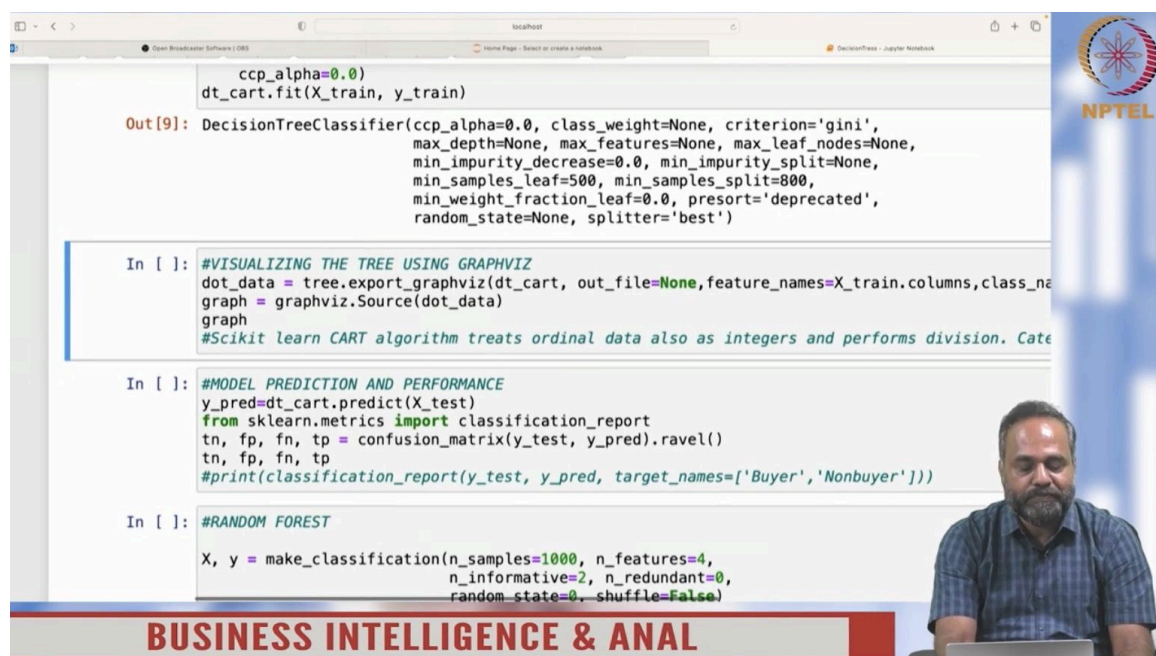


Course Name: Business Intelligence and Analytics
Professor Name: Prof. Saji.K.Mathew
Department Name: Department of Management Studies
Institute Name: Indian Institute of Technology Madras
Week: 08
Lecture: 29

DECISION TREE APPLICATION PART 2



```
ccp_alpha=0.0)
dt_cart.fit(X_train, y_train)

Out [9]: DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                               max_depth=None, max_features=None, max_leaf_nodes=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=500, min_samples_split=800,
                               min_weight_fraction_leaf=0.0, presort='deprecated',
                               random_state=None, splitter='best')
```

