DAVID SONTAG: So I'll begin today's lecture by giving a brief recap of risk stratification. We didn't get to finish talking survival modeling on Thursday, and so I'll go a little bit more into that, and I'll answer some of the questions that arose during our discussions and on Piazza since. And then the vast majority of today's lecture we'll be talking about a new topic-- in particular, physiological time series modeling. I'll give two examples of physiological time series modeling-- the first one coming from monitoring patients in intensive care units, and the second one asking a very different type of question-- that of diagnosing patients' heart conditions using EKGs.

And both of these correspond to readings that you had for today's lecture, and we'll go into much more depth in these-- of those papers today, and I'll provide much more color around them. So just to briefly remind you where we were on Thursday, we talked about how one could formalize risk stratification instead of as a classification problem of what would happen, let's say, in some predefined time period, rather thinking about risk stratification as a regression question, or regression task.

Given what you know about a patient at time zero, predicting time to event-- so for example, here the event might be death, divorce, college graduation. And patient one-- that event happened at time step nine. Patient two, that event happened at time step 12. And for patient four, we don't know when that event happened, because it was censored. In particular, after time step seven, we no longer get to view any of the patients' data, and so we don't know when that red dot would be-- sometime in the future or never.

So this is what we mean by right censor data, which is precisely what survival modeling is aiming to solve. Are there questions about this setup first?

AUDIENCE: You flipped the x on--

DAVID SONTAG: Yeah, I realized that. I flipped the x and the o in today's presentation, but that's not relevant. So f of t is the probability of death, or the event occurring at time step t. And although in this slide I'm showing it as an unconditional model, in general, you should think about this as a conditional density. So you might be conditioning on some covariates or features that you have for that patient at baseline.

And very important for survival modeling and for the next things I'll tell you are the survival function, to note it as capital S of t. And that's simply 1 minus the cumulative density function.

So it's the probability that the event occurring, which is time-- which is denoted here as capital T, occurs greater than some little t. So it's this function, which is simply given to you by the integral from 0 to infinity of the density.

So in pictures, this is the density. On the x-axis is time. The y-axis is the density function. And this black curve is what I'm denoting as f of t. And this white area is capital s of c, the survival probability, or survival function. Yes?

AUDIENCE: So I just want to be clear. So if you were to integrate the entire curve, [INAUDIBLE] by infinity you're going to be [INAUDIBLE].

DAVID SONTAG: In the way that I described it to here, yes, because we're talking about the time to event. But often we might be in scenarios where the event may never occur, and so that-- you can formalize that in a couple of different ways. You could put that at point mass at s of infinity, or you could simply say that the integral from 0 to infinity is some quantity less than 1.

And in the readings that I'm referencing in the very bottom of those slides-- it shows you how you can very easily modify all of the frameworks I'm telling you about here to deal with that scenario where the event may never occur. But for the purposes of my presentation, you can assume that the event will always occur at some point. It's a very minor modification where you, in essence, divide the densities by a constant, which accounts for the fact that it wouldn't integrate to one otherwise.

Now, a key question that has to be solved when trying to use a parametric approach to survivor modeling is, what should that f of t look like? What should that density function look like? And what I'm showing you here is a table of some very commonly used density functions. What you see in these two columns-- on the right hand column is the density function f of t itself. Lambda denotes some parameter of the model. t is the time.

And on this second middle column is the survival function. So this is obtained for these particular parametric forms by an analytical solution solving that integral from t to infinity. This is the analytic solution for that. And so these go by common names of exponential, weeble, log-normal, and so on. And critically, all of these have support only on the positive real numbers, because the event can ever occur at negative time.

Now, we live in a day and age where we no longer have to make standard parametric assumptions for densities. We could, for example, try to formalize the density as some output of some deep neural network. But if we don't use a parametric approach, so there are two ways to try to do that. One way to do that would be to say that we're going to model the post-- the distribution, f of t, as one of these things, where lambda or whatever the parameters of distribution are given to by the output of, let's say, a deep neural network on the covariate x. So that would be one approach.

A very different approach would be a non-parametric distribution where you say, OK, I'm going to define f of t extremely flexibly, not as one of these forms. And there one runs into a slightly different challenge, because as I'll show you in the next slide, to do maximum likelihood estimation of these distributions from censor data, one needs to get-- one needs to make use of this survival function, s of t.

And so if you're f if t is complex, and you don't have a nice analytic solution for s of t, then you're going to have to somehow use a numerical approximation of s of t during limiting. So it's definitely possible, but it's going to be a little bit more effort. So now here's where I'm going to get into maximum likelihood estimation of these distributions, and to define for you the likelihood function, I'm going to break it down into two different settings. The first setting is an observation which is uncensored, meaning we do observe when the event-- death, for example-- occurs. And in that case, the probability of the event-- it's very simple. It's just probability of the event occurring at capital-- at capital T, random variable T, equals a little t-- is just f or t. Done.

However, what happens if, for this data point, you don't observe when the event occurred because of censoring? Well, of course, you could just throw away that data point, not use it in your estimation, but that's precisely what we mentioned at the very beginning of last week's lecture-- was the goal of survival modeling to not do that, because if we did that, it would introduce bias into our estimation procedure.

So we would like to be able to use that observation that this data point was censored, but the only information we can get from that observation is that capital T, the event time, must have occurred some time larger than the observed-- the time of censoring, which is little t here. So we don't know precisely when capital T was, but we know it's something larger than the observed centering time little of t.

And that, remember, is precisely what the survival function is capturing. So for a censored observation, we're going to use capital S of t within the likelihood. So now we can then combine these two for censored and uncensored data, and what we get is the following likelihood objective.

This is-- I'm showing you here the log likelihood objective. Recall from last week that little b of i simply denotes is this observation censored or not? So if bi is 1, it means the time that you're given is the time of the censoring event. And if bi is 0, it means the time you're given is the time that the event occurs. So here what we're going to do is now sum over all of the data points in your data set from little i equals 1 to little n of bi times log of probability under the censored model plus 1 minus bi times log of probability under the uncensored model. And so this bi is just going to switch on which of these two you're going to use for that given data point.

So the learning objective for maximum likelihood estimation here is very similar to what you're used to in learning distributions with the big difference that, for censored data, we're going to use the survival function to estimate its probability. Are there any questions? And this, of course, could then be optimized via your favorite algorithm, whether it be stochastic gradient descent, or second order method, and so on. Yep?

AUDIENCE: I have a question about the a kind of side project. You mentioned that we could use [INAUDIBLE].

DAVID SONTAG: Yes.

