PROFESSOR: OK, so the last topic for the class is interpretability. As you know, the modern machine learning models are justifiably reputed to be very difficult to understand. So if I give you something like the GPT2 model, which we talked about in natural language processing, and I tell you that it has 1.5 billion parameters and then you say, why is it working?

Clearly the answer is not because these particular parameters have these particular values. There is no way to understand that. And so the topic today is something that we raised a little bit in the lecture on fairness, where one of the issues there was also that if you can't understand the model you can't tell if the model has baked-in prejudices by examining it.

And so today we're going to look at different methods that people have developed to try to overcome this problem of inscrutable models. So there is a very interesting bit of history. How many of you know of George Miller's 7 plus or minus 2 result? Only a few.

So Miller was a psychologist at Harvard, I think, in the 1950s. And he wrote this paper in 1956 called "The Magical Number 7 Plus or Minus 2-- Some Limits On Our Capacity for Processing Information." It's quite an interesting paper.

So he started off with something that I had forgotten. I read this paper many, many years ago. And I'd forgotten that he starts off with the question of how many different things can you sense? How many different levels of things can you sense?

So if I put headphones on you and I ask you to tell me on a scale of 1 to n how loud is the sound that I'm playing in your headphone, it turns out people get confused when you get beyond about five, six, seven different levels of intensity. And similarly, if I give you a bunch of colors and I ask you to tell me where the boundaries are between different colors, people seem to come up with 7 plus or minus 2 as the number of colors that they can distinguish.

And so there is a long psychological literature of this. And then Miller went on to do experiments where he asked people to memorize lists of things. And what he discovered is, again, that you could memorize a list of about 7 plus or minus 2 things. And beyond that, you couldn't remember the list anymore.

So this tells us something about the cognitive capacity of the human mind. And it suggests that if I give you an explanation that has 20 things in it, you're unlikely to be able to fathom it because you can't keep all the moving parts in your mind at one time.

Now, it's a tricky result, because he does point out even in 1956 that if you chunk things into bigger chunks, you can remember seven of those, even if they're much bigger. And so people who are very good at memorizing things, for example, make up patterns. And they remember those patterns, which then allow them to actually remember more primitive objects.

So you know-- and we still don't really understand how memory works. But this is just an interesting observation, and I think plays into the question of how do you explain things in a complicated model? Because it suggests that you can't explain too many different things because people won't understand what you're talking about.

OK. So what leads to complex models? Well, as I say, overfitting certainly leads to complex models. I remember in the 1970s when we started working on expert systems in healthcare, I made a very bad faux pas. I went to the first joint conference between statisticians and artificial intelligence researchers.

And the statisticians were all about understanding the variance and understanding statistical significance and so on. And I was all about trying to model details of what was going on in an individual patient. And in some discussion after my talk, somebody challenged me. And I said, well, what we AI people are really doing is fitting what you guys think is the noise, because we're trying to make a lot more detailed refinements in our theories and our models than what the typical statistical model does.

And of course, I was roundly booed out of the hall. And people shunned me for the rest of the conference because I had done something really stupid to admit that I was fitting noise. And of course, I didn't really believe that I was fitting noise. I believed that what I was fitting was what the average statistician just chalks up to noise. And we're interested in more details of the mechanisms.

So overfitting we have a pretty good handle on by regularization. So you can-- you know, you've seen lots of examples of regularization throughout the course. And people keep coming up with interesting ideas for how to apply regularization in order to simplify models or make them fit some preconception of what the model ought to look like before you start learning it from data.

But the problem is that there really is true complexity to these models, whether or not you're fitting noise. There's-- the world is a complicated place. Human beings were not designed. They evolved. And so there's all kinds of bizarre stuff left over from our evolutionary heritage. And so it is just complex. It's hard to understand in a simple way how to make predictions that are useful when the world really is complex.

So what do we do in order to try to deal with this? Well, one approach is to make up what I call just-so stories that give a simplified explanation of how a complicated thing actually works. So how many of you have read these stories when you were a kid? Nobody? My God. OK. Must be a generational thing.

So Rudyard Kipling was a famous author. And he wrote the series of just-so stories, things like *How the Lion Got His Mane* and *How the Camel Got His Hump* and so on. And of course, they're all total bull, right? I mean, it's not a Darwinian evolutionary explanation of why male lions have manes. It's just some made up story. But they're really cute stories.

And I enjoyed them as a kid. And maybe you would have, too, if your parents had read them to you. So I mean, I use this as a kind of pejorative because what the people who follow this line of investigation do is they take some very complicated model. They make a local approximation to it that says, this is not an approximation to the entire model, but it's an approximation to the model in the vicinity of a particular case.

And then they explain that simplified model. And I'll show you some examples of that through the lecture today. And the other approach which I'll also show you some examples of is that you simply trade off somewhat lower performance for a simple-- a model that's simple enough to be able to explain. So things like decision trees and logistic regression and so on typically don't perform quite as well as the best, most sophisticated models, although you've seen plenty of examples in this class where, in fact, they do perform quite well and where they're not outperformed by the fancy models.