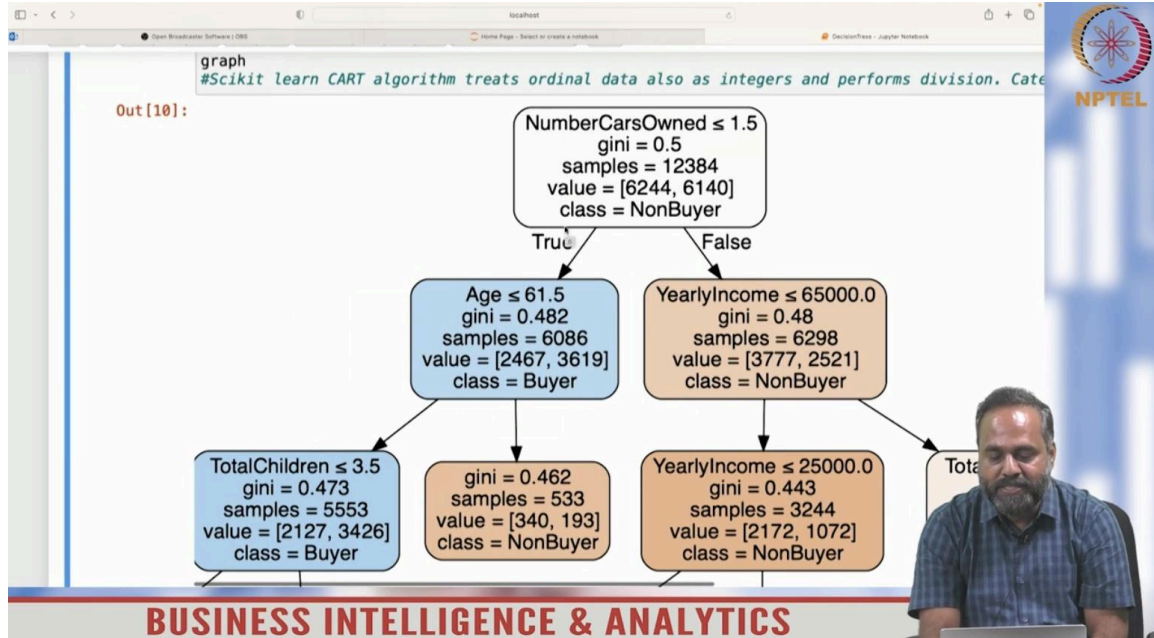
```
In [ ]: #VISUALIZING THE TREE USING GRAPHVIZ
dot_data = tree.export_graphviz(dt_cart, out_file=None, feature_names=X_train.columns, class_name=
graph = graphviz.Source(dot_data)
graph
#Scikit learn CART algorithm treats ordinal data also as integers and performs division. Cate

In [ ]: #MODEL PREDICTION AND PERFORMANCE
y_pred=dt_cart.predict(X_test)
from sklearn.metrics import classification_report
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
tn, fp, fn, tp
#print(classification_report(y_test, y_pred, target_names=['Buyer', 'Nonbuyer']))

In [ ]: #RANDOM FOREST
X, y = make_classification(n_samples=1000, n_features=4,
                          n_informative=2, n_redundant=0,
                          random_state=0, shuffle=False)
```

BUSINESS INTELLIGENCE & ANAL

And well, you have built a tree, you have trained a tree. Training, building all that means you are using only training data, you have not predicted yet, you have built a model, you have fit data on to a decision tree model. Decision tree has now already extracted rules, that is what has happened now. So, let us explore the tree now, the graph is, basically is for a good colorful visualization of the tree and it looks very nice. And that is only reason why I use graph, is because many of the other visualizations are very dry, but it has problem also. So, what you see here is the tree, you see it is all binary tree, in the sense binary tree splits and it is split from top or the root node down to several layers down, but it is, the tree is not too large, it is visualizable, it is a few levels.



And you can see that sometimes the tree splitting stops and sometimes it continues to split. Depending on the size of the leaf that is formed or terminal or sorry, size of the node that is formed. We have given that criteria 800 and 500. So, that is always kept. So, and it also colors the different nodes.

What is the color scheme based on? What does a deep blue indicate here? Class is buyer, if a blue classes or blue groups are all buyer groups where the probability of 1 is greater than 0.5, that is what you can imagine. And when it comes to brown the probability is less or these are 0 classes or non buyer classes. And the size and the proportion of each class in the group is also given. So, we calculated Gini score in the class.

You can see the size of the node and proportion of each class label in the node and from that you can see this also reports the Gini value, Gini score which is given here as 0.473. Is 0.473 a good Gini score? 0.47. Let us see some good scores as we go down, 0.413. When it comes to 0.364, the nodes becomes purer and purer as you go down, Gini is 0.364 it becomes deeper blue.

