Lecture – 38

I am going to teach couple of machine learning techniques called Decision Trees and random for us in this lecture.

(Refer Slide Time: 00:25)

We will start with a decision tree. So, decision tree as the name suggest is a tree like structure that you generate from data to do some machine learning tasks. And, as we have seen before, the most common machine learning tasks are either classification or regression or function approximation.

So, you can generate these decision trees for both these types of problems, both classification problems and regression problems. Just to recap for people who have not seen this many times. A classification problem is one, where when you are given a data you are trying to classify it to one of the predefined classes that you have. And, regression problem is one where given a data you are trying to predict an output or target feature value. Now, you could build a decision tree for either a classification problem our regression problem.

And there are multiple algorithms for building these trees, which are all mentioned here. But, what I would like to mention is at the end of it what you are going to get is like a tree structure which is going to let you either predict a value for a target or provide a classification. The reason why I mentioned many algorithms is that when you use a particular software or a package these would become options that you can choose. And there are multiple metrics that are used in building this decision tree; and I will demonstrate how one metric works in this lecture and you can see that the other metrics will work similarly.

So, once you understand one of these you will be able to read on your own and understand the other metrics and how they might be used. So, these metrics basically, we will talk about them in more detail as we go along. These metrics basically allow you to unravel the tree so, to speak. So, if you think about a decision tree as a tree, with some top node and then there are multiple nodes that are being unraveled and formed; there are must be some logical procedure for this unraveling and making this tree from data.

And, there could be multiple matrix and Gini impurity information entropy or variance reduction or metrics that are used. Typically Gini impurity and information entropy are used when you have classification problems. And variance reduction is a metric that is used, when you have regression problems that you are trying to model. So, in this lecture I will focus on classification problems and Gini impurity isometric. But what you need to see through this lecture is how this tree is formed? And once you understand that it will be very easy to see how you can develop a tree like that, for regression problems.

Describing decision trees through an example

- Consider the following classification problem
- Discriminate between three different species of Iris flower
- The training data contains 49-setosa, 50-versicolor and 50- virginica species
- The features that are available are sepal length, sepal width, petal length and petal width
- The ranges for these feature values are

	setosa	versicolor	virginica
S.L	[4.3,5.8]	[4.9,7]	[4.9,7.9]
S.W	[2.3,4.4]	[2,3.4]	[2.2,3.8]
P.L	[1,1.9]	[3,5.1]	[4.5,6.9]
P.W	[0.1,0.6]	[1,1.8]	[1.4,2.5]

Decision Tree and Random Forest

So, the way I am going to teach this development of a decision tree or understanding decision trees is through an example. So, this will make it a lot more concrete in terms of your understanding in terms of what the key ideas are in these decision trees ok. So, let us start with a classification problem as I mentioned in previous slide, I am going to focus on a classification problem. So, let us assume that we have this problem this is a very very well known problem.

If you just search for this you will see this being described in multiple papers it used to be a data set that many people tried the algorithms on for has been for a while. Now so, the idea is there is this iris flower and there are multiple species of this, given a new Iris flower would you be able to classify it as one of the three species that we are considering here. So, the flower type is iris and the species are setosa, versicolor or virginica species. So now, when we say that given as a flower would you be able to classify it into one of these categories, we should have some training data to do this.

So, let us assume in this case for me to explain decision trees to you. Let us assume that there are above 149 data points and; that means, I have 149 samples of this flower which are pre classified as being setosa, versicolor or virginica species. Now when we say data what does that mean? So, for each sample what we might do is we might compute certain characteristics of that sample. In this data set there are four features there are computed for each sample. So, one is called the sepal length, the other one is sepal width petal length

and petal width. So, these are actual geometric features that that you compute for the sample. And then let us say you have this 49 already I characterized setosa you take each sample measure this and then put it into data and then say this is from setosa and so on.

Now, the problem is I have to come up with a classifier, which will learn how to classify the data when I when I have a new data point. Based on already pre-classified data that is used to train my algorithm; and the algorithm that we are going to look at in this case is a decision tree because which is what we are going to use in the example that we are going to look at. Now, the ranges for these feature values are several length if its a setosa flower, we notice that it is between 4.3 and 5.8.

