

Introduction to Machine Learning
Prof. Sudeshna Sarkar
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Module – 2
Lecture - 06
Introduction to Decision Trees

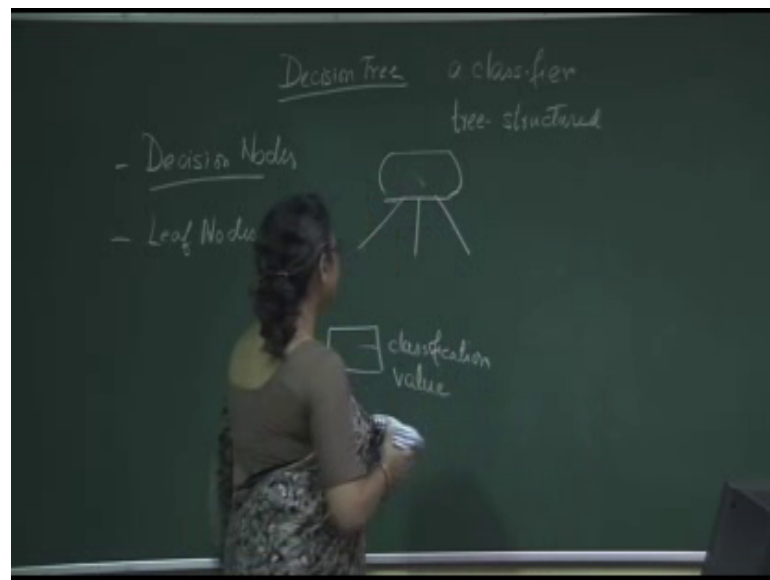
Good morning. So, today we will start the second part of the module introduction to linear regression and decision trees. In today's lecture we will give a brief introduction to decision trees. In the last class we saw, we talked about linear regression, which learns the linear function. The learning algorithm, that we will start today, the representation is a decision tree, which is a non-linear function. So, first let us define what is a decision tree?

(Refer Slide Time: 00:57)



I hope all of you know what is a computer science tree. A tree has nodes and branches. A rooted tree, you have a root node and you have children and then you have leafs, which do not have any children. Now, a decision tree is a tree, is also a classifier. A decision tree is a classifier in the form of a tree and the tree has two types of nodes, decision nodes, so is a classifier.

(Refer Slide Time: 01:57)

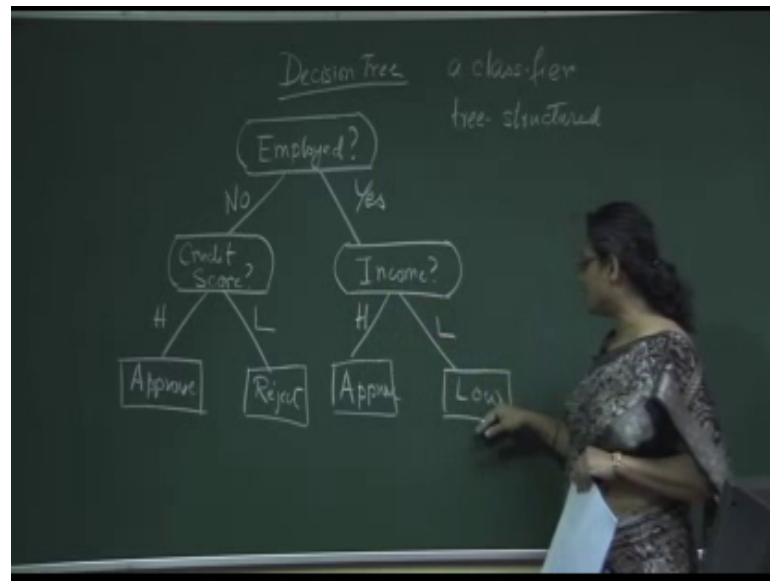


It is a tree structured classifier, which is tree structured and it has two types of nodes, decision nodes and leaf nodes. So, in decision nodes they specify a choice or a test based on this you can decide which direction you can go. So, in a decision tree we test something and that test may have more than one result and the based on the value of this test, you either follow this branch or this branch.

