake I think I am making mistake five example anyway does not matter.

So, now, this 4, 4, 5, let me see what is that yes no distribution tennis. So, out of the four sunny this tick three are yes, one is no, out of this four rain the crosses the four rain the crosses all four are yes zero no and out of the five rain four are yes and one is no. So, you see this is a relatively better. So, if I just this is a relatively better discriminator between yes and no. I missed something I missed something this is a dot. So, this I actually have I actually have two no’s here, this is six probably, my handwriting is very poor, pardon it. So, I have you see a relatively better discriminator in the sense that only one example I am making a mistake, no example I am making a mistake, and only two examples having mistake making a mistake; whereas, in the case of wind I was it was more confusing it was more confusing.

So, maybe this outlook attribute is a better predictor, it is a better it is the first thing that you should check, when you want to build a decision tree, you should check it at the root. Once that is done maybe you can check maybe for furthermore among these four examples which have outlook equal to sunny you can further check humidity. And among this you can so here everything is yes. So, I can directly put yes as a leaf node here this I can further check maybe humidity these I can further check maybe temperature or something and so on, I can build a tree further, but as a first cut this is the best attribute.

In other words, if you see what is my property of when I am calling something as a leaf and not splitting it further. I am calling something as a leaf when all the examples which come to that node belong to a single class, among these three possibilities only this belongs to yes single, class yes. So, moment it becomes a single class or a pure class, it is a leaf. So, in other words, I want a leaf finally, I want a leaf where all the examples which will be pushed down to that leaf will belong to only one class. And I do this progressively in the sense that I do it in a greedy way; that means, first I try to make a split using the root which makes each of the child’s of the root as pure as possible. And each of that child if they are not pure if they are pure at is a leaf if they are not pure I further split, so that they become even more pure till they are absolutely pure, pure means here that they belong to a single class.

(Refer Slide Time: 15:02)

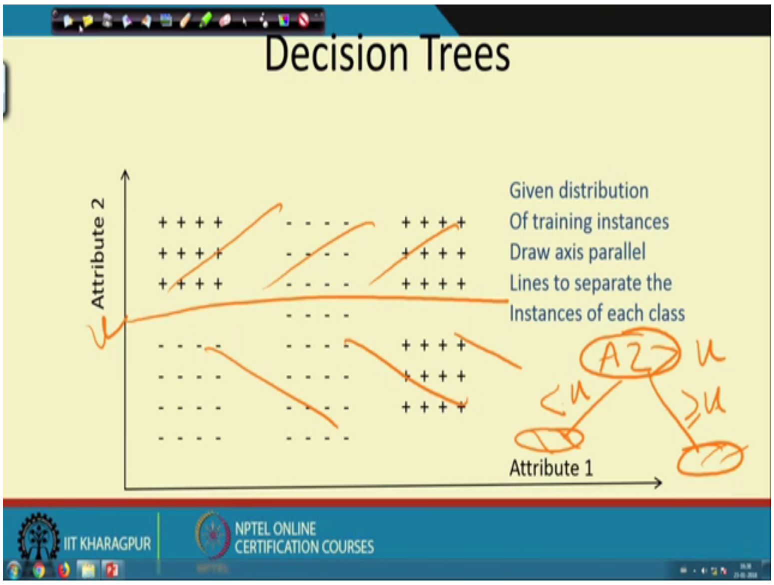


So, let me try to explain you this way, explain it to you this way. I said that the this each instance I can represent as a four-dimensional vector for example, D 1 I can represent as a four-dimensional vector having attribute value sunny, hot, high and weak - four tuple. And for ease of visualization let me say and not four I have only two attributes attribute one and attribute two and. So, any instance is nothing but a tuple of two values one value and attribute one value and attribute two value. So, I can always also visualize them as points in two-dimensional space coordinates at these values at the coordinate values. So, I can actually plot them as my points. So, I can plot them as this points, this points all the training set at these points in two-dimensional space. And in this example what I have done is that I have marked all the s class points as plus and I have marked all the no class point as minus.

So, what is my decision tree node split doing actually let us see. Suppose I have a decision tree node which says that attribute one has I check at root node attribute 1, and I split it into say three branches; one branch having attribute one value greater than something, another less than something, another between these two values. So, suppose I call this as V 1 this has V 2. And if attribute one greater than V 1 I check on attribute one if the value is greater than V 1 1 branch less than V 2; and another branch between V 1 and V 2 is the third branch. And let me see among these points that I have drawn which satisfies this, which satisfies this, and which satisfies these. So, all the points here will be pushed down to this branch, all the points here will be pushed down to the branch and all the points here will be pushed down to this branch.

