

Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 14

Principal Component Analysis and Regression Applications in Agriculture (Contd.)

Welcome friends to this NPTEL online certification course of Machine Learning for Soil and Crop Management. And in this week, we are discussing Principal Component Analysis and Regression Applications in Agriculture. Today, we are going to discuss our lecture number 14. And this is the fourth lecture of week 3.

In our previous three lectures, we have already discussed some of the important aspects of Principal Component Analysis, what is principal component analysis, how we can utilize principal component analysis for dimensionality reduction. We have also seen, how to use the principal component analysis for clustering, what are the PLS PCA score plot, what is loading plot and what is the PCA by plot.

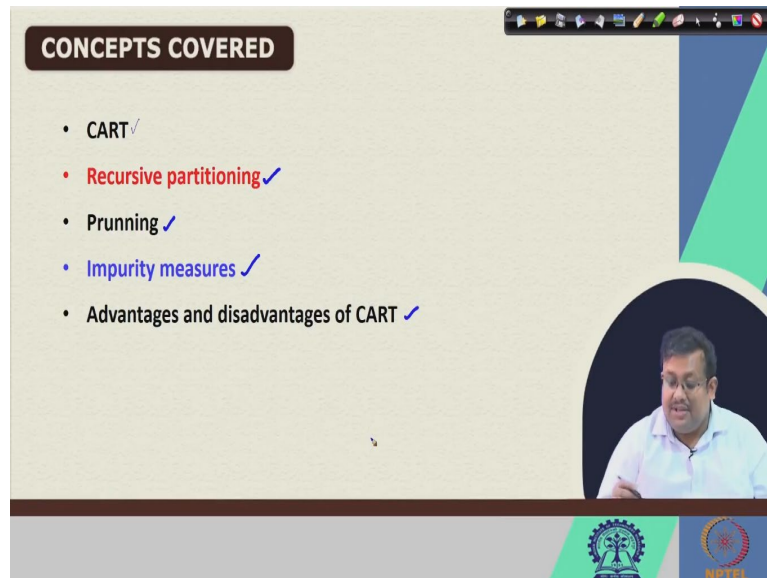
And then we have seen, what is the Principal Component Regression, which is an application of PCA for predicting certain predictors. So, and how to select the important principal component based on the Eigen values in the screeplot and also based on the cumulative percentage, the cumulative percentage variation explaining power of the principal components.

In our last lecture, we have also discussed one of the major important machine learning regression algorithm that is partial least squares regression for predicting several soil and crop properties. We have seen the major features of Partial Least Squares Regression and how we can use the Partial Least Squares Regression, what are the important consideration of partial least squares regression, how it differs from principal component regression, how we can calculate the Latin factors of partial least squares regression.

And we have also seen the mathematical representation and matrix based representation or partial least squares regression. So, remember the principal component regression or partial least squares regression which we have discussed so far, both of them are the parametric regression and today we are going to start another method, which is a nonlinear nonparametric method of regression.

We are going to discuss today the classification and regression tree. So, this is a very important aspect of machine learning application for soil and crop.

(Refer Slide Time: 03:05)



So, these are the major concepts which we are going to discuss today, we are going to first discuss, what is classification and regression tree and then we are going to also discuss, how this classification regression tree at generated by, how the software generates this classification regression tree by recursive partitioning.

We are also going to see, what is pruning for reducing the overfitting. We are also going to see, what are the different impurity measures and finally, we are going to see some advantages and disadvantages of classification regression tree. And we also we are going to discuss the some applications of classification regression tree algorithm.

(Refer Slide Time: 03:58)

KEYWORDS

- CART ✓
- Recursive partitioning ✓
- Entropy ✓
- Gini Index ✓
- Pruning ✓

The slide features a dark blue header with the word 'KEYWORDS' in white. Below the header, a list of five keywords is presented in blue text, each followed by a blue checkmark. The keywords are: CART, Recursive partitioning, Entropy, Gini Index, and Pruning. In the bottom right corner, there is a circular inset showing a man in a white shirt speaking. At the bottom of the slide, there are two logos: the Indian Institute of Technology (IIT) logo on the left and the NPTEL logo on the right.

Now, these are the keywords which we are going to discuss today in this lecture, the CART, also recursive partitioning, then entropy, Gini Index and Pruning. So, let us start with the classification and regression tree.

(Refer Slide Time: 04:17)

CLASSIFICATION AND REGRESSION TREES (CART)

- GOAL: classify or predict an outcome based on a set of predictors
- Output: a set of rules

The slide features a dark blue header with the title 'CLASSIFICATION AND REGRESSION TREES (CART)'. Below the header, two bullet points are listed in blue text. The first bullet point is 'GOAL: classify or predict an outcome based on a set of predictors', with 'classify or predict an outcome based on a set of predictors' underlined. The second bullet point is 'Output: a set of rules', with 'a set of rules' underlined. Below the text, there is a large decision tree diagram with nodes and branches. At the bottom of the tree, there are several bar charts showing the distribution of data points for different classes (A, C, F) at various nodes. In the bottom right corner, there is a circular inset showing a man in a white shirt speaking. At the bottom of the slide, there are two logos: the Indian Institute of Technology (IIT) logo on the left and the NPTEL logo on the right.