But in general, you can do a little better by tweaking a fancy model. But then it becomes incomprehensible. And so people are willing to say, OK, I'm going to give up 1% or 2% in performance in order to have a model that I can really understand. And the reason it makes sense is because these models are not self-executing. They're typically used as advice for some human being who makes ultimate decisions.

Your surgeon is not going to look at one of these models that says, take out the guy's left kidney and say, OK, I guess. They're going to go, well, does that make sense? And in order to answer the question of, does that make sense? It really helps to know what the model is-- what the model's recommendation is based on. What is its internal logic? And so even an approximation to that is useful.

So the need for trust, clinical adoption of ML models-- there are two approaches in this paper that I'm going to talk about where they say, OK, what you'd like to do is to look at case-specific predictions. So there is a particular patient in a particular state and you want to understand what the model is saying about that patient.

And then you also want to have confidence in the model overall. And so you'd like to be able to have an explanatory capability that says, here are some interesting representative cases. And here's how the model views them. Look through them and decide whether you agree with the approach that this model is taking.

Now, remember my critique of randomized controlled trials that people do these trials. They choose the simplest cases, the smallest number of patients that they need in order to reach statistical significance, the shortest amount of follow-up time, et cetera. And then the results of those trials are applied to very different populations.

So Davids talked about the cohort shift as a generalization of that idea. But the same thing happens in these machine learning models that you train on some set of data. The typical publication will then test on some held-out subset of the same data. But that's not a very accurate representation of the real world. If you then try to apply that model to data from a totally different source, the chances are you will have specialized it in some way that you don't appreciate.

And the results that you get are not as good as what you got on the held-out test data because it's more heterogeneous. I think I mentioned that Jeff Drazen, the editor-in-chief of the New England Journal, had a meeting about a year ago in which he was arguing that the journal shouldn't ever publish a research study unless it's been validated on two independent data sets because he's tired of publishing studies that wind up getting retracted because-- not because of any overt badness on the part of the investigators. They've done exactly the kinds of things that you've learned how to do in this class.

But when they go to apply that model to a different population, it just doesn't work nearly as well as it did in the published version. And of course, there are all the publication bias issues about if 50 of us do the same experiment and by random chance some of us are going to get better results than others. And those are the ones that are going to get published because the people who got poor results don't have anything interesting to report.

And so there's that whole issue of publication bias, which is another serious one. OK. So I wanted to just spend a minute to say, you know, explanation is not a new idea. So in the expert systems era that we talked about a little bit in one of our earlier classes, we talked about the idea that we would take medical-- human medical experts and debrief them of what they knew and then try to encode those in patterns or in rules or in various ways in a computer program in order to reproduce their behavior.

So Mycin was one of those programs-- [INAUDIBLE] PhD thesis-- in 1975. And they published this nice paper that was about explanation and rule acquisition capabilities of the Mycin system. And as an illustration, they gave some examples of what you could do with the system. So rules, they argued, were quite understandable because they say if a bunch of conditions, then you can draw the following conclusion.

So given that, you can say, well, when the program comes back and says, in light of the site from which the culture was obtained and the method of collection, do you feel that a significant number of organism 1 were detected-- were obtained? In other words, if you took a sample from somebody's body and you're looking for an infection, do you think you got enough organisms in that sample?

And the user says, well, why are you asking me this question? And the answer in terms of the rules that the system works by is pretty good. It says it's important to find out whether there's therapeutically significant disease associated with this occurrence of organism 1. We've already established that the culture is not one of those that are normally sterile and the method of collection is sterile.

Therefore, if the organism has been observed in significant numbers, then there's strongly suggestive evidence that there's therapeutically significant disease associated with this occurrence of the organism. So if you find bugs in a place carefully collected, then that suggests that you ought to probably treat this patient if there are were bunch of-- enough bugs there.

And there's also strongly suggestive evidence that the organism is not a contaminant, because the collection method was sterile. And you can go on with this and you can say, well, why that?

So why that question? And it traces back in its evolution of these rules and it says, well, in order to find out the locus of infection, it's already been established that the site of the culture is known. The number of days since the specimen was obtained is less than 7.

Therefore, there is therapeutically significant disease associated with this occurrence of the organism. So there's some rule that says if you've got bugs and it happened within the last seven days, the patient probably really does have an infection. And I mean, I've got a lot of examples of this. But you can keep going why.

You know, this is the two-year-old. But why, daddy? But why? But why? Well, why is it important to find out a locus of infection? And, well, there's a reason, which is that there is a rule that will conclude, for example, that the abdomen is a locus of infection or the pelvis is a locus of infection of the patient if you satisfy these criteria.

And so this is a kind of rudimentary explanation that comes directly out of the fact that these are rule-based systems and so you can just play back the rules. One of the things I like is you can also ask freeform questions. 1975, the natural language processing was not so good. And so this worked about one time in five. But you could walk up to it and type some question.

And for example, do you ever prescribe carbenicillin for pseudomonas infections? And it says, well, there are three rules in my database of rules that would conclude something relevant to that question. So which one do you want to see? And if you say, I want to see rule 64, it says, well, that rule says if it's known with certainty that the organism is a pseudomonas and the drug under consideration is gentamicin, then a more appropriate therapy would be a combination of gentamicin and carbenicillin.

