[CLICK]

[SQUEAK]

[PAGES RUSTLING]

[MOUSE DOUBLE-CLICKS]

PROFESSOR: So today we'll be continuing along the theme of risk stratification. I'll spend the first half to 2/3 of today's lecture continuing where we left off last week before the discussion. I'll talk about how does one derive the labels that one uses within a supervised machine learning approach. I'll continue talking about how one evaluates risk stratification models. And then I'll talk about some of the subtleties that arise when you want to use machine learning for health care, specifically for risk stratification. And I think that's going to be one of the most interesting parts of today's lecture. In the last third of today's lecture, I'll be talking about how one can rethink the supervised machine learning problem, not to be a classification problem, but be something closer to a regression problem.

And one now thinks about not will someone, for example, develop diabetes within one to three years from now, but when precisely will they develop diabetes-- so the time to event. Then one has to start to really think very carefully about the censoring issues that I alluded to last week. And so I'll formalize those notions in the language of survival modeling. And I'll talk about how one can do maximum likelihood estimation in that setting, and how one should do evaluation in that setting.

So in our lecture last week, I gave you this example of risk stratification for type 2 diabetes. The goal, just to remind you, was as follows. 25% of people in the United States have undiagnosed type 2 diabetes. If we could take health insurance claims data that's available for everyone who has health insurance, and use that to predict who, in the near-term-- next one to three years-- is likely to be newly diagnosed with type 2 diabetes, then we could use it to risk-stratify patient population. We could use that, then, to figure out who is most at risk, do interventions for those patients, to try to get them diagnosed and get them started on treatment if relevant.

But what I didn't talk much about was where did those labels come from. How do we know that someone had a diabetes onset in that window that I show up there on the top? So what are the answers? I mean, all of you should have read the paper by Razavian. And then also you should hopefully have some ideas. Thoughts? A hint-- it was in supplementary material.

How did we define a positive case in that paper? Yep.

AUDIENCE: Drugs they were on.

PROFESSOR: Drugs they were on. OK, yeah, so for example, metformin, glucose-- sorry, insulin.

AUDIENCE: I think they did include metformin actually.

PROFESSOR: Metformin is a tricky case. Because metformin is often used for alternative indications. But there are many medications, such as insulin, which are used pretty exclusively for treating diabetes. And so you can look to see, does a patient have a record of taking one of these diabetic medications in that window that we're using to define the outcome? If you see a record of a medication, you might conjecture, this patient probably has diabetes.

But what about it they don't have any medication listed in that time window? What could you conclude then? Any ideas? Yeah.

AUDIENCE: If you look at the HBA1C value, and you know the normal range, and if you see the [INAUDIBLE] above like 7.5 or 7.

PROFESSOR: So you're giving me an alternative approach, not looking at medications, but looking at laboratory test results. Look at their HBA1C results, which measures approximately an average of three-month glucose values. And if that's out of range, then they're diabetic. And that's, in fact, usually used as a definition of diabetes.

But that didn't answer my original question. Why is just looking at diabetic medications not enough?

AUDIENCE: Some of the diabetic medications can be used to treat other conditions.

PROFESSOR: Sometimes there's ambiguity in diabetic medications. But we've sort of dealt with that already by trying to choose an unambiguous set. What are other reasons?

AUDIENCE: You're starting with the medicine at the onset of diabetes [INAUDIBLE].

PROFESSOR: Oh, that's a really interesting point-- not the one I was thinking about, but I like it-- which is that a patient might have been diagnosed with type 2 diabetes, but they, for whatever reason, in that communication between provider and patient, they decided we're not going to start treatment yet. So they might not yet be on treatment for diabetes, yet the whole health care system might be very well aware that the patient is diabetic, in which case doing these interventions for that patient might be irrelevant. Yep, another reason?

AUDIENCE: So a lot of people are just not diagnosed for diabetes. So they have it. So one label means that they have diabetes, and the other label is a combination of people who have and don't have diabetes.

PROFESSOR: So the point was, often you just might not be diagnosed for diabetes. That, unfortunately, is not something that we're going to able to solve here. It is an issue, but we have no solution for it.

No, rather there's a different point that I want to get at, which is that this data has biases in it. So even if a patient is on a diabetes medication, for whatever reason-- maybe they are paying cash for those medications. If they're paying cash for those medications, then there's not going to be any record for the patient taking those medications in the health insurance claims. Because the health insurer didn't have to pay for it.

But the reason that you gave is also a very interesting reason. And both of them are valid. So for all of these reasons, just looking at the medications alone is going to be insufficient. And as was just suggested a moment ago, looking at other indicators, like, for example, does the patient have an abnormal blood glucose value or HBA1C value would also provide information.

So it's non-trivial, right? And part of what you're going to be doing in your next problem set, problem set 2, is going to be thinking through how does one actually do this cohort construction, not just what is your inclusion/exclusion criteria, but also how do you really derive those labels from that data set.

Now the traditional answer to this has two steps to it. Step 1 is to actually manually label some patients. So you take a few hundred patients, and you go through their data. You actually look at their data, and decide, is this patient diabetic or are they not diabetic? And the reason why you have to do that is because often what you might think of is obvious-- like, oh, if they're on diabetes medication, they're diabetic-- has flaws to it. And until you really dig down and look at the data, you might not recognize that that criteria has a flaw in it. So that chart review is really an essential part of this process.

Then the second step is, how do you generalize to get that label now for everyone in your population. And again, there, there are usually two different types of approaches. The first approach is to come up with some simple rule to try to then extrapolate to everyone. For example, if they have, A, diabetes medication, or an abnormal lab test result, that would be an example of a rule. And then you could then apply that to everyone.

But even those rules can be really tricky to derive. And I'll show you some examples of that in just a moment. And as we know, machine learning is sometimes good as an alternative for coming up with a rule. So there's often now a second approach to this being more and more commonly used in the literature, which is to actually use machine learning itself to derive the labels.

And this is a bit subtle, because it's machine learning for machine learning. So I want to break that down for one second. When you're trying to derive the labels, what you want to know is not, at time T, what's going to happen at time T plus W and onwards-- that's the original machine learning task that we set out to solve-- but rather, given everything you know about the patient, including the future data, is this patient newly diagnosed with diabetes in that window that I show in black there, between T plus W and onward. OK?

So for example, this machine learning problem, this new machine learning problem, could take, as input, lab test results, and medications, and a whole bunch of other data. And you then use the few examples you labeled in step 1 to try to predict, is this patient currently diabetic or not. You then use that model to extrapolate to the whole population. And now you have your outcome label. It might be a little bit imperfect, but hopefully it's much better than what you could have gotten with a rule. And then, now using those outcome labels, you solve your original machine learning problem. Is that clear? Any questions?

AUDIENCE: I have one.

PROFESSOR: Yep.

AUDIENCE: How do you evaluate yourself then, if you have these labels that were produced with machine learning, which are probabilistic?

PROFESSOR: So that's where this first step is really important. You've got to get ground truth somehow. And of course once you get that ground truth, you create a train-and-validate set of that ground truth. You run your machine learning algorithm with the trained one. You'd look at its performance metrics on that validate set for the label prediction problem. And that's how you get confidence in it.

