0:0:0.0 --> 0:0:0.850  
Anand sivaraman  
Good afternoon, team.

0:0:1.300 --> 0:0:6.60  
Anand sivaraman  
Uh, we'll just give couple more minutes for folks to join, otherwise we'll start by four or five.

0:0:8.330 --> 0:0:9.470  
Muniaraj Thangavelu  
Good afternoon, Alan, sure.

0:0:18.820 --> 0:0:19.470  
Anand sivaraman  
Now not good.

0:0:19.480 --> 0:0:20.230  
Anand sivaraman  
Profile picture, Sir.

0:0:24.30 --> 0:0:24.860  
Nawnath Laxmanrao  
I guess I think it's.

0:0:27.70 --> 0:0:28.550  
Anand sivaraman  
Yeah, I say it has stereo. Bye.

0:0:51.590 --> 0:0:58.100  
Anand sivaraman  
OK, while while people are joining, if any of you have any questions, happy to answer any concepts that you want to cover.

0:0:58.110 --> 0:0:58.720  
Anand sivaraman  
Happy to answer.

0:0:58.730 --> 0:1:1.200  
Anand sivaraman  
I know some of you probably would have joined in the middle.

0:1:1.250 --> 0:1:8.780  
Anand sivaraman  
I'm not sure how many of you attended it from the beginning, and I have not been very uh punctual about these sessions.

0:1:8.790 --> 0:1:12.0  
Anand sivaraman  
I was delaying it clear on there, so sorry about it.

0:1:12.40 --> 0:1:13.500  
Anand sivaraman  
But if you have any questions, happy to answer.

0:1:34.820 --> 0:1:37.70  
Muniaraj Thangavelu  
I am part of this session.

0:1:37.120 --> 0:1:42.870  
Muniaraj Thangavelu  
We are covering what is vector index and augment the retrieval method.

0:1:42.960 --> 0:1:50.740  
Muniaraj Thangavelu  
Those things will be in future covered or only models which we see in that only.

0:1:51.840 --> 0:1:52.130  
Anand sivaraman  
No.

0:2:1.480 --> 0:2:1.780  
Muniaraj Thangavelu  
Ohh.

0:1:52.140 --> 0:2:2.230  
Anand sivaraman  
So I think if you all saw the syllabus that I shared sometime back late November money, I just give me one second.

0:2:4.180 --> 0:2:5.590  
Anand sivaraman  
Yeah, it raining.

0:2:5.660 --> 0:2:7.960  
Anand sivaraman  
I think Shibani Sad shared the syllabus, I believe.

0:2:10.970 --> 0:2:11.500  
Muniaraj Thangavelu  
Okano.

0:2:11.370 --> 0:2:16.720  
Anand sivaraman  
So if you go to this AI training for data science, the very beginning, there was a syllabus shared.

0:2:18.200 --> 0:2:22.70  
Anand sivaraman  
Umm, probably back in October, November time frame.

0:2:23.300 --> 0:2:24.850  
Anand sivaraman  
So Shibani shared the syllabus.

0:2:24.900 --> 0:2:25.590  
Anand sivaraman  
It has.

0:2:26.80 --> 0:2:27.890  
Anand sivaraman  
We will be covering generative AI.

0:2:27.960 --> 0:2:31.560  
Anand sivaraman  
I'll be covering prompt engineering and and.

0:2:40.780 --> 0:2:41.140  
Muniaraj Thangavelu  
Uh.

0:2:33.190 --> 0:2:41.260  
Anand sivaraman  
Retrieve a lot my augmented generation methodologies we will cover but but for that to happen you need to understand some basics right?

0:2:41.270 --> 0:2:42.190  
Anand sivaraman  
So yeah.

0:2:44.50 --> 0:2:44.920  
Muniaraj Thangavelu  
Yeah, got it. And.

0:2:44.210 --> 0:2:45.460  
Anand sivaraman  
Because because without.

0:2:45.470 --> 0:2:50.710  
Anand sivaraman  
Without understanding the models, without understanding how to run, you know you will not.

0:2:50.930 --> 0:2:57.210  
Anand sivaraman  
The other problem is when we reach prompt engineering, it cannot be done on 9 right.

0:3:0.340 --> 0:3:0.670  
Muniaraj Thangavelu  
Umm.

0:2:57.580 --> 0:3:1.500  
Anand sivaraman  
It has to be on Python or pythonic code, so I will.

0:3:1.510 --> 0:3:2.810  
Anand sivaraman  
I will have to shift gears there.

0:3:4.410 --> 0:3:4.770  
Muniaraj Thangavelu  
Sure.

0:3:5.30 --> 0:3:7.170  
Muniaraj Thangavelu  
Just to get to know I asked.

0:3:7.830 --> 0:3:8.120  
Anand sivaraman  
No, no.

0:3:8.130 --> 0:3:9.210  
Anand sivaraman  
Yeah, I think you should go back.

0:3:9.220 --> 0:3:11.180  
Anand sivaraman  
There was a syllabus that Shibani had shared.

0:3:11.890 --> 0:3:15.80  
Anand sivaraman  
Umm if if that syllabus does not have it, please add it.

0:3:15.90 --> 0:3:17.690  
Anand sivaraman  
I I at least was thinking that we will cover all of that.

0:3:18.840 --> 0:3:20.650  
Muniaraj Thangavelu  
OK, I will search for.

0:3:27.40 --> 0:3:28.130  
Anand sivaraman  
OK, fine.

0:3:28.360 --> 0:3:29.240  
Anand sivaraman  
Just give me one second.

0:3:50.130 --> 0:3:51.460  
Anand sivaraman  
OK, let's start.

0:3:51.470 --> 0:3:52.300  
Anand sivaraman  
Thanks for joining.

0:3:52.350 --> 0:3:53.680  
Anand sivaraman  
I'll share my screen.

0:3:58.990 --> 0:4:1.160  
Anand sivaraman  
So where are we in the last session?

0:4:2.530 --> 0:4:6.570  
Anand sivaraman  
If can one of you recap what we learned, at least at a high level.

0:4:13.60 --> 0:4:15.320  
Anand sivaraman  
I mean anything that you can remember is fine.

0:4:16.180 --> 0:4:31.80  
Bhagyalakshmi Ganapathy  
We were like seeing about Decision tree and entropy information gain and have to like with that how to choose the correct features and we saw in nine years you showed us on don't example.

0:4:32.230 --> 0:4:33.0  
Anand sivaraman  
OK, great.

0:4:33.130 --> 0:4:33.530  
Anand sivaraman  
Fine.

0:4:33.540 --> 0:4:33.980  
Anand sivaraman  
Thank you.

0:4:34.590 --> 0:4:37.480  
Anand sivaraman  
So we did not cover random forest, right?

0:4:37.590 --> 0:4:38.520  
Anand sivaraman  
I just wanted to be sure.

0:4:39.250 --> 0:4:40.50  
Bhagyalakshmi Ganapathy  
No, no, wait.

0:4:41.350 --> 0:4:41.780  
Anand sivaraman  
OK.

0:4:41.830 --> 0:4:42.960  
Anand sivaraman  
OK, just give me one second.

0:4:42.970 --> 0:4:44.380  
Anand sivaraman  
I'm just setting up my.

0:4:47.60 --> 0:4:47.510  
Anand sivaraman  
Next.

0:4:52.690 --> 0:4:53.590  
Anand sivaraman  
Just give me one second, OK?

0:4:59.310 --> 0:5:1.330  
Anand sivaraman  
OK, I chatting now.

0:5:4.380 --> 0:5:6.230  
Anand sivaraman  
OK, so decision tree concepts.

0:5:7.200 --> 0:5:9.660  
Anand sivaraman  
You guys are OK, at least at a conceptual level.

0:5:15.60 --> 0:5:15.590  
Bhagyalakshmi Ganapathy  
Isn't.

0:5:16.580 --> 0:5:17.230  
Anand sivaraman  
OK, fine.

0:5:20.430 --> 0:5:21.370  
Anand sivaraman  
Is my screen result.

0:5:23.770 --> 0:5:24.240  
Bhagyalakshmi Ganapathy  
Yes it is.

0:5:25.600 --> 0:5:26.500  
Anand sivaraman  
OK, cool.

0:5:27.620 --> 0:5:28.410  
Anand sivaraman  
Yes.

0:5:28.860 --> 0:5:30.60  
Anand sivaraman  
What I'll do is.

0:5:32.320 --> 0:5:36.690  
Anand sivaraman  
We will pick up uh, we'll pick up random forest, so we covered decision tree.

0:5:37.0 --> 0:5:41.10  
Anand sivaraman  
If you remember, let me very quickly recap what we learned with this shantry.

0:5:41.500 --> 0:5:51.270  
Anand sivaraman  
So this was the use case that we saw and the data set that we had was delinquency prediction.

0:5:51.620 --> 0:6:0.310  
Anand sivaraman  
What is delinquency prediction if a client, the bank will predict whether a customer can pay back a particular loan or not.

0:6:0.620 --> 0:6:6.980  
Anand sivaraman  
If the customer is going to default that loan, it means that the there is a high likelihood that the customer is delinquent.

0:6:7.290 --> 0:6:9.440  
Anand sivaraman  
Delinquent is unable to pay, right?

0:6:9.450 --> 0:6:11.300  
Anand sivaraman  
That's that's what delinquency means.

0:6:11.690 --> 0:6:12.580  
Anand sivaraman  
Or bankruptcy.

0:6:12.590 --> 0:6:14.960  
Anand sivaraman  
You can think of that as individual bankruptcy.

0:6:15.170 --> 0:6:39.860  
Anand sivaraman  
So from that standpoint you will see that, uh, you will see that thanks before before they actually approved for loans or credit cards or even accounts for that matter, they do a KYC check and as part of the KYC check they do credit rating check and delinquency, delinquency quotient or delinquency score is very, very important for them.

0:6:41.180 --> 0:6:45.830  
Anand sivaraman  
So the data set has columns like serious delinquency in two years.

0:6:47.180 --> 0:6:54.860  
Anand sivaraman  
That means that in the next two years or in the last two years, as the customer shown, any patterns of delinquency, that is what?

0:6:56.460 --> 0:7:13.170  
Anand sivaraman  
Uh, so in the next two years, is there is a possibility, is there a possibility that the customer can become delinquent, so that is that is one second thing is age is a column number of times the customer has not paid the dues between 30 to 59 days.

0:7:13.440 --> 0:7:29.300  
Anand sivaraman  
So after the due date, if the customer has taken 30 to 59 days, how many times the customer has delayed between 30 and 59 days debt ratio is another uh column number of times the customer is 90 days late after the due date.

0:7:29.310 --> 0:7:38.900  
Anand sivaraman  
How many times has the has the customer delayed the payment 90 days after the due date and then the third one is number of times 60 to 89 days right?

0:7:38.910 --> 0:7:39.260  
Anand sivaraman  
So.

0:7:39.790 --> 0:7:44.980  
Anand sivaraman  
So these are all data points that are collected monthly income, and then finally number of dependents.

0:7:44.990 --> 0:7:45.680  
Anand sivaraman  
How many dependents?

0:7:45.690 --> 0:7:46.780  
Anand sivaraman  
So this is the data set.

0:7:47.90 --> 0:8:0.900  
Anand sivaraman  
We use this data set for and the data set has missing values, so we we removed the missing values by the statistical method and I think these are aspects that you will you will learn.