Again, this is medical knowledge as of 1975. But my guess is the real underlying reason is that there probably were pseudomonas that were resistant by that point, to gentamicin, and so they used a combination therapy. Now, notice, by the way, that this explanation capability does not tell you that, right? Because it doesn't actually understand the rationale behind these individual rules. And at the time there was also research, for example, by one of my students on how to do a better job of that by encoding not only the rules or the patterns, but also the rationale behind them so that the explanations could be more sensible.

OK. Well, the granddaddy of the standard just-so story approach to explanation of complex models today comes from this paper and a system called LIME-- Locally Interpretable Model-agnostic Explanations. And just to give you an illustration, you have some complicated model and it's trying to explain why the doctor or the human being made a certain decision, or why the model made a certain decision.

And so it says, well, here are the data we have about the patient. We know that the patient is sneezing. And we know their weight and their headache and their age and the fact that they have no fatigue. And so the explainer says, well, why did the model decide this patient has the flu?

Well, positives are sneeze and headache. And a negative is no fatigue. So it goes into this complicated model and it says, well, I can't explain all the numerology that happens in that neural network or Bayesian network or whatever network it's using. But I can specify that it looks like these are the most important positive and negative contributors. Yeah?

AUDIENCE: Is this for notes only, or it's for all types of data?

PROFESSOR: I'll show you some other kind of data in a minute. I think they originally worked it out for notes, but it was also used for images and other kinds of data, as well. OK. And the argument they make is that this approach also helps to detect data leakage, for example in one of their experiments, the headers of the data had information in them that that correlated highly with the result.

I think there-- I can't remember if it was these guys, but somebody was assigning study IDs to each case. And they did it a stupid way so that all the small numbers corresponded to people who had the disease and the big numbers corresponded to the people who didn't. And of course, the most parsimonious predictive model just used the ID number and said, OK, I got it.

So this would help you identify that, because if you see that the best predictor is the ID number, then you would say, hmm, there's something a little fishy going on here. Well-- so here's an example where this kind of capability is very useful. So this was another-- this was from a newsgroup. And they were trying to decide whether a post was about Christianity or atheism.

Now, look at these two models. So there's algorithm 1 and algorithm 2 or model 1 and model 2. And when you explain a particular case about using model 1, it says, while the words that I consider important are God, mean, anyone, this, Koresh, and through-- does anybody remember who David Koresh was?

He was some cult leader who-- I can't remember if he killed a bunch of people or bad things happened. Oh, I think he was the guy in Waco, Texas that the FBI and the ATF went in and set their place on fire and a whole bunch of people died. So the prediction in this case is atheism. And you notice that God and Koresh and Mean are negatives. And anyone this and through are positives.

And you go, I don't know, is that good? But then you look at algorithm 2 and you say, this also made the correct prediction, which is that this particular article is about atheism. But the positives were the word by and in, not terribly specific. And the negatives were things like NNTP. You know what that is? That's the Network Time Protocol. It's some technical thing, and posting and host.

So this is probably like metadata that got into the header of the articles or something. So it happened that in this case, algorithm 2 turned out to be more accurate than algorithm 1 on their held out test data, but not for any good reason. And so the explanation capability allows you to clue in on the fact that even though this thing is getting the right answers, it's not for sensible reasons.

OK. So what would you like from an explanation? Well, they say you'd like it to be interpretable. So it should provide qualitative understanding of the relationship between the input variables and the response. But they also say that that's going to depend on the audience. It requires sparsity for the George Miller argument that I was making before.

You can't keep too many things in mind. And the features themselves that you're explaining must make sense. So for example, if I say, well, the reason this decided that is because the eigenvector for the first principle component was the following, that's not going to mean much to most people. And then they also say, well, it ought to have local fidelity. So it must correspond to how the model behaves in the vicinity of the particular instance that you're trying to explain.

And their third criterion, which I think is a little iffier, is that it must be model-agnostic. In other words, you can't take advantage of anything you know that is specific about the structure of the model, the way you trained it, anything like that. It has to be a general purpose explainer that works on any kind of complicated model. Yeah?

AUDIENCE: What is the reasoning for that?

PROFESSOR: I think their reasoning for why they insist on this is because they don't want to have to write a separate explainer for each possible model. So it's much more efficient if you can get this done. But I actually question whether this is always a good idea or not. But nevertheless, this is one of their assumptions.

OK. So here's the setup that they use. They say, all right, x is a vector in some D-dimensional space that defines your original data. And what we're going to do in order to make the data explainable, in order to make the data, not the model, explainable, is we're going to define a new set of variables, x prime, that are all binary and that are in some space of dimension D prime that is probably lower than D.

So we're simplifying the data that we're going to explain about this model. Then they say, OK, we're going to build an explanation model, g, where g is a class of interpretable models. So what's an interpretable model? Well, they don't tell you, but they say, well, examples might be linear models, additive scores, decision trees, falling rule lists, which we'll see later in the lecture.

And the domain of this is this input, the simplified input data, the binary variables in D prime dimensions, and the model complexity is going to be some measure of the depth of the decision tree, the number of non-zero weights, and the logistic regression-- the number of clauses in a falling rule list, et cetera.

