

03-Live1 Decision Trees

Compiled by Shipra De, Fall 2016

- Okay, we're live now so everybody out of TV Land can see us which means the other 300 people in the class taking it online, so I want everybody here to wave at them. Hold on. Ok everybody. Okay. Hello out there.
- I'm going to turn on QA and I would appreciate it from someone out there if they can hear what we're saying that they post a questions and say yes I can hear. The first post is I'm not seeing questions. So you're right you should be able to hear it now. There's a lag of a couple minutes between what I say here and when you see it.
- Anyways today I'm going to start with the first of several things. I'll start with the first of several lectures on decision trees. There's going to be three. This first one is kind of a demo of decision trees and then some examples of how to use a decision tree if you had it.
- Once I finish that, I expect it's not going to take whole period, I'll try to allot some time to on-campus here talking about the lesson how to assess learners. But right now I'm going to focus on decision tree learning.



- Now decision trees can be used for classification and regression. And as a way to sort of demonstrate decision trees I have this fantastic piece of technology. Has anyone seen one of these before? No. Okay. You've seen one, really? It is a Jinn. You probably haven't seen this Jinn. I didn't even know it was related to something real.
- Okay, you may be wondering what is it. Okay, well packed into this little piece of plastic is [discussion with class]. Well so I'd never seen or heard of this thing until we were shopping at Wal-Mart one day and my daughter made me buy it. Just to get her to leave me alone I bought it. But it's like twenty dollars.
- Anyways, she started playing with it and I was amazed at what it could do. And you'll see in a minute. But anyways it has the ability to synthesize speech which really means

that it plays recorded messages. It can understand speech. Things like yes and no and it can reason to some extent.

- Now who's heard of the 20 questions, the animal guessing game? Here are the rules. I think the traditional American version is one person...uh...we do this a lot when we're like driving cross-country trying to keep the kids occupied, right? Like long road trip.
- Anyways, one kid thinks of something, and then the other kids get to ask questions one after the other to try and figure out what it is. And they have to ask good yes or no questions. So, as an example the game that this one plays, is it plays essentially the animal guessing game. So, you, the user, think of an animal and then it asks you questions, like, is it bigger than a microwave? All the questions are sort of binary questions. You'll hear the questions in a second. Through this process it eventually figures out which animal you're thinking of.
- Okay now to demonstrate this I need a volunteer from the audience. I have randomly selected three potential animals and I need a volunteer to select from these random animals. So you sir come on down. Okay I'll raise them up one at a time and you tell me when to stop.
- Okay, to make it more fun I should have another volunteer open up the envelope. As far as you know, I've put the same animal in all three. Now show it to them. I have to look at it too because I'm going to answer the questions. Well we don't want this thing to hear because it would be cheating. Okay, now we'll play the game.
- [Starts playing game]

>> You can say, Yes, No, Maybe, or it Depends

>> Does it have hair or fur? **Yes.**

>> Does this animal mainly eat plants, grass, and fruits? **Yes.**

>> Does it live mainly in Africa? **Yes.**

>> Very interesting. Does it live in the forest? **No.**

>> Ah ha. I see. Does it have hooves? **Yes.**

>> Does this animal have horns? **Yes.**

>> That's what I thought. Is this animal very big? **Yes.**

>> Alright. Is it friendly? **Yes.**

>> Okay. Does it have spots? **Yes.**

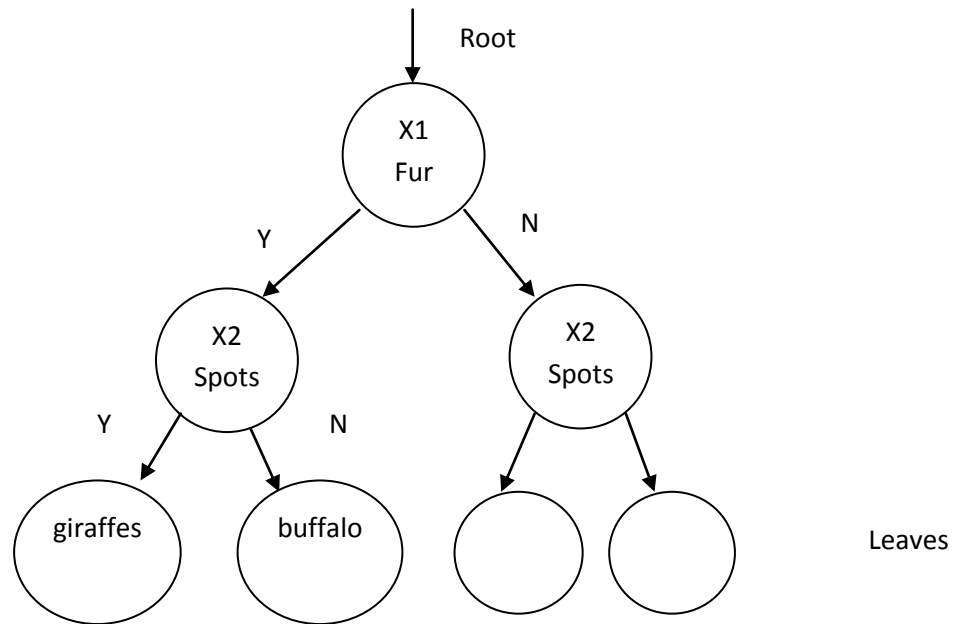
>> Very interesting. I know, don't tell me. If my powers aren't mistaken, you were thinking of a giraffe. Am I right? **Yes.**