So, the classification regression trees, which we can use the short form that is CART. It is an important machine-learning algorithm for classifying or predicting an outcome. When our target variable is a categorical variable, then this tree is known as the classification tree. And when our target is a numerical variable, then we call it or continuous variable, then we call it a regression tree.

The approaches are same, however, the final output calculation is somewhat different, we are going to also discuss that. So, here the goal of classification and regression trees is to classify or predict an outcome based on the set of predictors. In case of regression tree, we use to predict outcome, whereas, in case of classification tree, we use to classify that target. So, what are the outputs from this regression tree or CART algorithm?

The output of this algorithm are a set of rules which we can use to predict or define any particular output. For example, here you can see, it is a classification tree, for the sake of simplicity I am just explaining this and we are using different element like Zinc, Potassium Zircon, Lead, copper, then Potassium, Manganese, to segregate soil samples coming from three different land use PET, land use class.

So, these three different land use class are, A stands for agriculture and C stands for converted land and F stands for forest area. So, you can see that using some set of rules at different points, we are getting some splits and using getting and these splits are continuing until these final endpoints are the squares, where no further splitting is possible.

So, this is called the classification tree. Similarly, there will be a regression tree, where the final output is some continuous numerical variable. Now, we will see the principle of this classification regression tree, in our coming slides.

(Refer Slide Time: 07:17)

CLASSIFICATION AND REGRESSION TREES (CART)

- Example:
- Goal: classify a record as "will belong to class a" or "will not belong to class a" Rule might be "IF (Zn>91) AND (Zr<328)" THEN Class = a
- Also called CART, Decision Trees, or just Trees
- Rules are represented by tree diagrams

But let us discuss them, what is our goal for example. You can see, our goal is to classify a record as whether the sample belongs to Class A or will not belong to class A. So, here if we

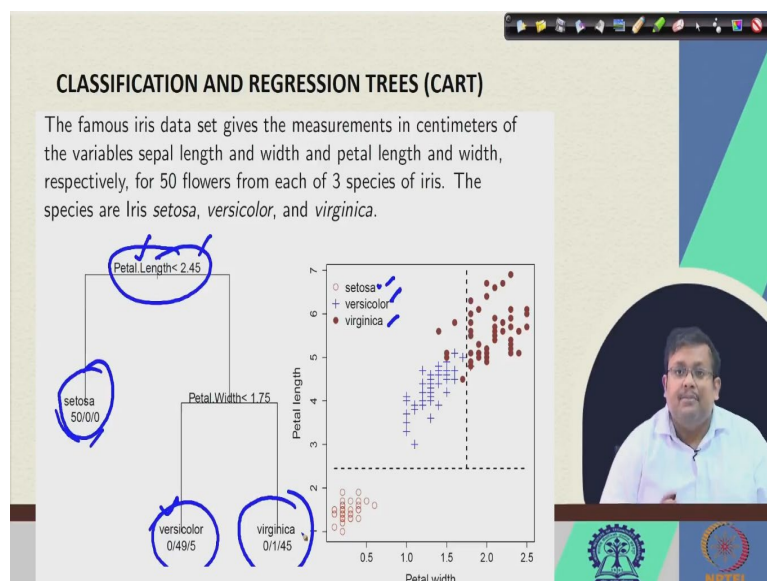
see, this is the first important feature which is basically segregating the samples. So, we can see if the Zinc content is greater than point, if the Zinc content is greater than equal to 91 ppm, and if the Zirconium content of the soil is less than 328 ppm, then we can get the majority of the class as the agricultural samples.

So, you can see these rule goes this way. So, you can see using the IF and AND algorithm rules, we can we can define any specific class. Similarly, for all other terminal nodes, these are called the terminal nodes or leaves, where these are not further split, we cannot split this farther. So, these are known as the terminal nodes. So, you can define these terminal nodes using a set of rules, okay. So, this is the classification regression tree in a nutshell.

So, this classification regression tree is also known as the CART decision tree or sometimes only just tree. So, you can see the rules are represented by tree diagram. Similarly, here you can see, if the Zinc content is less than 91, less than equal to 91 and the Zirconium content is greater than that 328 and if the Potassium content is greater than 35196 ppm, then we will get, most of the soil samples will belong to these converted land use region.

So, this is how we can define any output based on the set of the rules and you can see, these splitting are going on, at different levels, based on certain criteria's which we are going to discuss. Sometime they are no splitting and some time, we can see, there are continuous splitting going on until we reach this final terminal nodes. So, what are the consideration for getting this type of splitting, we are going to discuss in our next slides.