So, this test is usually done on the value of a feature or attribute of the instance. So, test is on some attribute and there is a branch for each outcome. So, there may be two outcomes or in some cases, you can have more than two outcome. And then, there are leaf nodes. So, leaf node indicate the classification of an example or the value of the example. Decision trees can be used both for classification and regression. However, it is more popularly used for classification though they can be used for regression also.

So, given an example, you start with the root of the tree and based on the value, based on the value of the test you go to the corresponding branch and you continue doing this until you come to a leaf node, and at the leaf node you have the value of the example. It can be the predicted value of the example for classification, regression or it can be a probability. So, let us make it clearer by drawing an example decision tree.

(Refer Slide Time: 04:15)



So, let us say, we want to draw a decision tree about whether to approve a loan. So, let us say, the first test that we will make is check if the applicant is employed. So, this is a decision node and we test whether the applicant is employed and there are two outcomes, no or yes. If the applicant is not employed, then we have another test, we check the credit score of the applicant.

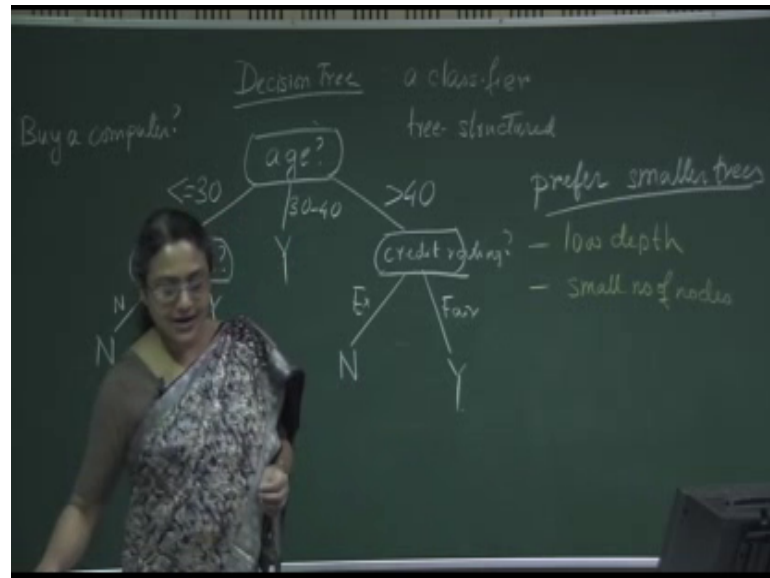
Does he have a high credit score? Now, if the credit score is high, then you approve the loan and if the credit score is low, then you reject the loan. If the applicant is employed, then you have another test. You check the income of the applicant and if the income is high, you approve the loan and if the income is low, you reject the loan. So, this is an example of a decision tree. We have three decision nodes and four leaf nodes.

Now, how do you use the decision tree? Suppose, that applicant has, is employed, has low income. So, he will come here, first, the applicant is employed. So, he will follow this branch. Check the income; income is low. So, he will follow this branch and the class here is, sorry, not approved, this is reject, so class here is reject. So, this applicant will be rejected.

So, what you want to do is that given the training example, in the training example you have, for past applicants you have the different attributes of the applicant including the income of the applicant, whether the applicant is employed, what is the credit score and several other attributes of the applicants are there and also the suggested action, whether

the loan should be approved or rejected, that is given in the training set. From the training set (Refer Time: 07:05) can come up the decision tree like this so that given a new applicant, you can find out whether you should accept or reject the loan.