If its versicolor 4.9 and 7; virginica 4.9 and 7.9. This is basic inspection of data and then seeing what are the ranges of these values for each of these attributes for each of these classes. So, this is one class this is another class another class; and the attributes are distributed in these ranges for this data. So, this is basic explanation of the problem that we are trying to solve. Now, you will notice as I described decision tree starting with this problem we will see that just from this data, we are going to generate a tree which is going to allow us to classify new data points.

So, the key things that I want you to notice as we go through this example or how this tree is generated at the end of it. If a new data point is given how would, that tree help us classify into one of these three classes is the important thing that I would like you to notice.

(Refer Slide Time: 07:23)

So, let me explain decision trees through this example of this iris classification. So, decision tree as a name suggests is tree like structure where you have nodes which open out into other nodes which might open into other nodes and so on. And, you start with one node and then you come to let us say certain number of nodes at the end.

And, how do you generate this tree and how it is useful for classification is what we are going to see in this lecture. So, to understand this easily let us think about, what each node means? So, what each node means is that each node has a collection of data points. So, if I start with the root node or the very first node, this node is going to have all the data points that are there in the problem.

So, in this case we know that there are 50 data points from versicolor; 50 data points from virginica; and 49 data points of setosa. Now if we did not build this tree at all and we just sat with this data; and then I am going to say let me do a classification whenever you give me a new data point; then the best way to do this would be to look at this data. And then see which class is the most occurring class in this data point under the assumption that the global data set will also has similar proportions.

So, whatever is occurring the most here is the most likely species type in terms of the total data set. Then we will say it has to be either versicolor or virginica. And if you are forced to give one decision you might randomly break the tie and then say a new data point is either versicolor or virginica right. So, if you were to just sit at the root node and then do this is the solution that you will come up with.

But this is not useful from a classification viewpoint what we are trying to do is, we are trying to use the features of these samples. And, then do some computations so that we get a good classification with a high accuracy is what we are looking at right. Because, if I just kept saying versicolor as the class without doing any more development of this decision tree. Then in the 149 sample points I will get 50 of them right because every time I am just going to say versicolor.

So, 50 I will get right, but 99 times I will get it wrong. So, that is very poor accuracy which is what you do not want. So, basically the idea is to somehow develop this tree. So, that whichever node I stopped and give an answer the accuracy improves. So, that is basic idea of this decision tree. The same notion also works with regression trees, if you had one

target value at the first node if I if you ask you what is likely to be the target value for a new data point.

If you do not want to do anything at all you might simply take the average of all the training data points and say that is a likely value for any new data point right, but that will give you a very poor model. So, in that case also you will try to develop this tree expand this tree so that your regression problem you can solve much better. So, the same idea works for both classification and regression problem is what I wanted to tell you.

Now we are going to learn how we are going to develop this tree starting with just this data node the first node. And these are concepts that we are going to introduce for us to explain to you how this tree is developed? But before we do that to understand this better, let us look at what might be the best case scenario for this example. If I were to develop a tree supposing I were to develop this tree at the end of it let us say I get three nodes.

And, this node has data of all versicolor and virginica is 0 this is 0 and this node has all data corresponding to 0 versicolor 50 virginica 0 setosa; and this has data corresponding to this 0, 0, 49. If this is the case then what happens is, as you start from here and go through these nodes we will explain what going through this nodes mean. As you start from the base node and use the features of the new data that has been given the feature values and you traverse this tree. And if you end up here, then you will say the new data belongs to versicolor sample. If you end up here you will say it belongs to virginica and if you end up here you will say it belongs to setosa.

Now, the question is how do you develop the tree? So, that this partition happens and what does traversing this tree mean. So, how do I start from here and decide whether I should go to this node or this node is basically what I am going to teach and that is the basic idea of decision tree as an algorithm itself. Now, remember that you might not always be able to classify this data into such distinct nodes, there might be some overlap which you might simply not be able to get rid of and all of this depends on the data.

So, in some data sets you can get a complete separation; and in some data set whatever you do you might not get complete separation. And actually you know from your machine learning knowledge that getting complete separation and training said by itself might not be very helpful, because basically we are over learning the training set. So, the ability to generalize when I get a new data point might be much less if this happens ok. Now, that

we have set up this basic tree structure and then explain to you the ideas of how we are going to use this tree to make the classification.