So it's some complexity measure. And you want to minimize complexity. So then they say, all right, the real model, the hairy, complicated full-bore model is f. And that maps the original data space into some probability. And for example, for classification, f is the probability that x belongs to a certain class.

And then they also need a proximity measure. So they need to say, we have to have a way of comparing two cases and saying how close are they to each other? And the reason for that is because, remember, they're going to give you an explanation of a particular case and the most relevant things that will help with that explanation are the ones that are near it in this high dimensional input space.

So they then define their loss function based on the actual decision algorithm, based on the simplified one, and based on the proximity measure. And they say, well, the best explanation is that g which minimizes this loss function plus the complexity of g. Pretty straightforward. So that's our best model.

Now, the clever idea here is to say, instead of using all of the data that we started with, what we're going to do is to sample the data so that we take more sample points near the point we're interested in explaining. We're going to sample in the simplified space that is explainable and then we'll build that g model, the explanatory model, from that sample of data where we weight by that proximity function so the things that are closer will have a larger influence on the model that we learn.

And then we recapture the-- sort of the closest point to this simplified representation. We can calculate what its answer should be. And that becomes the label for that point. And so now we train a simple model to predict the label that the complicated model would have predicted for the point that we've sampled. Yeah?

AUDIENCE: So the proximity measure is [INAUDIBLE]?

PROFESSOR: It's a distance function of some sort. And I'll say more about it in a minute, because that's one of the critiques of this particular method has to do with how do you choose that distance function? But it's basically a similarity. So here's a nice, graphical explanation of what's going on. Suppose that the actual model-- the decision boundary is between the blue and the pink regions. OK. So it's this god awful, hairy, complicated decision model.

And we're trying to explain why this big, red plus wound up in the pink rather than in the blue. So the approach that they take is to say, well, let's sample a bunch of points weighted by shortest distance. So we do sample a few points out here. But mostly we're sampling points near the point that we're interested in.

We then learn a linear boundary between the positive and the negative cases. And that boundary is an approximation to the actual boundary in the more complicated decision model. So now we can give an explanation just like you saw before which says, well, this is some D prime dimensional space. And so which variables in that D prime dimensional space are the ones that influence where you are on one side or another of this newly computed decision boundary, and to what extent? And that becomes the explanation. OK? Nice idea.

So if you apply this to text classification-- yes?

AUDIENCE: I was just going to ask if the-- there's a worry that if explanation is just fictitious, like, we can understand it? But is there reason to believe that we should believe it if that's really the true nature of things that the linear does-- you know, it would be like, OK, we know what's going on here. But is that even close to reality?

PROFESSOR: Well, that's why I called it a just-so story, right? Should you believe it? Well, the engineering disciplines have a very long history of approximating extremely complicated phenomena with linear models. Right? I mean, I'm in a department of electrical engineering and computer science. And if I talk to my electrical engineering colleagues, they know that the world is insanely complicated. Nevertheless, most models in electrical engineering are linear models.

And they work well enough that people are able to build really complicated things and have them work. So that's not a proof. That's an argument by history or something. But it's true. Linear models are very powerful, especially when you limit them to giving explanations that are local. Notice that this model is a very poor approximation to this decision boundary or this one, right?

And so it only works to explain in the neighborhood of the particular example that I've chosen. Right? But it does work OK there. Yeah.

AUDIENCE: [INAUDIBLE] very well there? [INAUDIBLE] middle of the red space then the--

PROFESSOR: Well, they did. So they sample all over the place. But remember that that proximity function says that this one is less relevant to predicting that decision boundary because it's far away from the point that I'm interested in. So that's the magic.

AUDIENCE: But here they're trying to explain to the deep red cross, right?

PROFESSOR: Yes.

AUDIENCE: And they picked some point in the middle of the red space maybe. Then all the nearby ones would be red and [INAUDIBLE].

PROFESSOR: Well, but they would-- I mean, suppose they picked this point, instead. Then they would sample around this point and presumably they would find this decision boundary or this one or something like that and still be able to come up with a coherent explanation.

OK, so in the case of text, you've seen this example already. It's pretty simple. For their proximity function, they use cosine distance. So it's a bag of words model and they just calculate cosine distance between different examples by how much overlap there is between the words that they use and the frequency of words that they use.

And then they choose k-- the number of words to show just as a preference. So it's sort of a hyperparameter. They say, you know, I'm interested in looking at the top five words or the top 10 words that are either positively or negatively an influence on the decision, but not the top 10,000 words because I don't know what to do with 10,000 words.

Now, what's interesting is you can also then apply the same idea to image interpretation. So here is a dog playing a guitar. And they say, how do we interpret this? And so this is one of these labeling tasks where you'd like to label this picture as a Labrador or maybe as an acoustic guitar. But some reason-- some labels also decide that it's an electric guitar.

And so they say, well, what counts in favor of or against each of these? And the approach they take is a relatively straightforward one. They say let's define a super pixel as a region of pixels within an image that have roughly the same intensity. So if you've ever used Photoshop, the magic selection tool can be adjusted to say, find a region around this point where all the intensities are within some delta of the point that I've picked.

And so it'll outline some region of the picture. And what they do is they break up the entire image into these regions. And then they treat those as if they were the words in the words style explanation. So they say, well, this looks like an electric guitar to the algorithm. And this looks like an acoustic guitar. And this looks like a Labrador.