AUDIENCE: And then combine it with the parametric approach.

DAVID SONTAG: Yes.

AUDIENCE: So is that true that we just still have the parametric assumption that we kind of map the input to the parameters?

DAVID SONTAG: Exactly. That's exactly right. So consider the following picture where for-- this is time, t. And this is f of t. You can imagine for any one patient you might have a different function. You might-- but they might all be of the same parametric form. So they might be like that, or maybe they're shifted a little bit. So you think about each of these three things as being from the same parametric family of distributions, but with different means.

And in this case, then the mean is given to as the output of the deep neural network. And so that would be the way it would be used, and then one could just back propagate in the usual way to do learning. Yep?

AUDIENCE: Can you repeat what b sub i is?

DAVID SONTAG: Excuse me?

AUDIENCE: Could you repeat what b sub i is?

DAVID SONTAG: b sub i is just an indicator whether the i-th data point was censored or not censored. Yes?

AUDIENCE: So [INAUDIBLE] equal it's more a probability density function [INAUDIBLE].

DAVID SONTAG: Cumulative density function.

AUDIENCE: Yeah, but [INAUDIBLE] probability. No, for the [INAUDIBLE] it's probability density function.

DAVID SONTAG Yes, so just to--

AUDIENCE: [INAUDIBLE]

DAVID SONTAG: Excuse me?

AUDIENCE: Will that be any problem to combine those two types there?

DAVID SONTAG: That's a very good question. So the observation was that you have two different types of probabilities used here. In this case, we're using something like the cumulative density, whereas here we're using the probability density function. The question was, are these two on different scales? Does it make sense to combine them in this type of linear fashion with the same weighting? And I think it does make sense.

So think about a setting where you have a very small time range. You're not exactly sure when this event occurs. It's something in this time range. In the setting of the censored data, where that time range could potentially be very large, your model is providing-- your log probability is somehow going to be much more flat, because you're covering much more probability mass.

And so that observation, I think, intuitively is likely to have a much-- a bit of a smaller effect on the overall learning algorithm. These observations-- you know precisely where they are, and so as you deviate from that, you incur the corresponding log loss penalty. But I do think that it makes sense to have them in the same scale. If anyone in the room has done work with [INAUDIBLE] modeling and has a different answer to that, I'd love to hear it. Not today, but maybe someone in the future will answer this question differently.

I'm going to move on for now. So the remaining question that I want to talk about today is how one evaluates survival models. So we talked about binary classification a lot in the context of risk stratification in the beginning, and we talked about how area under the ROC curve is one measure of classification performance, but here we're doing more-- something more akin to regression, not classification.

A standard measure that's used to measure performance is known as the C-statistic, or concordance index. Those are one in the same-- and is defined as follows. And it has a very intuitive definition. It sums over pairs of data points that can be compared to one another, and it says, OK, what is the likelihood of the event happening for an event that occurs before an event-- another event. And what you want is that the likelihood of the event that, on average, in essence, should occur later should be larger than the event that should occur earlier.

I'm going to first illustrate it with this picture, and then I'll work through the math. So here's the picture, and then we'll talk about the math. So what I'm showing you here are every single observation in your data set, and they're sorted by either the censoring time or the event time. So by black, I'm illustrating uncensored data points. And by red, I'm denoting censored data points.

Now, here we see that this data point-- the event happened before this data point's censoring event. Now, since this data point was censored, it means it's true event time you could think about as sometime into the far future. So what we would want is that the model gives that the probability that this event happens by this time should be larger than the probability that this event happens by this time, because this actually occurred first. And these two are comparable together-- to each other.

On the other hand, it wouldn't make sense to compare y2 and y4, because both of these were censored data points, and we don't know precisely when they occurred. So for example, it could have very well happened that the event 2 happened after event 4. So what I'm showing you here with each of these lines are the pairwise comparisons that are actually possible to make. You can make pairwise comparisons, of course, between any pair of events that actually did occur, and you can make pairwise comparisons between censored events and events that occurred before it.

Now, if you now look at this formula, the formula in this indicate-- this is looking at an indicator of survival functions between pairs of data points, and which pairs of data points? It was precisely those pairs of data points, which I'm showing comparisons of with these blue lines here.

So we're going to sum over i such that bi is equal to 0, and remember that means it is an uncensored data point. And then we look at-- we look at yi compared to all other yj that's great-- that has a value greater than-- both censored and uncensored. Now, if your data had no sensor data points in it, then you can verify that, in fact, this corresponds-- so there's one other assumption one has to make, which is that-- suppose that your outcome is binary. And so if you might wonder how you get a binary outcome from this, imagine that your density function looked a little bit like this, where it could occur either at time 1 or time 2. So something like that.

So if the event can occur at only two times, not a whole range of times, then this is analogous to a binary outcome. And so if you have a binary outcome like this and no censoring, then, in fact, that C-statistic is exactly equal to the area under the ROC curve. So that just connects it a little bit back to things we're used to. Yep?

AUDIENCE: Just to make sure that I understand. So y1 is going to be we observed an event, and y2 is going to be we know that no event occurred until that day?

DAVID SONTAG: Every dot corresponds to one event, either censored or not.

AUDIENCE: Thank you.

DAVID SONTAG: And they're sorted. In this figure, they're sorted by the time of either the censoring or the event occurring. So I talked to-- when I talked about C-statistic, it-- that's one way to measure performance of your survival modeling, but you might remember that I-- that when we talked about binary classification, we said how area under there ROC curve in itself is very limiting, and so we should think through other performance metrics of relevance.

So here are a few other things that you could do. One thing you could do is you could use the mean squared error. So again, thinking about this as a regression problem. But of course, that only makes sense for uncensored data points. So focus just in the uncensored data points, look to see how well we're doing at predicting when the event occurs.

The second thing one could do, since you have the ability to define the likelihood of an observation, censored or not censored, one could hold out data, and look at the held-out likelihood or log likelihood of that held-out data. And the third thing you could do is you can-- after learning using this survival modeling framework, one could then turn it into a binary classification problem by, for example, artificially choosing time ranges, like greater than three months is 1. Less than three months is 0.

That would be one crude definition. And then once you've done a reduction to a binary classification problem, you could use all of the existing performance metrics they're used to thinking about for binary classification to evaluate the performance there-- things like positive predictive value, for example. And you could, of course, choose different reductions and get different performance statistics out. So this is just a small subset of ways to try to evaluate survivor modeling, but it's a very, very rich literature. And again, on the bottom of these slides, I pointed you to several references that you could go to to learn more.