- Wow. Okay I won't subject you to any more. The secret is to pull out the battery. Anyways my daughter memorized all of its sayings and so on. Now when we play 20 questions in the car, she's speaking with that accent.
- Okay now to be honest I'm not certain that inside this thing it uses a decision tree. I think it does but at least that's consistent with the way it behaves. I'm going to tell you how we could build a decision tree to solve this problem.
- [Audience question: if you gets some of the questions wrong, how does it work?]
- In my experience with this thing, if you get the question wrong then it degenerates. It continues going on but it gives up. One thing that's interesting about it that one of the optional answers is I don't know or it depends. So in some sense, if it is using decision trees, it actually is using decision trees with potentially three outgoing edges—yes, I don't know, and no. But for the next hour we're going to focus on binary decision trees. I'm gonna move the camera over to here.
- Okay somebody says they can hear us. That's great. We have 18 viewers. That's awesome. so I'm going to be switching back and forth from writing on the board and looking at my computer, so just bear with me when I do that.

Factors

X1	skin texture	fur, no fur
X2	spots	yes, no
X3	size	larger than microwave, larger than washing machine, very big

- If you think about the decision tree that this thing was using, you can try to figure out what's inside it. Well one thing it's got is a list of factors, right? So those are quantitative metrics that we can check and ask questions about that.
- Sometimes they are binary, so like does it have spots? Let me list a couple of them. I'll call x1 skin texture. So that's like fur or no fur. X2 might be spots. It frequently asks questions about size. So the three sorts of questions I've heard it ask about size are is it larger than a microwave? Is it larger than a washing machine? and is it very large or something like that. So essentially we have microwave, washing machine size, and very large.
- So you get the idea. So in its system somewhere it has a number of these factors and then it asks questions regarding them. Now my hypothesis is that internally it's got a decision tree for asking questions and evaluating these issues.



- So a decision tree works like this. It's composed of nodes and there's two types of nodes. There's decision nodes and leaves.
- One of these nodes is the root node. And inside each one of these decision nodes is essentially a binary question. So let's say it's going to ask a question about factor-1. So we wouldn't need to create some sort of data structure that remembers "oh this question is about factor-1" and then there's some sort of value that you split on.
- So in this case it's essentially fur and if the answer is yes we go down this way and if the answer is no we go down this way. And we continue down to another node each direction. And in our case we had fur, so maybe this next one is x2, no spots, and yes and no.
- I'm going to make sort of an abbreviated tree just for the purpose of this discussion because we're gonna have some more detailed trees in a moment. But let's say we only studied a few animals and the ones that have fur and have spots, those are always giraffes. And ones that have fur and don't have spots, those are say buffalo.
- And similarly we might have more branches down this right-hand side and these last nodes are our leaves.
- Now today I'm going to focus on talking about the structure of a decision tree and how to use one if you have one, but the next lecture is going to be about, suppose we have the data, how do we build a tree from that data?
- So like I said today we're focusing on how to use one if you have, but anyways, I wanted to identify these key parts of the decision tree. Let me pause for a second see if anyone has any questions. I'll check online too.

Decision Tree

Factors X1 X2 X3 ...