0:8:1.110 --> 0:8:6.600  
Anand sivaraman  
So one of the things that I also wanted to show you is this give me one second, please.

0:8:6.800 --> 0:8:7.250  
Anand sivaraman  
Yeah.

0:8:7.300 --> 0:8:24.530  
Anand sivaraman  
So the decision tree learner is that is the mod is the model that we are going to use to train this data set, train on the data set partitioning is anyway 7030 is what we had taken that 70% training, 30% test and the decision tree learner.

0:8:24.540 --> 0:8:25.860  
Anand sivaraman  
If you see there are configurations.

0:8:26.390 --> 0:8:44.880  
Anand sivaraman  
Umm the first part, the class column is the column serious delinquency in years is the class column and we have we have used that column to help as the as the column as a target variable right that we are going to predict quality measure is Gini index.

0:8:45.550 --> 0:8:49.900  
Anand sivaraman  
Can can someone recap what genie indexes at a very very high level?

0:8:49.910 --> 0:8:51.330  
Anand sivaraman  
I'm not expecting the formula.

0:8:51.660 --> 0:8:55.60  
Anand sivaraman  
Why do you need Genie index and similar to Genie index?

0:8:55.70 --> 0:8:55.990  
Anand sivaraman  
What else did we learn?

0:9:1.940 --> 0:9:2.940  
Anand sivaraman  
What does gene? That's.

0:9:1.780 --> 0:9:5.300  
Bhagyalakshmi Ganapathy  
It is like uh to calculate the measure of like impurity.

0:9:6.770 --> 0:9:7.360  
Anand sivaraman  
Very good.

0:9:7.370 --> 0:9:7.620  
Bhagyalakshmi Ganapathy  
We.

0:9:7.410 --> 0:9:9.840  
Anand sivaraman  
So what is impurity, ma'am, in your mind?

0:9:9.850 --> 0:9:10.480  
Anand sivaraman  
What is impurity?

0:9:11.280 --> 0:9:13.240  
Bhagyalakshmi Ganapathy  
It is the randomness in the data.

0:9:13.980 --> 0:9:16.430  
Anand sivaraman  
And So what is randomness in the data?

0:9:16.440 --> 0:9:18.940  
Anand sivaraman  
Can you be just just give a very simple example, right?

0:9:20.850 --> 0:9:21.60  
Bhagyalakshmi Ganapathy  
Yeah.

0:9:18.950 --> 0:9:21.710  
Anand sivaraman  
So that everyone can understand and so that you can.

0:9:21.780 --> 0:9:23.430  
Anand sivaraman  
You can also remember understanding it.

0:9:23.960 --> 0:9:31.740  
Bhagyalakshmi Ganapathy  
Like if you have like 10 data sets and we have 5 to S or no we can we take the example of Sr?

0:9:31.750 --> 0:9:31.960  
Bhagyalakshmi Ganapathy  
No.

0:9:31.970 --> 0:9:32.950  
Bhagyalakshmi Ganapathy  
We have 5S5.

0:9:32.960 --> 0:9:35.390  
Bhagyalakshmi Ganapathy  
No, it is like half half here.

0:9:35.400 --> 0:9:38.910  
Bhagyalakshmi Ganapathy  
If we have like 7 is or a homogeneous.

0:9:38.970 --> 0:9:43.480  
Bhagyalakshmi Ganapathy  
If you have it like it is pure data and if it is heterogeneous, it is like included.

0:9:44.300 --> 0:9:45.50  
Anand sivaraman  
Very good, very good.

0:9:45.60 --> 0:9:49.560  
Anand sivaraman  
I understood that you understood, so let me let me also give some other example.

0:9:49.570 --> 0:9:49.870  
Anand sivaraman  
Right.

0:9:50.570 --> 0:9:50.740  
Bhagyalakshmi Ganapathy  
Yeah.

0:9:49.960 --> 0:9:51.880  
Anand sivaraman  
So let's say you are given a.

0:9:51.920 --> 0:9:57.540  
Anand sivaraman  
Let's say you are given a task to so split a team of men and women.

0:9:58.300 --> 0:10:1.90  
Anand sivaraman  
Uh, in such a way there are about 20 people.

0:10:1.320 --> 0:10:6.230  
Anand sivaraman  
If you are supposed to split a team of men and women in two rooms, right?

0:10:6.480 --> 0:10:16.110  
Anand sivaraman  
So you can, let's say if you go and split the team with five women, five men in one room, right, that is heterogeneous data, right?

0:10:16.200 --> 0:10:20.670  
Anand sivaraman  
So it means that data set is impure, meaning you have both men and women combined there, right?

0:10:20.680 --> 0:10:37.570  
Anand sivaraman  
So but if in one room you have 10 men and another room, you have 10 women that is homogeneous data, which is you have a, you have people of like minded nature in the same room, right or similar characteristics in the same room.

0:10:37.700 --> 0:10:38.440  
Anand sivaraman  
So that is what?

0:10:40.510 --> 0:10:51.840  
Anand sivaraman  
That is what impurity means when you have a mix of values of the target variable, then it becomes heterogeneous when the same value is there then becomes homogeneous.

0:10:51.990 --> 0:10:53.920  
Anand sivaraman  
Why do we need to calculate impurity?

0:10:54.190 --> 0:10:55.530  
Anand sivaraman  
Why do we need to calculate impurity?

0:10:57.610 --> 0:10:58.730  
Anand sivaraman  
For what purpose?

0:11:2.180 --> 0:11:4.10  
Muniaraj Thangavelu  
The route will not be.

0:11:4.60 --> 0:11:19.540  
Muniaraj Thangavelu  
Further, Sir means it will not divide further, though we need to find the root of that so that root will be the impurity one and it will delete the data and so like that it will.

0:11:20.460 --> 0:11:21.810  
Anand sivaraman  
OK, good try Muni.

0:11:21.930 --> 0:11:23.710  
Anand sivaraman  
A slight change in the understanding.

0:11:24.90 --> 0:11:24.450  
Muniaraj Thangavelu  
OK.

0:11:24.60 --> 0:11:26.890  
Anand sivaraman  
So see at the at the beginning.

0:11:26.900 --> 0:11:35.70  
Anand sivaraman  
This is my root node, OK in the root node I have 14 days worth of data and in the 14 days I have heterogeneous data 9 days.

0:11:35.380 --> 0:11:36.850  
Anand sivaraman  
The play has happened 5 days.

0:11:36.860 --> 0:11:37.850  
Anand sivaraman  
The play did not happen.

0:11:38.380 --> 0:11:45.310  
Anand sivaraman  
Now, with this heterogeneous data I need to make a prediction and the prediction can be what on the 15th day.

0:11:45.660 --> 0:11:48.210  
Anand sivaraman  
Either the play can happen or the play cannot happen.

0:11:48.220 --> 0:11:49.930  
Anand sivaraman  
Only one result can come right.

0:11:49.940 --> 0:11:51.940  
Anand sivaraman  
Both the results cannot come alright.

0:11:52.660 --> 0:11:53.490  
Anand sivaraman  
Is that correct money.

0:11:53.110 --> 0:11:54.140  
Muniaraj Thangavelu  
Yes, yes.

0:11:54.0 --> 0:11:57.810  
Anand sivaraman  
Right, so to so, which means the prediction and needs to be homogeneous.

0:11:58.20 --> 0:12:7.750  
Anand sivaraman  
So the algorithm what the algorithm will do is the algorithm will need to start with some independent variable that can allow it to reach the leaf node.

0:12:7.760 --> 0:12:8.720  
Anand sivaraman  
This is the leaf node.

0:12:8.730 --> 0:12:13.110  
Anand sivaraman  
The leaf node is the last node beyond which you cannot divide the data right?

0:12:13.200 --> 0:12:15.530  
Anand sivaraman  
So the leaf node is the pure node.

0:12:15.760 --> 0:12:32.220  
Anand sivaraman  
So by using a variable, the algorithm will try to find out which all the variables it will try to calculate all the variables, and whichever variable has the least impurity and highest information gain, right, it will use that variable to achieve the leaf node faster.

0:12:32.370 --> 0:12:42.360  
Anand sivaraman  
So to calculate what independent variable will allow it to reach the pure node or the leaf node quickly, it need it will use entropy and information gate.

0:12:44.500 --> 0:12:44.630  
Muniaraj Thangavelu  
Yeah.

0:12:42.370 --> 0:12:54.920  
Anand sivaraman  
OK, now entropy and entropy is the measurement of impurity, while information gain is the ability for that one variable to contribute or explain the data set, right?

0:12:55.10 --> 0:13:2.120  
Anand sivaraman  
So whichever variable, let's say, if the algorithm has chosen Outlook, outlook has the least impurity and highest information gain.

0:13:2.470 --> 0:13:3.660  
Anand sivaraman  
Now what is?

0:13:3.710 --> 0:13:5.420  
Anand sivaraman  
I was talking about entropy, right?

0:13:5.530 --> 0:13:7.40  
Anand sivaraman  
But what is Gini index here?

0:13:7.50 --> 0:13:10.420  
Anand sivaraman  
So if you see in the configuration, 9 provides 2 options.

0:13:10.430 --> 0:13:13.980  
Anand sivaraman  
Either I can take Gini index or I can use gain ratio.

0:13:14.550 --> 0:13:17.530  
Anand sivaraman  
Gain ratio is similar to information gain as of now.

0:13:17.540 --> 0:13:33.30  
Anand sivaraman  
You can assume that Jeannie index is equivalent to entropy, so when you use gain ratio when you use gain ratio, it means that the algorithm has calculated the entropy and from the entropy it has calculated the information gain.

0:13:33.40 --> 0:13:37.690  
Anand sivaraman  
So if you see here I explained the information gain is calculated from entropy.

0:13:37.820 --> 0:13:44.320  
Anand sivaraman  
So if you use the configuration as gained ratio, the algorithm algorithm will first calculate the entropy and then the information gain.

0:13:44.600 --> 0:13:49.200  
Anand sivaraman  
But if you use Genie index, the algorithm does not need to calculate information gain and all.

0:13:49.290 --> 0:13:57.620  
Anand sivaraman  
It will simply identify the genie impurity by by way of genie index and then identify and make predictions accordingly.

0:13:57.810 --> 0:13:58.610  
Anand sivaraman  
Is that clear to everyone?

0:14:1.250 --> 0:14:1.910  
Muniaraj Thangavelu  
Yes or no.

0:14:4.300 --> 0:14:6.210  
Anand sivaraman  
OK, what is pruning?

0:14:6.580 --> 0:14:16.470  
Anand sivaraman  
Pruning is the concept where when the tree is growing larger, we need we need a process or a methodology to make sure that the tree trees growth is limited.

0:14:17.50 --> 0:14:18.290  
Anand sivaraman  
So what do I mean by that?

0:14:18.440 --> 0:14:28.670  
Anand sivaraman  
So if my tree is like this, instead of reaching the instead of reaching the leaf node in one within in one branch node, if it takes like 100 branch nodes, right?

0:14:28.880 --> 0:14:31.830  
Anand sivaraman  
How long can I allow the algorithm to run right?

0:14:31.880 --> 0:14:41.840  
Anand sivaraman  
So I need some methodology to cut the tree in such a way that it is it's progress is managed OK why is it called as pruning?

0:14:41.850 --> 0:14:51.400  
Anand sivaraman  
Pruning is similar to, let's say if you have a garden and you have a gardener, the gardener will typically take a garden scissors and make sure that the trees and the leaves are kept in one shape.