(Refer Slide Time: 10:14)



So, if we see another very good example of CART using the famous Iris dataset, which gives the measurement in centimeter of the variable sepal length and width of the petal length and, petal length and width respectively for 50 flowers, each from three different varieties. What are these three different varieties? These three different varieties of Setosa, then Versicolor and Virginica.

And once we use this classification tree, since these are the categorical variables, so, we are using the classification tree. So, you can see at the first split the petal length, based on the petal length, which is less than 2.545 centimeter, you will see that all 50 samples will be classified as the setosa samples. So, this is the first splitting criteria, petal length combination of the feature as well as the value.

So, the here the feature is petal length and the value is 2.45. So, this combination of feature and its value will segregate these will, will cluster the Setosa, Setosa grouped into one cluster and here you can see, in the second splitting, based on the petal width, which is less than we do petal width of 1.75, we are getting two more classes.

So, when the petal width is less than 1.75, we are getting most of the samples classified as versicolor and when the petal width is greater than 1.75, we are getting most of the samples classified as Virginica. So, this is how we are getting, this is called the recursive partitioning, continuous partitioning, recurring partitioning, until we reach a point where no further splitting is possible.

There are a set of rules where we should stop this type of partitioning and we are going to discuss in our coming slides.

(Refer Slide Time: 12:23)

CLASSIFICATION AND REGRESSION TREES (CART)

- ▶ Tree partitions the feature space into a set of rectangles, and fit a simple model in each leaf (terminal node).
 - ▶ a constant in regression
 - ▶ a class in classification
- ▶ A tree is constructed in two steps:
 - ▶ growing: binary split on each region repeatedly.
 - ▶ pruning: weakest link pruning (collapse internal node).
- ▶ A key advantage of the CART is its interpretability. The feature space partition is fully described by a single tree (a nice graphical representation).
- ▶ CART is implemented in `rpart` library in R.
- ▶ A major competitor of CART is C5.0

So, tree partitioning, the tree partitioning is a phenomenon where the feature space is getting partitioned into a set of rectangles and fit a sample model in each of the leaf or terminal node. So, let us see, let us consider that there is a feature space and here we have y we have x and we have number of samples, two different types of samples, one is hollow, another is solid.

And in the first iteration, what this classification CART algorithm basically tries? It tries to find out the combination of the variables and their values. Suppose, there is a value of S . So, this combination of x variable and S value can separate out this, this whole rectangle, into two dotted rectangles or two terminal nodes, okay.

Then, remember these terminal nodes are selected based on the homogeneity among each terminal nodes and among each classes. It is not necessarily that in these data terminal node there will be equal distribution of the samples. No, it is not necessary. So, in the second split, you can see, based on another y and another value of y , we can see another split. So, in this next iteration, you can see there may be another split here.

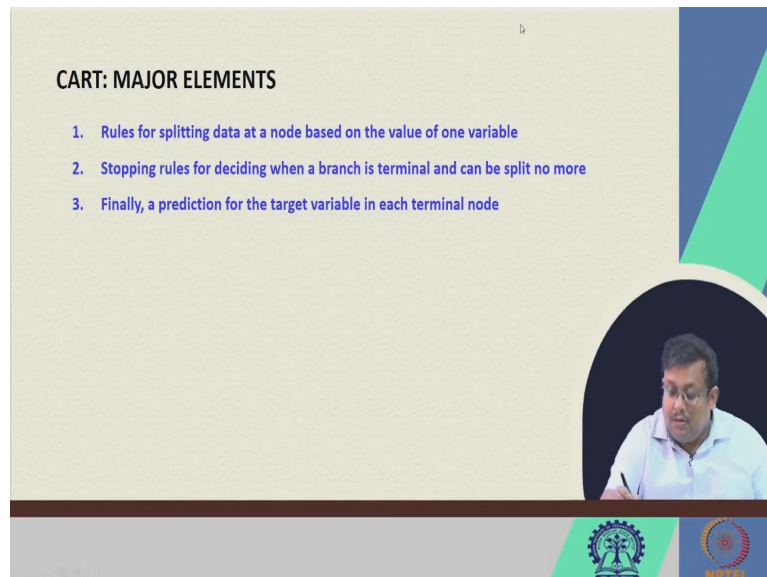
So, this type of splitting will continuously go on and ultimately you can see here 1, 2, 3, 4 ultimately rectangles are there, based on the recursive splitting. So, there will be four terminal nodes and the splitting in each point is going on, based on certain criteria which we are going to discuss by measuring some purity measures or in other words by measuring some impurity measures, which we are going to discuss.

Now, so, these are the three partition in the feature space is set of rectangles and if it is simple model in each of the each leaf, so, in case of a regression, there is a constant and in case of a classification the final output is the classification. So, a tree will generally, tree growth is generally content, if there are generally two steps is being done, one is growing by binary splitting of the each region repeatedly.