(Refer Slide Time: 07:32)

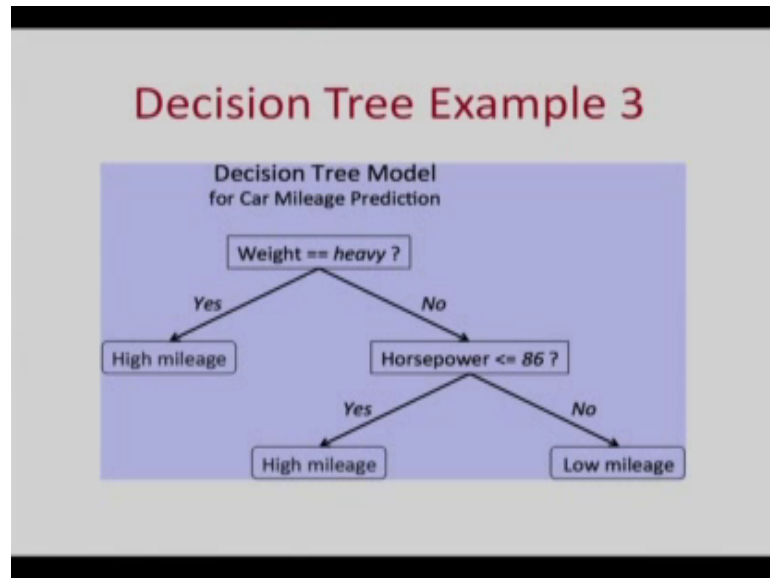


Similarly, you can have a decision tree to decide whether a person is likely to buy a computer. So, let us say, you can check the age of a person. Now, age is a continuous valued attribute. So, what you can do is, one of the things you can do is, that you can break the entire age range into two or more classes. For example, you can say, you can divide them into three classes: one class is age less than 30, another class is age less than equal to 30. So, age is between 30 to 40. Another class is age greater than 40, right. So, this is how you can also accommodate continuous variables as attributes in a decision tree.

Now, suppose if the age is less than 30, you have another decision variable checking if the applicant is a student. If us, if he is a student, then let us say, we say, that he is likely to buy a computer. So, this is the decision tree whether an applicant, whether a person is likely to buy a computer. So, if he is a student, yes; if not a student, no, okay. If the age is between 30 to 40, let us say yes, all persons between 30 to 40 are likely to buy a computer. If the age is greater than 40, like mine, so you check the credit rating of the applicant. And let us say, if the credit rating is excellent, then is not likely to buy and if

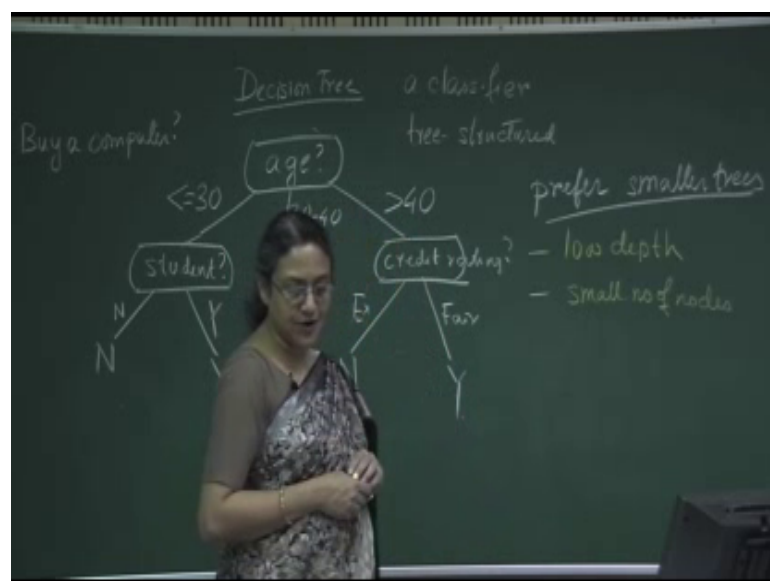
the credit rating is fair, then he is likely to buy a computer. This is another example for decision tree.