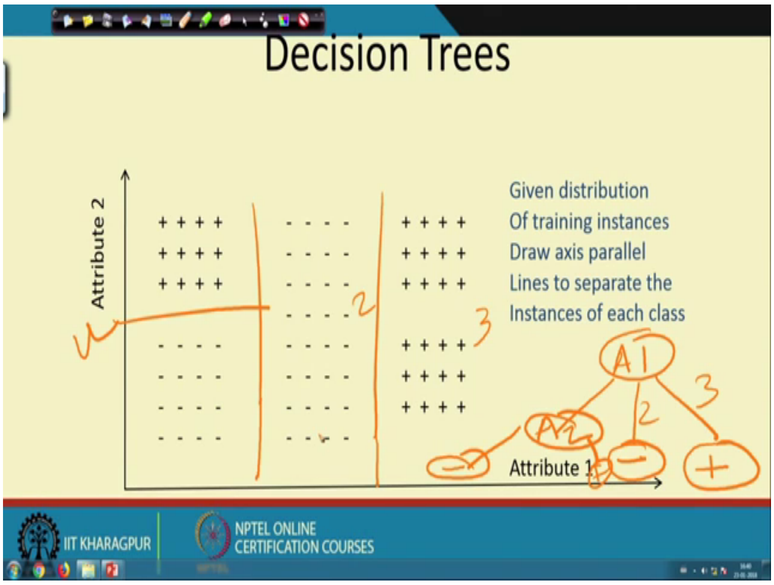
In other words, I can think of like this, each of the branch is drawing a vertical line like this. So, this is branch 1, this is branch 2, and this is branch 3. So, and all the points lying right of this branch 1 is in this branch in this line right of this line is in this, left of this line is in this, and in between is this. So, this splitting is equivalent to drawing these vertical lines. So, this vertical lines will splitting separates the training examples into three categories, these vertical lines also separate this training line training points into three categories. So, they are equivalent.

(Refer Slide Time: 19:47)



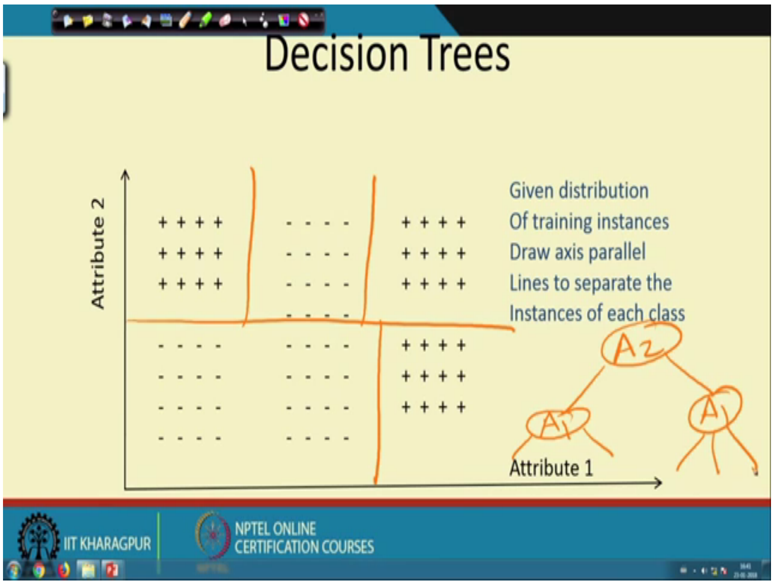
Note that I could have also done the following. Instead of attribute 1, I could have asked a question on attribute 2 also, maybe I could have made a two way split say value u 1, u 2 or only u 1 sorry. So, I have a value, I ask a question whether A 2 is greater than this value u, and all the points less than u lie in this branch, greater than equal to u lie in this branch. So, you see all these points will come here and all these points will come here. So, I could have if I split using A 2, I will get some split like this.

(Refer Slide Time: 20:58)