Now, it is easy for me to describe how this tree itself is generated purely from data; and how these algorithms work to develop this tree. A tree that is developed like this from data is called a decision tree and it is called a decision tree because at every node based on the feature value we make a decision traverse the tree till you stop at one point which you where you give the final decision in terms of what class this data belongs to ok. So, it basically mimics how we take decisions if this is this and that is that then do this and so on. The same notion is captured here in the tree ok.

So, now to develop this tree we are going to introduce some basic terminology as I mentioned before, I am going to describe this notion of Gini impurity. So, if you notice here Gini impurity for every node so, you can you can define a Gini impurity for every node in the tree. And, for every node in the tree Gini impurity is different using this equation. Let us try and understand what this equation means. So, to understand that equation, let us take this node here and then we will come back to one of these nodes and then kind of contrast what happens.

So, if I take a Gini index for this node, this formula basically says Gini index $= 1 - \sum_{i=1}^{3} f_i^2$. So, basically we have to define what f is. So, there are three classes, each f basically computes a fraction of samples from a particular class in the total set. So, for example, if we compute a fraction for versicolor f 1; the number of samples in this data set of versicolor is 50, the total number of samples is 149. So, f 1 will be 50 by 149 and f 2 will also be 50 by 149. And f 3 will be 49 by 149.

So, once we have these three numbers we can compute a Gini impurity for this node as $1 - \left(\frac{50}{149}\right)^2 - \left(\frac{50}{149}\right)^2 - \left(\frac{49}{149}\right)^2$. So, that is how you compute the Gini impurity of this node ok. Similarly, once we get to each of these nodes based on the number of data points of these classes in the total data we can compute Gini impurity. And, you will notice and in this example it will be nicely seen as we go through this. As we go to different nodes because your partitioning this data this Gini impurity will keep changing for each one of these nodes.

Now, if you are defining Gini impurity in a decision tree for each of these nodes we might also ask this question as to what is a best node right in this decision tree? So, if you go back and look at these nodes so, for example, let us try and compute a Gini impurity for this node. Why am I picking this node this node is what I am going to call as pure node because if you look at this data point. If I come to this node I can categorically say the sample belongs to versicolor right.

If I somehow land up in this node, then I can say the sample belongs to virginica. And if I land up here I will say the sample belongs to setosa right. So, these are what are called pure nodes; that means, these nodes have some more collected data, corresponding to only very specific classes where the other classes are excluded in the data. So, if I have a node like this a pure node a how would I compute the Gini index I use the same formula?

But what will that value be we can see this here. In this case again I have $1-\sum_{i=1}^{3} f_i^2$. In this case now, f_1 for versicolor is 50 divided by 50 because $50 + 0 + 0$. There are only 50 samples this is equal to 1, f_2 will be 0 by 50 and f_3 will be 0 by 50 ok. So, the fractions will be 1 0 0; and once you use this and then calculate the Gini impurity index. So, the Gini impurity will be 0 ok. So, whenever the Gini impurity is 0; that means, we have got pure sets were only one class is represented and the other classes are all left out. So, ideally then the goal of the whole decision tree is to start with the data itself which gives me a Gini impurity based on doing nothing with this data.

And then somehow unravel this tree and get two nodes at the bottom, which are all pure are as close to pure as possible right. So, here the top of this decision tree there will be a positive value for this Gini impurity. And our goal is to come down to the lowest level of the tree where all the nodes have Gini impurity of 0 or pure nodes. So, in other words what we are trying to do is we are trying to keep reducing this Gini impurity as we go down this tree to get to 0 Gini impurity so that is what we are trying to do.

So, now we come to this question of how do we decide how to traverse this tree? So, then we ask this question what does it mean when we say I want to traverse this tree. So, basically as I said before each one of this is a decision. So, at this point I have to take a decision and the way I take a decision here is I pick one feature from the data set and then do some computation with that feature. And, the result of this computation allows me to go either here or here.

So, I have one example might be take one feature and if that feature is greater than 5; go to this node and if the feature is less than 5 go to this node might be one way to do this. So, in others other words whenever we are trying to traverse on unravel will this tree, what we are looking at is we are looking at a future value and making some decisions based on that future value. So, this whole tree itself is developed like that and each one of these decision points will have a feature and something that you compare it with, so that you can develop this decision tree.