So some of that makes sense. I mean, you know, that dog's face does kind of look like a Lab. This does look kind of like part of the body and part of the fret work of a guitar. I have no idea what this stuff is or why this contributes to it being a dog. But such is-- such is the nature of these models.

But at least it is telling you why it believes these various things. So then the last thing they do is to say, well, OK, that helps you understand the particular model. But how do you convince yourself-- I mean, a particular example where a model is applied to it. But how do you convince yourself that the model itself is reasonable?

And so they say, well, the best technique we know is to show you a bunch of examples. But we want those examples to kind of cover the gamut of places that you might be interested in. And so they say, let's create this matrix-- an explanation matrix where these are the cases and these are the various features, you know, the top words or the top pixel elements or something, and then we'll fill in the element of the matrix that tells me how strongly this feature is correlated or anti-correlated with the classification for that model.

And then it becomes a kind of set covering issue of find a set of models that gives me the best coverage of explanations across that set of features. And then with that, I can convince myself that the model is reasonable. So they have this thing called the sub modular pick algorithm. And you know, probably if you're interested, you should read the paper.

But what they're doing is essentially doing a kind of greedy search that says, what features should I add in order to get the best coverage in that space of features by documents? And then they did a bunch of experiments where they said, OK, let's compare the results of these explanations of these simplified models to two sentiment analysis tasks of 2,000 instances each.

Bag of words as features-- they compared it to decision trees, logistic regression, nearest neighbors, SVM with the radial basis function, kernel, or random forests that use word to vacuum beddings-- highly non-explainable-- with 1,000 trees and K equal 10. So they chose 10 features to explain for each of these models.

They then did a side calculation that said, what are the 10 most suggestive features for each case? And then they said, does that covering algorithm identify those features correctly? And so what they show here is that their method line does better in every case than a random sampling-- that's not very surprising-- or a greedy sampling or a partisan sampling, which I don't know the details of.

But in any case, there's what this graph is showing is that of the features that they decided were important in each of these cases, they're recovering. So their recall is up around 90, 90-plus percent. So in fact, the algorithm is identifying the right cases to give you a broad coverage across all the important features that matter in classifying these cases.

They then also did a bunch of human experiments where they said, OK, we're going to ask users to choose which of two classifiers they think is going to generalize better. So this is like the picture I showed you of the Christianity versus atheism algorithm, where presumably if you were a Mechanical Turker and somebody showed you an algorithm that has very high accuracy but that depends on things like finding the word NNTP in a classifier for atheism versus Christianity, you would say, well, maybe that algorithm isn't good to generalize very well, because it's depending on something random that may be correlated with this particular data set. But if I try it on a different data set, it's unlikely to work.

So that was one of the tasks. And then they asked them to identify features like that that looked bad. They then ran this Christianity versus atheism test and had a separate test set of about 800 additional web pages from this website. The underlying model was a support vector machine with RBF kernels trained on the 20 newsgroup data-- I don't know if you know that data set, but it's a well-known, publicly available data set.

They got 100 Mechanical Turkers and they said, OK, we're going to present each of them six documents and six features per document in order to ask them to make this. And then they did an auxiliary experiment in which they said, if you see words that are no good in this experiment, just strike them out. And that will tell us which of the features were bad in this method.

And what they found was that the human subjects choosing between two classifiers were pretty good at figuring out which was the better classifier. Now, this is better by their judgment. And so they said, OK, this submodular pick algorithm-- which is the one that I didn't describe in detail, but it's this set covering algorithm-- gives you better results than a random pick algorithm that just says pick random features. Again, not totally surprising.

And the other thing that's interesting is if you do the feature engineering experiment, it shows that as the Turkers interacted with the system, the system became better. So they started off with real world accuracy of just under 60%. And using the better of their algorithms, they reached about 75% after three rounds of interaction.

So the users could say, I don't like this feature. And then the system would give them better features. Now, they tried a similar thing with images. And so this one is a little funny. So they trained a deliberately lousy classifier to classify between wolves and huskies. This is a famous example. Also it turns out that huskies live in Alaska and so-- and wolves-- I guess some wolves do, but most wolves don't.

And so the data set on which that-- which was used in that original problem formulation, there was an extremely accurate classifier that was trained. And when they went to look to see what it had learned, basically it had learned to look for snow. And if it saw snow in the picture, it said it's a husky. And if it didn't see snow in the picture, it said it's a wolf.

So that turns out to be pretty accurate for the sample that they had. But of course, it's not a very sophisticated classification algorithm because it's possible to put a wolf in a snowy picture and it's possible to have your Husky indoors with no snow. And then you're just missing the boat on this classification.

So these guys built a particularly bad classifier by having all wolves in the training set had snow in the picture and none of the huskies did. And then they presented cases to graduate students like you guys with machine learning backgrounds. 10 balance test predictions. But they put one ringer in each category. So they put in one husky in snow and one wolf who was not in snow.

And the comparison was between pre and post experiment trust and understanding. And so before the experiment, they said that 10 of the 27 students said they trusted this bad model that they trained. And afterwards, only 3 out of 27 trusted it. So this is a kind of sociological experiment that says, yes, we can actually change people's minds about whether a model is a good or a bad one based on an experiment.