0:14:51.510 --> 0:14:53.520  
Anand sivaraman  
That is what is called as pruning right.

0:14:53.570 --> 0:14:58.60  
Anand sivaraman  
And similarly because we are talking about a tree here, the tree also needs to be pruned.

0:14:58.190 --> 0:14:58.560  
Anand sivaraman  
Why?

0:14:58.570 --> 0:15:10.810  
Anand sivaraman  
Why do we need the trim to be treated to be pruned #1 to reduce the training time #2 to make sure that the algorithm is able to but come arrive at some level of leaf nodes very quickly, right?

0:15:10.820 --> 0:15:12.290  
Anand sivaraman  
So that is the reason why throning.

0:15:12.460 --> 0:15:20.210  
Anand sivaraman  
So in this configuration we have Genie index as the quality measure or the impurity measure pruning methodology.

0:15:20.310 --> 0:15:22.20  
Anand sivaraman  
We can or we don't need to choose.

0:15:22.70 --> 0:15:27.110  
Anand sivaraman  
We can use this as a parameter to improve the model number of records per node.

0:15:27.720 --> 0:15:28.610  
Anand sivaraman  
What does this mean?

0:15:29.60 --> 0:15:31.250  
Anand sivaraman  
What do we understand by number of records per node?

0:15:33.230 --> 0:15:34.280  
Anand sivaraman  
What do I mean by that?

0:15:34.330 --> 0:15:34.810  
Anand sivaraman  
Any thoughts?

0:15:45.410 --> 0:15:46.440  
Anand sivaraman  
Uh hmm.

0:15:39.660 --> 0:15:46.570  
Muniaraj Thangavelu  
The classes maximum is the subdivided, here the sunny and those 3 categories right?

0:15:46.840 --> 0:15:51.70  
Muniaraj Thangavelu  
So like that five categories, the maximum that's.

0:15:51.210 --> 0:15:51.600  
Anand sivaraman  
OK.

0:15:51.650 --> 0:15:54.80  
Anand sivaraman  
So it's not category money, it's a number of records.

0:15:53.520 --> 0:15:54.820  
Muniaraj Thangavelu  
OK. Ohh.

0:15:54.90 --> 0:16:5.540  
Anand sivaraman  
So for example, for example, if I have let's say in this node, in the branch node I'm checking outlook OK Outlook has sunny, overcast and raining three options.

0:16:5.910 --> 0:16:21.410  
Anand sivaraman  
So if what this particular node root node will say is if I'm saying number of records per node, number of records per node is 5, it means for this root node to process it at least needs a minimum of five records.

0:16:21.800 --> 0:16:25.490  
Anand sivaraman  
It has got nothing to do with the target variable.

0:16:25.760 --> 0:16:36.910  
Anand sivaraman  
It will have minimum of five records only if there is a minimum of five records in this data, which and and then you know the decision node can probably operate right?

0:16:36.920 --> 0:16:38.110  
Anand sivaraman  
So that is what it means.

0:16:38.220 --> 0:16:54.910  
Anand sivaraman  
If you look at nines description, go to view and you and description, it tells you that number, number of records per node, minimum records per node is to select the minimum number of records at least required for each node.

0:16:55.100 --> 0:17:5.570  
Anand sivaraman  
If the number of records is smaller or equal to this number, the tree will not move forward any further, so it is the number of records when it does not number of target value variables but it.

0:17:6.260 --> 0:17:6.620  
Muniaraj Thangavelu  
OK.

0:17:9.550 --> 0:17:14.500  
Anand sivaraman  
So and the next, the next is number of records to store for view.

0:17:14.750 --> 0:17:18.580  
Anand sivaraman  
This is not a parameter that is used to improve the model.

0:17:18.590 --> 0:17:22.580  
Anand sivaraman  
This is purely a parameter for 9 to display the number of records.

0:17:22.590 --> 0:17:24.320  
Anand sivaraman  
How many records you want to display?

0:17:24.430 --> 0:17:26.240  
Anand sivaraman  
So this is not a big deal, right?

0:17:26.690 --> 0:17:28.720  
Anand sivaraman  
I I'm saying display 10,000 records.

0:17:28.730 --> 0:17:29.100  
Anand sivaraman  
That's it.

0:17:31.760 --> 0:17:34.410  
Anand sivaraman  
This will this has got nothing to do with the model itself.

0:17:34.460 --> 0:17:37.970  
Anand sivaraman  
OK, now number of records average split point.

0:17:38.60 --> 0:17:39.470  
Anand sivaraman  
So what this means?

0:17:39.480 --> 0:17:41.410  
Anand sivaraman  
This is an interesting feature.

0:17:41.860 --> 0:17:53.310  
Anand sivaraman  
Average split point means if the split value of the number numeric attribute is determined according to the mean value of two attribute values, right?

0:17:53.380 --> 0:17:54.910  
Anand sivaraman  
That separate the two partitions.

0:17:55.0 --> 0:17:59.770  
Anand sivaraman  
If unchecked, the split value is set to the largest value of the lower partition.

0:17:59.860 --> 0:18:1.570  
Anand sivaraman  
OK, So what does this mean?

0:18:1.840 --> 0:18:6.250  
Anand sivaraman  
So if you look at this a simple way to understand this is you look at humidity.

0:18:6.260 --> 0:18:7.990  
Anand sivaraman  
Humidity is a numeric data item.

0:18:8.440 --> 0:18:15.730  
Anand sivaraman  
Now when the when the decision tree is running in this branch node, it is checking whether humidity is less than or equal to 70 or greater than 70.

0:18:16.220 --> 0:18:19.330  
Anand sivaraman  
But there is no value called 70 here you see this?

0:18:19.400 --> 0:18:22.930  
Anand sivaraman  
There is no value called 70 here, so how did it arrive at 70?

0:18:23.140 --> 0:18:25.610  
Anand sivaraman  
So that is based on the average split point.

0:18:25.680 --> 0:18:36.90  
Anand sivaraman  
So if you see what average split point means is, if it is a numeric variable, then the mean of two attribute values that separate the two partitions will be considered.

0:18:36.100 --> 0:18:42.60  
Anand sivaraman  
So in this case, two separate variables 80 and two separate attribute values, right?

0:18:42.70 --> 0:18:50.130  
Anand sivaraman  
Two separate 2 attribute values AT and sixty or two attribute values mean of 80 and 60 is 7080 + 60 by 270.

0:18:50.280 --> 0:19:2.710  
Anand sivaraman  
So the algorithm has automatically taken an average split point between 80 and 60, and then it is using that split point to check for conditions that can allow it to reach the leaf node quickly.

0:19:2.880 --> 0:19:3.320  
Anand sivaraman  
Is that clear?

0:19:5.550 --> 0:19:7.40  
Anand sivaraman  
So average split point.

0:19:7.50 --> 0:19:10.800  
Anand sivaraman  
Is there another parameter, SO3 parameters that we can use?

0:19:10.890 --> 0:19:12.920  
Anand sivaraman  
The other parameters are very simple.

0:19:12.930 --> 0:19:18.200  
Anand sivaraman  
If you look at the roots split by default, let's say the algorithm has chosen some variable.

0:19:18.270 --> 0:19:19.560  
Anand sivaraman  
Let's say the algorithm has chosen.

0:19:19.570 --> 0:19:23.900  
Anand sivaraman  
By default some variable as the variable to be used at the root node.

0:19:23.950 --> 0:19:24.940  
Anand sivaraman  
Here. Right.

0:19:24.950 --> 0:19:28.360  
Anand sivaraman  
Let's say by default the algorithm has chosen some variable like.

0:19:28.810 --> 0:19:32.200  
Anand sivaraman  
Let's say it has chosen number of times 90 days late, right?

0:19:32.210 --> 0:19:41.80  
Anand sivaraman  
It has chosen if you don't if, as a subject matter expert, you think that the algorithm is not correct, you can force the root split column.

0:19:41.470 --> 0:19:45.680  
Anand sivaraman  
You can force it to any column that you want, but you have to be very, very careful about doing this.

0:19:48.190 --> 0:19:49.530  
Anand sivaraman  
Is everyone clear with the parameters?

0:19:51.570 --> 0:19:52.110  
Anand sivaraman  
Any questions?

0:19:56.30 --> 0:19:56.600  
Anand sivaraman  
Any thoughts?

0:19:56.610 --> 0:19:57.180  
Anand sivaraman  
Any questions?

0:19:57.190 --> 0:19:58.690  
Anand sivaraman  
Any anything that you don't understand?

0:20:1.320 --> 0:20:4.610  
Anand sivaraman  
OK, so but there are no questions.

0:20:4.680 --> 0:20:5.840  
Anand sivaraman  
Then let's see.

0:20:6.850 --> 0:20:7.230  
Anand sivaraman  
Let's see.

0:20:7.240 --> 0:20:7.570  
Anand sivaraman  
What?

0:20:8.260 --> 0:20:9.250  
Anand sivaraman  
What else we can cover?

0:20:9.260 --> 0:20:13.930  
Anand sivaraman  
So if you look at the decision tree after the configuration, what we have done is we have.

0:20:15.10 --> 0:20:18.700  
Anand sivaraman  
If you see that there are multiple views, there is a tree view.

0:20:18.750 --> 0:20:21.340  
Anand sivaraman  
There is a simple view and this is a tree model.

0:20:21.390 --> 0:20:31.520  
Anand sivaraman  
OK, tree model is more like just a read only point for us, but the tree view or the simple view will tell you a few things right?

0:20:31.590 --> 0:20:42.80  
Anand sivaraman  
So if you look at this, the root node, it says Class 0 and it says it says Class Zero and there's just give me one second.

0:20:42.90 --> 0:20:42.730  
Anand sivaraman  
Just give me one SEC.

0:21:27.770 --> 0:21:28.900  
Anand sivaraman  
Sorry, was a call.

0:21:29.570 --> 0:21:31.340  
Anand sivaraman  
So what is the root class?

0:21:31.350 --> 0:21:39.450  
Anand sivaraman  
Say the root class says Class Zero, 24,502 out of 31,544, right?

0:21:39.590 --> 0:21:40.580  
Anand sivaraman  
So what does that mean?

0:21:40.590 --> 0:21:45.940  
Anand sivaraman  
It means the root node has a.

0:21:46.0 --> 0:21:47.620  
Anand sivaraman  
Let let me give you another view.

0:21:47.670 --> 0:21:49.800  
Anand sivaraman  
Actually, let's look at the complex view here, right?

0:21:50.10 --> 0:22:2.790  
Anand sivaraman  
I've used the simple view, but if you look at it, Class 0 has 24,500 and two class one has seven thousand 40042 total number of training records is 31,544, right?

0:22:3.850 --> 0:22:13.900  
Anand sivaraman  
And what this 0 means is out of out of what do you call 0 and one class the THIS class 0 is.

0:22:13.990 --> 0:22:14.800  
Anand sivaraman  
What does this mean?

0:22:14.950 --> 0:22:24.80  
Anand sivaraman  
It is the majority class because there are 24,502 rows with Class 0 and 7042 rows with class one.

0:22:24.470 --> 0:22:25.440  
Anand sivaraman  
What does 0 mean?

0:22:25.510 --> 0:22:30.240  
Anand sivaraman  
0 means no, darling, no delinquency, 1 means delinquent.

0:22:30.390 --> 0:22:32.560  
Anand sivaraman  
OK, so that's what it means.