The final comment I wanted to make is that I only told you about one estimator in today's lecture, and that's known as the likelihood based estimator. But there is a whole other estimation approach for survival modelings, which is very important to know about, that are called partial likelihood estimators. And for those of you who have heard of Cox proportional hazards models-- and I know they were discussed in Friday's recitation-- that's an example of a class of model that's commonly used within this partial likelihood estimator.

Now, at a very intuitive level, what this partial likelihood estimator is doing is it's working with something like the C-statistic. So notice how the C-statistic only looks at relative orderings of events-- of their event occurrences. It doesn't care about exactly when the event occurred or not.

In some sense, there's a constant. There's-- in this survival function, which could be divided out from both sides of this inequality, and it wouldn't affect anything about the statistic. And so one could think about other ways of learning these models by saying, well, we want to learn a survival function such that it gets the ordering correct between data points.

Now, such a survival function wouldn't do a very good job. There's no reason it would do any good at getting the precise time of when an event occurs, but if your goal were to just figure out what is the sorted order of patients by risk so that you're going to do an intervention on the 10 most risky people, then getting that order incorrect is going to be enough, and that's precisely the intuition used behind these partial likelihood estimators-- so they focus on something which is a little bit less than the original goal, but in doing so, they can have much better statistical complexity, meaning the amount of data they need in order to fit this models well. And again, this is a very rich topic. All I wanted to do is give you a pointer to it so that you can go read more about it if this is something of interest to you.

So now moving on into the recap, one of the most important points that we discussed last week was about non-stationarity. And there was a question posted to Piazza, which was really interesting, which is how do you actually deal with non-stationarity. And I spoke a lot about it existing, and I talked about how to test for it, but I didn't say what to do if you have it.

So I thought this was such an interesting question that I would also talk about it a bit during lecture. So the short answer is, if you have to have a solution that you deploy tomorrow, then here's the hack that sometimes works. You take your most recent data, like the last three months' data, and you hope that there's not much non-stationarity within last three months. You throw out all the historical data, and you just train using the most recent data. So a bit unsatisfying, because you might have now extremely little data left to learn with, but if you have enough volume, it might be good enough. But the real interesting question from a research perspective is how could you optimally use that historical data.

So here are three different ways. So one way has to do with imputation. Imagine that the way in which your data was non-stationary was because there were, let's say, parts of time when certain features were just unavailable. I gave you this example last week of laboratory test results across time, and I showed you how there are sometimes these really big blocks of time where no lab tests are available, or very few are available.

Well, luckily we live in a world with high dimensional data, and what that means is there's often a lot of redundancy in the data. So what you could imagine doing is imputing features that you observed to be missing, such that the missingness properties, in fact, aren't changing as much across time after imputation. And if you do that as a pre-processing step, it may allow you to make use of much more of the historical data.

A different approach, which is intimately tied to that, has to do with transforming the data. Instead of imputing it, transforming it into another representation altogether, such that that presentation is invariant across time. And here I'm giving you a reference to this paper by Ganin et al from the *Journal of Machine Learning Research* 2016, which talks about how to do domain and variant learning of neural networks, and that's one approach to do so. And I view those two as being very similar-- imputation and transformations.

A second approach is to re-weight the data to look like the current data. So imagine that you go back in time, and you say, you know what? I ICD-10 codes, for some very weird reason-- this is not true, by the way-- ICD-10 codes in this untrue world happen to be used between March and April of 2003. And then they weren't used again until 2015.

So instead of throwing away all of the previous data, we're going to recognize that those-- that three month interval 10 years ago was actually drawn from a very similar distribution as what we're going to be testing on today. So we're going to weight those data points up very much, and down weight the data points that are less like the ones from today. That's the intuition behind these re-weighting approaches, and we're going to talk much more about that in the context of causal inference, not because these two have to do with each other, but they have-- they end up using a very similar technique for how to deal with datas that shift, or covariate shift.

And the final technique that I'll mention is based on online learning algorithms. So the idea there is that there might be cut points, change points across time. So maybe the data looks one way up until this change point, and then suddenly the data looks really different until this change point, and then suddenly the data looks very different on into the future.

So here I'm showing you there are two change points in which data set shift happens. What these online learning algorithms do is they say, OK, suppose we were forced to make predictions throughout this time period using only the historical data to make predictions at each point in time. Well, if we could somehow recognize that there might be these shifts, we could design algorithms that are going to be robust to those shifts. And then one could try to analyze-- mathematically analyze those algorithms based on the amount of regret they would have to, for example, an algorithm that knew exactly when those changes were. And of course, we don't know precisely when those changes were. And so there's a whole field of algorithms trying to do that, and here I'm just give me one citation for a recent work.

So to conclude risk stratification-- this is the last slide here. Maybe ask your question after class. We've talked about two approaches for formalizing risk stratification-- first as binary classification. Second as regression. And in the regression framework, one has to think about censoring, which is why we call it survival modeling.

Second, in our examples, and again in your homework assignment that's coming up next week, we'll see that often the variables, the features that are most predictive make a lot of sense. In the diabetes case, we said-- we saw how patients having comorbidities of diabetes, like hypertension, or patients being obese were very predictive of patients getting diabetes.

So you might ask yourself, is there something causal there? Are those features that are very predictive in fact causing-- what's causing the patient to develop type 2 diabetes? Like, for example, obesity causing diabetes. And this is where I want to caution you. You shouldn't interpret these very predictive features in a causal fashion, particularly not when one starts to work with high dimensional data, as we do in this course.

The reason for that is very subtle, and we'll talk about that in the causal inference lectures, but I just wanted to give you a pointer now that you shouldn't think about it in that way. And you'll understand why in just a few weeks. And finally we talked about ways of dealing with missing data. I gave you one feature representation for the diabetes case, which was designed to deal with missing data. It said, was there any diagnosis code 250.01 in the last three months? And if there was, you have a 1. If you don't, 0.

So it's designed to recognize that you don't have information, perhaps, for some large chunk of time in that window. But that missing data could also be dangerous if that missingness itself has caused you to non-stationarity, which is then going to result in your test distribution looking different from your training distribution. And that's where approaches that are based on imputation could actually be very valuable, not because they improve your predictive accuracy when everything goes right, but because they might improve your predictive accuracy when things go wrong.

And so one of your readings for last week's lecture was actually an example of that, where they used a Gaussian process model to impute much of the missing data in a patient's continuous vital signs, and then they used a recurrent neural network to predict based on that imputed data. So in that case, there are really two things going on. First is this robustness to data set shift, but there's a second thing, which is going on as well, which has to do with a trade-off between the amount of data you have and the complexity of the prediction problem. By doing imputations, sometimes you make your problem look a bit simpler, and simpler algorithms might succeed where otherwise they would fail because not having enough data. And that's something that you saw in that last week's reading.

