w10_1

Tue, Nov 22, 2022 12:46PM 1:02:38

SUMMARY KEYWORDS

model, explanation, counterfactuals, explainer, properties, user, explain, insights, rely, features, people, suitable, case, transparent, prediction, important, deploy, algorithm, decision, predictive model

00:03

computer audio chat okay it will be magical with understanding other people it's not wrong to get an extension late but I think what we should do is we will get you going over to Russia and Isaac but wait one more what the topic is

00:52

don't be shy I will come around and fix that for you in a moment

02:07

you're asking get this set up yeah I didn't know better it's even nearly there interview I will make resume oops that is definitely not where I need to get Hi slash the following me under the tribute is be fun if somebody asks a question repeated into the microphone

03:55

is excited to do it. Okay have you joined the meeting over email normally? Not yet super Solar System Noobest it's been said it has really never once not worked. Well actually I will. Ever oh my god, this mic is not switched on. Um I'll just turn it on when it's q&a time so when you've got a question just yell it out properly or not all that loudly but this mic will pick it up

05:11

that's what they did have you joined the Zoom meeting I'm testing my two minute tune here and what I'm going to do is make your co host okay now if you share your screen perfect okay no Mitch

07:07

yep all right have you people watching online can you hear me speaking into the microphone now okay people online Can you still hear me yelling from over here?

07:29

Sounds good. All right. She's really really fantastic. Now Alright, if I didn't know any better, I'd say me We still got shopping going on. Do we need the other blog?

08:17

Still plenty of chocolate chocolate. Brilliant. Okay, let me know if you run out. Everybody. It is my great pleasure to introduce Casper circle all the way from Poland by hand over to give us our final lecture for this course today. Thank you everyone for being here. And for everyone online. Thank you to so in an absolutely fantastic course. Of course. We'll have a break in the middle today's lecture about five or 10 minutes. And I'll be around for questions afterwards. And Simplecast. All right, Catherine, over to you.

08:47

Right, shall we start them waiting until campus? Get on go

08:52

until around one o'clock, a little quick break or when you're around them, whatever works. All