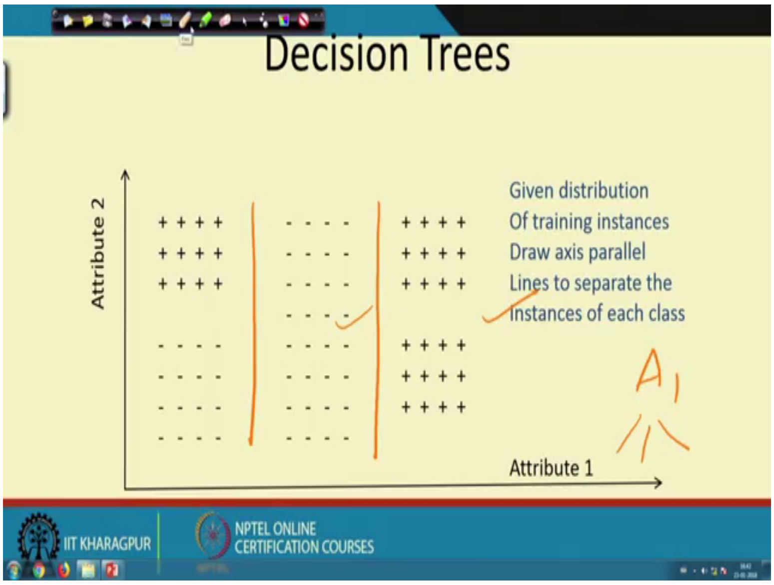
If I split using A 1, I will get a split like this. Now, you tell me if my goal so one thing you think also is that I could have not stopped here. So, using A 1, I first make a split three groups and I notice that this middle group is already pure, I put it as negative leaf. The third group is already pure, I put it as plus. Whereas, this first group, so third group, second group, the first group is not pure. How to make it pure, split further on attribute 2. So, again check on A 2, again check on A 2, and suppose for the value u. And if the value is less than u, I call it minus; greater than, u I call it plus. Now, you see that this decision tree correctly classifies all the examples, correctly classifies all the example and corresponds to a axis parallel splits looking like this, axis parallel splits looking like this. So, if I this tree, so first A 1 then A 2 will do it like this. Suppose, I reverse it I make it first A 2 them A 1.

(Refer Slide Time: 22:55)



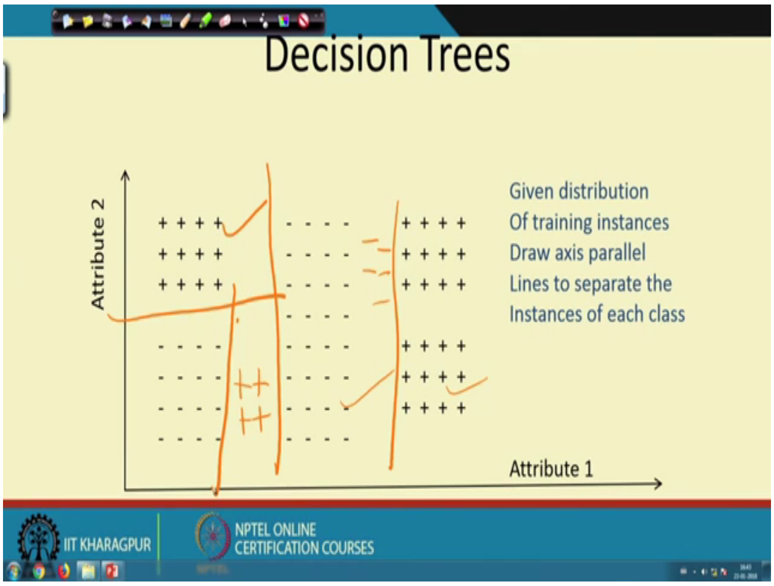
So, first on A 2 then on A 1. So, I will do maybe I will do this and then again on A 1 I will split this again on A 1 I will split this. So, A 2 then on A 1 A 1 split A 1, this into two, this again into three, I have to do this. So, I need more number of splits. So, maybe the other tree first it A 1 then A 2 is bigger. Maybe if I have asked you to draw axis parallel lines to sort of split this up into two classes, you have to have done the first tree only, you would have probably done the first tree only.

(Refer Slide Time: 24:18)



So, why I have drawn the first tree because splitting on A 1 on three values give me better purities in the leafs they gave me better purities, this is pure, this is pure this is impure then the other one. Also note that both these trees are equally correct, maybe one is more complex than the bigger than the deeper than the other, but both are equally correct. So, for a single data set, you can actually have you can actually have multiple trees which are correct, there may be many number of trees, which are correct.

(Refer Slide Time: 25:22)



Also note one more thing that whatever be the data distribution whatever be the data distribution. Suppose I have some positive point sitting here, and some positive point sitting here, maybe not this let me draw it like this. If I have some positive point sitting here, I could have also classified, or maybe some negative point sitting here it could have also classified. How would I classified maybe I will get a longer tree, what I will do is that I will split like this, this is pure, this is pure, this is pure, this is pure. This I will split like this, so this is pure this is impure this ill further split. So, one extra split. So, now I get. So, what is the equivalent tree if you draw you will get a deeper tree. So, this is the intuition.

So, what I will do in my next lecture is to quantify this notion of purity of a class by a formula by a function by a measure which we will use to construct the algorithm. So, we will call it as a decision tree construction algorithm, but the intuition is this try to make the best split first then the next split then the next greedy algorithm till you get a pure class in the leaf, do it by the smallest tree possible all it.

Thank you for today.