So I'm done with risk stratification. I'll take a one minute breather for everyone in the room, and then we'll start with the main topic of this lecture, which is physiological time-series modeling. Let's say started.

So here's a baby that's not doing very well. This baby is in the intensive care unit. Maybe it was a premature infant. Maybe it's a baby who has some chronic disease, and, of course, parents are very worried. This baby is getting very close monitoring. It's connected to lots of different probes.

In number one here, it's illustrating a three probe-- three lead ECG, which we'll be talking about much more, which is measuring its heart, how the baby's heart is doing. Over here, this number three is something attached to the baby's foot, which is measuring its-- it's a pulse oximeter, which is measuring the baby's oxygen saturation, the amount of oxygen in the blood. Number four is a probe which is measuring the baby's temperature and so on.

And so we're really taking really close measurements of this baby, because we want to understand how is this baby doing. We recognize that there might be really sudden changes in the baby's state of health that we want to be able to recognize as early as possible. And so behind the scenes, next to this baby, you'll, of course, have a huge number of monitors, each of the monitors showing the readouts from each of these different signals.

And this type of data is really prevalent in intensive care units, but you'll also see in today's lecture how some aspects of this data are now starting to make its way to the home, as well. So for example, EKGs are now available on Apple and Samsung watches to help understand-- help to help with diagnosis of arrhythmias, even for people at home.

And so from this type of data, there are a number of really important use cases to think about. The first one is to recognize that often we're getting really noisy data, and we want to try to infer the true signal. So imagine, for example, the temperature probe. The baby's true temperature might be 98.5, but for whatever reason-- we'll see a few reasons here today-- maybe you're getting an observation of 93.

And you didn't know. Is that actually the true baby temperature? In which case we-- it would be in a lot of trouble. Or is that an anomalous reading? So we like t be able to distinguish between those two things. And in other cases, we are interested in not necessarily fully understanding what's going on with the baby along each of those axes, but we just want to use that data for predictive purposes, for risk stratification, for example.

And so the type of machine learning approach that we'll take here will depend on the following three factors. First, do we have label data available? For example, do we know the ground truth of what the baby's true temperature was, at least for a few of the babies in the training set? Second.

Do we have a good mechanistic or statistical model of how this data might evolve across time? We know a lot about hearts, for example. Cardiology is one of those fields of medicine where it's really well studied. There are good simulators of hearts, and how they beat across time, and how that affects the electrical stimulation across the body.

And if we have these good mechanistic or statistical models, that can often allow one to trade off not having much label data, or just not having much data period. And it's really these three points which I want to illustrate the extremes of in today's lecture-- what do you do when you don't have much data, and what you do when-- what you can do when you have a ton of data. And I think it's going to be really informative for us as we go out into the world and will have to tackle each of those two settings.

So here's an example of two different babies with very different trajectories. One in the x-axis here is time in seconds. The y-axis here-- I think seconds, maybe minutes. The y-axis here is beats per minute of the baby's heart rate, and you see in some cases it's really fluctuating a lot up and down. In some cases, it's sort of going in a similar-- in one direction, and in all cases, the short term observations are very different from the long range trajectories.

So the first problem that I want us to think about is one of trying to understand, how do we deconvolve between the truth of what's going on with, for example, the patient's blood pressure or oxygen versus interventions that are happening to them? So on the bottom here, I'm showing examples of interventions.

Here in this oxygen uptake, we notice how between roughly 1,000 and 2,000 seconds suddenly there's no signal whatsoever. And that's an example of what's called dropout. Over here, we see a different type of-- the effect of a different intervention, which is due to a probe recalibration. Now, at that time, there was a drop out followed by a sudden change in the values, and that's really happening due to a recalibration step. And in both of these cases, what's going on with the individual might be relatively constant across time, but what's being observed is dramatically affected by those interventions.

So we want to ask the question, can we identify those artifactual processes? Can we identify that these interventions were happening at those points in time? And then, if we could identify them, then we could potentially subtract their effect out. So we could impute the data, which we know-- now know to be missing, and then have this much higher quality signal used for some downstream predictive purpose, for example.

And the second reason why this can be really important is to tackle this problem called alarm fatigue. Alarm fatigue is one of the most important challenges facing medicine today. As we get better and better in doing risk stratification, as we come up with more and more diagnostic tools and tests, that means these red flags are being raised more and more often. And each one of these has some associated false positive rate for it. And so the more tests you have-- suppose the false positive rate is kept constant-- the more tests you have, the more likely it is that the union of all of those is going to be some error.

And so when you're in an intensive care unit, there are alarms going off all the time. And something that happens is that nurses end up starting to ignore those alarms, because so often those alarms are false positives, are due to, for example, artifacts like what I'm showing you here.

And so if we had techniques, such as the ones we'll talk about right now, which could recognize when, for example, the sudden drop in a patient's heart rate is due to an artifact and not due to the patient's true heart rate dropping-- if we had enough confidence in that-- in distinguishing those two things, then we might not decide to raise that red flag. And that might reduce the amount of false alarms, and that then might reduce the amount of alarm fatigue. And that could have a very big impact on health care.

So the technique which we'll talk about today goes by the name of switching linear dynamical systems. Who here has seen a picture like this on-- this picture on the bottom before. About half of the room. So for the other half of the room, I'm going to give a bit of a recap into probabilistic modeling. All of you are now familiar with general probabilities. So you're used to thinking about, for example, univariate Gaussian distributions.

We talked about how one could model survival, which was an example of such a distribution, but for today's lecture, we're going to be thinking now about multivariate probability distributions. In particular, we'll be thinking about how a patient's state-- let's say their true blood pressure-- evolves across time. And so now we're interested in not just the random variable at one point in time, but that same random variable at the second point in time, third point in time, fourth point in time, fifth point in time, and so on.

So what I'm showing you here is known as a graphical model, also known as a Bayesian network. And it's one way of illustrating a multivariate probability distribution that has particular conditional independence properties. Specifically, in this model, one node corresponds to one random variable. So this is describing a joint distribution on x1 through x6, y1 through y6. So it's this multivariate distribution on 12 random variables.

The fact that this is shaded in simply denotes that, at test time, when we use these models, typically these y variables are observed. Whereas our goal is usually to infer the x variables. Those are typically unobserved, meaning that our typical task is one of doing posterior inference to infer the x's given the y's.