But let's try to break this down a little bit. So first of all, what does this chart review step look like? Well, if it's an electronic health record system, what you often do is you will pull up Epic, or Cerner, or whatever the commercial EHR system is. And you will actually start looking at the patient data. You'll read notes written by previous doctors about this patient. And you'll look at their blood test results across time, medications that they're on. And from that you can usually tell pretty coherent story what's going on with your patient.

Of course even better-- or the best way to get data is to do a prospective study. So you actually have a research assistant standing in the room when a patient walks into a provider. And they talk to the patient, and they take down really very clear notes what this patient has, what they don't have. But that's usually too expensive to do prospectively. So usually what we do is do this retrospectively.

Now, if you're working with health insurance claims data, you usually don't have the luxury of looking at notes. And so what, in my group, we type typically do is we build, actually, a visualization tool. And by the way, I'm a machine learning person. I don't know anything about visualization. Neither do I claim to be good at it.

But you can't do the machine learning work unless you really understand your data. So we had to build this tool in order to look at the data, in order to try to do that first step of understanding, did we even characterize diabetes correctly.

So I'm not going go deep into it. By the way, you can download this. It's an open source tool. But ballpark what I'm showing you here is one patient's data. I'm showing on this x-axis, time, going from April to December. And on the y-axis, I'm showing events as they occurred.

So in orange are diagnosis codes that were recorded for the patient. In green are procedure codes. In blue are laboratory tests. And if you see, on a given line, multiple dots along that same line, it means that same lab test was performed multiple times. And you could click on it to see what the results were. And in this way, you could start to tell a coherent story what's going on with your patient.

All right, so tools like this is what you're going to need to able to do that first step from something like health insurance claims data.

Now, traditionally, that first step, which then leads you to label some data, and then, from there, you go and come up with these rules, or do a machine learning algorithm to get the label, usually that's a paper in itself. Of course, not of interest to the computer science community, but of extreme interest to the health care community. So usually there's a first paper, academic paper, which evaluates this process for deriving the label, and then there are much later papers which talk about what you could do with that label, such as the machine learning problem we originally set out to solve.

So let's look at an example of one of those rules. Here is a rule, to derive from health insurance claims data whether a patient has type 2 diabetes. Now, this isn't quite the same one that we used in that paper, but it gets the idea across. First you look to see, did the patient have a diagnosis code for type 1 diabetes. If the answer is no, you continue. If the answer is yes, you've sort of ruled out. Because you say, OK, this patient's abnormal blood test results are because they have type 1 diabetes, not type 2 diabetes. Type 1 diabetes usually is what you can think of as juvenile diabetes, is diagnosed much earlier. And there's a different mechanism behind it.

Then you look at other things-- OK, is there a diagnosis code for type 2 diabetes somewhere in the patient's data? If so, you go to the right, and you look to see, is there a medication, an Rx, for type 1 diabetes in the data. If the answer is no, you continue down this way. If the answer is yes, you go this way. A yes of a type 1 diabetes medication doesn't alone rule out the patient. Because maybe the same medications are used for type 1 as for type 2. So there's some other things you need to do there.

Right, you can see that this starts to really quickly become complicated. And these manual-based approaches end up having pretty bad positive-- so they're designed usually to have pretty high positive predictive value. But they end up having pretty bad recall, in that they don't end up finding all of the patients. And that's really why the machine-learning-based approaches end up being very important for this type of problem.

Now, this is just one example of what I call a phenotype. I call this a phenotype. That's just what the literature calls it. It's a phenotype for type 2 diabetes. And the word, phenotype, in this context is exactly the same thing as the label. Yep.

AUDIENCE: What is abnormal mean?

PROFESSOR: For example, if the HA1C result is 6.5 or higher, you might say the patient has diabetes.

AUDIENCE: OK, so this is a lab result, not a medical--

PROFESSOR: Correct, yeah, thanks. Other questions.

AUDIENCE: What's the phenotype, which part exactly is the phenotype, like, the whole thing?

PROFESSOR: The whole thing, yeah. So the construction, where you say-- you follow this decision tree, and you get to a conclusion, which is case, which means, yes they're type 2 diabetic. And if ever you don't reach this point, then the answer is no, they're not type 2 diabetic. That's what I mean by-- so that labeling is what we're calling the phenotype of type 2 diabetes.

Now later in the semester, people will use the word, phenotype, to mean something else. It's an overloaded term. But this is what it's called in this context as well.

Now here's an example of a website-- it's from the PheKB project-- where you will find tens to close to 100 of these phenotypes that have been arduously created for a whole range of different conditions. OK, so if you go to this website, and you click on any one of these conditions, like appendicitis, autism, cataracts, you'll see a different diagram of this sort I just showed you. So this is a real thing. This is something that the medical community really needs to do in order to try to derive the label that we can then use in our machine learning task.

AUDIENCE: I'm just curious, is the lab value ground truth? Like if somebody has diabetes, then they must have [INAUDIBLE]. It means they have been diagnosed, and they must have--

PROFESSOR: Well, so, for example, you might have an abnormal glucose value for a variety of reasons. One reason is because you might have what's called gestational diabetes, which is diabetes that's induced due to pregnancy. But those patients typically-- well, although it's a predictive factor, they don't always have long-term type 2 diabetes. So even the laboratory test alone doesn't tell the whole story.

AUDIENCE: You could be diagnosed without having abnormal diabetic?

PROFESSOR: That's much less common here. The story will change in the future, because there will be a whole range of new diagnosis techniques that might use new modalities, like gene expression, for example. But typically, today, the answer is yes to that. Yep.

AUDIENCE: So if these are made by doctors, does that mean, for every single disease, there's one definitive phenotype?

PROFESSOR: These are usually made by health outcomes researchers, which usually have clinicians on their team. But the type of people who often work on these often come from the field of epidemiology, for example. And so what was your question again?

AUDIENCE: Is there just one phenotype for every single disease?

PROFESSOR: Is there one phenotype for every different disease? In the ideal world, you'd have at least one phenotype for every single disease that could possibly exist. Now, of course, you might be interested in different aspects. Like you might be interested in not knowing just does the patient have autism, but where they are on their autism spectrum. You might not be interested in knowing just, do they have it now, but you also might want to know when did they get it. So there's a lot of subtleties that could go into this.

But building these up is really slow. And validating them to make sure that they're going to work across multiple data sets is really challenging, and usually is a negative result. And so it's been a very slow process to do this manually, which has led me and many others to start thinking about the machine learning approaches for how to do it automatically.

AUDIENCE: Just as a follow-up, is there any case where there's, like, five autism phenotypes, for example, or multiple competing ones?

PROFESSOR: Yes. So there are often many different such rule-based systems that give you conflicting results. Yes, that happens all the time.

AUDIENCE: Can these rule-based systems provide an estimate of when their condition was onset?

PROFESSOR: Right, so that's getting at one of the subtleties I just mentioned-- can these tell you when the onset happened? They're not typically designed to do that, but one can come up with one to do it. And so one way to try to do that is you change those rules to have a time period associate to it. And then you can imagine applying those rules in a sliding window to the patient data to see, when is the first time that it triggers. And that would be one way to try to get a sense of when onset was. But there's a lot of subtleties to that, too.