Labels Y

Nodes Factor used

Split value

Left link

Right link

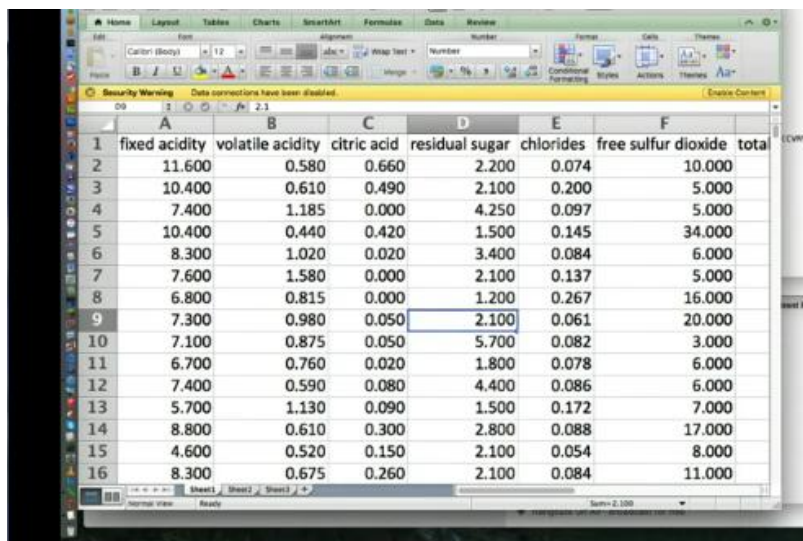
Root

Leaves

- Ok let me get my notes over here. So let me just formally write down again what makes up a decision tree. A long, long time ago I took a course on information theory and the professor could write on the board faster than I could write my notes. You're not gonna have that problem with me. But anyways I couldn't both keep up with him taking notes and understand what he was talking about so eventually I just quit taking notes and tried to understand him. But anyways, because I take notes so slowly, you can take notes.
- Okay so what makes up a decision tree? A list of factors and these are our Xs. And again there might be many of them. Might be x1, x2, x3...
- Labels. This is our Y. So that can either be a label like giraffe or it can be a label like 2.3. Could be numerical if we if we're building a regression tree.
- Then we have a set of nodes and each node includes the factor used and what I'm going to call a split val or split value. So we're looking at a particular value and then we're going to go one way down the tree if it's less than or equal to that value and another way down the tree if it's greater than that value.
- Also coming out of the nodes are essentially edges pointing us to the next nodes to the left and the right.
- Ok one of these nodes is designated as the root node and then finally we have our leaves. Okay everybody get that? Those are all the things you need to define a decision tree.
- Ok now I'm going to give you a concrete example of some of the data we can work with. Actually for the decision tree assignment coming up, we're going to use this data.
- Ok at UC Irvine there is a group of people and a website dedicated to providing data sets for machine learning testing. And they have a famous suite of many different types of data and the one that we're going to use is one concerning wine and I want to go ahead and cite properly; credit people who have created this. So I'll do it again when we

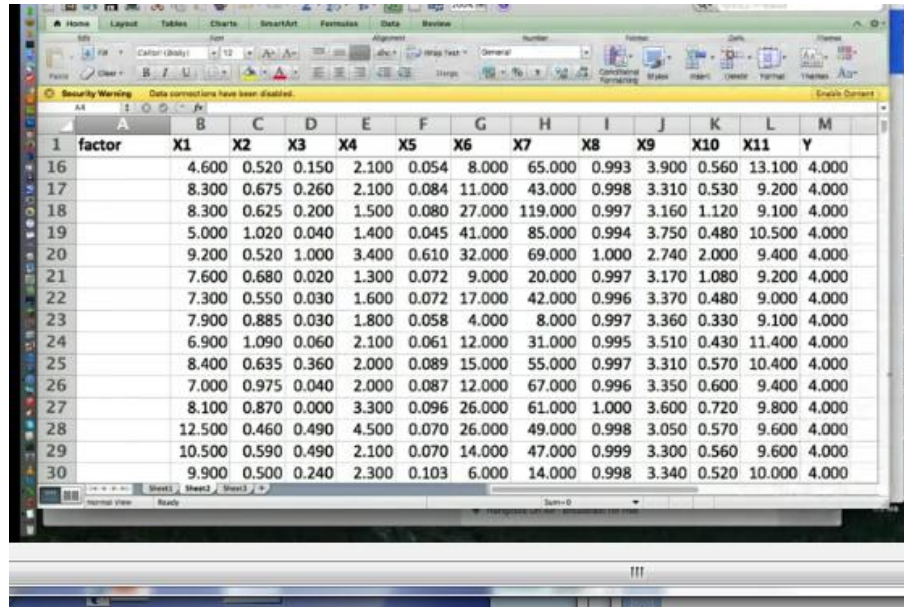
issue the assignment but anyways this is from a paper in 2009. Dataset is created by Cortez Cardeira. Anyways, they're Portuguese people and it turns out that they did the tests on port wine so that's the reason for the connection. Probably I'm guessing like on a grant from the port wine Federation Portugal or something. Anyways, thank-you for your data.

- Ok let's talk about the data and more properly, what they're trying to do. So I'm making a little bit of this up but it's consistent with whatever I've said so far.
- But they were looking for quantitative or sort of automated ways that they could evaluate the quality of wine. So what they did is they got a number of volunteers to taste several different Wines. I'm not sure how many different, but probably 10s or more and they asked each person to score each glass of wine on a scale from 0 to 10. Then they separately measured for each glass of wine of 10 different quantities. Things that they could just shove into a machine that could measure them chemically.