Now, associated with this graph, I already told you the nodes correspond to random variables. The graph tells us how is this joint distribution factorized. In particular, it's going to be factorized in the following way-- as the product over random variables of the probability of the i-th random variable. I'm going to use z to just denote a random variable. Think of z as the union of x and y. zi conditioned on the parents-- the values of the parents of zi.

So I'm going to assume this factorization, and in particular for this graphical model, which goes by the name of a Markov model, it has a very specific factorization. And we're just going to read it off from this definition. So we're going to go in order-- first x1, then y1, then x2, then y2, and so on, which is going based on a root to children transversal of this graph.

So the first random variable is x1. Second variable is y2, and what are the parents of y-- sorry, what are the parents of y1. Everyone can say out loud.

AUDIENCE: x1.

DAVID SONTAG: x1. So y1 in this factorization is only going to depend on x1. Next we have x2. What are the parents of x2? Everyone say out loud?

AUDIENCE: x1.

DAVID SONTAG: x1. Then we have y2. What are the parents of y2. Everyone say out loud.

AUDIENCE: x2.

DAVID SONTAG: x2 and so on. So this joint distribution is going to have a particularly simple form, which is given to by this factorization shown here. And this factorization corresponds one to one with the particular graph in the way that I just told you.

And in this way, we can define a very complex probability distribution by a number of much simpler conditional probability distributions. For example, if each of the random variables were binary, then to describe probability of y1 given x1, we only need two numbers. For each value of x1, either 0 or 1, we give the probability of y1 equals 1. And then, of course, probably y1 equals 0 is just 1 minus that. So we can describe that very complicated joint distribution by a number of much smaller distributions.

Now, the reason why I'm drawing it in this way is because we're making some really strong assumptions about the temporal dynamics in this problem. In particular, the fact that x3 only has an arrow from x2 and not from x1 implies that x3 is conditionally independent of x1. If you knew x2's value. So in some sense, think about this as cutting. If you're to take x2 out of the model and remove all edges incident on it, then x1 and x3 are now separated from one another. They're independent. Now, for those of you who do know graphical models, you'll recognize that that type of independent statement that I made is only true for Markov models, and the semantics for Bayesian networks are a little bit different. But actually for this model, it's-- they're one and the same.

So we're going to make the following assumptions for the conditional distributions shown here. First, we're going to suppose that xt is given to you by a Gaussian distribution. Remember xt--t is denoting a time step. Let's say 3-- it only depends in this picture-- the conditional distribution only depends on the previous time step's value, x2, or xt minus 1.

So you'll notice how I'm going to say here xt is going to distribute as something, but the only random variables in this something can be xt minus 1, according to these assumptions. In particular, we're going to assume that it's some Gaussian distribution, whose mean is some linear transformation of xt minus 1, and which has a fixed covariance matrix q. So at each step of this process, the next random variable is some random walk from the previous random variable where you're moving according to some Gaussian distribution.

In a very similar way, we're going to assume that yt is drawn also as a Gaussian distribution, but now depending on xt. So I want you to think about xt as the true state of the patient. It's a vector that's summarizing their blood pressure, their oxygen saturation, a whole bunch of other parameters, or maybe even just one of those. And y1 are the observations that you do observe. So let's say x1 is the patient's true blood pressure. y1 is the observed blood pressure, what comes from your monitor.

So then a reasonable assumption would be that, well, if all this were equal, if it was a true observation, then y1 should be very close to x1. So you might assume that this covariance matrix is-- the covariance is-- the variance is very, very small. y1 should be very close to x1 if it's a good observation. And of course, if it's a noisy observation-- like, for example, if the probe was disconnected from the baby, then y1 should have no relationship to x1. And that dependence on the actual state of the world I'm denoting here by these superscripts, s of t. I'm ignoring that right now, and I'll bring that in in the next slide.

Similarly, the relationship between x2 and x1 should be one which captures some of the dynamics that I showed in the previous slides, where I showed over here now this is the patient's true heart rate evolving across time, let's say. Notice how, if you look very locally, it looks like there are some very, very big local dynamics. Whereas if you look more globally, again, there's some smoothness, but there are some-- again, it looks like some random changes across time. And so those-- that drift has to somehow be summarized in this model by that A random variable. And I'll get into more detail about that in just a moment.

So what I just showed you was an example of a linear dynamical system, but it was assuming that there were none of these events happening, none of these artifacts happening. The actual model that we were going to want to be able to use then is going to also incorporate the fact that there might be artifacts. And to model that, we need to introduce additional random variables corresponding to whether those artifacts occurred or not. And so that's now this model.

So I'm going to let these S's-- these are other random variables, which are denoting artifactual events. They are also evolving with time. For example, if there's artifactual factual event at three seconds, maybe there's also an artifactual event at four seconds. And we like to model the relationship between those. That's why you have these arrows.

And then the way that we interpret the observations that we do get depends on both the true value of what's going on with the patient and whether there was an artifactual event or not. And you'll notice that there's also an edge going from the artifactual events to the true values to note the fact that those interventions might actually be affecting the patient. For example, if you give them a medication to change their blood pressure, then that procedure is going to affect the next time step's value of the patient's blood pressure.

So when one wants to learn this model, you have to ask yourself, what types of data do you have available? Unfortunately, it's very hard to get data on both the ground truth, what's going on with the patient, and whether these artifacts truly occurred or not. Instead, what we actually have are just these observations. We get these very noisy blood pressure draws across time.

So what this paper does is it uses a maximum likelihood estimation approach, where it recognizes that we're going to be learning from missing data. We're going to explicitly think of these x's and the s's as latent variables. And we're going to maximize the likelihood of the whole entire model, marginalizing over x and s. So just maximizing the marginal likelihood over the y's.

Now, for those of you who have studied unsupervised learning before, you might recognize that as a very hard learning problem. In fact, it's-- that likelihood is non-convex. And one could imagine all sorts of a heuristics for learning, such as gradient descent, or, as this paper uses, expectation maximization, and because of that non-convexity, each of these algorithms typically will only reach a local maxima of the likelihood.

So this paper uses EM, which intuitively iterates between inferring those missing variables-- so imputing the x's and the s's given the current model, and doing posterior inference to infer the missing variables given the observed variables, using the current model. And then, once you've imputed those variables, attempting to refit the model. So that's called the m-step for maximization, which updates the model and just iterates between those two things. That's one learning algorithm which is guaranteed to reach a local maxima of the likelihood under some regularity assumptions.

And so this paper uses that algorithm, but you need to be asking yourself, if all you ever observe are the y's, then will this algorithm ever recover anything close to the true model? For example, there might be large amounts of non-identifiability here. It could be that you could swap the meaning of the s's, and you'd get a similar likelihood on the y's.