(Refer Slide Time: 09:38)



And this is another example in the slide here. This is the decision tree to predict the car mileage prediction. Is the weight of the car heavy? Yes, then high mileage. Is it no? Then, check the horsepower. If horsepower is less than equal to 86, then high mileage; if no, low mileage. This is another example of a decision tree.

(Refer Slide Time: 10:05)



Now, what we have to do is, given some training examples we have to generate a decision tree. Now, given a training, given some training examples it is possible, that there can be many decision trees, which fit the training examples. Then, our question is, which decision tree should we choose among the many possible decision trees. You know, in our last class we saw, that in linear regression we want to find the equation of a line and we chose that line for which the sum of squared errors is minimum.

Now, given some examples, if the examples are noisy it could be, that there is no decision trees, which exactly fit the data. Then, you have to choose which one would we choose that have no error or it could be that there are many decision trees that fit the data, then you have to find out which one we should choose.

So, in the week 1, we talked about bias. We said, by using bias we restrict the hypotheses space or we put preferences on the hypotheses space; once we have chosen decision tree as the hypotheses space we can put some preference. So, commonly the preference that is put for decision tree is to have tree with smaller trees. So, you say, prefer smaller trees. So, this is the bias that you can select.

And then, what do we mean by smaller trees? You can define smaller trees as trees with smaller number of nodes or trees with smaller depth. So, low depth trees or small number of nodes. So, we want to come up with a decision tree and then we have to now come up with an algorithm. This algorithm will search the space of decision trees and come up with a small tree. Ideally, given the set of training examples, if there is no noise we want to come up with a decision tree, smallest decision tree that fits the data, but finding the smallest decision tree that fits the data is a computationally hard problem. Therefore, we look for some greedy algorithms, we search for a good tree and we have to decide how we can come up with a good tree for learning the decision tree.

(Refer Slide Time: 13:01)

Example Data

Training Examples:

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	Home
e2	reads	unknown	new	short	Work
e3	skips	unknown	old	long	Work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

New Examples:

e7	???	known	new	short	work
e8	???	unknown	new	short	work

Now, this is some example data on which we can use a decision tree and so, these are some training examples. These are certain attributes: author, thread, length, where. So, you want to know whether a user reads a thread or skips a thread and given the attributes who is the author of the post, whether the thread is new or old, what is the length of the post and where the user currently is, you want to decide the action of the user, skips or reads, right. So, given this attributes you want to learn a decision tree so that when you get some new examples you can find out, in the case of e7 whether the reader will read or skip.

(Refer Slide Time: 14:23)



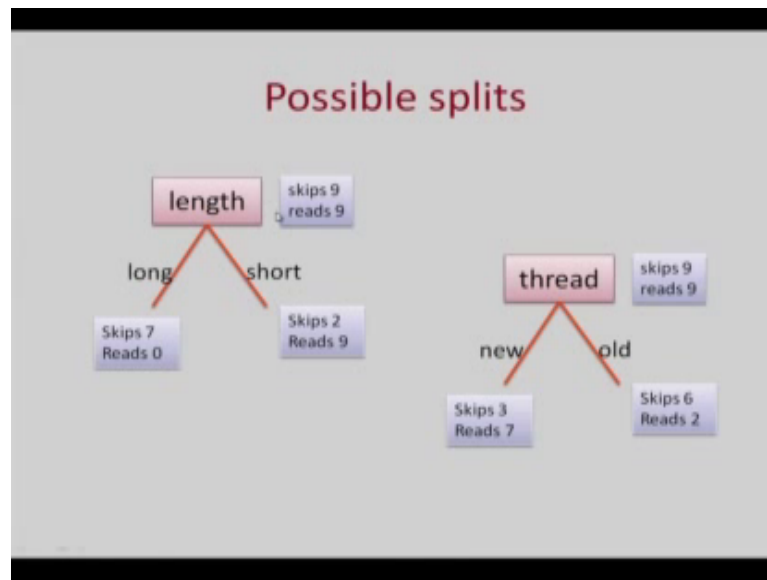
Now, so let us see how we can learn decision trees. So, we are given training examples. For example, we are given this sort of training example. Suppose D is the set of training examples. So, we have all the training examples in the beginning and we have to choose a test here.