You can see that when the CART algorithm that data is getting binary splitted, okay. One is higher of that splitting value; another is lower than the splitting value. And pruning is the weakest link, pruning which is collapsing of the internal node. We are going to discuss what is pruning in our upcoming slides.

So, a key advantage of classification regression tree or CART, it is interpretability and the feature space partitioning is fully described by a single tree in a graphical representation, okay. So, in our software, we can use the rpart library, to generate this CART algorithm. And another major competitor of CART is C5 algorithm, which we are going to discuss in our DSM, Digital Soil Mapping module.

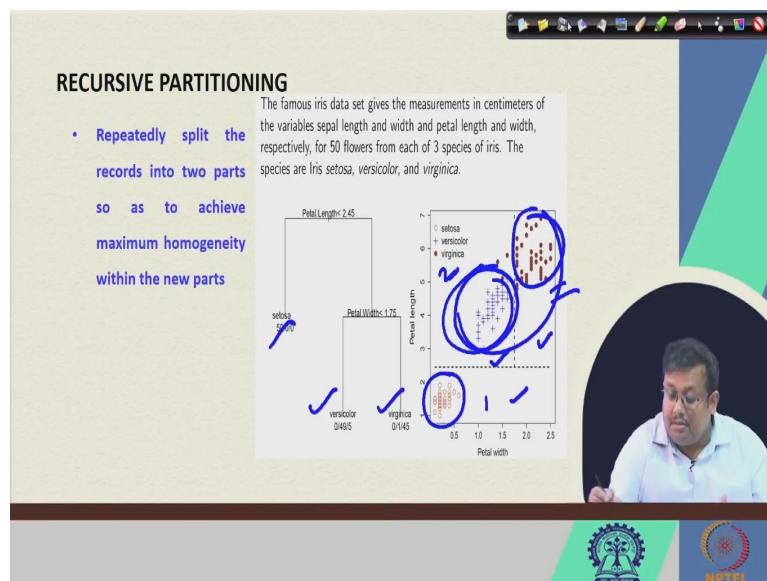
(Refer Slide Time: 16:16)



So, what are the major elements in case of a CART? There are three major elements in the class. First of all rules for splitting data at a node based on the values of one variable, and secondly stopping rules for deciding when a branch is terminal and can be split no more. Three major consideration and the final consideration is a prediction for the target variable each of the terminal node.

So, you can see here, first, we are talking about rules for splitting the data at a node based on the value of one variable we have already seen, then we can see, how to stop this type of recursive partitioning when a branch is terminal and can be split no more, we are going to also discuss that. And finally, the prediction of the target variable in each of the terminal node, when no further splitting is possible in the terminal node then we predict the target variable in each of the terminal node or leaf.

(Refer Slide Time: 17:25)



Now, here we can see here, this is the example of recursive partitioning in case of this Iris dataset. We know we can see that the CART algorithm is repeatedly splitting the records into two parts, so as to achieve maximum homogeneity within the new parts. I have already told you, while I was showing you the, I was showing you the rectangle, here you can see, this is the another rectangle, so, here two features are there, petal width and petal length, okay.

So, these are two features in earlier, there are two features x and y or x_1 and x_2 , you can think of anything, either x_1 or x_2 or X and Y maybe two features. So, here also you can see, petal width and petal length are two features. So, here based on at the first split, we are here, the petal length is the first and petal length of 2.45.

So, if you see the petal length of 2.45, it is distributing the samples into two major categories here. So, this is the Setosa class and other two classes are in the other part that is when the petal length is greater than 2.45. And in the second split again you can see just like I have showed you, based on the petal width of 1.75, we are again classifying the samples into two data nodes or two terminal nodes.

So, ultimately we are getting here 1, 2, 3 terminal node, 1, 2, 3 terminal nodes. And this separation is going on or binary splitting is going on, based on the maximum homogeneity, you can see here in each of these terminal nodes, one class is maximally homogeneous. Here, in this, if we consider this the first terminal node, here the setosa is more homogeneous.

In the second terminal node, this versicolor is more homogeneous and the third node or third terminal node, the virginica is more homogeneous. So based on these relative homogeneity, these the samples are divided into several classes or recursively.

(Refer Slide Time: 20:00)

TREE BUILDING BY RECURSIVE PARTITIONING

1. Pick one of the predictor variables, x_i
2. Pick a value of x_i , say s_i , that divides the training data into two (not necessarily equal) portions
3. Measure how "pure" or homogeneous each of the resulting portions are
4. "Pure" = containing records of mostly one class
5. Algorithm tries different values of x_i , and s_i to maximize purity in initial split
6. After you get a "maximum purity" split, repeat the process for a second split, and so on

The slide features a video inset of a man in a light blue shirt speaking. At the bottom, there are logos for IIT Bombay and NPTEL.