That's where bringing in domain knowledge becomes critical. So this is going to be an example where we have no label data or very little label data. And we're going to do unsupervised learning of this model, but we're going to use a ton of domain knowledge in order to constrain the model as much as possible.

So what is that domain knowledge? Well, first we're going to use the fact that we know that a true heart rate evolves in a fashion that can be very well modeled by an autoregressive process. So the autoregressive process that's used in this paper is used to model the normal heart rate dynamics. In a moment, I'll tell you how to model the abnormal heart rate observations.

And intuitively-- I'll first go over the intuition, then I'll give you the math. Intuitively what it does is it recognizes that this complicated signal can be decomposed into two pieces. The first piece shown here is called a baseline signal, and that, if you squint your eyes and you sort or ignore the very local fluctuations, this is what you get out.

And then you can look at the residual of subtracting this signal, subtracting this baseline from the signal. And what you get out looks like this. Notice here it's around 0 mean. So it's a 0 mean signal with some random fluctuations, and the fluctuations are happening here at a much faster rate than-- and for the original baseline.

And so the sum of bt and this residual is a very-- it looks-- is exactly equal to the true heart rate. And each of these two things we can model very well. This we can model by a random walk with-- which goes very slowly, and this we can model by a random walk which goes very quickly. And that is exactly what I'm now going to show over here on the left hand side.

bt, this baseline signal, we're going to model as a Gaussian distribution, which is parameterized as a function of not just bt minus 1, but also bt minus 2, and bt minus 3. And so we're going to be taking a weighted average of the previous few time steps, where we're smoothing out, in essence, the observation-- the previous few observations. If you were to-- if you're being a keen observer, you'll notice that this is no longer a Markov model.

For example, if this p1 and p2 are equal to 2, this then corresponds to a second order Markov model, because each random variable depends on the previous two time steps of the Markov chain. And so after-- so you would model now bt by this process, and you would probably be averaging over a large number of previous time steps to get this smooth property. And then you'd model xt minus bt by this autoregressive process, where you might, for example, just be looking at just the previous couple of time steps. And you recognize that you're just doing much more random fluctuations.

And then-- so that's how one would now model normal heart rate dynamics. And again, it's just-- this is an example of a statistical model. There is no mechanistic knowledge of hearts being used here, but we can fit the data of normal hearts pretty well using this. But the next question and the most interesting one is, how does one now model artifactual events? So for that, that's where some mechanistic knowledge comes in.

So one models that the probe dropouts are given by recognizing that, if a probe is removed from the baby, then there should no longer be-- or at least if you-- after a small amount of time, there should no longer be any dependence on the true value of the baby. For example, the blood pressure, once the blood pressure probe is removed, is no longer related to the baby's true blood pressure.

But there might be some delay to that lack of dependence. And so-- and that is going to be encoded in some domain knowledge. So for example, in the temperature probe, when you remove the temperature probe from the baby, it starts heating up again-- or it starts cooling, so assuming that the ambient temperature is cooler than the baby's temperature. So you take it off the baby. It starts cooling down.

How fast does it cool down? Well, you could assume that it cools down with some exponential decay from the baby's temperature. And this is something that is very reasonable, and you could imagine, maybe if you had label data for just a few of the babies, you could try to fit the parameters of the exponential very quickly. And in this way, now, we parameterize the conditional distribution of the temperature probe, given both the state and whether the artifact occurred or not, using this very simple exponential decay.

And in this paper, they give a very similar type of-- they make similar types of-- analogous types of assumptions for all of the other artifactual probes. You should think about this as constraining these conditional distributions I showed you here. They're no longer allowed to be arbitrary distributions, and so that, when one does now expectation maximization to try to maximize the marginal likelihood of the data, you've now constrained it in a way that you hopefully are moved on to identifyability of the learning problem. It makes all of the difference in learning here.

So in this paper, their evaluation did a little bit of fine tuning for each baby. In particular, they assumed that the first 30 minutes near the start consists of normal dynamics so that's there are no artifacts. That's, of course, a big assumption, but they use that to try to fine tune the dynamic model to fine tune it for each baby and for themselves.

And then they looked at the ability to try to identify artifactual processes. Now, I want to go a little bit slowly through this plot, because it's quite interesting. So what I'm showing you here is a ROC curve of the ability to predict each of the four different types of artifacts. For example, at any one point in time, was there a blood sample being taken or not? At any one point in time, was there a core temperature disconnect of the core temperature probe?

And to evaluate it, they're assuming that they have some label data for evaluation purposes only. And of course, you want to be at the very far top left corner up here. And what we're showing here are three different curves-- the very faint dotted line, which I'm going to trace out with my cursor, is the baseline. Think of that as a much worse algorithm. Sorry. That's that line over there. Everyone see it?

And this approach are the other two lines. Now, what's differentiating those other two lines corresponds to the particular type of approximate inference algorithm that's used. To do this posterior inference, to infer the true value of the x's given your noisy observations in the model given here is actually a very hard inference problem. Mathematically, I think one can show that it's an NP-hard computational problem.

And so they have to approximate it in some way, and they use two different approximations here. The first approximation is based on what they're calling a Gaussian sum approximation, and it's a deterministic approximation. The second approximation is based on a Monte Carlo method.

And what you see here is that the Gaussian sum approximation is actually dramatically better. So for example, in this blood sample one, that the ROC curve looks like this for the Gaussian sum approximation. Whereas for the Monte Carlo approximation, it's actually significantly lower. And this is just to point out that, even in this setting, where we have very little data, we're using a lot of domain knowledge, the actual details of how one does the math-- in particular, the proximate inference-- can make a really big difference in the performance of this system. And so it's something that one should really think deeply about, as well.

I'm going to skip that slide, and then just mention very briefly this one. This is showing an inference of the events. So here I'm showing you three different observations. And on the bottom here, I'm showing the prediction of when artifact-- two different artifactual events happened. And these predictions were actually quite good, using this model.

So I'm done with that first example, and-- and the-- just to recap the important points of that example, it was that we had almost no label data. We're tackling this problem using a cleverly chosen statistical model with some domain knowledge built in, and that can go really far.

So now we'll shift gears to talk about a different type of problem involving physiological data, and that's of detecting atrial fibrillation. So what I'm showing you here is an AliveCore device. I own one of these. So if you want to drop by my E25 545 office, you can-- you can play around with it. And if you attach it to your mobile phone, it'll show you your electric conductance through your heart as measured through your two fingers touching this device shown over here. And from that, one can try to detect whether the patient has atrial fibrillation.