So I'm going to move on now. I just want to give it some sense of what that deriving the labels ends up looking like. Let's now turn to evaluation. So a very commonly used approach in this field is to compute what's known as the Receiver-Operator Curve, or ROC curve.

And what this looks at is the following. First of all, this is well-defined for a binary classification problem. For a binary classification problem when you're using a model that outputs, let's say, a probability or some continuous value, then you could use that continuous valid prediction. If you wanted to make a prediction, you usually threshold it, right? So you say, if it's greater than 0.5, it's a prediction of 1. If it's less than 0.5, prediction of zero.

But here we might be interested in not just what minimizes, let's say, 0-1 loss, but you might also be interested in trading off, let's say, false positives for false negatives. And so you might choose different thresholds. And you might want to quantify how do those trade-offs look for different choices of those thresholds of this continuous value prediction. And that's what the ROC curve will show you.

So as you move along the threshold, you can compute, for every single threshold, what is the true positive rate, and what is the false positive rate. And that gives you a number. And you try all possible thresholds, that gives you a curve. And then you can compare curves from different machine learning algorithms.

For example, here, I'm showing you, in the green line, the predictive model obtained by using what we're calling the traditional risk factors, so something like eight or 10 different risk factors for type 2 diabetes that are very commonly used in the literature. Versus in blue, it's showing you what you'd get if you just used a naive L1-regularized logistic regression model with no domain knowledge, just sort of throw in the bag of features.

And you want to be up there. You want to be in that top left corner. That's the goal here. So you would like that blue curve to be up there, and then all the way to the right.

Now, one way to try to quantify in a single number how useful any one ROC curve is is by looking at what's called the area under the ROC curve. And mathematically, this is exactly what you'd expect. This area is the area under the ROC curve. So you could just integrate the curve, and you get that number out. Now, remember, I told you you want to be in the upper left quadrant. And so the goal was to get an area under the ROC curve of a 1.

Now, what would a random prediction give you? Any idea? So if you're to just flip a coin and guess-- what do you think?

AUDIENCE: 0.5.

PROFESSOR: 0.5?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Well, so I was a little bit misleading when I said you just flip a coin. You got to flip a coin with sort of different noise rates. And each one of those will get you sort of a different place along this curve. And if you look at the curve that you get from these random guesses, it's going to be the straight line from 0 to 1. And as you said, that will then have an AUC of 0.5. So 0.5 is going to be random guessing. 1 is perfect. And your algorithm is going to be somewhere in between.

Now, of relevance to the rest of today's lecture is going to be an alternative definition-- alternative way of computing the area under the ROC curve. So one way to compute it is literally as I said. You create that curve, and you integrate to get the area under it. But one can show mathematically-- I'm not going to give you the derivation here, but you can look it up on Wikipedia. One can show mathematically that an equivalent way of computing the area under the ROC curve is to compute the probability that an algorithm will rank a positive-labeled patient over a negative-labeled patient.

So mathematically what I'm talking about is the following thing. You're going to sum over pairs of patients where-- I'm going to call x1 is a patient with label y1 equals 1. And x2 is a patient with label y-- actually, I'll call it-- yeah, with label x2 equals 1. So these are two different patients.

I think I'm going to rewrite it like this-- xi and xj, just for generality's sake. So we're going to sum this up over all choices of i and j such that yi and yj have different labels. So that should say yj equals 0.

And then you're going to look at-- what you want to happen, like suppose that you're using a linear model here. So your prediction is given to you by, let's say, w.xj. What you want is that this should be smaller than w.xi. So the j data point, remember, was the one that got the 0-th and the i-th data point is the one that got the 1 label. So we want the score of the data point that should've been 1 to be higher than the score of the data point which should've gotten the label 0. And you just count up-- this is an indicator function. You just count up how many of those were correctly ordered. And then you're just going to normalize by the total number of comparisons that you do. And it turns out that that is exactly equal to the area under the ROC curve. And it really makes clear that this is a notion that really cares about ranking. Are you getting the ranking of patients correct? Are you ranking the ones who should have been given 1 higher than the ones that should have gotten the label 0.

And importantly, this whole measure is actually invariant to the label imbalance. So you might have a very imbalanced data set. But if you were to re-sample with now making it a balanced data set, your AUC of your predictive model wouldn't change. And that's a nice property to have when it comes to evaluating settings where you might have artificially created a balanced data set for computational concerns. Even though the true setting is imbalanced, there at least you know that the numbers are going to be the same in both settings.

On the other hand, it also has lots of disadvantages. Because often you don't care about the performance of the whole entire curve. Often you care about particular parts along the curve. So for example, in last week's lecture, I argued that really what we often care about is just the positive predictive value for a particular threshold. And we want that to be as high as possible for as few people as possible. Like, find the 100 most risky people, and look at what fraction of them actually developed type 2 diabetes. And that setting, what you're really looking at is this part of the curve.

And so it turns out there are generalizations of area under the curve that focus on parts of the curve. And that goes by the name of partial AUC. For example, if you just integrated from 0 to, let's say, 0.1 of the curve, then you could still get a number to compare two different curves, but it's focusing on the area of that curve that's actually relevant for your predictive purposes, for your task at hand.

So that's all I want to say about receiver-operator characteristic curves. Any questions? Yep.

AUDIENCE: Could you talk more about what the drawbacks were of using this. Does the class imbalance-- is the class imbalance, then, always a positive effect?

PROFESSOR: So the thing is, when you want to use this approach, depending on how you're using the [INAUDIBLE], you might not be able to tolerate a 0.8 false positive rate. So in some sense, what's going on in this part of the curve might be completely irrelevant for your task. And so one of the algorithms-- so one of these curves-- might look like it's doing really, really well over here, and pretty poorly over here. But if you're looking at the full area under the ROC curve, you won't notice that. And so that's one of the big problems. Yeah.

AUDIENCE: And when would you use this versus precision recall or--

PROFESSOR: Yeah, so a lot of the community is interested in precision recall curves. And precision recall curves, as opposed to receiver-operator curves, have the property that they are not invariant to class imbalance, which in many settings is of interest, because it will allow you to capture these types of quantities. I'm not going to go into depth about your reasons for one or the other. But that's something that you could read up about, and I encourage you to post to Piazza about, and we have discussion on Piazza.

So the last evaluation quantity that I want to talk about is known as calibration. And calibration, as I've defined it here, has to do with binary classification problems. Now, before you dig into this figure, which I'll explain in a moment, let me just give you the gist of what I mean by calibration. Suppose that your model outputs a probability. So you do logistic regression. You get a probability out. And your model says, for these 10 patients, that their likelihood of dying in the next 48 hours is 0.7. Suppose that's what your model output. If you were on the receiving end of that result, so you heard that, 0.7, what should you expect about those 10 people? What fraction of them should actually die in the next 48 hours? Everyone could scream out loud.

[INTERPOSING VOICES]

PROFESSOR: So seven of them. Seven of the 10 you would expect to die in the next 48 hours if the probability for all of them that was output was 0.7. All right, that's what I mean by calibration.