Now, as you notice even in this case there are four features right petal width, petal length, sepal width and sepal length. So, you could choose any one of these four features for opening out this tree. And each of these features have different ranges of values as we saw in the previous slide. So, there has to be some partition point for those values so, that we make a decision to go to one or the other part of the tree. So, how do we do that ok, that is what we are going to see.

So, once we have this Gini impurity, we are also going to discuss something called Gini split index. So, let us assume I choose some feature and then decide to split this data ok. Now, what does it mean, when I say one feature and somehow I am going to make this decision to split the data we will see in the next slide? But for now bear with me I am just think about this supposing I say I am going to pick one feature in this case you know sepal length; and then I am going to say if sepal length is less than some value a; go here and if it is greater than equal to some value a, go here right that is something that I can make a decision.

So, why should I only choose sepal length not sepal width why petal length petal width? And so on are questions that we are going to answer, but just to explain this whole process here I am explaining this ok. Then if this is how I am going to unravel this tree, then what I do is of all the samples here I find the samples that would satisfy this condition and then put them here ok; maybe of the 50 samples we will see the actual values I am just explaining this. So, maybe of this 50 samples you know if 30 of these samples are such that sepal length is less than a; then the remaining 20 will go here.
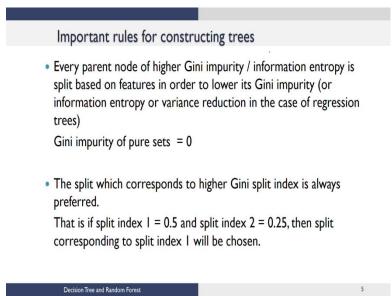
So, this way you split the whole sample of 149 data points into some number. Now it might be just that this has maybe 99 data points come here 50 come here it all depends on how what we choose here. Maybe 60 of these data points come here and the remaining go there

and so on ok. So, now, you have this because we have already defined the Gini impurity for each node based on this we can actually compute a Gini impurity for this node. And based on what comes here we can compute a Gini impurity for this node.

So, now if we use this sepal length and this number a; as the choices to make this split; then we can compute something called Gini split index which is basically the difference between the Gini index of this original node minus p_1 and p_2 will come back to we can compute a Gini index for this node same formula based on how this split happened. And we can compute a Gini index for this node based on how split this split happened.

The only thing that we need to understand at this formula is what are p_1 and p_2 that is very simple. If I start with 149 data points, let us say 99 data points come here and 50 data points go here. p_1 is the fraction of data points that came to this node which will be 99 by 149. And p_2 will be the fraction of data points that went to this node which will be 50 by 149. So, we can compute this fraction and we can also compute the Gini index for the each of these once we have that we can compute this Gini split index ok. Now what this value is will depend very critically on what feature we chose here to unravel this. And also what is the value of the feature that we chose to make that partition.

Now, the whole notion of decision tree is how do you come up with the sequence of features so, that you unravel the tree and the values that you use to partition this all of those are automated. And these algorithms these packages will give you the best solution for these splits and so on. You do not have to do it, but I am explaining this, so that you understand what basically happens when you finally, see a result for a decision tree example that you might work with.

So, basically the important rules for constructing this is so, every parent node which will have a higher Gini impurity remember I said at the at the top is where I have not done any splitting. And ideally what I am looking for is the bottom level nodes which are pure so; that means, for then the Gini impurity is 0 there they are pure sets. So, basically Gini impurity keeps coming down as I go down the tree. And there are multiple options for doing this splitting.

So, what we do is, whenever there are multiple options we could enumerate all of those options or we could use some smart algorithm to pick options that are likely to be very good. And let us say I have multiple options, I will compute the Gini split index for each of these options and then I will always choose the Gini split index which is maximum right. Now, why do we want to choose a Gini split index which is maximum that comes from your previous equation here.

Now, if I am having a node with a certain Gini impurity and I want to make children nodes of those what I want to do is, I want to get to these pure nodes as quickly as possible right, only then I can do perfect classification. So, ideally if I could split in such a way that the two children node that come out of this main node, have Gini impurity of 0 right that would be the best solution. In which case, the Gini split index will be the Gini impurity value of the original node because this will be 0 this will be 0 right, so that is the best solution.