So what is atrial fibrillation? Good question. It's [INAUDIBLE]. So this is from the American Heart Association. They defined atrial fibrillation as a quivering or irregular heartbeat, also known as arrhythmia. And one of the big challenges is that it could lead to blood clot, stroke, heart failure, and so on.

So here is how a patient might describe having atrial fibrillation. My heart flip-flops, skips beats, feels like it's banging against my chest wall, particularly when I'm carrying stuff up my stairs or bending down. Now let's try to look at a picture of it.

So this is a normal heartbeat. Hearts move-- pumping like this. And if you were to look at the signal output of the EKG of a normal heartbeat, it would look like this. And it's roughly corresponding to the different-- the signal is corresponding to different cycles of the heartbeat. Now for a patient who has atrial fibrillation, it looks more like this.

So much more obviously abnormal, at least in this figure. And if you look at the corresponding signal, it also looks very different. So this is just to give you some intuition about what I mean by atrial fibrillation.

So what we're going to try to do now is to detect it. So we're going to take data like that and try to classify it into a number of different categories. Now this is something which has been studied for decades, and last year, 2017, there was a competition run by Professor Roger Mark, who is here at MIT, which is trying to see, well, how could-- how good are we at trying to figure out which patients have different types of heart rhythms based on data that looks like this?

So this is a normal rhythm, which is also called a sinus rhythm. And over here it's atrial-- this is an example one patient who has atrial fibrillation. This is another type of rhythm that's not atrial fibrillation, but is abnormal. And this is a noisy recording-- for example, if a patient's-- doesn't really have their two fingers very well put on to the two leads of the device.

So given one of these categories, can we predict-- one of these signals, could predict which category it came from? So if you looked at this, you might recognize that they look a bit different. So could some of you guess what might be predictive features that differentiate one of these signals from the other? In the back?

AUDIENCE: The presence and absence of one of the peaks the QRS complex are [INAUDIBLE].

DAVID SONTAG: So speak in English for people who don't know what these terms mean.

AUDIENCE: There is one large piece, which can-- probably we can consider one mV and there is another peak, which is sort of like-- they have reverse polarity between normal rhythm and [INAUDIBLE].

DAVID SONTAG: Good. So are you a cardiologist?

AUDIENCE: No.

DAVID SONTAG: No, OK. So what the student suggested is one could look for sort of these inversions to try to describe it a little bit differently. So here you're suggesting the lack of those inversions is predictive of an abnormal rhythm. What about another feature that could be predictive? Yep?

AUDIENCE: The spacing between the peaks is more irregular with the AF.

DAVID SONTAG: The spacing between beats is more irregular with the AF rhythm. So you're sort of looking at this. You see how here this spacing is very different from this spacing. Whereas in the normal rhythm, sort of the spacing looks pretty darn regular. All right, good.

So if I was to show you 40 examples of these and then ask you to classify some new ones, how well do you think you'll be able to do? Pretty well? I would be surprised if you couldn't do reasonably well at least distinguishing between normal rhythm and AF rhythm, because there seem to be some pretty clear signals here.

Of course, as you get into alternatives, then the story gets much more complex. But let me dig in a little bit deeper into what I mean by this. So let's define some of these terms. Well, cardiologists have studied this for a really long time, and they have-- so what I'm showing you here is one heart cycle. And they've-- you can put names to each of the peaks that you would see in a regular heart cycle-- so that-- for example, that very high peak is known as the R peak.

And you could look at, for example, the interval-- so this is one beat. You could look at the interval between the R peak of one beat and the R peak of another peak, and define that to be the RR interval. In a similar way, one could take-- one could find different distinctive elements of the signal-- by the way, each-- each time step corresponds to the heart being in a different position. For a healthy heart, these are relatively deterministic. And so you could look at other distances and derive features from those distances, as well, just like we were talking about, both within a beat and across beats. Yep?

AUDIENCE: So what's the difference between a segment and an interval again?

DAVID SONTAG: I don't know what the difference between a segment and an interval is. Does anyone else know? I mean, I guess the interval is between probably the heads of peaks, whereas segments might refer to within a interval. That's my guess.

Does someone know better? For the purpose of today's class, that's a good enough understanding. The point is this is well understood. One could derive features from this.

AUDIENCE: By us.

DAVID SONTAG: By us. So what would a traditional approach be to this problem? So this is-- I'm pulling this figure from a paper from 2002. What it'll do is it'll take in that signal. It'll do some filtering of it. Then it'll run a peak detection logic, which will find these peaks, and then it'll measure intervals between these peaks and within a beat. And it'll take those computations or make some decision based on it. So that's a traditional algorithm, and they work pretty reasonably.

And so what do I mean by signal processing? Well, this is an example of that. I encourage any of you to go home today and try to code up a peaked finding algorithm. It's not that hard, at least not to get an OK one. You might imagine keeping a running tab of what's the highest signal you've seen so far. Then you look to see what is the first time it drops, and the second time-- and the next time it goes up larger than, let's say, the previous-- suppose that one of-- you want to look for when the drop is-- the maximum value-- recent maximum value divided by 2. And then you-- then you reset. And you can imagine in this way very quickly coding up a peak finding algorithm.

And so this is just, again, to give you some intuition behind what a traditional approach would be. And then you can very quickly see that that-- once you start to look at some intervals between peaks, that alone is often good enough for predicting whether a patient has atrial fibrillation. So this is a figure taken from paper in 2001 showing a single patient's time series. So the x-axis is for that single patient, their heart beats across time. The y-axis is just showing the RR interval between the previous beat and the current beat.

And down here in the bottom is the ground truth of whether the patient is assessed to have-- to be in-- to have a normal rhythm or atrial fibrillation, which is noted as this higher value here. So these are AF rhythms. This is normal. This is AF again.

And what you can see is that the RR interval actually gets you pretty far. You notice how it's pretty high up here. Suddenly it drops. The RR interval drops for a while, and that's when the patient has AF. Then it goes up again. Then it drops again, and so on. And so it's not deterministic, the relationship, but there's definitely a lot of signal just from that.

So you might say, OK, well, what's the next thing we could do to try to clean up the signal a little bit more? So flash backwards from 2001 to 1970 here at MIT, studied by-- actually, no, this is not MIT. This is somewhere else, sorry. But still 1970-- where they used a Markov model very similar to the Markov models we were just talking about in the previous example to model what a sequence of normal RR intervals looks like versus what a sequence of abnormal, for example, AF RR intervals looks like.