Before only 12 out of 27 students mentioned snow as a potential feature in this classifier, whereas afterwards almost everybody did. So again, this tells you that the method is providing some useful information. Now this paper set off a lot of work, including a lot of critiques of the work. And so this is one particular one from just a few months ago, the end of December.

And what these guys say is that that distance function, which includes a sigma, which is sort of the scale of distance that we're willing to go, is pretty arbitrary. In the experiments that the original authors did, they set that distance to 75% of the square root of the dimensionality of the data set. And you go, OK. I mean, that's a number. But it's not obvious that that's the best number or the right number.

And so these guys argue that it's important to tune the size of the neighborhood according to how far z, the point that you're trying to explain, is from the boundary. So if it's close to the boundary, then you ought to take a smaller region for your proximity measure. And if it's far from the boundary, this addresses the question you guys were asking about what happens if you pick a point in the middle.

And so they show some nice examples of places where, for instance, if you compare this explaining this green point, you get a nice green line that follows the local boundary. But explaining the blue point, which is close to a corner of the actual decision boundary, you got a line that's not very different from the green one. And similarly for the red point.

And so they say, well, we really need to work on that distance function. And so they come up with a method that they call LEAFAGE, which basically says, remember, what LINE did is it sampled nonexistent cases, simplified nonexistent cases. But here they're going to sample existing cases.

So they're going to learn from the training-- the original training set. But they're going to sample it by proximity to the example that they're trying to explain. And they argue that this is a good idea because, for example, in law, the notion of precedent is that you get to argue that this case is very similar to some previously decided case, and therefore it should be decided the same way.

I mean, Supreme Court arguments are always all about that. Lower court arguments are sometimes more driven by what the law actually says. But case law has been well established in British law, and then by inheritance in American law for many, many centuries. So they say, well, case-based reasoning normally involves retrieving a similar case, adapting it, and then learning that as a new precedent.

And they also argue for contrastive justification, which is not only why did you choose x, but why did you choose x rather than y as giving a more satisfying and a more insightful explanation of how some model is working? So they say, OK, similar setup. f solves the classification problem where x is the data and y is some binary classifier, you know 0, 1, if you like.

The training set is a bunch of x's. y sub true is the actual answer. y predicted is what f predicts on that x. And to explain f of z equals some particular outcome, you can define the allies of a case as ones that come up with the same answer. And you can define the enemies as one that wants to come up with a different answer.

So now you're going to sample both the allies and the enemies according to a new distance function. And the intuition they had is that the reason that the distance function in the original line work wasn't working very well is because it was a spherical distance function in n dimensional space. And so they're going to bias it by saying that the distance, this b, is going to be some combination of the difference in the linear predictions plus the difference in the two points.

And so the contour lines of the first term are these circular contour lines. This is what lime was doing. The contour lines of the second term are these linear gradients. And they add them to get sort of oval-shaped things. And this is what gives you that desired feature of being more sensitive to how close this point is to the decision boundary.

Again, there are a lot of relatively hairy details, which I'm going to elide in the class today. But they're definitely in the paper. So they also did a user study on some very simple prediction models. So this was how much is your house worth based on things like how big is it and what year was it built in and what's some subjective quality judgment of it?

And so what they show is that you can find examples that are the allies and the enemies of this house in order to do the prediction. So then they apply their algorithm. And it works. It gives you better answers. I'll have to go find that slide somewhere.

All right. So that's all I'm going to say about this idea of using simplified models in the local neighborhood of individual cases in order to explain something. I wanted to talk about two other topics. So this was a paper by some of my students recently in which they're looking at medical images and trying to generate radiology reports from those medical images.

I mean, you know, machine learning can solve all problems. I give you a collection of images and a collection of radiology reports, should be straightforward to build a model that now takes new radiological images and produces new radiology reports that are understandable, accurate, et cetera. I'm joking, of course.

But the approach they took was kind of interesting. So they've taken a standard image decoder. And then before the pooling layer, they take essentially an image embedding from the next to last layer of this image encoding algorithm. And then they feed that into a word decoder and word generator. And the idea is to get things that appear in the image that correspond to words that appear in the report to wind up in the same place in the embedding space.

And so again, there's a lot of hair. It's an LSDM based encoder. And it's modeled as a sentence decoder. And within that, there is a word decoder, and then there's a generator that generates these reports. And it uses reinforcement learning. And you know, tons of hair. But here's what I wanted to show you, which is interesting.

So the encoder takes a bunch of spatial image features. The sentence decoder uses these image features in addition to the linguistic features, the word embeddings that are fed into it.

And then for ground truth annotation, they also use a remote annotation method, which is this chexpert program, which is a rule-based program out of Stanford that reads radiology reports and identifies features in the report that it thinks are important and correct.

So it's not always correct, of course. But that's used in order to guide the generator. So here's an example. So this is an image of a chest and the ground truth-- so this is the actual radiology report-- says cardiomegalia is moderate. Bibasilar atelectasis is mild. There's no pneumothoraxal or cervical spinal fusion is partially visualized. Healed right rib fractures are incidentally noted.