So, actually the split index should be as high as possible which basically means that these values are as low as possible. So, this has low impurity or there as close to pure sets are possible. So, when I have multiple possibilities one thing I can do is I can you numerate these possibilities; and then for each one of these possibilities I compute a Gini split index. And, then I find out the possibility for which the Gini split index is a maximum and then I say this is the choice I am going to make ok.

(Refer Slide Time: 26:57)

So, that is what is seen here. So, let us look at this now with this example now that I have explained all of this, you will understand this hopefully much better. So, the root node as we showed before in this example is with this number of data points ok. So, we do 49, 50, 50 in this case for 49 setosa 50 versicolor and you know 50 virginica. So, if you know look at this node here you see these numbers of data points ok; and also you see this node has something. So, which says versicolor that basically means if you are sitting in the node for a data point and I asked you what do you classify the sampler.

So, you are going to say I am going to classify the samplers versicolor; that means, at the beginning if I do not do anything at all any new sample you are giving me I am just going to close my I sense and say its over versicolor. And why am I saying its versicolor I could have said versicolor are virginica because there are 50 data points I have just randomly broken this tie and then said its for versicolor ok.

So, this now you see in this node the none of the data is lost ok. So, all the data is still there here right, 49 setosa 50 versicolor and 50 virginica. Now, remember in the tree thing I said the first decision we have to take. So, the decision means I have to choose one of the features there are four different possibilities. What the algorithms do is that, they look at all these multiple possibilities and find the best split somehow to come up with the most compact tree that you can build. But here in this lecture I am trying to explain the ideas behind it. So, I am going to take a couple of examples to show you what happens.

So, for example: if the algorithm had actually chosen the petal length as basically the feature of choice here. Now, you look at petal length and then. So, you see this here. So, if its setosa the petal length range is 1 to 1.9 versicolor its 3.5 to1 and this is 3 to 5.1 sorry. And this range is 4.5 to 6.9. Now, if you notice all of this are automatically done by the algorithm here I am just trying to explain this with this data so that you understand there is this logic behind how I break this and so on.

And, once you look at a tree you will be able to see what is actually happening here. Now if you look at this range 3 to 5.1, 4.5 to 6.9 you will see that there is an overlap right. So, any data value that I pick which partitions this, there is likely to be both versicolor and virginica and the partition data. But if you look at this here there is a clean partition because for setosa, petal length is between 1 and 1.9. For versicolor is 3 and 5.1 for virginica its 4.5 and 6.9.

So, if you take any value between 1.9 and 3 ok, you would be able to separate out setosa from the other species right. So, one value you can take is roughly in the middle of 1.9 and 3.5 so that you get good classification. So, you might say if petal length is less than 2.4 go to this node and if petal length is greater than 2.4 go to this node right. If that is the decision that you are making then let us see how the data will get partitioned right.
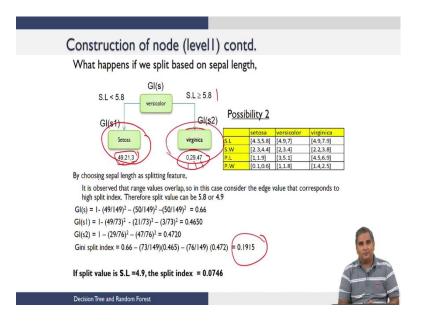
So, in the training data if you pick all the data points where petal length is less than 2.4 and bring it to this node, you will notice all the setosa data will come here to this node. And all the versicolor and virginica data will go to this node ok. So, remember this 149 data points now has been split it into two nodes. One node has retained 49 data points and the other node has retained 100 data points. Now you notice that this 49 data point node is a pure node whereas, the 100 data point node is not a pure node.

Nonetheless, if you make this decision that I am going to choose petal length as the first feature based on which I am going to separate this and 2.4 as the number then you will get this. And, now you can quite easily compute the Gini index for this which is sorry Gini impurity for this which is $1 - \left(\frac{49}{149}\right)^2 - \left(\frac{50}{149}\right)^2 - \left(\frac{50}{149}\right)^2$ which is what I had mentioned, this will turn out to be 0.66. Now since this is a pure node, you will get the value to be 0 here because at the three fractions or 49 by 49, 0 by 49 and 0 by 49 so, you will get a value of 0 here. And, if you look at this here you will get a value of 0 by 100, 50 by 100 and 50 by 100 as the three fractions.