So if, on the other hand, what you found was that only one of them died, then it would be a very weird number that you're outputting. And so the reason why this notion of calibration, which I'll define formally in a second, is so important, is when you're out putting a probability, and when you don't really know how that probability is going to be used. If you knew-- if you had some task loss in mind. And you knew that all that mattered was the actual prediction, 1 or 0, then that would be fine.

But often predictions in machine learning are used in a much more subtle way. Like for example, often your doctor might have more information than your computer has. And so often they might want to take the result that your computer predicts, and weigh that against other evidence. Or in some settings, it's not just weighting about other evidence. Maybe it's also about making a decision. And that decision might take exertion-- a utility, for example, a patient preference for suffering versus getting a treatment that could have big, adverse consequences.

And that's something that Pete is going to talk about much more later in the semester, I think, how to formalize that notion. But at this point, I just want to sort of get out the point that the probabilities themselves could be important. And having the probabilities be meaningful is something that one can now quantify.

So how do we quantify it? Well, one way to try to quantify it is to create the following prompt. Actually, we'll call it a histogram. So on the x-axis is the predicted probability. So that's what I meant by p-hat. On the y-axis is the true probability. It's what I mean when I say the fraction of individuals with that predicted probability that actually got the positive outcome. That's going to be the y-axis. So I'll call that the true probability.

And what we would like to see is that this is a line, a straight line, meaning these two should always be equal. And in the example I gave, remember I said that there were a bunch of people with 0.7 probability predicted, but for whom only one out of them actually got the positive end. So that would have been something like over here. Whereas you would have expected it to be over there.

So you might ask, how do I create such a plot from finite data? Well, a common way to do so is to bin your data. So you'll create intervals. So this bin is the bin from 0 to 0.1. This bin is the bin from 0.1 to 0.2, and so on.

And then you'll look to see, OK, how many people for whom the predicted probability was between 0 and 0.1 actually died? And you'll get a number out. And now here's where I can go to this plot. That's exactly what I'm showing you here.

So for now, ignore the bar charts and the bottom, and just look at the line. So let's focus on just the green line. Here I'm showing you several different models. For now, just focus on the green line. So the green line, by the way, notice it looks pretty good. It's almost a straight line. So how did I compute it? Well, first of all, notice the number of ticks are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. OK, so there are 10 points along this line. And each of those corresponds to one of these bins. So the first point is the 0 to 0.1 bin. The second point is the 0.1 to 0.2 bin, and so on. So that's how it computed this.

The next thing you notice is that I have confidence intervals. And the reason I compute these confidence intervals is because sometimes you just might not have that much data in one of these bins. So for example, suppose your algorithm almost never said that someone has a predictive probability of 0.99. Then until you get a ton of data, you're not going to know what fraction of those individuals actually went on to develop the event.

And you should be looking at sort of the confidence interval of this line, which should take that into consideration. And a different way to try to understand that notion, now looking at the numbers, is what I'm showing you in the bar charts in the bottom. On the bar charts, I'm showing you the number of individuals or the fraction of individuals who actually got that predicted probability.

So now let's start comparing the lines. So the blue line shown here is a machine learning algorithm which is predicting infection in the emergency rooms. It's a slightly different problem than the diabetes one we looked at earlier. And it's using a bag of words model from clinical text. The red line is using just chief complaint. So it's using one piece of structured data that you get at one point of time in the ER. So it's using very little information. And you can see that both models are somewhat well calibrated.

But the intervals-- the confidence intervals of both the red and the purple lines gets really big towards the end. And if you look at these bar charts, it explains why, because the models that use less information end up being much more risk-averse. So they will never predict a very high probability. They will always sort of stay in this lower regime. And that's why we have very big confidence intervals there.

OK, so that's all I want to say about evaluation. And I won't take any questions on this right now, because I really want to get on to the rest of the lecture. But again, if you have any questions, post to Piazza, and I'm happy to discuss them with you offline.

So, in summary, we've talked about how to reduce risk stratification to binary classification. I've told you how to derive the labels. I've given you one example of machine learning algorithm you can use, and I talked to you about how to evaluate it. What could possibly go wrong?

So let's look at some examples. And these are a small number of examples of what could possibly go wrong. There are many more.

So here's some data. I'm showing you-- for the same problem we looked at before, diabetes onset, I'm showing you the prevalence of type 2 diabetes as recorded by, let's say, diagnosis codes across time. All right, so over here is 1980. Over here is 2012. Look at that. It is not a flat line. Now, what does that mean? Does that mean that the population is eating much more unhealthy from 1980 to 2012, and so more people are becoming diabetic? That would be one plausible answer.

Another plausible explanation is that something has changed. So in fact I'm showing you with these blue lines, well, in fact, there was a change in the diagnostic criteria for diabetes.

And so now the patient population actually didn't change much between, let's say, this time point at that time point. But what really led it to this big uptick, according to one theory, is because the diagnostic criteria changed. So who we're calling diabetic has changed. Because diseases are, at the end of the day, a human-made concept, you know, what do we call some disease.

And so the data is changing, as you see here. Let me show you another example. Oh, by the way, so the consequence of that is that automatically-derived labels-- for example, if you use one of those phenotyping algorithms I showed you earlier, the rules-- what the label is derived for over here might be very different from the label that's derived from over here, particularly if it's using data such as diagnosis codes that have changed in meaning over the years. So that's one consequence. There'll be other consequences I'll tell you about later.

Here's another example. And by the way, this notion is called non-stationarity, that the data is changing across time. It's not stationary.

Here's another example. On the x-axis again I'm showing you time. Here each column is a month, from 2005 to 2014. And on the y-axis, for every sort of row of this table, I'm showing you a laboratory test. And here we're not looking at the results of the lab test, we're only looking at what fraction of-- at how many lab tests of that type were performed at this point in time.

And now you might expect that, broadly speaking, the number of glucose tests, the number of white blood cell count tests, the number of neutrophil tests and so on might be pretty constant across time, on average, because you're averaging over lots of people. But indeed what you see here is that, in fact, there is a huge amount of non-stationarity. Which tests are ordered dramatically changes across time. So for example you see this one line over here, where it's all blue, meaning no one is ordering the test, until this point in time, when people start using it. What could that be? Any ideas? Yeah.

AUDIENCE: [INAUDIBLE]

PROFESSOR: So the test was used less, or really, in this case, not used at all. And then suddenly it was used. Why might that happen? In the back.

AUDIENCE: A new test.

PROFESSOR: A new test, right, because technology changes. Suddenly we come up with a new diagnostic test, a new lab test. And we can start using it, where it didn't exist before. So obviously there was no data on it before. What's another reason why it might have suddenly showed up? Yep.

AUDIENCE: It could be like annual check-ups become mandatory, or that it's part of the test admission at hospital. Like, it's an additional test.

PROFESSOR: I'll stick with your first example. Maybe that test becomes mandatory. OK, so maybe there's a clinical guideline that is created at this point in time, right there. And health insurers decide we're going to reimburse for this test at this point in time. And the test might've been really expensive. So no one would have done it beforehand. And now that the health insurance companies are going to pay for it, now people start doing it. So it might have existed beforehand. But if no one would pay for it, no one would use it.

What's another reason why you might see something like this, or maybe even a gap like this? Notice, here in the middle, there's this huge gap in the middle. What might have explained that?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Hold on. Yep, over here.