Now, when we chose a test, suppose the test has two outcomes, yes and no. So, some of the training examples will satisfy this outcome, some will satisfy this outcome. Suppose, the test is on an attribute A_5 . So, A_5 is one of the features and let us say, this feature has value yes or no. So, some of this examples in D will have A_5 equal to yes, some will have A_5 equal to no. So, D_1 is the subset of D for which A_5 equal to yes; D_2 is a subset for A_5 equal to no. So, this number of training examples will come here. Now, here again we can decide, that if all the examples in D_1 have the same output y , then we need not expand the node D_1 corresponding to D_1 further, but if they have different values, then we can split this node further and we have to choose another attribute on which to split the node.

Suppose you choose A_2 and suppose A_2 is also bullion, it has two values. Now, part of D_1 will come here, D_{11} and part of D_1 will come here, D_{12} . And then, you look at all the examples here and suppose all the examples in D_{11} are positive, then you say positive and you stop. And suppose D_{12} has a mixture of positive or negative examples, you again choose an attribute to split on and then you proceed further. So, this is how we recursively build a decision tree.

We do the same things at all the nodes. So, at every step we have to make a decision whether to stop growing the tree at that node or whether to continue. If you want to continue growing the tree we have to decide which attribute to split on. So, these are the decisions that we have to make.

(Refer Slide Time: 16:59)

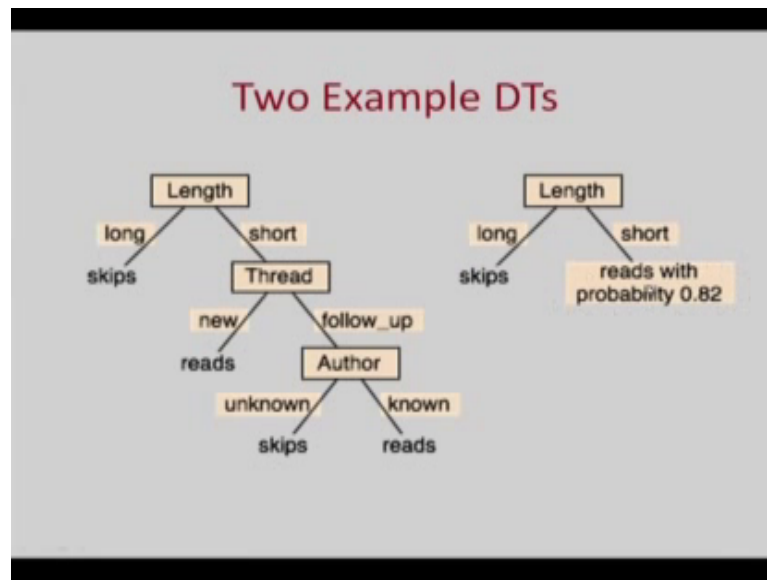


For example, on this examples, suppose we take length as the attribute and let us say, the examples that are there in the node, 9 of them has the action skip and 9 of them has action read. So, we split on length and length has two values, long and short. For length equal to long, there are 7 examples, all of them have skip. So, we can stop growing the tree here. For length equal to short, there are 11 examples, two of them are skip 9 of them are read. So, we have to decide whether to continue the tree here and then which attribute to use here.

On the other hand, on the same examples, if you use the attribute thread to split, then thread has two values, new and old. For thread equal to new, there are 10 examples, three of them skip and 7 of them read. And thread equal to old, there are 6 of them are skip and 2 of them are read.