So, I will have 1 minus 0 minus 0.5 square minus 0.5 square which is what is shown here its 0.5. So, I have a Gini impurity for this I have Gini impurity for this I have a Gini impurity for this. Now, for this option of petal length and value of 2.4, if I were to compute a Gini split index remember Gini split inductors in indexes the Gini impurity value of this minus whatever fraction of data came here times the Gini impurity of this minus whatever fraction of data came here times the Gini impurity of this. We have already computed the Gini impurity of these three nodes here. So, the Gini split index is the original nodes Gini impurity minus 49 points came here.

So, 49 by 149 and the Gini impurity is 0 times 0 minus 100 points came here. So, 100 by 149 times the Gini impurity of this which is 0.5. So, if we compute this you get 0.324 ok. So, please look at this computation and if you understand this computation every other time the same computation is done there is no difference at all. So, all you need to know is that every node has a Gini impurity and then based on this partition you can compute a Gini split index.

(Refer Slide Time: 33:39)

So, if I were to do this and said instead of petal length let me use sepal length ok; and then I decide sepal length is what I use. And then let us say I come up with this value of 5.8, if you asked me how did you come up with this 5.8 there are multiple heuristic in these things. And, you can use a very simple heuristic here to come up with 5.8. Now, the reason why I do not want to go too much into this heuristic is there are multiple possibilities and these algorithms automatically figure out what are good possibility.

So, we really do not need to worry about how this 5.8 comes about, but you want to vary think about how once I have a final solution how I understand that solution. So, let us say somehow I have come up with this 5.8 as the as the best split value here. So, basically I say sepal length is 5.8 remember this is again the top node, which is whatever we had before without any partition. Now, what I do is I partition data such that all the data which have sepal length less than 5.8, I bring to this node and all the data which have sepal length greater than 5.8 I bring to this node.

And it might turn out when I do this that 49 samples of setosa comes here and 21 from the next one and 3 from the next one. And notice what I do here ok, I basically have this node saying setosa. And why does it say setosa? Because if you look at this data point the maximum number of class from which this data comes is setosa right. So, if I land up here after at the end of my decision tree process then I will say that setosa that sample is most
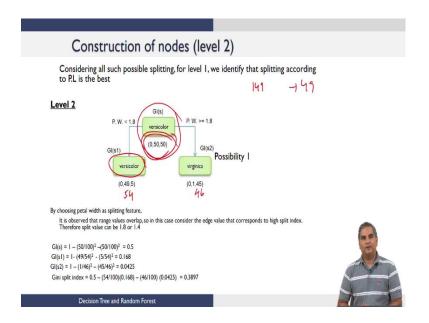
likely to be setosa right. And if you look at this here this is virginica that is because 0 of setosa 29 of versicolor and 47 of virginica.

So, the maximum number of samples come from the virginica species. So, I will say the sample that I have is a virginica sample. Now, I do not want to go through these computations, if you do the same computations as I showed before you will get a split index value of 0.1915. And, you will notice the previous value was different from this. The key things that I want to notice look at the mechanics of this very simple all that is happening is your original data set is being split into multiple data sets right.

So, in the first case the original data set was split into two data set in a particular fashion and in the second choice the same data set is split into two data sets which have different from what I got for the first data set right. So, the way the decision tree works is actually splitting this data set into multiple nodes. And in each node depending on whatever class is preponderant in the data I am going to classify that that is it right. So, if I have more nodes in the tree if I start from here now what will happen is this data set will be split more and this data set will be split more.