By the way, I've stared at hundreds of radiological images like this. I could never figure out that this image says that. But that's why radiologists train for many, many years to become good at this stuff. So there was a previous program done by others called TieNet which generates the following report. It says AP portable upright view of the chest. There's no call no focal consolidation, effusion, or pneumothorax.

The cardio mediastinal silhouette is normal. Imaged osseous structures are intact. So if you compare this to that, you say, well, if the cardio mediastinal silhouette is normal, then where would the lower cervical spinal fusion, being partially visualized, because that's along the middle. And so these are not quite consistent.

So the system that these students built says there's mild enlargement of the cardiac silhouette. There is no pleural effusion or pneumothorax. And there's no acute osseous abnormalities. So it also missed the healed right rib fractures that were incidentally noted. But anyway, it's-- you know, the remarkable thing about a singing dog is not how well it sings but the fact that it sings at all.

And the reason I included this work is not to convince you that this is going to replace radiologists anytime soon, but that it had an interesting explanation facility. And the explanation facility uses attention, which is part of its model, to say, hey, when we reach some conclusion, we can point back into the image and say what part of the image corresponds to that part of the conclusion.

And so this is pretty interesting. You say in upright and lateral views of the chest in red, well, that's kind of the chest in red. There's moderate cardiomegaly, so here the green certainly shows you where your heart is. OK. About there and a little bit to the left. And there's no pleural effusion or pneumothorax.

This one is kind of funny. That's the blue region. So how do you show me that there isn't something? And we were surprised, actually, the way it showed us that there isn't something is to highlight everything outside of anything that you might be interested in, which is not exactly convincing that there's no pleural effusion.

And here's another example. There is no relevant change, tracheostomy tube in place, so that roughly is showing a little too wide. But it's showing roughly where a tracheostomy tube might be. Bilateral pleural effusion and compressive atelectasis. Atelectasis is when your lung tissues stick together. And so that does often happen in the lower part of the lung. And again, the negative shows you everything that's not part of the action. Yeah?

AUDIENCE: [INAUDIBLE].

PROFESSOR: Yes.

AUDIENCE: [INAUDIBLE]

PROFESSOR: No. It's trying to predict the whole model-- the whole node.

AUDIENCE: And it's not easier to have, like, one node for, like, each [INAUDIBLE]?

PROFESSOR: Yeah. But these guys were ambitious. You know, they-- what was it? Jeff Hinton said a few years ago that he wouldn't want his children to become radiologists because that field is going to be replaced by computers. I think that was a stupid thing to say, especially when you look at the state of the art of how well these things work.

But if that were true, then you would, in fact, want something that is able to produce an entire radiology report. So the motivation is there. Now, after this work was done, we ran into this interesting paper from Northeastern, which says-- but listen guys-- attention is not explanation. OK. So attention is clearly a mechanism that's very useful in all kinds of machine learning methods. But you shouldn't confuse it with an explanation.

So they say, well, assumption-- it's the assumption that the input units are accorded high attention-- that are accorded high attention weights are responsible for the model outputs. And that may not be true. And so what they did is they did a bunch of experiments where they studied the correlation between the attention weights and the gradients of the model parameters to see whether, in fact, the words that had high attention were the ones that were most decisive in making a decision in the model.

And they found that the evidence that correlation between intuitive feature importance measures, including gradient and feature erasure approaches-- so this is ablation studies and learn detention weights is weak. And so they did a bunch of experiments. There are a lot of controversies about this particular study. But what you find is that if you calculate the concordance, you know, on different data sets using different models, you see that, for example, the concordance is not very high. It's less than a half for this data set.

And you know, some of it below 0, so the opposite for this data set. Interestingly, things like diabetes, which come from the mimic data, have narrower bounds than some of the others. So they seem to have a more definitive conclusion, at least for the study.

OK. Let me finish off by talking about the opposite idea. So rather than building a complicated model and then trying to explain it in simple ways, what if we just built a simple model? And Cynthia Rudin, who's now at Duke, used to be at the Sloan School at MIT, has been championing this idea for many years.

And so she has come up with a bunch of different ideas for how to build simple models that trade off maybe a little bit of accuracy in order to be explainable. And one of her favorites is this thing called a falling rule list. So this is an example for a mammographic mass data set. So it says, if some lump has an irregular shape and the patient is over 60 years old, then there's an 85% chance of malignancy risk, and there are 230 cases in which that happened.

If this is not the case, then if the lump has the speculated margin-- so it has little spikes coming out of it-- and the patient is over 45, then there's a 78% chance of malignancy. And otherwise, if the margin is kind of fuzzy, the edge of it is kind of fuzzy, and the patient is over 60, then there's a 69% chance.

And if it has an irregular shape, then there's a 63% chance. And if it's lobular and the density is high, then there's a 39% chance. And if it's round and the patient is over 60, then there's a 26% chance. Otherwise, there's a 10% chance. And the argument is that that description of the model, of the decision-making model, is simple enough that even doctors can understand it. You're supposed to laugh.

Now, there are still some problems. So one of them is-- notice some of these are age greater than 60, age greater than 45, age greater than 60. It's not quite obvious what categories that's defining. And in principle, it could be different ages in different ones. But here's how they build it. So this is a very simple model that's built by a very complicated process.