	A	B	C	D	E	F	G
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total
1	11.600	0.580	0.660	2.200	0.074	10.000	
2	10.400	0.610	0.490	2.100	0.200	5.000	
3	7.400	1.185	0.000	4.250	0.097	5.000	
4	10.400	0.440	0.420	1.500	0.145	34.000	
5	8.300	1.020	0.020	3.400	0.084	6.000	
6	7.600	1.580	0.000	2.100	0.137	5.000	
7	6.800	0.815	0.000	1.200	0.267	16.000	
8	7.300	0.980	0.050	2.100	0.061	20.000	
9	7.100	0.875	0.050	5.700	0.082	3.000	
10	6.700	0.760	0.020	1.800	0.078	6.000	
11	7.400	0.590	0.080	4.400	0.086	6.000	
12	5.700	1.130	0.090	1.500	0.172	7.000	
13	8.800	0.610	0.300	2.800	0.088	17.000	
14	4.600	0.520	0.150	2.100	0.054	8.000	
15	8.300	0.675	0.260	2.100	0.084	11.000	

- Here some of those factors. Fixed acidity. Volatile acidity I'm not sure what the difference between that is. Citric acid. Residual sugar. Chlorides and so on. So these first factors are our Xs and then this last one, quality, is the Y we're going to use to train our system.
- Now short quiz. Which of these factors do you think is most strongly correlated with the volunteers' perception of quality. I'm deliberately not showing you all the factors.
- [Audience: Residual sugar]
- No, but good guess. Somebody else make a guess. Why do you drink wine as opposed to grape juice? Of all these factors, percentage of alcohol was the most strongly correlated with the people's judgment of quality.
- That's not necessarily causality. There may be other factors that are related to alcohol as well but just pointing that out.

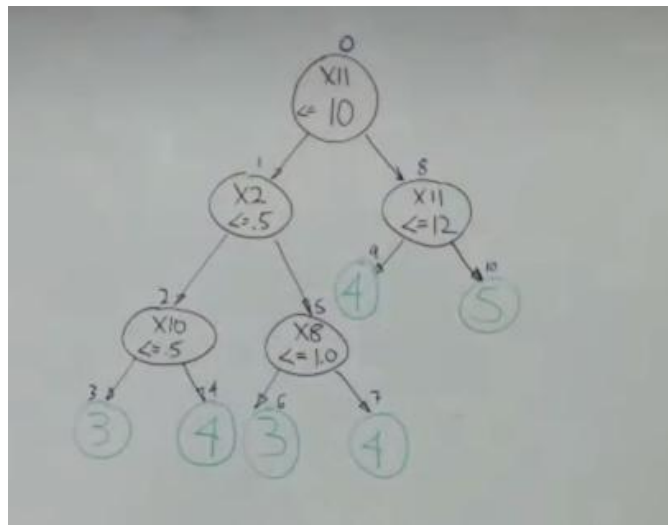


1	factor	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	Y
16		4.600	0.520	0.150	2.100	0.054	8.000	65.000	0.993	3.900	0.560	13.100	4.000
17		8.300	0.675	0.260	2.100	0.084	11.000	43.000	0.998	3.310	0.530	9.200	4.000
18		8.300	0.625	0.200	1.500	0.080	27.000	119.000	0.997	3.160	1.120	9.100	4.000
19		5.000	1.020	0.040	1.400	0.045	41.000	85.000	0.994	3.750	0.480	10.500	4.000
20		9.200	0.520	1.000	3.400	0.610	32.000	69.000	1.000	2.740	2.000	9.400	4.000
21		7.600	0.680	0.020	1.300	0.072	9.000	20.000	0.997	3.170	1.080	9.200	4.000
22		7.300	0.550	0.030	1.600	0.072	17.000	42.000	0.996	3.370	0.480	9.000	4.000
23		7.900	0.885	0.030	1.800	0.058	4.000	8.000	0.997	3.360	0.330	9.100	4.000
24		6.900	1.090	0.060	2.100	0.061	12.000	31.000	0.995	3.510	0.430	11.400	4.000
25		8.400	0.635	0.360	2.000	0.089	15.000	55.000	0.997	3.310	0.570	10.400	4.000
26		7.000	0.975	0.040	2.000	0.087	12.000	67.000	0.996	3.350	0.600	9.400	4.000
27		8.100	0.870	0.000	3.300	0.096	26.000	61.000	1.000	3.600	0.720	9.800	4.000
28		12.500	0.460	0.490	4.500	0.070	26.000	49.000	0.998	3.050	0.570	9.600	4.000
29		10.500	0.590	0.490	2.100	0.070	14.000	47.000	0.999	3.300	0.560	9.600	4.000
30		9.900	0.500	0.240	2.300	0.103	6.000	14.000	0.998	3.340	0.520	10.000	4.000