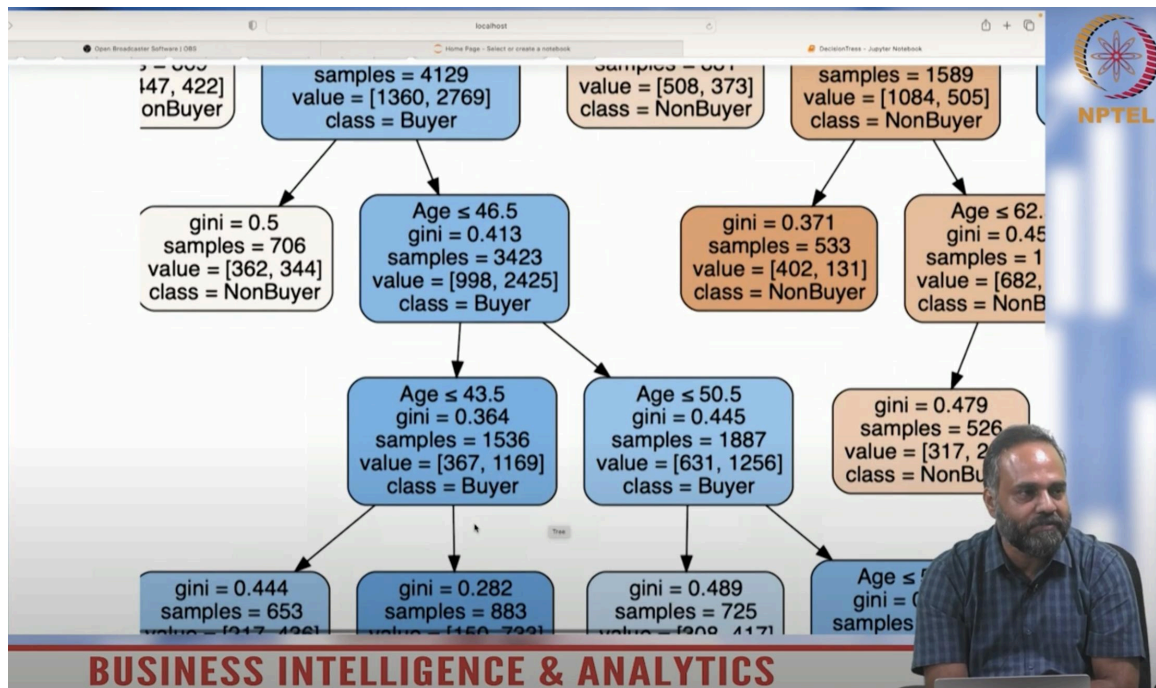
And when it becomes 0.282 it is much deeper or much purer. And this shows that this has more buyers in it than non buyers. 733 buyers and 150 non buyers. It is a predominantly or the probability of buyers in this particular group is quite high or this is a useful rule.

This is a good rule. This leads to buyers. And the criteria for split is of course, is given

at the top of each node description. So, you can interpret this tree or you can understand this tree and if you are given time, you will also write what are the rules. For example, what are the rules for the best leaf nodes, two best leaf nodes.

Then you will look for a Gini score, lowest Gini score and try to describe the rule and then you will know that the rule. One rule is age is less than or equal to 43.5. Yes, if age is less than 43.5, that is a good rule, but that is not the rule alone.

And it is the same. Yearly income less than 25000, that rule comes again age is used. That is the way the split happens, it may reuse the same variable. Total children criteria, number of cars owned criteria. When you put all these rules together it is a composite rule, many conditions that leads to a bike buyer, bike buyer subgroup.



So essentially you can see that each node has different sizes, each node also has different probabilities. You can also describe a node by its probability. Gini score is one measure of purity. Can you describe this node by its probability? Probability of buyers in this node or probability of buyers for this rule. Just look at this and tell me immediately.