So, how these trees are being built actually. So, as I have told you, I have showed you. So, first you have to pick one variable that is x_i and then pick a value of the variable say a s_i that divides the training data into two, remember as I have mentioned it that data or the terminal nodes for these binary splitting should not be necessarily equal, in equal portion.

So, it may not be necessarily equal splitting. So, basically in each of these split, it basically measured, how pure or homogeneous each of these resulting portions are? We can see in case of Iris dataset, how homogeneous the, how the splitting was going on, based on the relative homogeneity of three different types of samples. So, what is pure? Pure means, containing the records of mostly one class.

Now, here algorithm tries different values of x_i and s_i , to maximize the purity on the initial state. So, whatever features are there, this CART algorithm will try to consider all the features and they are combination with their values, so, that they can identify the best

variable or feature and their value combination to get, to give us the maximum purity in the initial split and so on so forth.

So, after you get the maximum purity split, repeat the process for a second split and so on. And this is how you do the recursive partitioning in a CART algorithm.

(Refer Slide Time: 21:41)

CART IMPURITY METRICS

- **ENTROPY:** It is used to measure the impurity or randomness of a dataset.

Imagine choosing a yellow ball from a box of just yellow balls (say 100 yellow balls). Then this box is said to have 0 entropy which implies 0 impurity or total purity.

$$\text{entropy}(A) = - \sum_{k=1}^m p_k \log_2(p_k)$$

p = proportion of cases (out of m) in rectangle A that belong to class k
Entropy ranges between 0 (most pure) and $\log_2(m)$ (equal representation of classes)

The slide also features a video inset of a man speaking and logos for IIT Bombay and NPTEL at the bottom.

So, how to compute the purity? It is based on certain metrics; the first important metrics is the Entropy. What is Entropy? Entropy is used to measure the impurity of randomness of a data set. Just like in case of entropy in thermodynamics, the entropy is the degree of randomness.

So, similarly, in CART algorithm, it is used to measure the impurity or randomness of the dataset. So, let us consider that if we imagine choosing a yellow ball from a box of just yellow ball, say there are 100 yellow balls and we want to select the some yellow balls from those all the 100 yellow balls then these boxes say to have zero entropy, which implies zero impurity or total purity, because all the balls are yellow balls.

But, if we replace 50 balls, yellow balls with 50 blue balls and then we want to see the probability of selecting yellow ball then of course, the entropy will go down. Because, then that way there is no total there is no zero entropy, there are randomness, when you are introducing some other types of balls.

So, this is called the entropy and entropy we measure by using this formula, this is the formula of entropy, which is denoted by A which is minus summation of k equal to 1 to m

which is $p_k \log_2 \frac{1}{p_k}$. So, p is here is the proportion of the cases out of m , in rectangle A that belongs to class k and entropy changes between 0, which is most pure which we have already seen, and $\log_2 \frac{1}{m}$, which is showing the equal representation of the classes.

So, this is how we calculate, this is an important impurity metrics. Another important metrics for CART algorithm is Gini Index.

(Refer Slide Time: 23:55)

CART IMPURITY METRICS

- Gini Index: It is calculated by subtracting the sum of squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values

$$I(A) = 1 - \sum_{k=1}^m p_k^2$$

p = proportion of cases in rectangle A that belong to class k

- $I(A) = 0$ when all cases belong to same class
- Max value when all classes are equally represented (= 0.50 in binary case)

So, what is Gini Index? It is calculated by subtracting the sum of squared probabilities of each class from 1. So, it favors the larger partitioning and easy to implement, whereas, information gain favors smaller partition with distinct values. So, this is the formula of this CART, Gini impurity or Gini index.



So, it is this is the formula where p is the proportion of the cases in rectangle A that belongs to class k , where these this Gini index will be 0 when all class belong to the same class and maximum values it will assume when all classes are equally represented in case of binary case it will be 0.50. So, this is how we calculate these entropy and Gini index in during these recursive partitioning of CART.

(Refer Slide Time: 24:53)

HOW IMPURITY IMPACTS RECURSIVE PARTITIONING?

1. Obtain overall impurity measure (weighted avg. of individual rectangles)
2. At each successive stage, compare impurity measure across all possible splits in all features
3. Select the split that reduces impurity the most
4. Selected split points become nodes on the tree



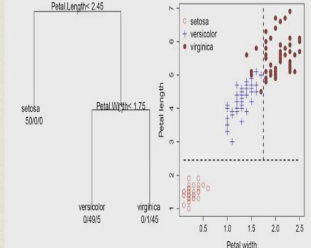
► Where to split? Find the variable j and splitting point s to minimize

$$\sum_{x_i \in R_1(j,s)} (y_i - \hat{c}_1)^2 + \sum_{x_i \in R_2(j,s)} (y_i - \hat{c}_2)^2$$


RECURSIVE PARTITIONING

The famous iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

- Repeatedly split the records into two parts so as to achieve maximum homogeneity within the new parts