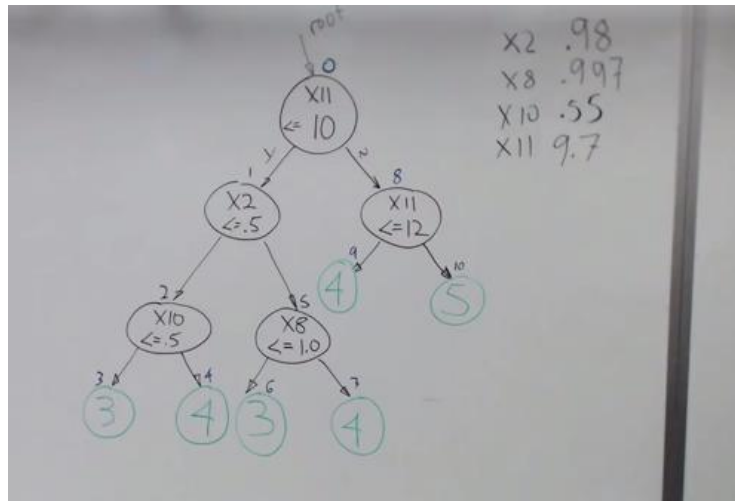
- Okay so let's look at this data a little bit more now. I have another view over here. Each row here represents the value of each of these 11 factors and then Y the judgment of the of the person. So each row represents one test. So let's suppose for a moment I'm going to map this to the S&P 500. Each row might represent the value of factors of each stock on a particular day and then Y might represent the return several days later. So what that means is for each day we have 500 individual rows, one for each stock.
- Coming back to the wine, each row is 1 person tasting one wine and their assessment over here. Now one thing to keep in mind is many people tasted exactly the same wine so we're gonna have a row for each person. So we'll have multiple rows for each particular wine where the Xs will be the same but the Ys might be different.
- Now just for fun I used excels correlation function for each factor to look at how strongly correlated it was with our Y. So you can see here factor 11 has a correlation of .47 with quality. If you're an engineer you might be thinking okay I don't want to build a bridge with a point four seven correlation that it won't fall down so that's typically a weak relationship for many uses. But for machine learning and especially classification and regression and so on then that's a fairly strong value.
- Now note that some of these are negative like this one. Negative 0.391. What that means is essentially, in general, the larger this number is, the lower the quality. It doesn't mean it's not a good factor for evaluating, it just means that it's inversely related.
- Ok so that's our data. Now I'm going to take a moment to draw a decision tree and then I'm gonna give you some example data rows and I want you all from my excellently drawn decision tree on the board, I'm gonna ask you to figure out what the decision tree says the quality of the wine should be.
- Ok but before I do that let me back up a little bit. The process for utilizing decision trees is the following. You start with data like this. You use that data through some mysterious process that you learn next week to build a decision tree. Then once you

have the decision tree, you can now query it. So you can say, okay, I have a wine with these different factors, what do we think that a human will say the quality is. And then we consult that tree and we end up at a leaf and that's what we estimate the quality will be.

- And by the way, I skipped over it, but presumably the utility for that is they can very quickly and automatically evaluate the quality of the wine without necessarily having to have a human taste it. So essentially they're putting some poor bastard out of work who spends all day tasting wine. But it also though provides a more objective measure of the quality of wine.
- Okay I'm going to take a moment to draw my decision tree on the board and in the meantime we'll have an interlude elevator music so I'll be back in just a moment.