Come on 733 buyers out of 883, 733/ 883. That is the probability score for the particular node. We talked about scoring model. This is actually scoring also, but it is a scoring. It is scoring a group.

Each member of this group get the same score. So, there are 883 members here. They all

get the same probability score. You have visualized the tree. This is the interpretation of the tree in short summary.

Now my next step, this is what I said some of this codes I wrote this time for you. I wrote means I used scikit-learn libraries to sort of do something more with the results we got. First thing is you can actually predict using this model. This is a model we built. Rules have been extracted.

Now my next step is, well I want to test how good a predictor is this model or how what is the prediction performance of this model. So, you can see that in the next module or in the next cell what am I doing? `y_pred = dt_cart.predict(x_test)`. That is my model. I have named this model as `dt_cart` previously.

You have seen that. This is my `dt_cart`. `dt_cart` is defined here, after I specified the model finally with the `CCP` alpha. So, I am predicting I am using it for predicting. So, prediction data is getting stored in a array `y_pred = dt_cart.predict(x_test)`. I extracted the test data already.

I am inputting the `x_test` data and asking to create the `y_pred` or the \hat{y} (y cap). You understood right? And here is the test of the model. When you do the prediction, how well it has predicted with respect to the actual `y` data. I do not know if you are following this. You have `y_pred` which I am generating out of the model.

You also have `y_test` which is your actual test data. Then you are going to compare the test data and the predicted data and see how accurate is the prediction in terms of the accuracy matrix or the different measures of accuracy. See so, all this is done by the confusion matrix function. We just call that function gave this arguments. It has generated the true positive, false positive, true negative and false negative.

That is done by a function. And now, with this now you can calculate the accuracy, error, sensitivity, specificity, recall and so on. You can manually calculate it using python or you can automate it. So far so good. So, if you have to calculate sensitivity, what is the sensitivity? Look at the data and can you tell me what is the sensitivity? Data has come in a particular order `tn, fp, fn, tp`. Tell me what is the sensitivity? You are not recalling anything.

15, 12 divided by, that is true positive divided by, come on should not take time. What is the denominator? Denominator is `p`, total `p`'s. Total `p`'s is what? `tp + fn`. Excellent. `fn` is nothing but positive which got misclassified.

$tp + fn$. So, that is your sensitivity. So, you can compute all this from this data. Now in the next module I am building a random forest. We discussed what is a random forest.

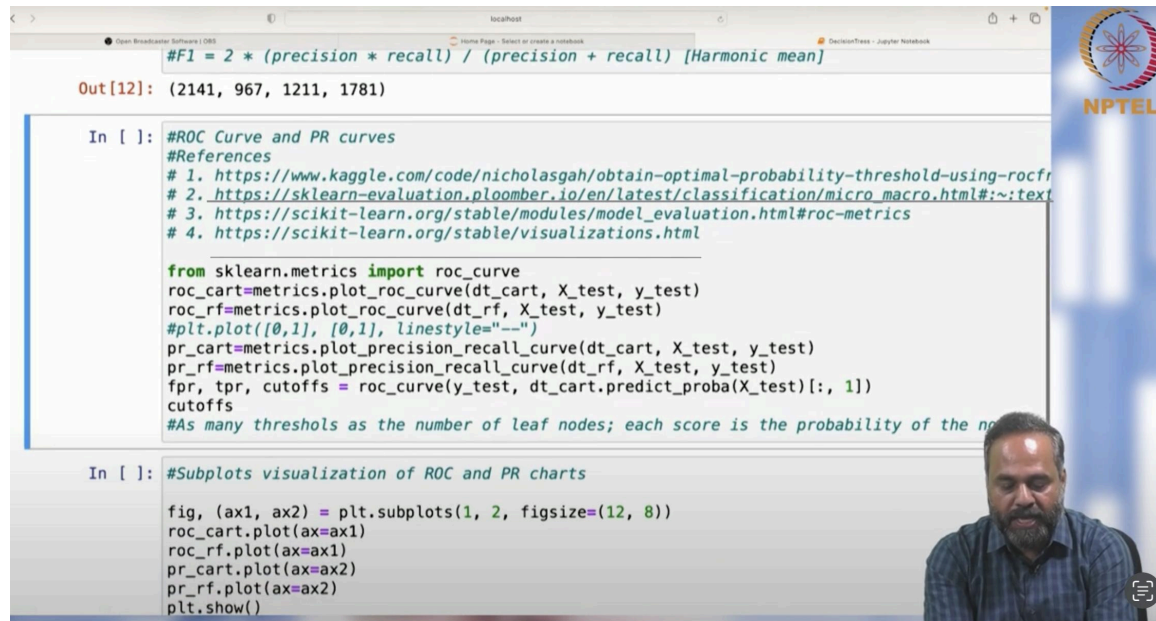
Its algorithm is different. But random forest generally outperforms R-CART, sorry, CART algorithm. So, the procedure is the same. You just have to call a different function called random forest classifier. And main classification will do the bootstrapping. Bootstrapping is done first and then random forest classifier is specified and then the fitting is done.