So, you can think of this original data as being whittle down into smaller and smaller data sets in each of these nodes and the way that smaller and smaller data sets are developed is through these choices that we make. So, in this case we set sepal length less than 5.8 then we go to the training data set. And, say all instances where sepal length was less than 5.8 I combine into this node and greater than 5.8, I combine into the other node.
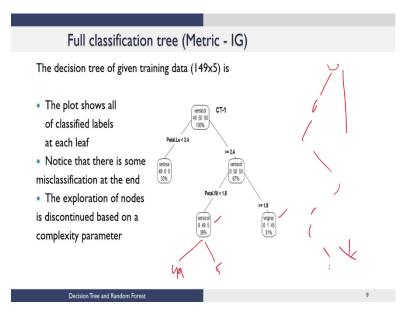
Now, after doing all of this possibilities we be might identify that petal length is the best possibility. So, which if you go back and notice here so, if I do petal length as the possibility then I have already classified all setosa samples. So, I do not need to explore this portion of the tree anymore, simply because there is nothing to do it is already a pure set right the only problem is here. So, what I want to do is I want to start exploring this part of the tree more; and if you notice this that what we are doing it.

So, we are starting with the node where I have this versicolor with 0 50 and 50. Now, again I have multiple choices right, again I can say I want to do petal length again there is nothing stopping you from that again I have this four choices and again I have to compare them some with some value. So, let us say I did choose petal width and then say less than 1.8. Now, what will happen is of the 149 samples 49 have already been classified. So, they are not any more under consideration. So, I start with this other 100 samples which is 50 versicolor and 50 virginica.

And, then what I do is I use this petal width and then say less than 1.8 I collect all of this data from this and see how many come here. So, about 54 of this 100 comes here and about 46 of the 100 comes here. So, 54 + 46 is 100 right. Now, notice what I call this node this node I call as versicolor because most of the samples in this node are from versicolor. And, this node I will call as virginica because most of the samples from this node are from virginica.

(Refer Slide Time: 39:09)

So, ultimately you do all of this and then basically you come up with this kind of decision tree. So, basically what it says is if I am here I have this versicolor as the decision this is setosa, this is versicolor versicolor virginica. So, now when I get a new data point, how will I use this decision tree? The way I will use this decision tree is, I will first take the data point and I will check what is the petal length in the data point right.

If that petal length is 2.4 then I will say for this new data point it is actually setosa and the problem is done right. Now, if it is greater than 2.4 then what I will do is I will look at the petal width of this new data point. And, if it is less than 1.8 I will say its versicolor if its greater than 1.8 I will say its virginica right. Knowing fully, well that I could have errors here and errors here. So, for example, you know 5 times when it is actually virginica this is being called versicolor even in the training data.

So, in the test data we do not know, that is where we will use a test data to check the decision tree and see how many times do I get this to be the correct number right. So, that is what I will look at. Now, there is always you know the possibility of some misclassification at the end the data might not be complete, so that you can clearly classify this problem all the time and you know you. If you are not happy with this and then you are saying ok, in this node its not still pure 49 and 5 can I actually break this down further. So, that I get 49 and 5 you have to look at other features.
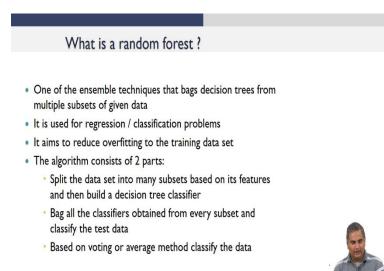
And, see whether the other features will allow you to do this in some cases they might allow you to do it in some cases the data might be such that you cannot do it. And, as I mentioned before even in cases where the data allows you to do this you do not want to keep building a tree which is very very complex right. So, you do not want to have a very very deep tree, which completely learns this training data set which will have no generalization capability.

So, you might want to stop at some point and then say in the training set I am willing to take certain amount of errors so that I have better generalization capability with a test data. Now, that we have seen decision trees random forest is a very very simple idea the key notion of random forest is the following. So, when you look at decision trees if there are minor changes in data or there are minor errors in data, the decision tree that comes about can be quite different.

So, decision trees are not generally very robust to errors in data right because you are choosing some numbers and in some cases if there are errors these numbers might not partition them very well and so on. So, in some sense what you want is if you give the data set and then you build a decision tree. And, if the data change set changes a little bit you do not want to see major changes in the decision tree. You do not want decision trees to completely change which is possible because of errors in data.