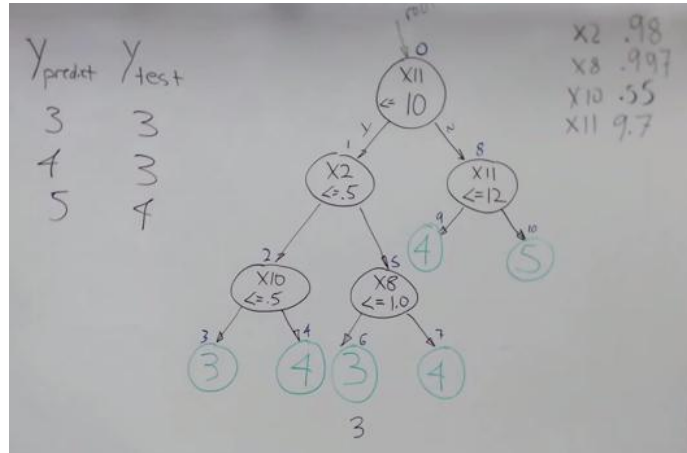


- Ok, so online you all should be able to see the tree I drew up. If not, complain with a question please. Anyways, I realize for some of you in the back it may be hard to see unless you have really good eyes, so I tried to draw this as large as I could.
- Anyways, each one each of these is a decision node and if the answer to the question we ask in each one is yes go to the left; if it's no we go to the right.
- So for instance this decision node which is the root, asks is factor 11 less than or equal to 10 and if it is we go down here and we ask about factor 2 and so on.
- Anyways we ask these questions. Eventually we reach a leaf node and boom that's our answer. Now couple things to point out here. We don't necessarily use all factors in a decision tree. Sometimes some factors are just so predictive...in fact sometimes for some data you might have 10 factors but only end up using one of them in your decision tree. Now you might still have that factor, like for instance we have factor 11 twice in this decision tree. You might have it at multiple levels within the decision tree and it might be repeated for instance.
- Now note if you go down say this path use factor 11, 2 and 8 but you don't use a factor 10 for instance. So anyway, I'm I going to put up an example piece of data and have you use this tree and tell me what you think the value is. So we need factor two, factor eight factor 10 and factor 11. So we're gonna query the tree right.



- Ok so these are the values of our four factors that make up this tree. I want everybody to take a moment themselves and try and figure out what does our model think the quality of the wine is. Okay raise your hand if you're done. Okay, we'll wait longer. How about you in TV land?
- Who wants me to work through it? Ok, I'm confused, so you're not finished but you don't want me to work through it. Let me ask again who's worked through it? Okay, where are you stuck? Alright what's the answer? <Three>.
- Yes, so working through it, the first question is about the factor 11, is it less than or equal to 10? Yes, so that's predictive of a sucky wine right? Well, I know because I looked at the data here and 10 is about the medium (alcohol level was 11), 10 is about the median value of alcohol. So if it's less than 10 it's probably headed towards suckiness.
- Ok factor 2. Is it less than or equal to 5? No. We're down here at this node factor eight. Is a less than or equal to 1? Yes. Boom the answer is three.
- Ok so that's how we query a tree that we have. Now I'll look at my sample data and tell you what the real answer was. Well the real answer was three, so a human who tasted the wine actually also evaluated it at three. Now I for the most part kind of invented this decision tree but I built it from a few examples so we're lucky that it turned out to match that.
- But here's something for you to think about. Let's suppose we take half of our data and we built a tree like this. In this case I think there's 1,600 rows, where each row is one sample. So let's suppose we took 800 of those samples and built a tree. Now most ways of building a tree, if you build an exact tree, which an exact tree means that each example is represented in one leaf, right? So if we took 800 samples, we would build a tree that had 800 leaves.
- Now suppose we query that tree with one of the data elements that we use to train it. Who thinks it would give us exactly the same value retrained it with? Yes? Well the answer is yes because the way that we build the trees, were guaranteed that if we query it with any of the samples from the training data, we will get exactly the answer that we trained it with. So it's called the in-sample testing.

- Ok now suppose we took the other eight hundred samples and we query the tree for each one of the samples one at a time and we compared what our tree thought the correct answer was with what was recorded in our data. How can we evaluate whether our tree was making good predictions or not?



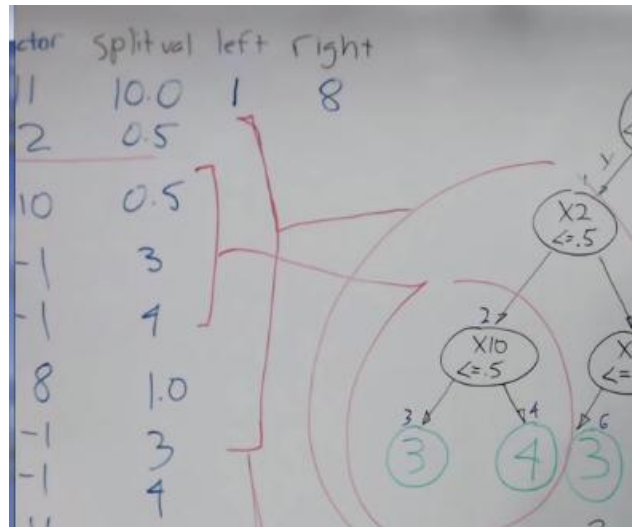
- So let me put it another way. So Y_{test} is the number that our model thinks that particular data element is and Y_{test} is the value that we know already. We hid that data from the system and now we're testing to see how good it is. So let's say our predict is three, our test is 3-3, 4-3, 5-4. So say we're making a prediction and we're comparing it to what the actual data was, what sorts of values would you like to see in these columns if your model is working well? <The Same.> Right.
- Or another way to think about it is you would like the values that the pairs of values in these columns to be correlated. So what's the best correlation we could have? What would indicate that our model's perfect? <One.> That's right.
- Ok so we'll talk later about how to assess models, but that's an important one.
- Now you're very rarely going to have a correlation of one when you test your data out-of-sample. But sometimes for certain problems it happens.
- The last topic I want to cover today on decision trees is I want to present to you a data structure that you can use to build and store your decision trees in.
- There are lots of different ways to do it. A Java or C++ programmer thinks object oriented and they want to make a node class and each of these nodes an instance and link them all together.
- That's ok but it ends up, especially for large datasets, in Python it ends up being really slow and burns up a lot of memory. You can do it in Python, but what I'm going to suggest to you is a matrix or an array-based representation that ends up being really, really fast.
- Now if you're if you're determined that you have to do an object-oriented manner that's ok but please try my way first and then do your object-oriented approach. Because I think once you get into it a little bit you'll discover that my recommendation is pretty easy and fast.
- Ok so here is the data structure. And the data structure is an ndarray with two dimensions.