That is all. Just run this and here you get the confusion matrix for the random forest. And of course, by looking at the data you cannot actually say which is a better classifier. You have to actually work on the, you have to work on the specific measures of accuracy, specific measures of classifier performance. Now you have both the data here.

Let me give you a 5 minutes work. Look at this data and tell me which model is useful for predicting credit risk. Which model would you deploy for credit risk? If your problem comes from credit risk, would random forest be better or CART algorithm be better? You have the confusion matrix here. Of course, you have to justify the answer, you have to use the right measure. Which model would you use for credit scoring or credit application? We discussed this already in the class, medical diagnosis versus financial credits. What would you minimize here in the case of credit? What is something that you do not desire at all? True positives and true negatives are always good.

You do not have to look at that, but you are worried about false reporting, right? Misclassification is what you are worried. So, false positives and false negatives you do not want, but which is a greater evil for credit? False positive, you do not want false positive. So, which measure will have false positive in it? Specificity. In this case, one is predicting a customer who is a buyer. One is what you are targeting, not zero, right? So, you will look at specificity.

Think about the reason as to why you are going to look at specificity not sensitivity. Specificity does not give you direct indication, but $1 - specificity$ is something that is useful. But each of you should be clear about this logic. Otherwise, you cannot use classifiers because the different measures are developed for different reasons.



```
#F1 = 2 * (precision * recall) / (precision + recall) [Harmonic mean]
Out[12]: (2141, 967, 1211, 1781)

In [ ]: #ROC Curve and PR curves
#References
# 1. https://www.kaggle.com/code/nicholasgah/obtain-optimal-probability-threshold-using-rocfr
# 2. https://sklearn-evaluation.ploomber.io/en/latest/classification/micro\_macro.html#:~:text=
# 3. https://scikit-learn.org/stable/modules/model\_evaluation.html#roc-metrics
# 4. https://scikit-learn.org/stable/visualizations.html

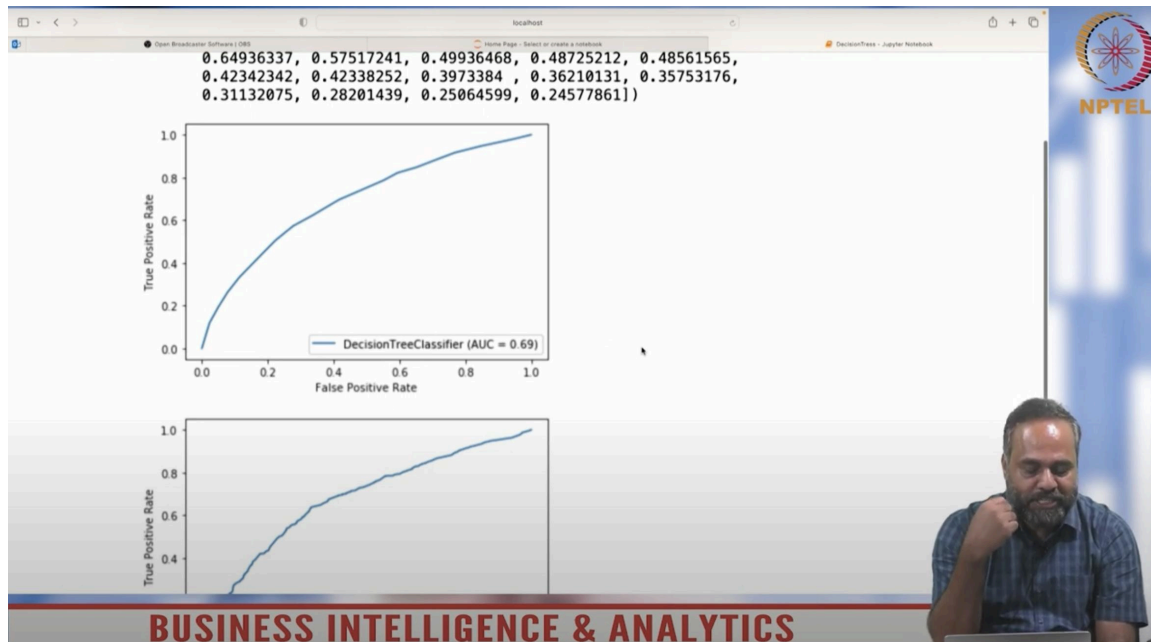
from sklearn.metrics import roc_curve
roc_cart=metrics.plot_roc_curve(dt_cart, X_test, y_test)
roc_rf=metrics.plot_roc_curve(dt_rf, X_test, y_test)
#plt.plot([0,1], [0,1], linestyle="--")