0:22:32.670 --> 0:22:35.780  
Anand sivaraman  
So if you look at it that uh, it gives you an idea about.

0:22:37.660 --> 0:22:44.750  
Anand sivaraman  
It gives you an idea about what is class Zero, what is class one, how the distribution is 77.77%.

0:22:44.820 --> 0:22:51.900  
Anand sivaraman  
Sorry, 77% is no, not delinquent they it's not, it's not delinquent.

0:22:51.910 --> 0:23:0.180  
Anand sivaraman  
Non delinquent, I'll call it as non D and 22% is delinquent KD.

0:23:0.370 --> 0:23:3.710  
Anand sivaraman  
Now why is 0 mentioned here?

0:23:3.720 --> 0:23:10.640  
Anand sivaraman  
It is called as the zero is the majority class majority class as it has more number of records.

0:23:12.380 --> 0:23:12.740  
Anand sivaraman  
OK.

0:23:12.930 --> 0:23:16.120  
Anand sivaraman  
So that is that is something that we need to understand. OK.

0:23:16.190 --> 0:23:17.300  
Anand sivaraman  
Now, what is this?

0:23:17.510 --> 0:23:21.880  
Anand sivaraman  
What is this variable number of times 90 days late?

0:23:22.270 --> 0:23:32.370  
Anand sivaraman  
Right, so algorithm has chosen this variable as the variable to be used for the root node at the root node for the split.

0:23:33.300 --> 0:23:35.690  
Anand sivaraman  
Why do you think the algorithm has chosen that?

0:23:36.560 --> 0:23:37.40  
Anand sivaraman  
Why do you?

0:23:37.50 --> 0:23:39.460  
Anand sivaraman  
On what basis you think the algorithm has chosen that?

0:23:41.960 --> 0:23:45.680  
Anand sivaraman  
On what basis do you think the algorithm has chosen that particular variable?

0:23:47.40 --> 0:23:47.900  
Anand sivaraman  
Any any thoughts?

0:23:51.160 --> 0:23:55.940  
Muniaraj Thangavelu  
That is The Hobbit abnormal route to find the leaf.

0:23:57.450 --> 0:23:58.280  
Anand sivaraman  
Yeah, correct.

0:23:58.320 --> 0:23:59.120  
Anand sivaraman  
So why?

0:23:59.130 --> 0:24:0.920  
Anand sivaraman  
How on so you?

0:24:0.930 --> 0:24:1.720  
Anand sivaraman  
You're right, money.

0:24:1.730 --> 0:24:6.280  
Anand sivaraman  
So what are the parameters we just what helps in finding them, yes.

0:24:3.940 --> 0:24:8.540  
Muniaraj Thangavelu  
The information gain entropy value is calculated I.

0:24:8.280 --> 0:24:9.420  
Anand sivaraman  
Yes and yes.

0:24:8.980 --> 0:24:9.470  
Muniaraj Thangavelu  
This is.

0:24:10.640 --> 0:24:10.950  
Anand sivaraman  
Correct.

0:24:10.960 --> 0:24:13.50  
Anand sivaraman  
So entropy and information gain is calculated.

0:24:13.60 --> 0:24:18.800  
Anand sivaraman  
So you can say that this particular column has the lowest entropy.

0:24:20.410 --> 0:24:23.650  
Anand sivaraman  
Right and highest information gain.

0:24:25.670 --> 0:24:27.830  
Anand sivaraman  
Alright, I just information gate.

0:24:30.580 --> 0:24:30.760  
Muniaraj Thangavelu  
Yes.

0:24:30.750 --> 0:24:42.540  
Anand sivaraman  
So as a result, the algorithm has found this column to be the primary column for for all the for for the root node, or the root split.

0:24:43.50 --> 0:24:43.500  
Anand sivaraman  
Is that clear?

0:24:45.0 --> 0:24:50.510  
Anand sivaraman  
OK, now if you look at, if you look at what the algorithm has done further, it's very interesting, correct.

0:24:50.590 --> 0:24:58.770  
Anand sivaraman  
So if you look at it, it says number of times 90 days late 0.5 and .5.

0:24:59.210 --> 0:25:5.330  
Anand sivaraman  
Is that a value called 0.5 and .54 point 5 for this?

0:25:6.300 --> 0:25:10.0  
Anand sivaraman  
This particular algorithm, do you think that value is there?

0:25:12.160 --> 0:25:13.770  
Anand sivaraman  
That is no such value, right?

0:25:13.600 --> 0:25:13.880  
Shubham Kumar  
No.

0:25:19.980 --> 0:25:20.810  
Shubham Kumar  
Maybe the mean value?

0:25:13.980 --> 0:25:22.240  
Anand sivaraman  
So then how did the algorithm pick .5 and -, .5, so .5? Sorry.

0:25:23.350 --> 0:25:24.110  
Shubham Kumar  
Maybe the mean value?

0:25:25.400 --> 0:25:31.330  
Anand sivaraman  
Yes, it is the mean value because we have chosen average split point here, right?

0:25:31.340 --> 0:25:37.30  
Anand sivaraman  
We have chosen average split point, so I'm for some reason I'm not able to close this.

0:25:37.40 --> 0:25:38.60  
Anand sivaraman  
Just give me one second guess.

0:25:40.510 --> 0:25:47.540  
Anand sivaraman  
We have chosen the average split point here and as a result the the algorithm has given.

0:25:53.120 --> 0:25:55.710  
Anand sivaraman  
OK, so we have chosen the average split point here.

0:25:55.900 --> 0:25:58.430  
Anand sivaraman  
As a result, the algorithm has.

0:26:0.160 --> 0:26:6.30  
Anand sivaraman  
I has split the data number of times between 0.5 and greater than .5.

0:26:6.280 --> 0:26:10.350  
Anand sivaraman  
Now if you look at how the tree is expanding, see I am.

0:26:10.400 --> 0:26:20.310  
Anand sivaraman  
It is able to just create now after 30 and after 90 days the next variable it has chosen us number of times 30 to 59 and then number of times 60 to 89.

0:26:20.420 --> 0:26:27.950  
Anand sivaraman  
So by looking at this, what can you understand by looking at this you can you can understand that you can.

0:26:27.960 --> 0:26:37.250  
Anand sivaraman  
You can understand that number of times the most important variable or number of times this variable has high information gain.

0:26:38.990 --> 0:26:47.200  
Anand sivaraman  
This variable has the second highest information gain and this variable has the third highest information gain, right?

0:26:47.300 --> 0:26:54.790  
Anand sivaraman  
So with all of this, you can at least understand that this particular variable you what can you what can you construe out of this?

0:26:55.20 --> 0:27:3.170  
Anand sivaraman  
You can construe that number of times 90 days late has the most important is the most important variable.

0:27:3.180 --> 0:27:12.940  
Anand sivaraman  
It's called feature importance feature importance number of times 90 days late is the most important feature, right?

0:27:13.190 --> 0:27:16.220  
Anand sivaraman  
That helps to decide, helps to predict.

0:27:19.700 --> 0:27:21.520  
Anand sivaraman  
Delinquency. OK.

0:27:24.580 --> 0:27:28.250  
Anand sivaraman  
So this is how you will construe you will end up doing it.

0:27:28.410 --> 0:27:28.730  
Anand sivaraman  
Got it.

0:27:30.150 --> 0:27:30.860  
Anand sivaraman  
Everyone with me.

0:27:30.870 --> 0:27:33.390  
Anand sivaraman  
Are you able to understand how the algorithms splitting this here?

0:27:35.890 --> 0:27:37.910  
Anand sivaraman  
Any questions you can keep?

0:27:38.70 --> 0:27:39.280  
Anand sivaraman  
So where is the leaf node?

0:27:39.290 --> 0:27:42.700  
Anand sivaraman  
Right, you will be interested to find out where is the leaf node.

0:27:42.870 --> 0:27:48.990  
Anand sivaraman  
So if you see, this is a very big tree, it is just going on and on and on because it has 35,000 rows.

0:27:49.310 --> 0:27:55.980  
Anand sivaraman  
This is a very big tree and and you might end up, so I'm I'm probably doing clicking it here.

0:27:56.30 --> 0:27:57.300  
Anand sivaraman  
OK, here is my leaf.

0:27:57.590 --> 0:27:58.840  
Anand sivaraman  
OK, here is my leaf.

0:27:59.70 --> 0:28:5.400  
Anand sivaraman  
It might not be a leaf, but because I have not because I have pruned it, is stopping here.

0:28:5.410 --> 0:28:6.110  
Anand sivaraman  
Think about this.

0:28:6.960 --> 0:28:8.770  
Anand sivaraman  
I started with look at this.

0:28:8.780 --> 0:28:25.550  
Anand sivaraman  
I started with 31,544 rows as the Afd as the tree was getting built, the number of rows are getting reduced obviously right because the conditions are the conditions are being applied and the number of rows are filtered based on those conditions.

0:28:25.810 --> 0:28:42.570  
Anand sivaraman  
So if you see here after number of first split number of times 90 days number of times 3030 to 59 number of times 60 to 89 from 24,000 rows it came to 2000 rows 23,000 rows and and then if you see.

0:28:44.420 --> 0:28:46.490  
Anand sivaraman  
That yes or no component here, right?

0:28:46.500 --> 0:28:51.870  
Anand sivaraman  
So yes, is 23,000 and no component is probably 2000.

0:28:52.120 --> 0:28:56.690  
Anand sivaraman  
Similarly, if it is 30 to 59, yes is 20,000.

0:28:56.820 --> 0:28:58.660  
Anand sivaraman  
No, is less right.

0:28:58.700 --> 0:29:22.360  
Anand sivaraman  
As you can see, this is just growing on and on and on and eventually eventually what you will see is eventually what you will see is the number of rows that it is finally coming is there are five, there are 8 rows of which one is a class, one is 5 and Class 0 is 3.

0:29:22.410 --> 0:29:24.80  
Anand sivaraman  
So the algorithm has stopped here.

0:29:24.140 --> 0:29:27.40  
Anand sivaraman  
Beyond this, it is unable to divide effectively.

0:29:27.310 --> 0:29:36.840  
Anand sivaraman  
You can make the algorithm divide further by changing the configuration, like for example instead of instead of saying reduced error pruning.

0:29:36.910 --> 0:29:38.860  
Anand sivaraman  
If you just probably remove this right?

0:29:39.410 --> 0:29:42.790  
Anand sivaraman  
I'm also interestingly think about this, right?

0:29:44.20 --> 0:29:46.500  
Anand sivaraman  
We say number of records per node is 5.

0:29:47.740 --> 0:29:52.540  
Anand sivaraman  
So I said result the algorithm has stopped here because we said minimally we need 5 rows.

0:29:52.920 --> 0:29:55.570  
Anand sivaraman  
So there is 8 rows beyond below this.

0:29:55.820 --> 0:29:58.250  
Anand sivaraman  
This cannot happen, so it has stopped here if I.

0:29:58.260 --> 0:30:6.390  
Anand sivaraman  
If I reduce the number of records as one right then you will see that the tree will grow further OK like for example if I run this again.

0:30:9.380 --> 0:30:13.40  
Anand sivaraman  
So if I run this and then, uh, this is there, right?

0:30:14.430 --> 0:30:17.210  
Anand sivaraman  
So if I run this and.

0:30:25.160 --> 0:30:25.810  
Anand sivaraman  
Right on this.