- Ok so each row in this ndarray corresponds to one of our decision nodes or a leaf. Let's start with this one. I'll use a different color. Oh and by the way, these little blue numbers are labeled to correspond to the rows that I'm gonna fill in over here. I did that little cheat to make this part of the lecture easier.

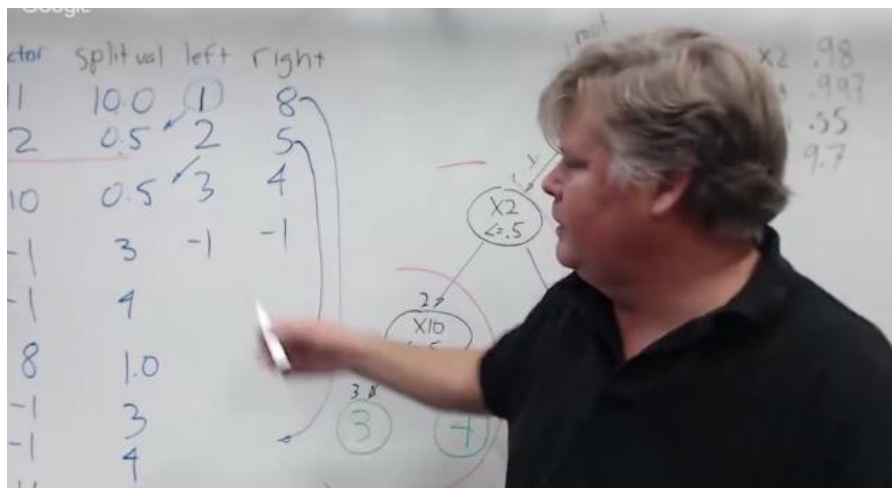
Ndarray

Row	Factor	Split Val	Left	Right
0	11	10.0	1	8
1	2	0.5		
2	10	0.5		
3	-1	3		
4	-1	4		
5	8	1.0		
6	-1	3		
7	-1	4		

- Anyways, in this first row, row 0, the factor that we're going to use is 11; the split val or the number that we're going to make our comparison on, is a 10. And if when we're querying, and oh by the way the this first row is always the root, and if we're querying the root and the value of factor 11 is less than or equal to 10 this column (left) tells us which row represents our next decision node. I've labeled it there. It's one and if we go to the right it's 8.
- Ok now the rest of the rows are going to be split into essentially two groups. So one set of rows is going to represent the left tree and some other set of nodes are going to represent the right tree. so these guys over here will be packed into this part of the array and these guys over here will be over here. That make sense? Okay.
- Let me flesh this out a little bit more.
- One additional thing I want to tell you. The way that we represent leaves in this data structure is we use a special number to represent the factor. We use negative 1. So if we get to a row and the factors negative 1, that means it's a leaf.
- Let me real quickly fill this in. Now how many rows should we have? 11. Right. And that's because we have 11 nodes in our graph. I know it was easy but just checking.