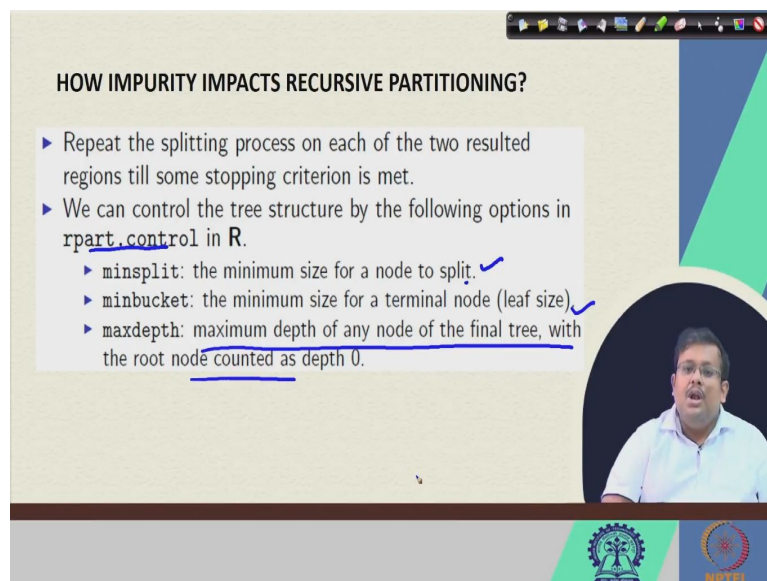


And how these impurity impacts is recursive partitioning? This is another question. So, We know we can identify the, obtain the overall impurity measure, which is the weighted average of the individual rectangles and then at each successive stage, we can compare the impurity measures across all possible split in all the features and then we select the split that reduces the impurity most and then select the split points become nodes of the tree.

So, the splitting points are becoming the nodes, we call them nodes. So, first of all in the first step, we in each split, we obtain the overall impurity of the measures which is a weighted average, then at each successive stage, we compare the impurity measures across all possible split, in all the features with their variable, with their values and select the split that reduces the impurity most and then select that splitting point will become the node of the tree.

So, if we go back, these are the nodes of the tree. So, this is one node this is another node, so where the tree divides or branches are coming out that is called the nodes of the trees, okay. Where to split? So, this this is the mathematical representation, which showing, reducing the impurity in each of the classes by will increase the purity in each of the, each stages of splitting. So, this is a mathematical representation. We have to find the variable j and the splitting point at value s to minimize this total mathematical expression, okay.

(Refer Slide Time: 26:40)



HOW IMPURITY IMPACTS RECURSIVE PARTITIONING?

- ▶ Repeat the splitting process on each of the two resulted regions till some stopping criterion is met.
- ▶ We can control the tree structure by the following options in rpart.control in R.
 - ▶ minsplit: the minimum size for a node to split. ✓
 - ▶ minbucket: the minimum size for a terminal node (leaf size) ✓
 - ▶ maxdepth: maximum depth of any node of the final tree, with the root node counted as depth 0.

The slide includes a video inset of a man speaking and logos for IIT Bombay and NPTEL at the bottom.

So, how impurity impacts recursive partitioning, we know that. We were we recursively partition or repeat the splitting process on each of the two resulted region till some, we know, we will go on partitioning these, this data set, and then ultimately we will stop at a point where some stopping criteria are met.

So, in our software, we generally use the following options called `rpart.control`, you know, this is an argument which we use in our to assign some stopping criteria. So, minimum split is the minimum size for a node to split and then minimum bucket is the minimum size for a terminal node which is leaf size and maximum depth is maximum depth of any node on the final tree with the node, root node set counted as 0.

So, if we can fix the maximum depth, the tree will go up to that depth and then it will stop. Minimum buckets, suppose, we assign the minimum size of a terminal node or leaf size then also the CART will stop once that number of terminal nodes m_{min} reached. So, as m_{min} reached. And finally, minimum split is also the minimum size for not to split. We can assign the minimum number of samples, which will be required for further splitting.

So, if we can specify these criteria, you know, the CART will not go indefinitely and it will stop at a particular point based on these features. So, why we need these kind of features, because we are using these we can prevent overfitting, because, if we do not give this stopping criteria, CART can perfectly create a rule for each of the samples, which are present in the dataset and in that way, it will train so, it will be trained on the noises of the data.

So, it becomes over fitted. So, that so, to prevent that overfitting, we go with this type of stopping criteria.

(Refer Slide Time: 29:06)

TREE STOPPING CRITERIA

1. Maximum tree depth
2. Only one case is left in a node
3. All other cases are duplicates of each other
4. The node is pure (all target values agree)

The slide is displayed in a video player interface. At the bottom right, there is a circular video feed of a man in a white shirt speaking. At the bottom of the slide, there are two logos: the NPTL logo on the left and the NPTL logo on the right.