0:30:25.820 --> 0:30:28.170  
Anand sivaraman  
And then I keep on expanding the tree.

0:30:32.220 --> 0:30:36.340  
Anand sivaraman  
So I might probably need to do 1020 rows so as you can see here.

0:30:38.630 --> 0:30:39.470  
Anand sivaraman  
Why Analytics?

0:30:40.440 --> 0:30:42.590  
Anand sivaraman  
It has come to a pruning node.

0:30:42.780 --> 0:30:47.790  
Anand sivaraman  
If I change the configuration to number of nodes, records per node is 1.

0:30:47.900 --> 0:30:53.60  
Anand sivaraman  
It can give me the purest node, otherwise it will stop at 5 or whatever you see the difference?

0:30:54.810 --> 0:30:55.350  
Anand sivaraman  
Like guys with me.

0:30:58.440 --> 0:30:59.290  
Anand sivaraman  
Any questions on this?

0:30:59.10 --> 0:31:4.310  
Shubham Kumar  
Click is it a good method to reduce the number of records up to one?

0:31:5.700 --> 0:31:6.530  
Anand sivaraman  
It is actually not.

0:31:6.540 --> 0:31:7.690  
Anand sivaraman  
That's a very good question.

0:31:8.180 --> 0:31:10.80  
Anand sivaraman  
So the it's actually not.

0:31:10.90 --> 0:31:10.310  
Anand sivaraman  
Why?

0:31:10.320 --> 0:31:15.830  
Anand sivaraman  
Because you are not even giving enough sample data or sample size to the algorithm to do its job.

0:31:17.330 --> 0:31:17.500  
Shubham Kumar  
Yeah.

0:31:16.740 --> 0:31:18.20  
Anand sivaraman  
Uh, so we should.

0:31:18.30 --> 0:31:21.690  
Anand sivaraman  
Ideally, we should ideally have a decent sample size, right?

0:31:21.840 --> 0:31:24.600  
Anand sivaraman  
What is a decent sample size, at least more than one record?

0:31:24.610 --> 0:31:34.350  
Anand sivaraman  
I would say to do it otherwise you know you, I mean you will you you will get some output, but you may not be very sure about the output.

0:31:34.440 --> 0:31:34.720  
Anand sivaraman  
Got it.

0:31:38.70 --> 0:31:38.230  
Shubham Kumar  
Yeah.

0:31:40.120 --> 0:31:42.570  
Anand sivaraman  
So, OK, so so that's one.

0:31:42.580 --> 0:31:49.910  
Anand sivaraman  
And then finally, what we do, we do prediction, the prediction is, is a very simple this thing.

0:31:50.180 --> 0:31:52.840  
Anand sivaraman  
So if I take this to Phi again.

0:32:4.210 --> 0:32:12.250  
Anand sivaraman  
So if you see in the predictor I'm, I'm just saying, predicting column name, nothing else, nothing really nothing is rocket science fair.

0:32:12.260 --> 0:32:13.460  
Anand sivaraman  
Look at my classified data.

0:32:13.470 --> 0:32:16.880  
Anand sivaraman  
The prediction also has this entry with the test data.

0:32:16.890 --> 0:32:22.480  
Anand sivaraman  
It will give me give you the tree view and then if you look at the classified data it gives you all the necessary.

0:32:25.240 --> 0:32:39.840  
Anand sivaraman  
Outcomes predicted what is actual serious delinquency and what is the predicted serious telling Quincy, if you want the probability scores you can go to the configuration again, append columns with normalized class distribution right?

0:32:40.150 --> 0:32:41.430  
Anand sivaraman  
If you if you do that.

0:32:44.860 --> 0:32:53.870  
Anand sivaraman  
Should be able to look at the classified data and as you can see it gives you the probability value based on which why the prediction happened.

0:32:54.80 --> 0:32:56.70  
Anand sivaraman  
OK so here is the problem.

0:32:56.80 --> 0:32:57.140  
Anand sivaraman  
Here is the actual value.

0:32:57.980 --> 0:33:3.670  
Anand sivaraman  
Uh here is the predicted value, so this is actual this is predicted.

0:33:6.70 --> 0:33:12.40  
Anand sivaraman  
And these are the probabilities likelihood based on which the prediction has happened.

0:33:12.310 --> 0:33:20.360  
Anand sivaraman  
OK, so so in this case the the value 0 has a higher probability, so the prediction is zero.

0:33:20.470 --> 0:33:26.600  
Anand sivaraman  
OK, so so when you when you are and you are actually zero, you're in your prediction is also zero.

0:33:27.50 --> 0:33:30.480  
Anand sivaraman  
So what is this in confusion matrix terms?

0:33:30.490 --> 0:33:34.200  
Anand sivaraman  
What is this true positive, true negative, false positive, false negative?

0:33:38.580 --> 0:33:41.640  
Anand sivaraman  
So assuming zero is -, 1 is positive.

0:33:42.120 --> 0:33:42.850  
Anand sivaraman  
What is this?

0:33:42.940 --> 0:33:45.690  
Anand sivaraman  
If the actual and prediction are same, what does this mean?

0:33:48.340 --> 0:33:48.970  
Shubham Kumar  
Extra cost.

0:33:51.570 --> 0:33:55.240  
Anand sivaraman  
Assuming zero is negative, Sir 1 means positive, zero is negative.

0:33:54.170 --> 0:33:56.790  
Shubham Kumar  
Much more then, yeah, then then it's video posted here.

0:33:57.800 --> 0:33:58.0  
Anand sivaraman  
Then.

0:33:59.760 --> 0:34:0.690  
Shubham Kumar  
Then it would be opposite.

0:34:2.20 --> 0:34:2.660  
Anand sivaraman  
Opposite is what?

0:34:5.170 --> 0:34:6.230  
Shubham Kumar  
Like two negative.

0:34:7.260 --> 0:34:7.550  
Anand sivaraman  
Yeah.

0:34:7.560 --> 0:34:9.490  
Anand sivaraman  
Why is it true and why is it negative?

0:34:9.820 --> 0:34:11.400  
Anand sivaraman  
Negative is zero class 0.

0:34:11.610 --> 0:34:12.230  
Anand sivaraman  
Why is it true?

0:34:11.830 --> 0:34:16.330  
Shubham Kumar  
Yeah, because like it's one, right.

0:34:18.210 --> 0:34:19.270  
Anand sivaraman  
Well, why is it true?

0:34:19.420 --> 0:34:20.270  
Anand sivaraman  
Negative is 0.

0:34:20.280 --> 0:34:20.630  
Anand sivaraman  
Set.

0:34:24.340 --> 0:34:24.550  
Shubham Kumar  
Umm.

0:34:20.740 --> 0:34:24.940  
Anand sivaraman  
You are setting up zero as negative, which is OK, but why is it true?

0:34:28.190 --> 0:34:29.290  
Anand sivaraman  
Why is it called true negative?

0:34:30.70 --> 0:34:34.690  
Anand sivaraman  
So what I'm asking negative because it has predicted zero class it's negative.

0:34:34.700 --> 0:34:35.480  
Anand sivaraman  
Why is it called true?

0:34:36.290 --> 0:34:39.610  
Shubham Kumar  
Because we, uh, we are getting what is expected.

0:34:36.970 --> 0:34:39.690  
Muniaraj Thangavelu  
Maybe it correctly identify?

0:34:41.50 --> 0:34:42.660  
Anand sivaraman  
Ah, the actual and prediction.

0:34:43.310 --> 0:34:44.450  
Shubham Kumar  
It seemed, yeah.

0:34:42.950 --> 0:34:57.160  
Anand sivaraman  
Both are same since it is so the easiest way to understand confusion matrix is when the algorithm correctly predicts a negative class when the algorithm correctly predicts a negative class, it is true negative.

0:34:57.670 --> 0:35:1.310  
Anand sivaraman  
So in this case, what is this actually is 1?

0:35:3.440 --> 0:35:4.370  
Anand sivaraman  
Predicted is 0.

0:35:4.420 --> 0:35:4.860  
Anand sivaraman  
What is this?

0:35:10.120 --> 0:35:10.580  
Shubham Kumar  
Can you repeat?

0:35:10.280 --> 0:35:14.210  
Anand sivaraman  
In this case, the algorithm has incorrectly predicted the negative class, correct?

0:35:16.220 --> 0:35:17.130  
Muniaraj Thangavelu  
False negative.

0:35:16.830 --> 0:35:18.900  
Anand sivaraman  
I've got, yes, false negative.

0:35:18.910 --> 0:35:20.0  
Anand sivaraman  
Very good, right?

0:35:20.50 --> 0:35:25.860  
Anand sivaraman  
And then if you see this is both are true, I mean both are correct one is.

0:35:25.950 --> 0:35:32.610  
Anand sivaraman  
So this is true positive, correct and then probably if you see down the line you will probably see.

0:35:34.310 --> 0:35:35.210  
Anand sivaraman  
False positives.

0:35:35.220 --> 0:35:37.740  
Anand sivaraman  
Somewhere zero and one will be false positive.

0:35:37.740 --> 0:35:39.410  
Anand sivaraman  
There should be a false positive value Subbaiah.

0:35:39.420 --> 0:35:39.960  
Anand sivaraman  
OK.

0:35:40.0 --> 0:35:45.200  
Anand sivaraman  
So so we are we are predicting the algorithm has predicted this way.

0:35:45.210 --> 0:35:54.910  
Anand sivaraman  
So if you look at the scorer, the scorer giving 78.882%, so how do I improve the model?

0:35:54.960 --> 0:36:2.550  
Anand sivaraman  
One way to improve the model is one way is here I had originally given 5 records per node.

0:36:3.40 --> 0:36:9.510  
Anand sivaraman  
The moment I increase the sample size to 10, right, if I increase the sample size to 10, let us see what has happened.

0:36:9.860 --> 0:36:10.450  
Anand sivaraman  
What happens?

0:36:11.740 --> 0:36:20.580  
Anand sivaraman  
And what I have done is I've increased the sample size to 10, use Genie index 10 and number of threads is 200.

0:36:20.590 --> 0:36:22.970  
Anand sivaraman  
So here I forgot to tell you about threads.

0:36:22.980 --> 0:36:24.980  
Anand sivaraman  
Threads is basically parallel processing.

0:36:25.150 --> 0:36:28.940  
Anand sivaraman  
OK, how many threads you want to run?

0:36:29.70 --> 0:36:32.780  
Anand sivaraman  
So this is actually giving more memory for the algorithm to perform.

0:36:32.790 --> 0:36:34.40  
Anand sivaraman  
So I've given a number of threads.

0:36:34.50 --> 0:36:35.780  
Anand sivaraman  
Is 200 OK?

0:36:35.830 --> 0:36:39.680  
Anand sivaraman  
I have increased the number of records per node to 10 from 5 it is 10.

0:36:39.930 --> 0:36:41.800  
Anand sivaraman  
So let's let's use the same thread.

0:36:41.810 --> 0:36:51.450  
Anand sivaraman  
OK, let's use whatever we use in the earlier here as well, just to see if the algorithm is really, uh, looking right?

0:36:51.540 --> 0:36:54.50  
Anand sivaraman  
So I mean if there is, if it's improving or not.

0:36:54.520 --> 0:37:9.630  
Anand sivaraman  
So in this case, yes, just by improving, just by increasing the number of records per node from 5 to 10, what has happened, you see a better score from 78 with the score has increased to 81.67.