AUDIENCE: Maybe your patient population is mostly of a certain age, and coverage for something changes once your age crosses a threshold.

PROFESSOR: Yeah, so one explanation-- I think it's not plausible in this data set, but it is plausible for some data sets-- is that maybe your patients at time 0 were all of exactly the same age. So maybe there's some amount of alignment. And suddenly, at this point in time, let's say, women only get, let's say, their annual mammography once they turn a certain age. And so that might be one reason why you would see nothing until one point in time. And maybe that would change across time as well. Maybe they'll stop getting it at some point after menopause. That's not true, but let's say.

So that's one explanation. In this case, it doesn't make sense, because the patient population is very mixed. So you could think about it as being roughly at steady state. So they're not-- you'll have patients of all ages here.

What's another reason? Someone raised their hand over here. Yep.

AUDIENCE: Yeah, I was just going to say, maybe the EMR shut down for awhile, and so they were only doing stuff on paper, and they only were able to record 4 things.

PROFESSOR: Ding ding ding ding ding. Yes, that's right. So maybe the EMR shut down. Or in this case, we had data issues. So this data was acquired somehow. For example, maybe it was required through a contract with something like Webquest or LabCorp. And maybe, during that four-month interval, there was contract negotiation. And so suddenly we couldn't get the

Data for that time period. Or maybe our databases crashed, and we suddenly lost all the data for that time period. This happens, and this happens all the time, and not just the health care industry, but other industries as well.

And as a result of those systemic-type changes, your data is also going to be non-stationary across time. So now we've seen three or four different explanations for why this happens. And the reality is really a mixture of all of these.

And just as in the previous-- so in the previous example, notice how what really changed here is that the derived labels might change meaning across time. Now the significance of the features used in the machine learning models would really change across time. And that's one of the consequences of this, particular if you're driving features from lab test values.

Here's one last example. Again, on the x-axis here, I have time. On the y-axis here, I'm showing the number of times that you observed some diagnosis code of some kind. This cyan line is ICD-9 codes. And this red line are ICD-10 codes. You might remember that Pete mentioned in an earlier lecture that there was a big shift from ICD-9 coding to ICD-10 coding at some point. When was that time? It was precisely this time.

And so if you think about the feature vector that you would derive for your machine learning problem, you would have one feature for all ICD-9 codes, and one-- a whole set of features for all ICD-10 codes. And those ICD-9-based features are going to be-- they're going to be used quite a bit in this time period. And then suddenly they're going to be completely sparse in this time period. And ICD-10 features start to become used. And you could imagine that if you did machine learning using just ICD-9 data, and then you tried to apply your model at this point in time, it's going to do horribly, because it's expecting features that it no longer has access to. And this happens all the time.

And in fact, what I'm describing here is actually a major problem for the whole health care industry. For the next five years, everyone is going to grapple with this problem, because they want to use their historical data for machine learning, but their historical data is very different from their recent data.

So now, in the face of all of this non-stationarity that I just described, did we do anything wrong in the diabetes risk stratification problem that I told you about earlier? Thoughts. That was my paper, by the way. Did I make an error? Thoughts. Don't be afraid. I'm often wrong. I'm just asking specifically about the way I evaluated the models. Yep.

AUDIENCE: This wasn't an error, but one thing, like if I was a doctor I would like to see is the sensitivity to-- like, the inclusion criteria if I remove the HBA1C for instance. Like most people, they have compared to having either Rx or [INAUDIBLE] then kind of evaluating the--

PROFESSOR: So understanding the robustness to changing the data a bit is something that would be of a lot of interest. I agree. But that's not immediately suggested by the non-stationarity results. Not something that's suggested by non-stationarity results.

Our TA in the front row has an idea. Yeah, let's hear it.

AUDIENCE: The train and test distributions were drawn from the same-- or the train and tests were drawn from the same distribution.

PROFESSOR: So in the way that we did our evaluation there, we said, OK, we're going to set it up such that on January 1, 2009, we're predicting what's going to happen in the following three years. And we segmented our patient population into train, validate, and test, but at all times, using that same setup, January 1 2009, as the prediction time.

Now, we learned this model, and it's now 2018. We want to apply this model today. And I computed an area under the ROC curve. I computed positive predictive values using that retrospective data. And I handed those off to my partners. And they might hope that those numbers are reflective of what their models would do today. But because of these issues I just told you about-- for example, that the number of people who have type 2 diabetes, and even the definition of it has changed.

Because of the fact that the laboratory-- ignore this part over here. That's just a fluke. But the fact, because of the laboratory tests that were available during training might be different from the ones that are available now, and because of the fact that we have only ICD-10 data now, and not ICD-9, for all of those reasons, our predictive performance is going to be really horrible now, Particularly because of this last issue of not having ICD-9s. Our predictive model is going to work horribly now if it was trained on data from 2008 or 2009. And so we would have never ever even recognized that if we used the validation set up that we had done there.

So I wrote that paper when I was young and naive.

[AUDIENCE CHUCKLING]

I'm a little bit more gray-haired now. And so in our more recent work-- for example, this is a paper which we're working on right now, done by a master's student of mine, Helen Zhou, and is looking at predicting antibiotic resistance, now we're a little bit smarter about over evaluation setup. And we decided to set it up a little bit differently.

So what I'm showing you now is the way that we chose, trained, validated and test for our population. So we segmented our data. So the x-axis here is time, and the y-axis here are people. So you can think of each person as being a different row. And you can imagine that we randomly sorted the rows.

What we did is we segmented our data into these four quadrants. The first two quadrants, we used for train and validate. Notice, by the way, that we have different people in the training set as we do in the validate set. That's important for another quantity which I'll talk about in a minute. So we used this data for train and validate. And that's, again, very similar to the way we did it in the diabetes paper.

But now, for testing, we use this future data. So we used data from 2014 to 2016. And one can imagine two different quadrants. You might be interested in knowing, for the same patients for whom you made predictions on during training, how would your predictions do for those same people at test time in the future data. And that's assuming that what we're predicting is something that's much more myopic in nature. In this case it was predicting, are they going to be resistant to some antibiotic?

But you can also look at it for a completely different set of patients, for patients who are not used during training at all. And suppose that this 2 bucket isn't used at all, for those patients, how do we do, again, using the future data for that.

And the advantage of this setup is that it can really help you assess non-stationarity. So if your model really took advantage of features that were available in 2007, 2008, 2009, but weren't available in 2014, you would see a big drop in your performance. Looking at the drop in performance from your validate set in this time period, to your test set from that time period, that drop in performance will be uniquely attributed to the non-stationarity. So it's a good way to diagnose it. Yep.

AUDIENCE: Just some clarification on non-stationarity-- is it the fact that certain data is just lost altogether, or is it the fact that it's just encoded differently, and so then it's difficult to get that mapping correct?

PROFESSOR: Both. Both of these happen. So I have a big research program now which is asking not just how-- so this is how you can evaluate and recognize there's a problem. But of course there's a really interesting research question, which is, how can you make use of the non-stationarity. Right, so for example, you had ICD-9/ICD-10 data. You don't want to just throw away the ICD-9 data. Is there a way to use it?