pr_cart=metrics.plot_precision_recall_curve(dt_cart, X_test, y_test)
pr_rf=metrics.plot_precision_recall_curve(dt_rf, X_test, y_test)
fpr, tpr, cutoffs = roc_curve(y_test, dt_cart.predict_proba(X_test)[: , 1])
cutoffs
#As many thresholds as the number of leaf nodes; each score is the probability of the node

In [ ]: #Subplots visualization of ROC and PR charts

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 8))
roc_cart.plot(ax=ax1)
roc_rf.plot(ax=ax1)
pr_cart.plot(ax=ax2)
pr_rf.plot(ax=ax2)
plt.show()
```

Good. Your assignment might help. And these are important topics for your examination. Please keep that also in mind. So, this time as I said, I have worked out some more lessons related to ROC curves and precision recall curves, ROC and precision. So, again there are standard functions available for plotting ROC curve and also plotting PR curve. So, these functions I have directly used here and the references are given right here, as comments you can look at that.

Now here is the key. So, here there are certain outputs of course, the first graphical outputs are easy to interpret. Before you go to the graphs, I wanted to understand what is this array that is displayed here. The array is actually a one result, one output or one return when I executed the function `ROC underscore curve y test dt cart dot predict proba x test`. So, essentially that particular function returns false positive rate, true positive rate and cut offs, probability cut offs. You understand what is true positive rate and false positive rate which we already discussed.



BUSINESS INTELLIGENCE & ANALYTICS

But what is a cut off? Because in order to plot an ROC curve, you need cut off. It is plotted at different cut off values. We discussed this in the class with respect to an example of the TSH value for thyroid problem, right. When you change the cut off, the false positives and true positives change. If you make the sensitivity very high, false positive is very low, then true positive is also very low.

When you make sensitivity relaxed or small, true positives increase, but false positives also increase. That is a phenomenon in classifier. So, therefore, the classifier can perform under different levels of cut offs. Can anyone based on this output correlate what is the output obtained here, in terms of a matrix of probability values with the decision tree that we created or we built sometime back? Can you correlate? Can you relate? But where did this values come from? Somebody has to say, a decision tree classifier generate different rules, correct.