So, what we have to decide is, at this particular case, you know, we have four attributes: author, thread, length, where, out of this 4 attributes which attribute should we use at the root. For example, length and thread are the two of the attributes. So, do you think we should use length or should we use thread? You see, if we use the attribute length, for one value of length we can immediately get to a leaf. Remember, we wanted to find a decision trees, which are smaller. So, the quicker or faster we reach the leaf, a smaller tree that we get. So, based on that, you know, the attribute length appears to be more promising.

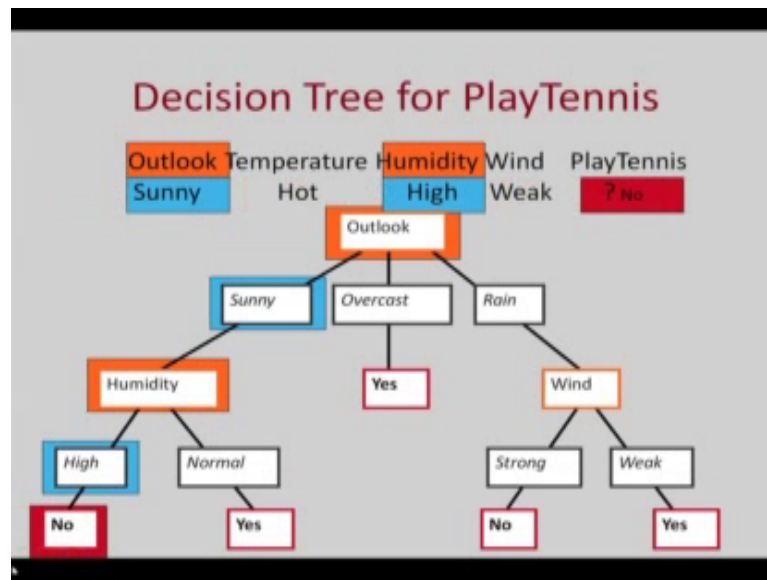
(Refer Slide Time: 18:48)



So, these are some examples of decision tree. This is a decision tree where the ones that we have seen earlier where each leaf is giving the class we can also have, you know. So, here we start with length. Length is a long, we say skip; if length is short, we further try to grow the tree. Or, this is another decision tree for the same examples where length is long, we have a leaf which says skip; if length is short we do not have a leaf because here we can either skip or read, but read is more probable. So, we can stop here saying, that this leaf is a read with probability 0.82.

So, you can grow the tree so that every leaf has a specific value or you can stop at a point where at a leaf there are more than one possible values, but one of them is dominant.

(Refer Slide Time: 19:47)



So, let us take one example. This example is taken by, from the book on machine learning by Tom Mitchell. So, where he looks at a decision tree to decide whether it is a good day to play tennis. The attributes used in the decision tree are outlook. Outlook can be sunny, overcast or rainy; humidity, which has values high and normal; wind has values strong and weak; and temperature has hot, mild and cool. And the target concept or why is whether it is good day to play tennis, yes or no and this is a sample decision tree to play tennis. If outlook is overcast, it is a good day; if it is sunny, if humidity is high, it is not a good day; if humidity is normal, it is a good day and so on.

Now, in this decision tree we have internal nodes at decision nodes, which test on attributes and branch corresponds to an attribute value node and there are leafs, which assign a classification. Now, given this decision tree and give a new example for which outlook is sunny, temperature is hot, humidity is high, wind is weak, you want to know whether it is a good day to play tennis. So, you first check the root, it says outlook. Because outlook is sunny, you go the left branch. Then, you check for humidity. If humidity is high, you take the left branch and then it says no, it is not a good day to play tennis. So, your output no. This is how you use a decision tree.


So, a decision tree can be expressed as a boolean function, which is a disjunction of conjunctions. So, decision tree is a very flexible function, which can represent

disjunction of conjunction and thus it can represent all boolean function. If a decision tree is of sufficiently large size, it can express all boolean functions.