Other stopping criteria as I have already mentioned, maximum tree depth, then when the only one case is left in the node, also all other cases are duplicates of each other and finally, the node is pure. So, all target values, when they agree, then this is the another stopping criteria. So, these are the stopping criteria, which we can apply in the CART to stop the recursive partitioning at some stage.

(Refer Slide Time: 29:37)

TREE PRUNING

- Simplify the tree by pruning peripheral branches to avoid overfitting

<https://www.maxixel.net/Tree-Tree-Cutter-Hedge-Trimmer-Pruning-Shears-4964455>

The slide features a central image of hands using pruning shears on a tree branch. A video inset in the bottom right shows a man speaking. Logos for IIT Bombay and NPTEL are at the bottom.

Now, what is pruning suppose, we have allowed the tree to grow into his full extent. And when we see a tree is grown to a full extent, there is always chance of getting overfitting. So, to reduce the chance of overfitting, we can simplify the tree by pruning the peripheral branches to avoid the overfitting.

So, we can reduce the number of peripheral branches just we have seen in case of gardening, when a tree grows indefinitely, we cut their terminal by peripheral branches to reduce the overfitting. So, similarly, here also we can do the overall you know pruning to simplify that to reduce or to avoid the overfitting.

(Refer Slide Time: 30:23)

ADANTAGES AND DISADVANTAGES OF CART

- ▶ Very easy to explain (IF/AND/THEN) to people (even easier to explain than linear models!)
- ▶ Some believe that decision trees mirror human decision-making.
- ▶ Trees can be displayed graphically, and are easily interpreted even by a non-expert (especially if they are small).
- ▶ Easily handle qualitative predictors without the need to create dummy variables.
- ▶ Can naturally handle the missing values in the predictors.
- ▶ Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches.

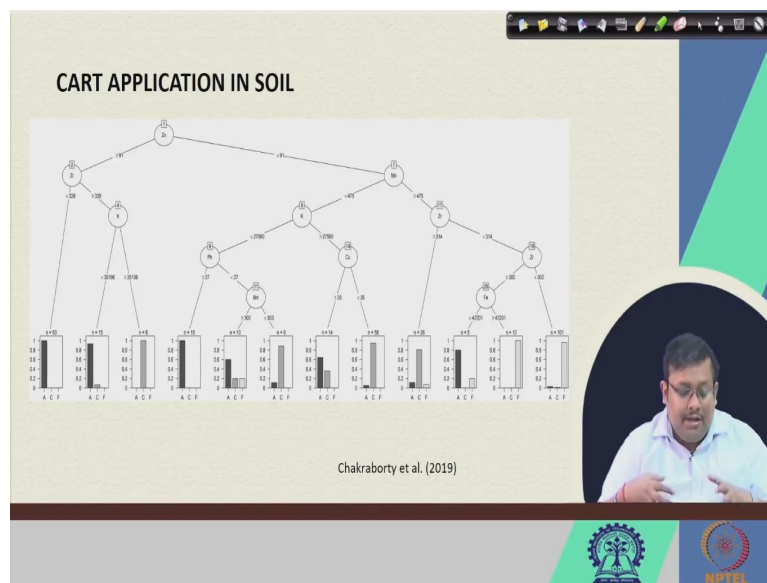
The slide features a video inset in the bottom right showing a man speaking. Logos for IIT Bombay and NPTEL are at the bottom.

What are the advantages and disadvantages? First of all, it is very easy to explain using these if and then rule, to people even easier to explain than the linear models. Some believe that decision tree mirror the human decision-making. It likes the human, how humans are making the decision, making some rules and step by step.

And then trees can be displayed graphically and are easily interpreted by a non-expert, especially, if they are small and then it can easily handle quantitative predictors without the need of creating any dummy variables and can naturally handle the missing values in the predictors entries generally do not have same level of predictive accuracy.

But, this is the only demerit of trees; the trees do not have the same level of predictive accuracy as some of the other some of the other regression classification approaches. So, these are some of the advantages and disadvantages of the tree.

(Refer Slide Time: 31:23)