So, to avoid that and somehow give it robustness we come up with this idea of random for us; and as the name suggests random forest it is a collection of decision trees right. So, you might ask the question how do I make multiple decision trees from the same data set. So, the way random forest work is a following.
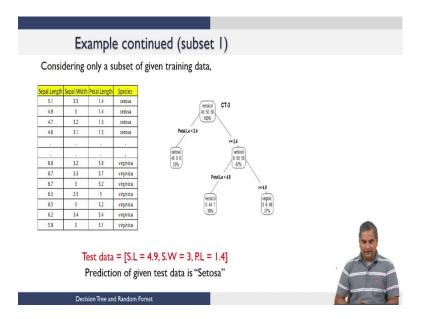
You have all this data, what you do is you make multiple data sets from your original data. So, how do you make multiple data sets from your original data? What you do is you sometimes sub select only a portion of the data right. And then say I am going to build a decision tree for this portion of the data, in some cases you might sub select only a certain features in the data. So, in the previous case we saw four features you might say I am going to drop one feature and then say that is my data set right.

So, you can sub select number of data points you can sub select number of features and so on. So, if I have one original data set from this I generate multiple data sets which are kind of marked form of the original data set, either through dropping some data or dropping some features and so on. Now, using the technique that I discussed for each of this data set you can come up with a decision tree right. So, if I make 10 different data sets from a original data then I can build 10 decision trees.
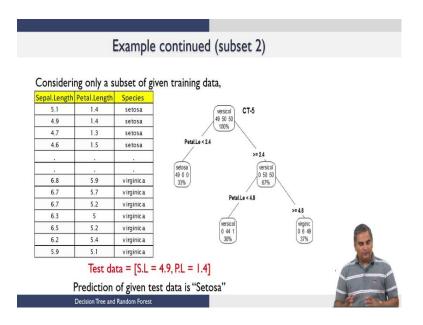
So, there those 10 decision trees together form a random forest ok. Now, you might ask the question has to I have now 10 decision trees which solution from which decision tree do I use?

(Refer Slide Time: 44:25)

So, in random forest what you do is, if you have 10 decision trees you go through all are those trees get a solution from all of them; and whatever is the majority decision of all these trees that is a solution of the random forest. So, for example, here you might get one subset of data where you only keep sepal length, sepal width and petal length and then build a tree. And, let us say you have a new test data point and let us say this three predicts setosa.
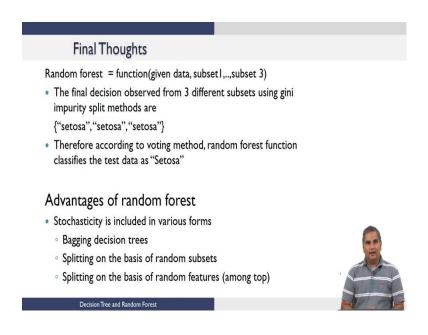
(Refer Slide Time: 44:43)

Now, you could have built another tree with only sepal length and petal length and that is another decision tree. And, when you give a new data point you only pick the sepal length petal length attributes of that and run it through this tree and then say let us say that is also saying it is setosa.

(Refer Slide Time: 44:59)

Now, you could have petal length and petal width and in these cases you could have dropped some data also right. So, you could sub-select data also and then send this new test data point and if that also says setosa.

(Refer Slide Time: 45:15)

Now, there are three trees and all of them said setosa so, the solution is set also. Now, if two of them had set setosa and one is virginica the solution is still setosa. So, basically that is what is called majority. Now, if one says setosa one says virginica and one says versicolor there is a problem, you have to break the tie somehow and then give a solution. But, in general when you have multiple trees you would hope that there will be some consensus among all of these three solutions and you say that solution is the solution for the random forest.

So, by doing this the stochasticity or the problems with the data can be addressed to a large extent in many problems. By basically bagging the trees there are multiple trees from which you are getting the result. And, each of these trees themselves are produced from splitting the data to through dropping of some data points are dropping of some features and so on. So, basically you try to make yourself immune to fluctuations in data by multiple data sets, that you generate and when all of them overwhelmingly say something is a result then it is likely to be correct than just depending on one decision tree, so that is the idea of random forests.

Now, all of this is for you to understand how these things work, but when you use one of these python packages they will do most of this work for you. All you need to know is when you look at it and see tree in to understand what that tree means. And you have to know the difference between random forest and decision tree and so on. So, hopefully this has been useful session for you, to understand decision trees and random for us.

Thank you.