(Refer Slide Time: 21:59)

Searching for a good tree

- The space of decision trees is too big for systematic search.
- **Stop** and
 - return a value for the target feature or
 - a distribution over target feature values
- **Choose** a test (e.g. an input feature) to split on.
 - For each value of the test, build a subtree for those examples with this value for the test.



Now, as we said, that the learning problem is, given a decision tree we have to find a good tree and there are two choices that you have to make. You have to decide at a particular point, whether you should stop or whether you should continue. If you want to continue we have to choose a test, that is, you have to choose an attribute or a feature to continue with.

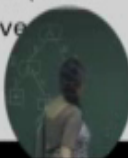
(Refer Slide Time: 22:21)

Top-Down Induction of Decision Trees ID3

1. Which node to proceed with?

1. $A \leftarrow$ the “best” decision attribute for next *node*
2. Assign A as decision attribute for *node*
3. For each value of A create new descendant
4. Sort training examples to leaf node according to the attribute value of the branch
5. If all training examples are perfectly classified (same value of target attribute) stop, else iterate over leaf nodes.

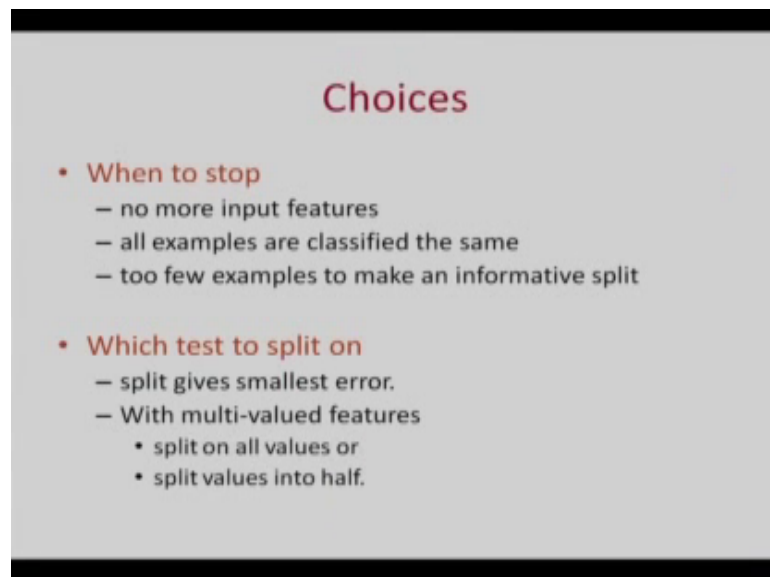
2. When to stop?



Now, we will just give the framework of a basic decision tree algorithm. This algorithm is called top down induction of decision trees and this is the basic ID3 algorithm, which was proposed by Quinlan. So, these are the steps of the algorithm. At the current node you choose the best decision attribute, then assign A as the decision attribute for the node. For each value of A, that is, the outcome you create a new descendant and then the training examples will get split into the different branches. So, you sort or split the training example to the leaves of the current node according to the attribute value of the branch. So, at a particular node if you find, that all the training examples have the same class, then you can stop otherwise you can again continue this process.

So, as we see, there are two choices that we have to take. We have to decide at a particular step if we have to continue which attribute to use for the test and we have to decide when to stop. We have also to decide when we have a partial decision tree, should we continue at this node or this node. So, we have to decide which node to continue with. Once we choose a node we have to choose the best attribute of that node and we have to decide when we want to stop. These are the decisions that we have to take in a decision tree and in the next class.

(Refer Slide Time: 24:08)



Choices

- **When to stop**
 - no more input features
 - all examples are classified the same
 - too few examples to make an informative split
- **Which test to split on**
 - split gives smallest error.
 - With multi-valued features
 - split on all values or
 - split values into half.

So, these are the two decisions that we have to take and in the next class we will look at some specific heuristics to may take a decision on these.

Thank you.