So the naive answer, which is what the community is largely using today, is come up with a mapping. Come up with a manual mapping from ICD-9 to ICD-10 so that you can sort of manually transform your data into this new format such that the models you learn from this older time is useful in the future time. That's the boring and simple answer.

But I think we could do much better. For example, we can learn new representations of the data. We can learn that mapping directly in order to optimize for your sort of most recent performance. And there's a whole bunch more that we can talk about later. Yep.

AUDIENCE: [INAUDIBLE] non-stationary change, this will [INAUDIBLE] does not ensure robustness to the future.

PROFESSOR: Correct. So this allows you to detect that a non-stationarity has happened. And it allows you to say that your model is going to generalize to 2014-2016. But of course, that doesn't mean that your model's going to generalize to 2016-2018.

And so how do you do that? How do you have confidence in that? Well, that's a really interesting research question. We don't have good answers to that today.

From a practical perspective, the best I can offer you today is, build in these checks and balances all the time. So continuously sort of evaluate how you're doing on the most recent data. And if you see big changes, throw a red flag. Build more checks and balances into your deployment process. If you see a bunch of patients who are getting predicted probabilities of 1, and in the past, you'd never predicted probability 1, that might tell you something.

Then much later in the semester, we'll talk about robust machine learning approaches, for example, approaches that have been designed to be robust against adversaries. And those type of approaches as well will allow you to be much more robust to particular types of data set shift, of which non-stationarity is one example. But it's a big, open research field. Yep.

AUDIENCE: So just to make sure I have the understanding correct, theoretically, if you could map everything from the old data set to the new data set, like the encodings, would it still be OK, like the results you get on the future data set?

PROFESSOR: If you could do a perfect mapping, and it's one to one, and the distributions of those things also didn't change, then yeah. Really what you need to assess is, is there data set shift? Is your training distribution, after mapping, the same as your testing distribution? If the answer is yes, you're all good. If you're not, you're in trouble. Yep.

AUDIENCE: What seems to be the test set of traits set here? Or what [INAUDIBLE]?

PROFESSOR: So 1 is using data only from 2007-2013, 3 is using data only from 2014-2016.

AUDIENCE: But in the case, like, the output we care about happened in, like, 2007-2013, then that observation would be not-- it wouldn't be useful.

PROFESSOR: Yeah, so for the diabetes problem, there's also just inclusion/exclusion criteria that you have to deal with. For what I'm showing you here, I'm talking about a setting where you might be making multiple predictions for patients across time. So it's a much more myopic prediction task.

But one could come up with an analogy to this for the diabetes setting. Like, for example, just hold out half of the patients at random. And then for your training set, use data up to 2009, and evaluate on data only up to 2013. And for your test set, pretend as if it was January 1, 2013, and look at performance up to 2017. And so that would be-- you're changing your prediction time to use more recent data.

So the next subtlety is-- it's a name that I put on to it. This isn't a standard name. This is what I'm calling intervention-tainted outcomes. And so the example here came from your reading for today. The reading was this paper on intelligible models for health care predicting pneumonia risk in hospital 30-day admissions from KDD 2015.

So in that paper, they give an example-- it's a very old example-- of trying to use a predictive model to understand a patient's risk of mortality when they come into the hospital. And what they learned-- and they used a rule-based learning algorithm-- and what they discovered was a rule that said if the patient has asthma, then they have low risk of dying. So these are all patients who have pneumonia. So a patient who comes in with pneumonia and asthma has a lower risk of dying than a patient who comes in with pneumonia and does not have a history of asthma. OK, that's what this rule says.

And this paper argued that there's something wrong with that learned model. Any of you remember what that was? Someone who hasn't talked today, please. Yeah, in the back.

AUDIENCE: It was that those with asthma had more aggressive treatment. So that means that they had a higher chance of survival.

PROFESSOR: Patients with asthma had more aggressive treatment. In particular, they might have been admitted to the intensive care unit for more careful vigilance. And as a result, they had better outcomes. Yes, that's exactly right.

So the real story behind this is that risk stratification, as we talked about the last couple weeks, it's used to drive interventions. And those interventions, if they happened in the past data, would change the outcomes.

So in this case, you might imagine using the learned predictive model to say, a new patient comes in, this new patient has asthma, and so we're going to say they're low risk. And if we took a naive action based on that prediction, we might say, OK, let's send them home. They're at low risk of dying. But if we did that, we could be killing people. Because the reason why they were low risk is because they had those interventions in the past.

So here's what's going on in that picture. You have your data, X. And you're trying to make a prediction at some point in time, let's say, emergency department triage. You want to predict some outcome Y, let's say, whether the patient dies at some defined point in the future.

Now, the challenge is that, as stated in the machine learning tasks that you saw there, all you had access to was X and Y, the covariance of the features and the outcome. And so you're predicting Y from X, but you're marginalizing over everything that happens in between, in this case, the treatment. So the good outcomes, people surviving, might have been due to what's going on in between. But what's going on in between is not even observed in the data necessarily.

So how do we address this problem? Well, the first thing I want you to think about is, can we even recognize that this is a problem? And that's where that article really suggests that using an unintelligible model, a model that you can introspect and try to understand a little bit, is actually really important for even recognizing that weird things are happening. And this is a topic which we will talk about in a lecture towards the end of the semester in much more-- Jack will talk about algorithms for interpreting machine learning models.

So that's important. You've got to recognize what's going on. But what do you do about it? So here are some hacks.

Hack number 1-- modify the model. This is the solution that is proposed in the paper you read. They said, OK, if it's a simple rule-based prediction that the learning algorithm outputs to you, you could see the rule that doesn't make sense, you could use your clinical insight to recognize it doesn't make sense. You might even be able to explain why it happened. And then you just remove that rule.

So you manually modify the model to push it towards something that's more sensible. All right, so that's what was suggested. And I think it's nonsense. I don't think that's ever going to work in today's world. In today's world of high-dimensional models, there's always going to be surrogates which are somehow picked up by a learning algorithm that you will not even recognize. And it will be really hard to modify it in the way that you want.

Maybe it's impossible using the simple approach, by the way. Another interesting research question-- how do you actually make this work in a high-dimensional setting?

But for now, let's say we don't know how to do it in a high-dimensional setting. So what are your other choices? Hack number 2 is to redefine the outcome altogether, to change what you're predicting. So for example, if you go back to this picture, and instead of trying to predict

Y, death, if you could try to find some surrogate for the thing you care about, which is pre-treatment, and you predict that thing instead, then you'll be back in business.

And so, for example, in one of the optional readings for-- or actually I think in the second required reading for today's class, it was a paper about risk revocation for sepsis, which is often caused by infection. And what they show in that article is that there are laboratory test results, such as lactate, and there are others, which can give you a hint that this patient might be on a path to clinical deterioration. And that test might precede the interventions to try to take care of that condition. And so if you instead change your outcome to be predicting that surrogate, then you're getting around this problem that I just pointed out.

Now, a third hack is from one of the optional readings from today's lecture, this paper by Suchi Saria and her students, from *Science Translational Medicine* 2015. It's a really well-written paper. I highly recommend reading it. In that paper, they suggest formalizing the problem as one of censoring, which is what we'll be talking about for the very last third of today's lecture.