Based on each rule, there is a leaf node. At the end, there is a leaf node. Your output is nothing but leaf nodes. Each leaf node is based on a rule. So, decision tree created a set of leaf nodes.

And we also saw that each leaf node has a probability. Some leaf nodes are deep blue, meaning high probability of buying. Some are deep browns means very low probability of buying. So, we also saw that in targeting, we target those nodes which have higher probability of buying or we pick those nodes which have higher probability of buying. Now, in this particular cut offs matrix, what you get or vector what you get, is the probabilities of different leaf nodes.

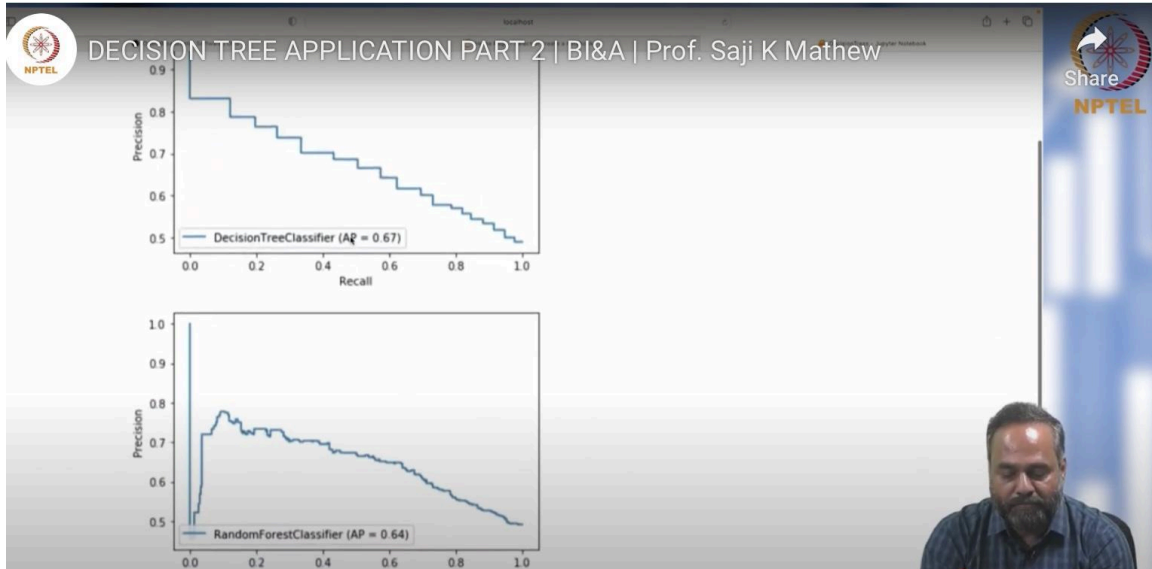
You can compare this each value. If you go and calculate the probability of each node, you will find that it is the probabilities of the leaf nodes that is being used to construct the ROC curve. So then you know if probability is 1, of course, you are the left side of the ROC curve. Probability is 0. You know you are including everything. 0 probability means you will include all the database for your targeting, which means that you know there is not targeting at all and you will be, it is not actually leading to, it leads to all the true positives, but it also bring all the negatives, all the negatives whom you should not have target.

So, our true positives increases, false positives also increase.. Here you get the maximum true positives, but unfortunately all the negatives have also gotten in here. It became the original database at this point. When your cut off is 0, you appreciate this point.

This is where probability is 0. This has included all the elements in, but no use. You have you are just wasting your money by sending your, you know you are targeting all those who are not going to respond to you. So, these two graphs help us understand how these two models are performing. They look alike, you know the cart algorithm and the you know the random forest both of them look alike.