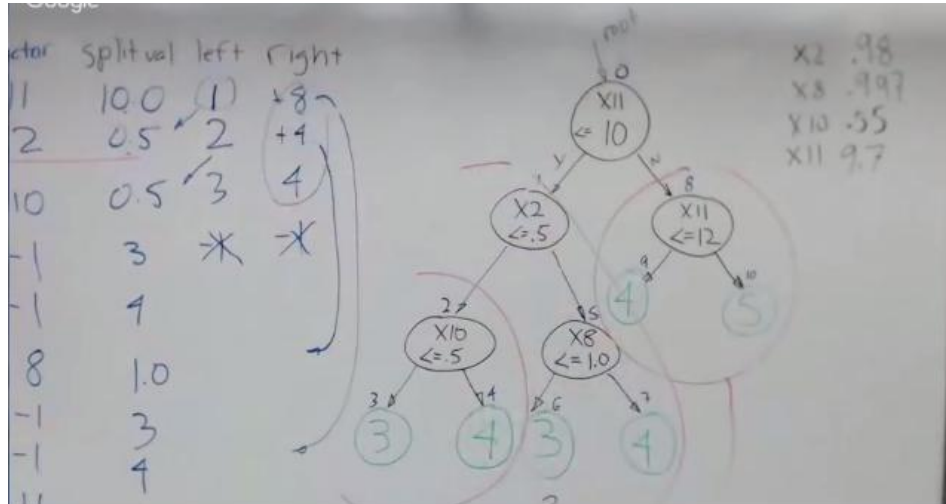


- It's gonna be too tedious to fill all this in but let me just highlight a few things about it. The left branch of this tree is represented by nodes 1 through 7. And of course the right by 8 through 9.
- Something to observe is, each of these subtrees is itself a tree. The root node is of the left tree is that first row. And then it's left tree is right here. So does everybody get the structure? Any questions about the data structure?
- We're gonna blaze into it more on Tuesday. I should fill in a few of these numbers on the left here as I want to illustrate something.



- So let's look at this for a second. So coming into the root node if factor 11 is less than or equal to 10 then we go to node 1 which is the next one. Otherwise we go to node eight which is way down here, right. So that redirects us to look at the right part of the tree.
- Similarly if factor 2 is less than or equal to 0.5 we go immediately to the next row that's our left tree, otherwise we go to 5 which is down here.
- Anyways, a couple things to point out. So the number here is essentially the index to the head of the next tree to look at, or the root of the next tree.

- Now when we have a leaf it doesn't really matter what values we put in here because we're never going to use them, right? We're at a leaf, we're done, we terminated. I put negative ones just for reference. The way I've shown this data structure here today, these are absolute references into this array.



- You could if you wanted to instead make them relative. Why is that? When you're building this tree, it's a recursive process, and like I said we'll get to that soon. But it's often convenient when you're building it, when you recurse to build the left tree, if it just pops up and tells you how big is the left tree, then you know immediately that however big the left tree is that's how far down we go to start the right tree.
- You don't have to worry about that too much right now but as an example, we know the right tree starts at cell eight here and that's because our left tree 7 elements. So we just know immediately to go to that to that point.
- Similarly, here is an absolute reference to five, but really if we want to store a relative number then we could use 4. That would get us there using relative.
- Just keep in mind that you might want to use relative numbers to those rows instead of absolute.
- Okay just a couple more things to cover today and then I'll take questions. Ok a couple of things I wanted to highlight and ask you some questions about. First question, which learning method of the three that I'm going to list, do you think is fastest? K nearest neighbor, decision trees, or linear regression?
- I didn't give you enough information. The question that would be reasonable to ask me is, do you mean the learning part or the querying part? For now say I'm asking about the query part.
- Once you have the model, the fastest learning method is linear regression because, think about it, how do you calculate the prediction? You got three X's or whatever. It's one parameter times x1 another parameter times x2, another parameter times x3 and together boom you're done. So that's linear regression.
- For k-nearest neighbor you have to go consult your thousand data elements, calculate the distance from your query to all of the other data elements, and then find the

closest three or five or whatever your k is. So you have to calculate distance to all of them, sort, and then take the mean of the ones that are closest. So KNN at query time is really slow.

- Okay I'll give you one more chance. Another question. Let's suppose we have a hundred samples and we build a perfect tree, on average how many binary questions will we have to ask for each query? $\log_2 100$ Yes. That's right you get an extra bonus prize. So what does that work out to be? Six or seven. So it turns out that this is pretty fast, right?
- Now let's say we go to a thousand. What's $\log_2 1000$? 10, so with a thousand elements we only have to ask 10 questions and binary questions are really, really fast. Processors are optimized to do that quickly. Also if you're using a GPU, the sort of topography of decision trees is well suited to GPUs as well so it can be very, very fast.
- Now decision trees are slow at learning time because it takes a lot of computational effort to build one of these trees and that's this we'll talk about Tuesday. I want to raise a couple more points I let everybody go.
- Sometimes we repeat factors in the tree. Remember that. Not all factors are guaranteed to be used. And one last question for thought. When I asked you how many questions we would have to answer for a 1000 node tree. You all said $\log_2 1000$. You're making an assumption. What's the assumption that you're making about the tree? It's a binary tree, that's one. There's another assumption you're making. That it's a balanced tree.
- So a lot of the algorithms involved in building these trees strive to keep them balanced and that'll be something else that we talk about.
- I'm gonna go real quick and see if we have any questions online. The only question we have is if I could move the camera closer to the whiteboard. Too late for that.
- Ok goodbye in TV land. I'll be on piazza to answer questions. See you all Tuesday.