0:37:10.80 --> 0:37:23.650  
Anand sivaraman  
Now what you can do, you can further improve the model and in this case, if you see ten, I increase the number of 10 threads to 200 right and then uh, you know, let us see if there is any change at all.

0:37:24.120 --> 0:37:24.450  
Anand sivaraman  
Right.

0:37:24.740 --> 0:37:25.830  
Anand sivaraman  
So when you look at that.

0:37:31.10 --> 0:37:36.630  
Anand sivaraman  
So 83, so number of threads has improved the performance a little more 83%, right?

0:37:36.710 --> 0:37:41.960  
Anand sivaraman  
So you can improve it further if you want increase the number so one of one of the other things.

0:37:42.10 --> 0:37:47.280  
Anand sivaraman  
OK, in this case what I changed is I actually changed the quality measure.

0:37:47.450 --> 0:37:53.620  
Anand sivaraman  
Remember, in the last two scenarios I only set up Gini index, but here I change the quality measure.

0:37:54.90 --> 0:38:3.510  
Anand sivaraman  
Let me keep this number, the thread measurement, multi threading or number of threads to only 16 OK and if I run it like this.

0:38:6.160 --> 0:38:8.460  
Anand sivaraman  
Right, it is still giving me 83.

0:38:8.470 --> 0:38:12.620  
Anand sivaraman  
So without even multi threading it is giving me 83% which is which is good right?

0:38:12.720 --> 0:38:18.900  
Anand sivaraman  
So what I can do is I can start improving based on this, maybe with gain ratio.

0:38:19.190 --> 0:38:21.680  
Anand sivaraman  
I've already on same records per node.

0:38:21.690 --> 0:38:22.450  
Anand sivaraman  
It is improving.

0:38:22.610 --> 0:38:24.90  
Anand sivaraman  
So if I say 50 records.

0:38:26.810 --> 0:38:29.280  
Anand sivaraman  
Is that something that can be doable?

0:38:29.470 --> 0:38:31.700  
Anand sivaraman  
Let's see if that gives better outcomes.

0:38:32.70 --> 0:38:34.520  
Anand sivaraman  
Yeah, 83.66 is what is coming.

0:38:34.530 --> 0:38:40.560  
Anand sivaraman  
So beyond a point, I doubt if the algorithm can be, uh, I'm.

0:38:40.570 --> 0:38:43.800  
Anand sivaraman  
I'm just changing the pruning method again just to see if this.

0:38:48.330 --> 0:38:51.40  
Anand sivaraman  
83.45 Right, it's slightly increasing.

0:38:51.210 --> 0:38:59.460  
Anand sivaraman  
So maybe I can I can increase this to 100 or 150 if I want to to see if that's making any difference.

0:39:4.850 --> 0:39:7.400  
Anand sivaraman  
It to the point forward 6 nothing, nothing much is happening.

0:39:7.410 --> 0:39:18.510  
Anand sivaraman  
So the algorithm decision Tree algorithm is no more capable of improving, at least for this data set with the given set of parameters.

0:39:18.920 --> 0:39:23.250  
Anand sivaraman  
Maybe maybe one another aspect that we can do is.

0:39:24.820 --> 0:39:33.630  
Anand sivaraman  
So let's say let's say if this works OK, one of the other parameters that we can handle is we can change the partition from draw random to stratified sampling.

0:39:34.200 --> 0:39:35.680  
Anand sivaraman  
What does stratified sampling mean?

0:39:38.770 --> 0:39:39.370  
Anand sivaraman  
We saw this.

0:39:39.20 --> 0:39:41.220  
Shubham Kumar  
Sims, the type of data similar type of data.

0:39:44.500 --> 0:39:46.180  
Shubham Kumar  
Will be grouped or yeah.

0:39:42.240 --> 0:39:48.520  
Anand sivaraman  
So Kumar type of data spread spread evenly across test, training and test.

0:39:49.500 --> 0:39:49.690  
Shubham Kumar  
Yeah.

0:39:50.780 --> 0:39:53.910  
Anand sivaraman  
OK, so for example it is not similar type of data.

0:39:56.480 --> 0:39:57.560  
Muniaraj Thangavelu  
Instead of random.

0:39:53.980 --> 0:39:59.490  
Anand sivaraman  
I mean not only similar type of data but data with the huh?

0:39:59.160 --> 0:39:59.740  
Muniaraj Thangavelu  
Ohh sorry.

0:40:0.880 --> 0:40:1.310  
Anand sivaraman  
Good, good.

0:40:2.70 --> 0:40:7.240  
Muniaraj Thangavelu  
Randomly it will pick but it should not be skewed in a particular.

0:40:7.250 --> 0:40:11.540  
Muniaraj Thangavelu  
One way it will be evenly distributed for all the scenarios.

0:40:11.550 --> 0:40:13.310  
Muniaraj Thangavelu  
That's the status.

0:40:13.0 --> 0:40:14.850  
Anand sivaraman  
Yes, yes, correct.

0:40:14.890 --> 0:40:17.900  
Anand sivaraman  
So if you look at the data explorer here.

0:40:22.440 --> 0:40:29.150  
Anand sivaraman  
To look at Data Explorer right and you look at this particular data ideally.

0:40:29.200 --> 0:40:32.470  
Anand sivaraman  
Ideally, if you are looking at random, this is this.

0:40:32.480 --> 0:40:33.830  
Anand sivaraman  
Partitioning is done randomly.

0:40:33.840 --> 0:40:46.950  
Anand sivaraman  
OK, so if you look at the Explorer view and you try to understand this nominal column, you see that nearly 224,000 records and 7000 records are there.

0:40:47.60 --> 0:41:0.300  
Anand sivaraman  
This is without stratified sampling, but if I do start if at sampling right and I choose the column, the target variable, I can do stratified sampling only on the target variable using the target variable.

0:41:0.310 --> 0:41:4.260  
Anand sivaraman  
I'm splitting the data set in a very stratified manner, right?

0:41:4.490 --> 0:41:10.420  
Anand sivaraman  
If I use the data explorer to view it, I may I may try to see some differences right?

0:41:10.430 --> 0:41:16.200  
Anand sivaraman  
So, for example, uh, let's say partitioning.

0:41:22.40 --> 0:41:22.990  
Anand sivaraman  
Just give me one second, OK?

0:41:36.410 --> 0:41:42.100  
Anand sivaraman  
So what I do is if I have this data, I'm using the training data here.

0:41:45.150 --> 0:41:58.400  
Anand sivaraman  
So if I sample this training data or if I if I stratify this data through this my training and test data sets are going to be having uh spread out equal spread out.

0:41:58.530 --> 0:42:17.720  
Anand sivaraman  
So in this case I'm just connecting to the training data here, so I might still see some value 24526 and 7018 slightly different, but ideally you should understand the stratified sampling will make sure that the difference between training and test is evenly balanced or not even even it's balanced in some form, right?

0:42:17.890 --> 0:42:29.360  
Anand sivaraman  
So you can look at the data Explorer view to understand this better, but anyway I am using stratified sampling to see if, let's say the model is giving me better results or not.

0:42:29.420 --> 0:42:32.790  
Anand sivaraman  
So this is this is with stratified sampling.

0:42:41.850 --> 0:42:43.20  
Anand sivaraman  
OK, so when I do this.

0:42:46.880 --> 0:42:58.730  
Anand sivaraman  
83.379 not much of a difference, so this is probably where the model you have to think about some other options rather than using the hyperparameter, you might want to think about feature engineering.

0:42:58.740 --> 0:43:4.190  
Anand sivaraman  
You might want to think about other things, so as of now this is where it is now.

0:43:4.200 --> 0:43:17.330  
Anand sivaraman  
This is a good time to move to on semble methods because you are, because you think that because we think that the decision tree model has reached its limit, we will be going to ensemble method.

0:43:18.30 --> 0:43:20.760  
Anand sivaraman  
So on Kamble methods to see if it is performing.

0:43:20.770 --> 0:43:25.740  
Anand sivaraman  
I'll show the ensemble methods same stuff instead of decision tree I'm going to use random forest.

0:43:26.150 --> 0:43:30.720  
Anand sivaraman  
Random Forest is an ensemble tree based method and it has these parameters.

0:43:30.770 --> 0:43:35.510  
Anand sivaraman  
I select the target column and then I select the columns that need.

0:43:38.160 --> 0:43:43.650  
Anand sivaraman  
To be included and then and then I have these limits, right?

0:43:43.660 --> 0:43:45.510  
Anand sivaraman  
I have limit number of levels.

0:43:45.830 --> 0:43:47.590  
Anand sivaraman  
What is tree depth here?

0:43:47.840 --> 0:43:55.470  
Anand sivaraman  
Tree depth is if, let's say here one root node, one branch node, so that the tree is 1 level deep.

0:43:55.680 --> 0:44:3.490  
Anand sivaraman  
If let's say there are, there are five branch notes, then you can say the tree is 6 level deep or sorry five level deep right?

0:44:3.600 --> 0:44:5.950  
Anand sivaraman  
So we are saying tree depth can be 10 levels.

0:44:5.960 --> 0:44:9.550  
Anand sivaraman  
So which means I am allowing for 10 branch nodes to be created right?

0:44:9.740 --> 0:44:17.560  
Anand sivaraman  
So 10, which means the more the more depth you have, the larger the Team tree will be and it will also take more time.

0:44:19.530 --> 0:44:22.400  
Anand sivaraman  
Which is OK, but still minimum node size.

0:44:22.410 --> 0:44:25.420  
Anand sivaraman  
I've put as one here number of models.

0:44:25.430 --> 0:44:28.150  
Anand sivaraman  
This is the main difference between Decision tree and random forest.

0:44:29.400 --> 0:44:31.270  
Anand sivaraman  
Random Forest is an ensemble method.

0:44:31.710 --> 0:44:37.970  
Anand sivaraman  
What is an ensemble method when you have more than one model running on the same data set, right?

0:44:38.420 --> 0:44:39.950  
Anand sivaraman  
You can call it as ensemble.

0:44:40.80 --> 0:44:42.210  
Anand sivaraman  
OK, ensemble methods are very interesting.

0:44:42.220 --> 0:44:43.870  
Anand sivaraman  
I'll first explain random forest.

0:44:44.220 --> 0:44:53.650  
Anand sivaraman  
I just say number of models is 100 which means I am running hundred decision tree models under this one random forest model random forest.

0:44:53.700 --> 0:44:54.30  
Anand sivaraman  
Why?

0:44:54.40 --> 0:44:57.370  
Anand sivaraman  
Why is it called as a forest when you're running more than one tree?

0:44:57.420 --> 0:44:59.680  
Anand sivaraman  
Many trees form a forest, right?

0:44:59.760 --> 0:45:3.820  
Anand sivaraman  
So decision tree when hundred decision trees are running, it forms a.

0:45:4.90 --> 0:45:5.100  
Anand sivaraman  
It forms a forest.

0:45:5.290 --> 0:45:6.480  
Anand sivaraman  
Why is it called random?

0:45:6.690 --> 0:45:11.130  
Anand sivaraman  
Because you are not supervising it, the data is randomly chosen.

0:45:11.200 --> 0:45:14.190  
Anand sivaraman  
The manner in which the models are picked up or randomly chosen.

0:45:14.380 --> 0:45:16.110  
Anand sivaraman  
So that's why it's called as random forest.

0:45:16.360 --> 0:45:27.850  
Anand sivaraman  
So list the list of number of levels is basically the depth of the tree as if you have many branch nodes it will be deeper right and minimum node size, 1 record per node, right and number of models.