In particular, what they say is suppose you see that a patient is treated for the condition. Let's say they're treated for sepsis. Then if the patient is treated for that condition, then we don't know what would have happened to them had they not been treated. So we don't observe the outcome, death given no treatment.

And so we're going to treat it as an unknown outcome. And for patients who were not treated, but ended up dying due to sepsis, then they're not censored. And what I'll show you in the later part of the class is how to learn from censored data. So this is another formalization which tries to address this problem that we pointed out.

Now, I call these hacks because, really, I think what we should be doing is formalizing it using the language of causality. Once you do this introspection and you realize that there is treatment, in fact, you should be rethinking about the problem as one of now having three quantities of interest. There's the patient, everything you know about them at triage. That's the X-variable I showed you before. There's the outcome, let's say, Y. And then there's that everything that happened in between, in particular the interventions that happened in between. We'll call that T, for treatment.

And the question that one would like to ask in order to figure out how to optimally care for the patient is one of, will admission to the ICU, which is the intervention that we're considering here, will that lower the likelihood of death for the patient? And now when I say lower, I don't mean correlation, I mean causation. Will it actually lower the patient's risk of dying? I think we need to hit these questions on the head with actually thinking about causality to try to formalize this properly. And if you do that, this will be a solution which will generalize to the high-dimensional settings that we care about in machine learning.

And this will be a topic that we'll talk really in-depth after spring break. But I wanted to give you this as one motivation for why it's so important-- there are many other reasons-- to really think about it from a causal perspective.

OK, so subtlety number 3-- there's been a ton of hype in the media about deep learning and health care. A lot of it is very well warranted. For example, the advances we're seeing in areas ranging from radiology and pathology to interpretation of EKGs are all really being transformed by deep learning algorithms.

But the problems I've been telling you about for the last couple of weeks, of doing risk stratification on electronic health record data, such as taxed notes, such as lab test results and vital signs, diagnosis codes, that's a different story. And in fact, if you look closely at all of the papers, all the papers that have been published in the last few years that have been trying to apply the gauntlet of deep learning algorithms at those problems, in fact, the gains are very small.

And so what I'm showing you here is just one example of such a paper. This is a paper that received a lot of media attention. It's a Google paper called "Scalable and Accurate Deep Learning with Electronic Health Records." And if you go across the United States, if you go internationally, you talk to chief medical information officers, they're all going to be telling you about this paper. They've all read it, they've all heard about it, and they all want to use it.

But what is this actually doing? What's going on behind the scenes? Well, this paper uses the same sorts of data we've been talking about. It takes vitals, notes, orders, medications, thinks about it as a timeline, summarizes it, then uses a recurrent neural network. It also uses attentional architectures. And there's some pretty smart people on this paper-- you know, Greg Corrado, Jeff Dean, are all co-authors of this paper. They know what they're doing.

All right, so they use these algorithms to predict a number of downstream problems-- readmission risk, for example, 30-day readmission, like you read about in your readings for this week. And they see they get pretty good predictions. But if you go to the supplementary material, which is a bit hard to find, but here's the link for all of you, and I'll post it to my slides. And if you look at the very last figure in that supplementary material, you'll see something interesting.

So here are those three different tasks that they studied-- inpatient mortality prediction, 30-day readmission, length-of-stay prediction. The first line each of these buckets is what your deep learning algorithm does. Over here, they have two different hospitals. I think it might have been University of Chicago and Stanford. And they're showing the area under the ROC curve, which we've talked about, performance for each of these tasks for their best models. And in the parentheses, they give confidence intervals-- let's say something like 95% confidence intervals-- for area under the ROC curve.

Now, the second line that you see is called full-feature enhanced baseline. It's using the same data, but it's using something very close to the feature represetnation that you saw in the paper by Narges Razavian, so that paper on diabetes prediction that I told you about and we've been criticizing. So it's using that L1-regularized logistic regression with a smart set of features.

And what you see across all three settings is that the results are not physically significantly different. So let's look at the first one, hospital A, deep learning, 0.95 AUC. This L1-regularized logistic regression, 0.93. 30-day readmission, 0.77, 0.75, 0.86, 0.85. And the confidence intervals are all overlapping.

So what's going on? So I think what you're seeing here, first of all, is a recognition by the machine learning community that-- in this case, a late recognition that simpler approaches tend to work well with this type of data. I don't think this was the first thing that they tried. They tried probably the deep learning algorithms first.

Second, we're all grasping at this, and we all want to come up with these better algorithms, but so far we're not doing that well. And I'll tell you more about that in just a second. But before I finish with the slide, I want to give you a punch line I think is really important. You might come home from this and say, you know what, it's not that much better, but it's a little bit better-- 0.95 to 0.93. Suppose it was tight confidence intervals, there might be a few patients whose lives you could save with that.

But because all the issues I've told you about up until now, of non-stationary, for example, those gains disappear. In many cases, they even reverse when you actually go to deploy these models because of that data set shift for non-stationarity. It so happens that the simpler models tend to generalize better when your data changes on you. And this is nicely explored in this paper from Kenneth Jung and Nigam Shah in *Journal of Biomedical Informatics,* 2015.

So this is something that I want you to think about. Now let's try to answer why. Well, the areas where we've been seeing recurrent neural networks doing really well-- in, for example, speech recognition, natural language processing, are areas where, often-- for example, you're predicting what is the next word in a sequence of words, the previous few words are pretty predictive. Like, what is the next [PAUSES] that I'm going to say? What is it?

AUDIENCE: Word.

PROFESSOR: Word, right, and you knew that, right, because it was pretty obvious to predict that. And so the models that are good at predicting for that type of data, it doesn't mean that they should be good for predicting for a different type of sequential data. Sequential data which, by the way, lives in many different time scales. Patients who are hospitalized, you get tons of data for them at a time, and then you might go months without any data on them. Data with lots of missing data. Data with multivariate observations at each point in time, not just a single word at that point in time.

All right, so it's a different setting. And we shouldn't expect that the same architectures that have been developed for other problems will generalize immediately to these problems.

Now, I do conjecture that there are lots of nonlinear attractions where deep neural networks could be very powerful at predicting for. But I think they're subtle. And I don't think that we have enough data currently to deal with the fact that the data is messy and that the non-linear interactions are subtle. We just can't find them right now. But this shouldn't mean that we're not going to find them a few years from now. I think this deservedly is a very interesting research direction to work on.

And a final reason to point out is that the features that are going into these types of models are actually really cleverly-chosen features. A laboratory test result, like looking at your A1C-- what is A1C? So it's something that had been developed over decades and decades of research, where you recognize that looking at a particular protein is actually informative as something about a patient's health.

So the features that we're using that go into these models were designed-- first, they were designed for humans to look at. And second, they were designed to really help you with decision-making, or largely independent features from other information that you have about a patient. And all of those are reasons, really, I think why we're observing these subtleties.

OK, so for the last 10 minutes of class-- I'm going to have to hold questions, because I want to get through all the material. But please post them to Piazza. For the last 10 minutes of class, I want to change gears a little bit, and talk about survival modeling.

So often we want want to talk about predicting time to some event. So this red dot here-- sorry, this black line here is what I mean by an event. That event might be, for example, a patient dying. It might mean a married couple getting divorced. It might mean the day that what you graduate from MIT. And the red dot here denotes censored events. So for whatever reason, we don't have data on this patient, patient S3, after time step 4. They were censored.