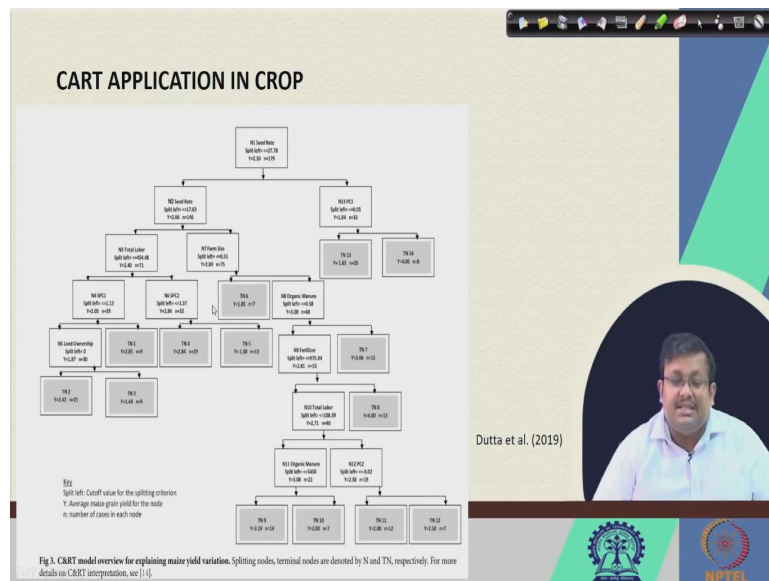
Let us see some example of classification regression tree application in soil. We have already discussed, how we have classified around 500 number of samples, you know one of our projects using this classification tree algorithm to. So, these are the splitting nodes, these are the nodes of the tree and these are the terminal nodes and you can see in the terminal nodes, how the samples coming from three different classes are distributed.

So, while we know based on certain stopping criteria, the fact that no further splitting was possible, and ultimately, we are able to define the soils coming from different land use by

using a set of rules and then set of rules you know, are really helpful for identifying the samples coming from different land use patterns.

So, if we can set we can if we can set up certain rules if we can define certain rules to divide the soil samples into different land use Styles, in future we can also bring some unknown samples and using the set of rules you can getting their elemental values and using that setup rules, we can also classify them into one of these three categories.

(Refer Slide Time: 32:38)



Another way, crop application I would like to show you here, when the maize crop, we have applied this classification and regression tree and we have applied different types of variables, not only the socio economic variables, we have also used some management variables, agronomic variables and soil variables to classify the crop yield into.

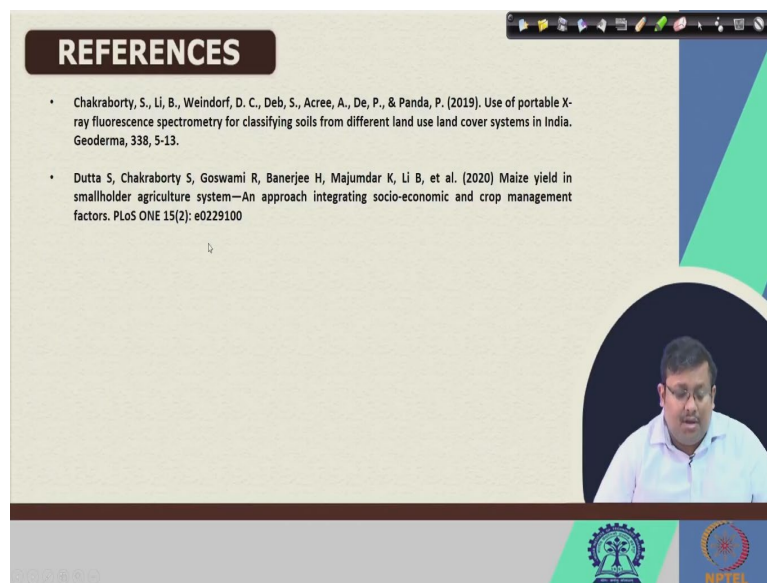
So, this is an example of, this is the perfect example of the decision tree where you can see the terminal nodes, our target is already given, y values are given, a number of samples are given, an each node you can see the splitting based on certain features and their value. For example, in this node, you can see splitting node; you can see seed rate is one of the major features.

So, first seed rate was a major feature with a value of 27.78 and then when it is less than 27.78, we are getting to further split. And in the second level, we can see here based on the principal component 1 of soil spectral data, we will divide them into again two terminal nodes. So, similarly, these are the final terminology. So, this type of algorithm helps us to

identify, what are the features and what are their values, which are very much important to define or to predict the ultimate the crop yield.

So, these type of applications have been done, you know, lots of application you can see in the literature, where people are using a scientist and using this CART algorithm for predicting the crop yield predicting the soil properties then we call it a decision tree. You will see they are using the decision tree model. So, these type of applications are there.

(Refer Slide Time: 34:34)



REFERENCES

- Chakraborty, S., Li, B., Weindorf, D. C., Deb, S., Acree, A., De, P., & Panda, P. (2019). Use of portable X-ray fluorescence spectrometry for classifying soils from different land use land cover systems in India. *Geoderma*, 338, 5-13.
- Dutta S, Chakraborty S, Goswami R, Banerjee H, Majumdar K, Li B, et al. (2020) Maize yield in smallholder agriculture system—An approach integrating socio-economic and crop management factors. *PLoS ONE* 15(2): e0229100

So, these are the references which I have used in this lecture. So, thank you. I hope that you have gathered some knowledge about CART and how it operates what are the basic features of CART and we have seen one or two examples of CART application on soil and crop based studies. So, let us stop here and in our next lecture we will discuss some more important machine learning algorithms for soil and crop application. Thank you.