0:45:27.860 --> 0:45:30.230  
Anand sivaraman  
How many decision tree models you are going to run?

0:45:33.670 --> 0:45:51.360  
Anand sivaraman  
So with all of this, I'm I'm running the same data and if you see if you see my output and my prediction right, you will see that immediately without even doing much of hyperparameter tuning, I'm already getting better results than they shun tree.

0:45:51.570 --> 0:46:1.420  
Anand sivaraman  
OK, so random forest is very powerful when it comes to improving the limitations when it comes to resolving the limitations that the decision tree have beyond a certain point that we cannot grow.

0:46:1.550 --> 0:46:4.620  
Anand sivaraman  
But in this case, random forest allows you to do this right?

0:46:4.750 --> 0:46:12.620  
Anand sivaraman  
And with all of this, I can probably, if you see, I can probably in this case I did not include the.

0:46:12.990 --> 0:46:18.630  
Anand sivaraman  
I only had 10 numbers and 100 models, hundred decision tree models and ensemble.

0:46:19.260 --> 0:46:20.710  
Anand sivaraman  
Here I'm.

0:46:21.280 --> 0:46:33.20  
Anand sivaraman  
I'm gonna have a 14 levels, which means 14 branch nodes, minimum Rd size is 13 and I'm gonna have 110 models right?

0:46:33.30 --> 0:46:39.160  
Anand sivaraman  
So when I when I change that parameter and then I'm going to have 110 or 114 models.

0:46:39.170 --> 0:46:39.670  
Anand sivaraman  
Or 114 models.

0:46:39.680 --> 0:46:40.600  
Anand sivaraman  
14 models.

0:46:41.80 --> 0:46:57.400  
Anand sivaraman  
Let's see how the output is so 84.244 it is improving slowly right here without all of this, with nothing, your model is able to at least predict better 8484.193 was the prior output with random forest.

0:46:57.790 --> 0:47:0.780  
Anand sivaraman  
Now the new one is 84.244, right?

0:47:0.850 --> 0:47:12.390  
Anand sivaraman  
So I'm pretty sure that if we go to the data and then add more information in terms of the data itself, umm you will be you will be seeing a better result right?

0:47:12.400 --> 0:47:17.990  
Anand sivaraman  
So when I say go to the data here, let's say if I increase it from 110 to 1:20 or 1:30 right?

0:47:18.260 --> 0:47:36.30  
Anand sivaraman  
So I'm I'm talking about 136 models, so when you look at 136 models and looking at the output, you see 84.23, so the accuracy might increase if you do this and one last approach that I have is called as bagging and I'll explain bagging in a bit.

0:47:36.100 --> 0:47:38.60  
Anand sivaraman  
So I'll explain bagging in a bit.

0:47:38.480 --> 0:47:38.660  
Anand sivaraman  
So.

0:47:39.230 --> 0:47:41.820  
Anand sivaraman  
Umm umm.

0:47:42.340 --> 0:47:47.50  
Anand sivaraman  
So with that bagging is called as bootstrap aggregation.

0:47:47.180 --> 0:47:56.230  
Anand sivaraman  
OK, So what this bagging, if you look at this, what I'm doing is I'm taking the same data, whatever data that I passed here, right?

0:47:56.280 --> 0:47:57.850  
Anand sivaraman  
I've created a meta node out of this.

0:47:57.860 --> 0:48:5.650  
Anand sivaraman  
Sorry, whatever data I took here, I'm taking the same data and what I'm doing is I'm doing some interesting stuff.

0:48:6.100 --> 0:48:8.250  
Anand sivaraman  
My bagging workflow looks like this right?

0:48:8.380 --> 0:48:10.430  
Anand sivaraman  
So when I click on this, it looks like this.

0:48:10.640 --> 0:48:28.330  
Anand sivaraman  
So I I shuffled the data all the data and then I send the I send the data into a loop, OK and in this loop I'm saying chunk the data whatever records I have remember bagging means bootstrap aggregation.

0:48:29.350 --> 0:48:36.200  
Anand sivaraman  
So what this means is I'm taking the data I'm splitting the data into chunks, chunks means apportion.

0:48:36.530 --> 0:48:40.320  
Anand sivaraman  
I am sending each portion into a separate decision tree model.

0:48:41.480 --> 0:48:46.90  
Anand sivaraman  
Think about it as I have a think about it as something like this guys right?

0:48:46.240 --> 0:48:47.420  
Anand sivaraman  
So I have a.

0:48:48.550 --> 0:48:50.80  
Anand sivaraman  
You know what does random what?

0:48:50.90 --> 0:48:51.80  
Anand sivaraman  
What is random forest?

0:48:51.90 --> 0:48:54.200  
Anand sivaraman  
I'll I'll explain random Forest now and then I'll explain bagging, OK.

0:48:54.290 --> 0:48:59.560  
Anand sivaraman  
So if you look at the random forest approach, why is it called as an ensemble method?

0:48:59.750 --> 0:49:7.860  
Anand sivaraman  
It's called as an ensemble method because it actually allows for multiple models to run, and in this case it allows for multiple decision trees to run.

0:49:8.130 --> 0:49:9.900  
Anand sivaraman  
So I have many decision trees.

0:49:10.270 --> 0:49:22.880  
Anand sivaraman  
Each tree will make a prediction and all the predictions will be put together under voting will happen, but whichever class is predicted by the most number of trees that will be the final class.

0:49:23.10 --> 0:49:25.320  
Anand sivaraman  
OK, so that is how random forest works.

0:49:25.330 --> 0:49:29.120  
Anand sivaraman  
OK, so for example let me let me add.

0:49:29.270 --> 0:49:32.420  
Anand sivaraman  
Let me give this information easily to be easy to consume.

0:49:32.590 --> 0:49:34.20  
Anand sivaraman  
So how does random forest work?

0:49:34.30 --> 0:49:35.700  
Anand sivaraman  
How does the decision tree work?

0:49:35.870 --> 0:49:37.150  
Anand sivaraman  
Decision tree is very simple.

0:49:37.160 --> 0:49:45.640  
Anand sivaraman  
We take one data set, we take one data set and in the same data set you have a a data set.

0:49:45.650 --> 0:50:1.350  
Anand sivaraman  
You pass the data set to a decision tree and then for each row in the data set, if the data set has multiple rows, let's say all the way till row hundred, you will have a actual value, actual target variable and the predicted target variable right?

0:50:1.640 --> 0:50:7.640  
Anand sivaraman  
So you will have for each of the rows you will have prediction and then from here you will calculate the accuracy and the error.

0:50:7.650 --> 0:50:14.630  
Anand sivaraman  
This is how lesion tree works, but the way random Forest works is very different in the random forest.

0:50:14.640 --> 0:50:22.590  
Anand sivaraman  
Here if you see we had mentioned that you know the number of models is 100 here correct.

0:50:22.820 --> 0:50:28.260  
Anand sivaraman  
If you see the number of models we have mentioned here as hundred hundred models correct.

0:50:30.700 --> 0:50:33.240  
Anand sivaraman  
We have mentioned the model number of model AS100 models.

0:50:35.290 --> 0:50:37.280  
Anand sivaraman  
100 or 130 or whatever.

0:50:37.350 --> 0:50:38.950  
Anand sivaraman  
100 plus models here, correct?

0:50:39.430 --> 0:50:43.720  
Anand sivaraman  
So if you look at it 100 models, how will the algorithm work?

0:50:43.770 --> 0:50:44.660  
Anand sivaraman  
Very simple.

0:50:44.950 --> 0:50:51.490  
Anand sivaraman  
The same data set, OK, same data set will be passed to the random forest model.

0:50:51.880 --> 0:51:12.110  
Anand sivaraman  
What the random forest model will do is for each row in the data set, it will run against hundred decision trees, OK, and then for each of those decision trees it will do a prediction, but each at all prediction 1 prediction, one for hundred decision trees.

0:51:13.30 --> 0:51:13.510  
Anand sivaraman  
OK.

0:51:13.590 --> 0:51:17.100  
Anand sivaraman  
Similarly, for each row we'll do prediction 2, prediction 2 whatever.

0:51:17.270 --> 0:51:28.780  
Anand sivaraman  
Right now, after all of this, let's say row 100 Rao 100, it will predict prediction 100 prediction 100 for each of the model.

0:51:29.30 --> 0:51:33.400  
Anand sivaraman  
It will predict now what will happen is it will do what is called as majority voting.

0:51:37.980 --> 0:51:39.30  
Anand sivaraman  
And already voting.

0:51:39.400 --> 0:51:43.360  
Anand sivaraman  
What majority voting does is out of the 100 models.

0:51:43.430 --> 0:51:48.720  
Anand sivaraman  
How many models had predicted the same value or the same class?

0:51:48.950 --> 0:51:55.820  
Anand sivaraman  
So for example, if we say, let's take this right instead of prediction one, let's take class zero, class one.

0:51:55.880 --> 0:52:3.600  
Anand sivaraman  
OK, we'll call it as we'll call class Zero as let us call 0 is delinquent.

0:52:6.520 --> 0:52:6.810  
Anand sivaraman  
Sorry.

0:52:10.40 --> 0:52:12.650  
Anand sivaraman  
Let's call 0 as delinquent, right?

0:52:13.20 --> 0:52:14.390  
Anand sivaraman  
Let's call 0 as delinquent.

0:52:16.660 --> 0:52:18.510  
Anand sivaraman  
And one as non delinquent.

0:52:19.460 --> 0:52:19.860  
Anand sivaraman  
OK.

0:52:20.310 --> 0:52:22.160  
Anand sivaraman  
Are are let's take the other way around. Sorry.

0:52:24.590 --> 0:52:31.600  
Anand sivaraman  
Let's say 0 is non delinquent, one is delinquent because we are going to predict whether the customer is going to be delinquent or not, right?

0:52:31.790 --> 0:52:36.700  
Anand sivaraman  
So let's say low one, row two, row 3 all the way till row 100.

0:52:36.710 --> 0:52:42.820  
Anand sivaraman  
The dataset has in row one for Decision Tree, one for the for the values of row one.

0:52:43.410 --> 0:52:51.540  
Anand sivaraman  
It is predicting 0 in this country, one in Decision Tree two it is predicting probably one in Decision Tree 3-4.

0:52:51.710 --> 0:52:55.60  
Anand sivaraman  
It is predicting 11 and Asian 300.

0:52:55.70 --> 0:53:8.840  
Anand sivaraman  
Also it is predicting 1, So what the algorithm will do is in the majority voting algorithm will find that nearly out of the 100 models, nearly 80 models are 70 models are predicting one more than zero.

0:53:8.990 --> 0:53:19.920  
Anand sivaraman  
So the majority voting will finally predict one, let's say for row two the prediction is 00000 and then probably let's say one or two models predict one.

0:53:20.150 --> 0:53:25.120  
Anand sivaraman  
So in this case the majority is 0, so the final outcome will also be final.

0:53:25.130 --> 0:53:26.600  
Anand sivaraman  
Majority voting will also be 0.

0:53:26.770 --> 0:53:31.240  
Anand sivaraman  
Similarly, for each of the row, the random forest algorithm will predict.

0:53:31.570 --> 0:53:43.510  
Anand sivaraman  
So which means what it is not only predicting once, it is predicting 100 times and then finally it is putting together all the predictions through a voting process and then finally comes out with the voting.