So the simple model is the one I've just showed you. There's a Bayesian approach, a Bayesian generative approach, where they have a bunch of hyper parameters, falling rule list parameters, theta-- they calculate a likelihood, which is given a particular theta, how likely are you to get the answers that are actually in your data given the model that you generate?

And they start with a possible set of if clauses. So they do frequent clause mining to say what conditions, what binary conditions occur frequently together in the database. And those are the only ones they're going to consider because, of course, the number of possible clauses is vast and they don't want to have to iterate through those.

And then for each set of-- for each clause, they calculate a risk score which is generated by a probability distribution under the constraint that the risk score for the next clause is lower or equal to the risk score for the previous clause. There are lots of details. So there is this frequent itemset mining algorithm.

It turns out that choosing r sub l to be the logs of products of real numbers is an important step in order to guarantee that monotonicity constraint in a simple way. l, the number of clauses, is drawn from a Poisson distribution. And you give it a kind of scale that says roughly how many clauses would you be willing to tolerate in your following rule list?

And then there's a lot of computational hair where they do-- they get mean a posteriori probability estimation by using a simulated annealing algorithm. So they basically generate some clauses and then they use swap, replace, add, and delete operators in order to try different variations. And they're doing hill climbing in that space.

There's also some Gibbs sampling, because once you have one of these models, simply calculating how accurate it is is not straightforward. There's not a closed form way of doing it. And so they're doing sampling in order to try to generate that. So it's a bunch of hair. And again, the paper describes it all.

But what's interesting is that on a 30 day hospital readmission data set with about 8,000 patients, they used about 34 features, like impaired mental status, difficult behavior, chronic pain, feels unsafe, et cetera. They mind rules or clauses with support more than 5% of the database and no more than two conditions.

They set the expected length of the decision list to be eight clauses. And then they compared the decision model they got to SVM's random force logistic regression cart and an inductive logic programming approach. And shockingly to me, their method-- the following rule list method-- got an AUC of about 0.8, whereas all the others did like 0.79, 0.75 logistic regression, as usual outperformed the one they got slightly. Right?

But this is interesting, because their argument is that this representation of the model is much more easy to understand than even a logistic regression model for most human users. And also, if you look at-- these are just various runs and the different models. And their model has a pretty decent AUC up here. I think the green one is the logistic regression one.

And it's slightly better because it outperforms their best model in the region of low false positive rates, which may be where you want to operate. So that may actually be a better model. So here's their readmission rule list. And it says if the patient has bed sores and has a history of not showing up for appointments, then there's a 33% probability that they'll be readmitted within 30 days.

If-- I think some note says poor prognosis and maximum care, et cetera. So this is the result that they came up with. Now, by the way, we've talked a little bit about 30 day readmission predictions. And getting over about 70% is not bad in that domain because it's just not that easily predictable who's going to wind up back in the hospital within 30 days.

So these models are actually doing quite well, and certainly understandable in these terms. They also tried on a variety of University of California-Irvine machine learning data sets. These are just random public data sets. And they tried building these falling rule list models to make predictions. And what you see is that the AUCs are pretty good. So on the spam detection data set, their system gets about 91. Logistic regression, again, gets 97.

So you know, part of the unfortunate lesson that we teach in almost every example in this class is that simple models like logistic regression often do quite well. But remember, here they're optimizing for explainability rather than for getting the right answer. So they're willing to sacrifice some accuracy in their model in order to develop a result that is easy to explain to people.

So again, there are many variations on this type of work where people have different notions of what counts as a simple, explainable model. But that's a very different approach than the

LIME approach, which says build the hairy model and then produce local explanations for why it makes certain decisions on particular cases.

All right. I think that's all I'm going to say about explainability. This is a very hot topic at the moment, and so there are lots of papers. I think there's-- I just saw a call for a conference on explainable machine learning models. So there's more and more work in this area. So with that, we come to the end of our course. And I just wanted-- I just went through the front page of the course website and listed all the topics.

So we've covered quite a lot of stuff, right? You know, what makes health care different? And we talked about what clinical care is all about and what clinical data is like and risk stratification, survival modeling, physiological time series, how to interpret clinical text in a couple of lectures, translating technology into the clinic.

The italicized ones were guest lectures, so machine learning for cardiology and machine learning for differential diagnosis, machine learning for pathology, for mammography. David gave a couple of lectures on causal inference and reinforcement learning where David and a guest-- which I didn't note here-- disease progression and sub typing.

We talked about precision medicine and the role of genetics, automated clinical workflows, the lecture on regulation, and then recently fairness, robustness to data set shift, and interpretability. So that's quite a lot. I think we're-- we the staff are pretty happy with how the class has gone. It was our first time as this crew teaching it. And we hope to do it again.

I can't stop without giving an immense vote of gratitude to Irene and Willy, without whom we would have been totally sunk.

[APPLAUSE]

And I also want to acknowledge David's vision in putting this course together. He taught a sort of half-size version of a class like this a couple of years ago and thought that it would be a good idea to expand it into a full semester regular course and got me on board to work with him. And I want to thank you all for your hard work. And I'm looking forward to--