So we do know that the event didn't occur prior to time step 4. But we don't know if and when it's going to occur after time step 4, because we have missing data there. OK, so this is what I mean by right-censored data.

So you might ask, why not just use classification-- like binary classification-- in this setting? And that's exactly what we did earlier. We thought about formalizing the diabetes risk stratification problem as looking to see what happens years 1 to 3 after the time of prediction. That was with a gap of one year.

And there a couple of reasons why that's perhaps not what you really wanted to do. First, you have less data to use during training. Because you've suddenly excluded patients for whom-- or to differently-- if you have patients for whom they were censored during that time window, you're throwing them out. So you have fewer data points there. That was part of our inclusion/exclusion criteria.

Also, when you go to deploy these models, your model might say, yes, this patient is going to develop type 2 diabetes between one and three years from now. But in fact what happened is they develop type 2 diabetes 3.1 years from now. So your model would count this as a negative. Or it would be a false positive. The prediction would be a false positive. But in reality, your model wasn't actually that bad. We did pretty well. We didn't quite get the right range, but they did get diagnosed diabetes right outside that time window.

And so your measures of performance are going to be pessimistic. You might be doing better than you thought. Now, you can try to address these two challenges in many ways. You can imagine a multi-task learning framework where you try to predict what's going to happen one to two years from now, what's going to happen two to three years from now, three to four, and so on. Each of those are different binary classification models. You might try to tie together the parameters of those models via a multi-task learning formulation. And that will get you closer to what you care about.

But what I'll tell you about in the last five minutes is a much more elegant approach to trying to deal with that. And it's akin to regression. So that leads to my second point-- why not just treat this as a regression problem? Predict time to event. You have some continuous valued outcome, the time until diagnosis diabetes. Just try to minimize mean squared-- minimize your squared error trying to predict that continuous value.

Well, the first challenge to think about is, remember where that mean squared error loss function came from. It came from thinking about your data as coming from a Gaussian distribution. And if you do maximum likelihood estimation of this Gaussian distribution, it turns out to look like minimizing a squared loss.

So it's making a lot of assumptions about the outcome. For one, it's making the assumption that outcome could be negative or positive. A Gaussian distribution doesn't have to be positive. But here we know that T is always non-negative. In addition, there might be long tails. We might not know exactly when the patient's going to develop diabetes, but we know it's not going to be now. It's going to be at some point in the far future. And that may also look very non-Gaussian. So typical regression approaches aren't quite what you want.

But there's another really important problem, which is that if you naively remove those censored points-- like, what do you do for the individuals where you never observe the time-- where the never get diabetes, because they were censored? Well, if you just remove those from your learning algorithm, then you're biasing your results.

So for example, if you think about the average age of diabetes onset, if you only look at people who actually were observed to get diabetes, it's going to be much closer to now. Because obviously the people who were censored are people who got it much later from the censoring time. So that's another serious problem.

So the way they we're trying to formalize this mathematically is as follows. Now we should think about having data which has, again, features x, outcome-- what we usually call Y for the outcome in regression, but here I'll call it capital T, because of the time to the event. And now we have an additional variable-- so it's no longer a two-point, now it's a triple-- b. And b is going to be a binary variable, which is saying, was this individual censored-- was the time, t, denoting a censoring event, or was it denoting the actual event happening? So it's distinguishing between the red and the black. So black is b equals 0. Red is b equals 1.

OK, so now we can talk about learning a density, P of t, which I'll also call f of t, which is the probability of death at time t. And associated with any density, of course, is the cumulative density function, which is the integral from 0 to any point of the density. Here we'll actually look at 1 minus the CDF, what's called the survival function. So it's looking at probability of T, actual time of the event, being larger than some quantity, little t. And that's, of course, just the integral of the density from little t to infinity.

All right, so this is the survival function. It's of a lot of interest. You want to know, is the patient going to be diagnosed with diabetes two or more years from now.

So pictorially, what you're interested in is something like this. You want to estimate these conditional distributions. So I call it conditional because you want to condition on the covariant to individual x. So what I'm showing you, this black line, is your density, little f of t. And this white area here, the integral from little t to infinity, meaning all this white area, is capital S of t. It's the probability of surviving longer than time little t.

OK, so the first thing you might do is say, we get these data, these tuples, and we want to try to estimate that function, little f, the probability of death at some time. Or, equivalently, you might want to estimate the survival time, capital S of t, which is the CDF version. And these two are related to another just by some calculus.

So a method called the Kaplan-Meier estimator is a non-parametric method for estimating that survival probability, capital S of t. So this is the probability that an individual lives more than some time period.

So first I'll explain to you this plot, then I'll tell you how to compute it. So the x-axis of this plot is time. The y-axis is this survival property, capital S of t. It's the probability that an individual lives more than this amount of time. I think this x-axis is in days, so 500, 1,000, 1,500, 2,000. This figure, by the way, was created by one of my students who's studying a multiple myeloma data set.

So you could then ask, well, under what covariants do you want to compute this survival? So here, this method I'll tell you about, is very good for when you don't have any features. So all you want to do is estimate that density by itself. And of course you could apply a method for multiple populations. So what I'm showing you here is applying it for two different populations. Suppose there's just a single binary feature. And we're going to apply it to the x equals 0 and to x equals 1. That gets you two different curves out. But here the estimator is going to work independently for each of the two populations.

So what you see here on this red line is for the x equals 0 population. We see that, at time 0, everyone is alive, as you would expect. And at time 1,000, roughly 60% individuals are still alive for time 1,000. And that sort of stays constant.

Now you see that, for the other subgroup, the x equals 1 subgroup, again, time step 0, as you would expect, everyone is alive. But they survive much longer. At time step 1,000, over 75% of them are still alive. And of course of interest here is also confidence balance. I'm not going to tell you how can you do that, but it's in some of the optional readings. And by the way, there are more optional readings given on the bottom of these slides.

And so you see that there is a statistically significant difference between x equals 1 and x equals 0. These people seem to be surviving longer than these people. And you get that immediately from this curve.

So how do we compute that? Well, we take those observed times, those capital Ts, and here I'm going to call them just y. I'm going to sort them. So these are sorted times. And I don't care whether they were censored or not censored.

So y is just all of the times for all of the patients, whether they are censored or not. dK I want you think about as 1. It's the number of events that occurred at that time. So if everyone had a unique time of censoring or death, then dK is always 1. K is indexing one of these things. n of K is the number of individuals alive and uncensored by the K-th time point.

Then what this estimator says is that S of t-- so the estimator at any point in time-- is given to you by the product over K such that y of K is less than or equal to t. So it's going over the observed times up to little t, of 1 minus the ratio of 1 over-- so I'm thinking about dK as 1-- 1 over the number of people who are alive and uncensored by that time. And that has a very intuitive definition.

And one can prove that this estimator gives you a consistent estimator of the number of people who are alive-- sorry, the number of survival probability at any one point in time for censored data. And that's critical. This works for censored data. So I'm past time today. So I'll finish the last few slides on Tuesday's lecture. So that's all for today. Thanks.