0:53:43.600 --> 0:53:44.180  
Anand sivaraman  
Is that clear?

0:53:44.190 --> 0:53:44.830  
Anand sivaraman  
Are you guys with me?

0:53:46.800 --> 0:53:47.200  
Muniaraj Thangavelu  
Yes, hello.

0:53:48.120 --> 0:53:50.190  
Anand sivaraman  
OK, now what is the advantage of this?

0:53:50.340 --> 0:53:51.450  
Anand sivaraman  
Now, what is the advantage of this?

0:53:52.270 --> 0:53:55.450  
Anand sivaraman  
Uh, do you all know what is distributed computing?

0:53:59.630 --> 0:53:59.880  
Shubham Kumar  
Yeah.

0:53:59.890 --> 0:54:1.640  
Shubham Kumar  
Where the computation happens.

0:54:2.130 --> 0:54:2.340  
Shubham Kumar  
Uh.

0:54:2.350 --> 0:54:3.950  
Shubham Kumar  
Parallelly on different computers.

0:54:5.700 --> 0:54:7.0  
Anand sivaraman  
But that's all.

0:54:7.390 --> 0:54:19.10  
Anand sivaraman  
So when when you have a client server setup, computation happens across very many different computers, the distribution of the computation happens across the board.

0:54:19.20 --> 0:54:21.830  
Anand sivaraman  
All cloud services work that way, right?

0:54:21.840 --> 0:54:27.210  
Anand sivaraman  
It the distribution happens across very many computers now, what is the advantage of distributed computing?

0:54:27.540 --> 0:54:36.680  
Anand sivaraman  
The advantage of distributed computing is even if one system goes down, the other systems are able to support and run the workflow, correct?

0:54:36.920 --> 0:54:47.700  
Anand sivaraman  
So for example, if all of you must have used IRCTC website for booking a ticket, I think IRCTC website is one of the best websites that India has ever built because it's highly distributed.

0:54:47.870 --> 0:54:49.820  
Anand sivaraman  
It is super fluid and it's very good, right?

0:54:49.830 --> 0:54:56.260  
Anand sivaraman  
There are scenarios where obviously the kind of volume that Indian Railways handles, it's unbelievably good.

0:54:56.270 --> 0:54:57.400  
Anand sivaraman  
I would say right.

0:54:57.930 --> 0:55:6.400  
Anand sivaraman  
There are always issues, but I've not seen something that perfect, but IRCTC will fail miserably if it is running on only one computer.

0:55:6.650 --> 0:55:9.900  
Anand sivaraman  
It is running on hundreds of parallel computers, right?

0:55:9.910 --> 0:55:15.770  
Anand sivaraman  
That's why it's able to take the load and even if one computer fails, that concept is called as fault tolerance.

0:55:16.90 --> 0:55:19.170  
Anand sivaraman  
Even if one computer fails, the other computers are able to manage.

0:55:19.180 --> 0:55:25.610  
Anand sivaraman  
The other servers are able to manage and pick up and run the show similarly, in this case, random forest is a distributed modeling.

0:55:25.980 --> 0:55:27.350  
Anand sivaraman  
That's why it's called ansamble.

0:55:27.480 --> 0:55:33.790  
Anand sivaraman  
It's called distributed modeling, so even if one model predicts very poorly, let's say this model predicts very poorly.

0:55:33.960 --> 0:55:41.130  
Anand sivaraman  
The other models will observe offset this model's performance so that the overall outcomes will always be good, right?

0:55:41.140 --> 0:55:47.910  
Anand sivaraman  
Many a times, that is why random forest is called as called as but nashia.

0:55:51.80 --> 0:55:56.600  
Anand sivaraman  
It's an English term for a medicine that can cure all the diseases, right?

0:55:56.800 --> 0:56:8.150  
Anand sivaraman  
So random forest is called as a panacea for all machine learning problems, because when it got introduced, it solved so many problems that the other supervised learning models were not able to solve one.

0:56:8.260 --> 0:56:24.260  
Anand sivaraman  
One such problem is offsetting the issues or the problems or the errors that one model is having by but with the help of the other models other issue other thing is decision tree models are very susceptible, are prone to get affected.

0:56:27.50 --> 0:56:27.820  
Anand sivaraman  
By outliers.

0:56:29.60 --> 0:56:30.70  
Anand sivaraman  
What are outliers?

0:56:30.340 --> 0:56:35.290  
Anand sivaraman  
Outliers are extreme values, random forest because it is running on multiple models.

0:56:35.560 --> 0:56:41.930  
Anand sivaraman  
It is extreme values, so random forest is not getting affected by will not get affected by.

0:56:43.830 --> 0:56:44.640  
Anand sivaraman  
Outliers, right?

0:56:44.650 --> 0:56:48.260  
Anand sivaraman  
So random Forest is a very powerful tree based ensemble model.

0:56:48.480 --> 0:56:50.600  
Anand sivaraman  
So why is it called random forest?

0:56:51.30 --> 0:56:54.280  
Anand sivaraman  
It is made of many decision trees, so it is called as a forest.

0:56:54.620 --> 0:57:6.460  
Anand sivaraman  
But remember, Random Forest is not just the only one simple method, it tangent for us is just one of the many ensemble methods and random forest is a tree based ensemble method.

0:57:6.750 --> 0:57:8.380  
Anand sivaraman  
OK, tree based ensemble method.

0:57:8.710 --> 0:57:12.200  
Anand sivaraman  
Now, what is an ensemble method or an ensemble technique?

0:57:12.450 --> 0:57:15.850  
Anand sivaraman  
An ensemble technique is typically something that is made of multiple models.

0:57:16.330 --> 0:57:18.810  
Anand sivaraman  
OK, what is an ensemble technique?

0:57:19.100 --> 0:57:21.330  
Anand sivaraman  
An ensemble technique is made of multiple models.

0:57:25.260 --> 0:57:26.260  
Anand sivaraman  
They made of.

0:57:28.610 --> 0:57:32.300  
Anand sivaraman  
Made of multiple sorry about my handwriting.

0:57:32.770 --> 0:57:34.490  
Anand sivaraman  
Multiple models.

0:57:34.830 --> 0:57:36.100  
Anand sivaraman  
So what does that mean?

0:57:36.310 --> 0:57:45.390  
Anand sivaraman  
So if I say in in the random forest, I have decision Tree one decision tree, two many decision tree models right?

0:57:46.190 --> 0:57:48.730  
Anand sivaraman  
Similarly, in another ensemble model, right?

0:57:48.740 --> 0:57:50.790  
Anand sivaraman  
Let's call this as ensemble model one.

0:57:51.310 --> 0:57:53.20  
Anand sivaraman  
I can have logistic regression.

0:57:53.90 --> 0:57:55.640  
Anand sivaraman  
I can have NAV by NAV bias.

0:57:55.940 --> 0:58:8.670  
Anand sivaraman  
I can have K nearest neighbors and then I can have let's say a decision tree and then I can have random forest and then I can ansamble put all of these together right to create an ensemble.

0:58:8.880 --> 0:58:13.270  
Anand sivaraman  
This is the this the this is another method of on several model.

0:58:13.600 --> 0:58:20.940  
Anand sivaraman  
The third method of ensemble model can also be something called as transfer learning transfer learning.

0:58:21.290 --> 0:58:26.40  
Anand sivaraman  
In a way, it is ansamble, but it cannot be completely called that way.

0:58:26.50 --> 0:58:27.570  
Anand sivaraman  
But in a way, I believe it's on sample.

0:58:27.580 --> 0:58:28.220  
Anand sivaraman  
What does that mean?

0:58:28.230 --> 0:58:32.760  
Anand sivaraman  
It means so I have my data set OK and in that data set.

0:58:36.150 --> 0:58:48.470  
Anand sivaraman  
And in that data set what I do is I in that data set what I do is I start building first or I run the data set against logistic regression.

0:58:48.840 --> 0:58:50.530  
Anand sivaraman  
Output of logistic regression.

0:58:50.820 --> 0:58:54.230  
Anand sivaraman  
I am putting it through naive bias output of naive bias.

0:58:54.320 --> 0:58:57.10  
Anand sivaraman  
I am putting it through KN output of KN.

0:58:57.20 --> 0:58:58.110  
Anand sivaraman  
I'm putting it through the Sherry.

0:58:58.120 --> 0:59:1.630  
Anand sivaraman  
Hypothetically, think of it this way and finally I get an outcome.

0:59:1.720 --> 0:59:5.90  
Anand sivaraman  
So there are multiple models model chain that is involved.

0:59:5.100 --> 0:59:6.510  
Anand sivaraman  
Some of you who are LLM.

0:59:6.520 --> 0:59:7.970  
Anand sivaraman  
Knowledge on large language models.

0:59:8.80 --> 0:59:9.570  
Anand sivaraman  
There's something called as LLM chain.

0:59:9.720 --> 0:59:11.590  
Anand sivaraman  
Similarly, a model chain is involved.

0:59:11.900 --> 0:59:17.970  
Anand sivaraman  
The output of 1 model is transferred to the input of another model transferred transferred here right?

0:59:18.40 --> 0:59:18.940  
Anand sivaraman  
And this is called as.

0:59:19.630 --> 0:59:22.960  
Anand sivaraman  
This is also one another form of ensemble method, right?

0:59:22.970 --> 0:59:26.580  
Anand sivaraman  
But mostly we will be focusing on random forest Ensemble method here.

0:59:26.630 --> 0:59:27.750  
Anand sivaraman  
Is that clear or you guys with me?

0:59:32.780 --> 0:59:33.220  
Ashok Reddy  
There's enough.

0:59:33.310 --> 0:59:33.860  
Bhagyalakshmi Ganapathy  
Yes, Anna.

0:59:33.510 --> 0:59:33.990  
Muniaraj Thangavelu  
Yes, ma'am.

0:59:34.610 --> 0:59:37.140  
Anand sivaraman  
OK, So what we'll do, we'll stop here tomorrow.

0:59:37.150 --> 0:59:41.360  
Anand sivaraman  
We will pick up on the 1 aspect of random forest called bagging.

0:59:42.120 --> 0:59:44.940  
Anand sivaraman  
I'll explain that and then we could take it forward from there, OK.

0:59:48.700 --> 0:59:49.140  
Shubham Kumar  
OK cannon.

0:59:48.70 --> 0:59:53.540  
Anand sivaraman  
So if you see here through bagging through bagging, I'm able to get about a better accuracy.

0:59:53.670 --> 0:59:55.280  
Anand sivaraman  
So we can always pick up from there.

0:59:59.820 --> 1:0:0.170  
Anand sivaraman  
OK.

1:0:0.220 --> 1:0:1.450  
Anand sivaraman  
So we'll connect tomorrow.

1:0:1.460 --> 1:0:1.960  
Anand sivaraman  
Thank you guys.

1:0:3.310 --> 1:0:3.690  
Bhagyalakshmi Ganapathy  
Thank you.

1:0:4.500 --> 1:0:5.50  
Muniaraj Thangavelu  
Thanks, honey.

1:0:5.60 --> 1:0:5.640  
Shubham Kumar  
Thank God no.

1:0:5.120 --> 1:0:5.900  
Muniaraj Thangavelu  
Thanks too.

1:0:5.730 --> 1:0:6.50  
Ashok Reddy  
Thank you.

1:0:5.840 --> 1:0:7.800  
Shubham Kumar  
And and now they because you're not.