And in that way, one can recognize that, for any one observation of an RR interval might not by itself be perfectly predictive, but if you look at sort of a sequence of them for a patient with atrial fibrillation, there is some common pattern to it. And you can-- one can detect it by just looking at likelihood of that sequence under each of these two different models, normal and abnormal. And that did pretty well-- even better than the previous approaches for-- for predicting atrial fibrillation.

This is the paper I wanted to say from MIT. Now 1991, this is also from Roger Mark's group. Now this is a neural network based approach, where it says, OK, we're going to take a bunch of these things. We're going to derive a bunch of these intervals, and then we're going to throw that through a black box supervised machine learning algorithm to predict whether a patient has AF or not.

So these are very-- first of all, there are some simple approaches here that work reasonably well. Using neural networks in this domain is not a new thing, but where are we as a field? So as I mentioned, there was this competition last year, and what I'm showing you here-- the citation is from one of the winning approaches.

And this winning approach really brings the two paradigms together. It extracts a large number of expert derived features-- so shown here. And these are exactly the types of things you might think, like proportion, median RR interval of regular rhythms, max RR irregularity measure. And there's just a whole range of different things that you can imagine manually deriving from the data. And you throw all of these features into a machine learning algorithm, maybe a random forest, maybe a neural network, doesn't matter. And what you get out is a slightly better algorithm than what if you had just come up with a simple rule on your own. That was the winning algorithm then.

And in the summary paper, they conjectured that, well, maybe it's the case that they were-- they'd expected that convolutional neural networks would win. And they were surprised that none of the winning solutions involved convolution neural networks. And they conjectured that may be the reason why is because maybe with these 8,000 patients that they had [INAUDIBLE] that just wasn't enough to give the more complex models advantage.

So flip forward now to this year and the article that you read in your readings in *Nature Medicine,* where the Stanford group now showed how a convolutional neural network approach, which is, in many ways, extremely naive-- all it does is it takes the sequence data in. It makes no attempt at trying to understand the underlying physiology, and just predicts from that-- can do really, really well.

And so there are couple of differences that I want to emphasize to the previous work. First, the censor is different. Whereas the previous work used this alive core censor, in this paper from Stanford, they're using a different censor called the Zio patch, which is attached to the human body and conceivably much less noisy. So that's one big difference.

The second big difference is that there's dramatically more data. Instead of 8,000 patients to train from, now they have over 90,000 records from 50,000 different patients to train from. The third major difference is that now, rather than just trying to classify into four categories-- normal, abnormal, other, or noisy-- now we're going to try to classify into 14 different categories. We're, in essence, breaking apart that other class into much finer grain detail of different types of abnormal rhythms.

And so here are some of those other abnormal rhythms, things like complete heart block, and a bunch of other names I can't pronounce. And from each one of these, they gathered a lot of data. And that actually did-- so it's not described in the paper, but I've talked to the authors, and they did-- they gathered this data in a very interesting way.

So they sort of-- they did their training iteratively. They looked to see where their errors were, and then they went and gathered more data from patients with that subcategory. So many of these other categories are very under-- might be underrepresented in the general population, but they actually gather a lot of patients of that type in their data set for training purposes. And so I think those three things ended up making a very big difference.

So what is their convolutional network? Well, first of all, it's a 1-D signal. So it's a little bit different from the con nets you typically see in computer vision, and I'll show you an illustration of that in the next slide. It's a very deep model. So it's 34 layers.

So the input comes in on the very top in this picture. It's passed through a number of layers. Each layer consists of convolution followed by rectified linear units, and there is sub sampling at every other layer so that you go from a very wide signal-- so a very long-- I can't remember how long-- 1 second long signal summarized down into sort of much-- just many smaller number of dimensions, which you then have a sort of fully connected layer at the bottom to do for your predictions. And then they also have these shortcut connections, which allow you to pass information from earlier layers down to the very end of the network, or even into intermediate layers. And for those of you who are familiar with residual networks, it's the same idea.

So what is a 1D convolution? Well, it looks a little bit like this. So this is the signal. I'm going to just approximate it by a bunch of 1's and 0's. I'll say this is a 1. This is a 0. This is a 1, 1, so on.

A convolutional network has a filter associated with it. That filter is then applied in a 1D model. It's applied in a linear fashion. It's just taken a dot product with the filter's values, with the values of the signal at each point in time.

So it looks a little bit like this, and this is what you get out. So this is the convolution of a single filter with the whole signal. And the computation I did there-- so for example, this first number came from taking the dot product of the first three numbers-- 1, 0, 1-- with the filter. So it's 1 times 2 plus 3 times 0 plus 1 times 1, which is 3.

And so each of the subsequent numbers was computed in the same way. And I usually have you figure out what this last one is, but I'll leave that for you to do at home. And that's what a 1D convolution is. And so they have-- they do this for lots of different filters. Each of those filters might be of varying lengths, and each of those will detect different types of signal patterns.

And in this way, after having many layers of these, one can, in an automatic fashion, extract many of the same types of signals used in that earlier work, but also be much more flexible to detect some new ones, as well. Hold your question, because I need to wrap up.

So in the paper that you read, they talked about how they evaluated this. And so I'm not going to go into much depth in it now. I just want to point out two different metrics that they used.

So the first metric they used was what they called a sequential error metric. What that looked at is you had this very long sequence for each patient, and they labeled different one second intervals of that sequence into abnormal, normal, and so on. So you could ask, how good are we at labeling each of the different points along the sequence? And that's the sequence metric.

The different-- the second metric is the set metric, and that looks at, if the patient has something that's abnormal anywhere, did you detect it? So that's, in essence, taking an or of each of those 1 second intervals, and then looking across patients. And from a clinical diagnostic perspective, the set metric might be most useful, but then when you want to introspect and understand where is that happening, then the sequential metric is important.

And the key take home message from the paper is that, if you compared the model's predictions-- this is, I think, using an f1 metric-- to what you would get from a panel of cardiologists, these models are doing as well, if not better than these panels of cardiologists. So this is extremely exciting. This is technology-- or variance of this is technology that you're going to see deployed now. So for those of you who have purchased these Apple watches, these Samsung watches, I don't know exactly what they're using, but I wouldn't be surprised if they're using techniques similar to this. And you're going to see much more of that in the future.

So this is going to be really the first example in this course so far of something that's really been deployed. And so in summary, we're very often in the realm of not enough data. And in this lecture today, we gave two examples how you can deal with that. First, you can try to use mechanistic and statistical models to try to work in settings where you don't have much data. And in other extremes, you do have a lot of data, and you can try to ignore that, and just use these black box approaches. That's all for today.