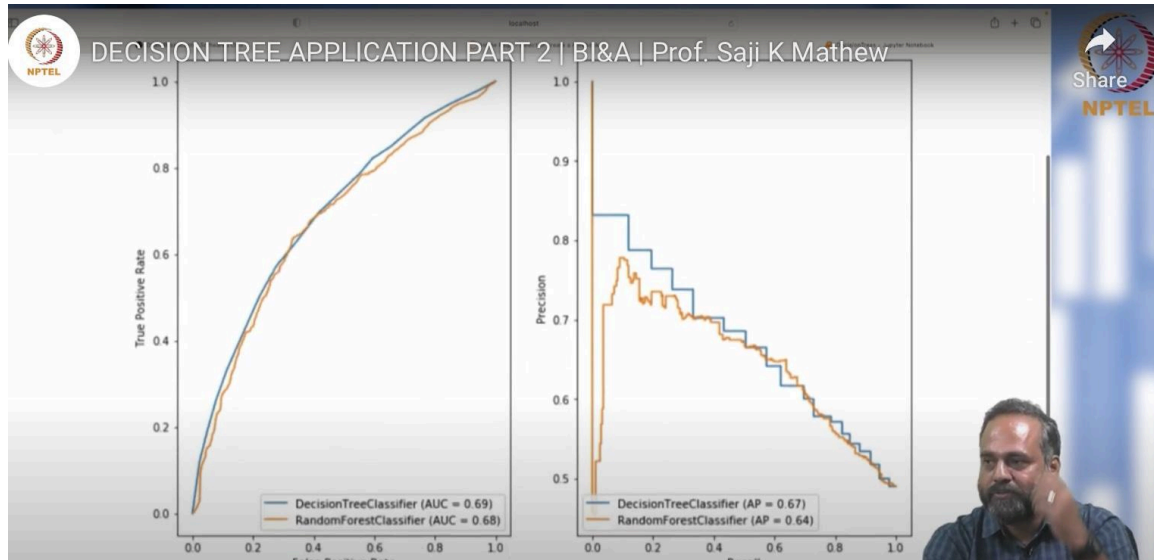
So, yeah. Probabilities The starting point, you have to forgive. It starts with 1. Just truncate it and assume that it is 1. It starts with 1, which is this point.

This is the starting point. It is 1. Probability cannot be more than 1, but this is the starting point. It goes up till 0. So, in terms of cut off, it travels from 1 to 0 in this range and then different false positive rate and true positive rates are plotted. Now, here is the P R curve, precision recall curve.



Precision recall curve gives you another sense. Precision is nothing but sensitivity right. And recall is, recall is true positive divided by true positive plus false positive. False positive is not something that you desire. So, therefore, when recall is having a low value, it shows that you may have some true positive, but a lot of false positives have also come in, right.

So, it is increasing your cost and it is not desirable right. So, what is a desirable shape of the curve for a precision recall curve? Should it be a graph like this or should it be a graph like this? You want it to be budging here or should it be like this? Yeah, the, a graph which should be budging right top, that is what a precision recall curve should be. Whereas, a ROC curve should be left up, those are the shapes. So, as soon as you look at the graph, you know whether the classifier is performing well or not. So, one way to do that is to get this plots as sub plots.



So, there is a subplot function in matplotlib. Matplotlib has a subplot function. So, I am using that. So, that both the graphs are plotted together that help you appreciate which algorithm is better. Now, prima facie some changes happen here. So, prima facie which algorithm is better? Marginally, the blue is slightly better, right.

Marginally, the blue is better, but by using a random forest there is no major change that has happened. I do not know why. So, but this shows that you have to try different algorithms, try different, possibly try different hyper parameters and see if the algorithm performance changes. And these graphical visualizations help you immediately to see whether the model is improving or not, by comparing them.

And you can see the area under the curve is reported below. AUC is 0.69, 0.68. As we said it is only a marginal change in the classifier performance. And AP is average precision which also shows some marginal difference, but not much. Now, rest is cross validation, you can just run that code, that is simple. But this is the heart of the matter as far as takeaway from this course is concerned.

Be able to build different models, validate them, compare them in terms of their performance. Specific performance measures like false positive rate and true positive rate, precision and recall which gives you a picture of whether the model is good for application. Alright, I will quit this screen.