08:58

right, amazing. Thanks for having me. Thank you all for coming. And thank you to all seven online participants for showing up. This is a very superficial introduction to explain ability in machine learning. Given that we only have two hours. It's split into three parts or less. The first part covers a bunch of general aspects, the some theories then in the second part. I'll show you some examples. And then in the third part, if we have enough time, we'll dive into one very specific explainer. And I'll show you all the ways in which it does not really produce useful explanations, even though they look very convincing. So in the first part, we'll go through the brief history of explainability then I'll try to motivate why we need explainability. I'll give you one example. Then some important developments in the past 10 years, specifically two important developments, then we'll spend quite a lot of time going through various properties of explainers. Then I'll try to define explainability for you. Finally, we'll have a brief look at how to evaluate on a very conceptual level. Then I'll summarize and share with you some useful resources if you want to learn some more. Right, so let's get on with it. explainability pretty much follow the same trajectory, as fairness in machine learning. Where it wasn't really important until one day people realize, oh, well, we probably should start caring more about what these models do, and how to communicate this to humans. This may coincide with the introduction of GDPR. In the European Union, which originally was intended to force, any entity that uses data driven modeling, to provide explanations of their systems to users. But in the end, this passage was abandoned before the legislation came into force. So GDPR actually did not enforce that. But even though there's been a lot of things happening in the past 10 years explainability predates that by nearly 50 years. Yeah, so it all pretty much started with expert systems back in 70s, and 80s, where people handcraft, and engineer all of this data driven systems that were very much aspiring and explainable. However, because they were handmade, they required a lot of effort. And they were designed on case by case basis. So they didn't really scale that well, and they didn't handle noise very well. So then, data driven machine learning came in. And here are just two examples. So on the left, you can see a simple decision tree. And then on the right, you can see a rule list. And this is inherently transparent, you can trace the disease process, from the very top to the very bottom, even if you can't comprehend it, you can still more or less just say, well, the decision is like that, because then you follow the path or you provide the rule. More complex models, data driven models, were there as well, like random forests and ensemble of trees. And while this may not necessarily be explainable anymore, it's still so to speak transparent, because you can look inside that model and see what what it does how it decides this stop being the case, with the rise of deep learning. So then we started having these huge models, you didn't really have to engineer features anymore. They had predictive power, which then and they also handled other tasks very well, which allowed them to spread and then also be deployed in real life. And I suppose that's when people started caring about this sort of aspects. Because once you have these models, in deployment affecting people affecting real life, then if something goes wrong, who do you blame? If you want to make sure that they are robust, you know, what's happening, you want to be able to debug it? And then you also need explainability. So once blackbox modeling became pervasive, then when people started asking questions, right, so what about the explainability? This is when DARPA has introduced their x ai project. So they said, Well, we have this high predictive power, everything is great. We're developing better and better models that provide us with better and better predictive power. But what about their explainability, especially if they are deployed in mission critical settings? So they wanted to develop something that is both that has both high predictive power, but it's also transparent explainable to people who use these models. So why do we need to explain it? We already touched upon that a bit. So first of all, it may offer us trustworthiness so we can avoid silly mistakes with these models. We can also sort of inspect and look into fairness of decisions made by these models, so that they don't discriminate against any groups or individuals. Also, they explainability can aid in scientific discovery. So it can extract new knowledge from these models if they have high predictive power, if they are better at certain costs than humans. If you could extract them Assuming that the leaves arrive at these decisions to achieve these goals, then you could possibly discover new knowledge. And then obviously, there's legislation coming in and you do not want to break any laws. There are also stakeholders. So it's not just engineers, and, and regulators who care about these things. There may be a whole pipeline and a bunch of stakeholders that wants something different from your data driven predictive pipeline. And they may have different expectations of this model. So you may have data scientists who use explainability, to understand the models that they're developing. But also to debug them, or improve their performance. If they pick up that, for example, the model relies on some noise via explainability, then they can fix that potentially very early on, then you obviously have somebody who assess the risk, are we are we ready to deploy it? Will it harm people, also business owners who might look into business case from others, but then regulators who come from the other end and say, Are you harming anybody? Is it? Is the mother reliable? Is it trustworthy? And then at the end, you have a consumer who may want to ask, right, so why did you What How did you arrive with this decision? Right, now, I'll walk you through a very simple example of explainability, just to set the scene for the rest of the talk. And that's basically a linear model. So even though this may seem transparent, and explainable, there are quite a few things that you need to take care of when explaining sectionals. So first of all, how do we explain a linear model like this? Well, you took the code, you take the coefficients, and this provides you with either the influence or the importance of each feature for this model, and also the base prediction. But you need to remember to think so first of all, the modeling assumptions, which in this case, assume feature independence. But also, if you want for these coefficients to be comparable, between themselves, the features have had to be normalized to the same range, right? Because if the features are in different ranges, then this case of these coefficients are uncomfortable. So you can't say that, for example. One of the features is twice as important as the other if the scale is different. So these are the two things that you need to remember in that case, and that you need in order to interpret a simple linear model. Right. So now that we have a an example arriving,

17:53

how can we process it? Well, we just multiply the feature values with the coefficients, and we can calculate the contribution of each feature towards the final prediction. So then you have open to us the base predictions and contribution of each feature positive or negative, towards the final outcome, right, that requires some background knowledge in computer science machine learning, in order to, to interpret. So can we simplify that if if it's not the technical audience that we're trying to explain it to? If this is not our target audience, can we do any better? Well, you can visualize the same example for this particular instance, which obviously removes some complexity, but also discards some information from the explanation itself. So here, you pretty much visualize the contribution of each feature, or the data point. So you can see here that open is the base prediction, and then the contribution of each feature with the final score, that's not ideal, it still requires you to know how to operate with this plot. So then again, you can use a different visualization mechanism. And this is a forest plot. It's extra i, this is usually used with sharp, which is another explainability method, but you can also use it to present other explanations. So here, this the exact same information presented as the first part, which again may be easier for some people to comprehend, which shows you your prediction here, the base value is marked here, and then it shows the magnitude of the contribution of each feature for the data point that we're explaining. So you can see that even with a simple simple linear model, you need to take care of your assumptions. And also, you may want to present it in different ways. depending on your audience, and the use case. Right, so, up until a certain point explainability was pretty much a technical endeavor, it was mostly studied within computer science. And people didn't really care about where it is deployed, how it is deployed, they had some idea of what the explainability is, how to achieve that. And they operated within this remit. And then around 2017, the Miller came around, and he said, Well, if you're explaining something, you're probably explaining it to a human. So you should also consider human when you're doing explainability, in machine learning and artificial intelligence. And there are two main findings, but two main propositions that he put there for human centered explainability, in AI and machine learning. And that's first of all, that humans really like contrast, if statements. And that translated later on in research into counterfactual explanations. So had one of the features has been different in some way, the prediction would be different. This has nothing to do with causality. So that's just how the explanation is presented as a contrast, if statement, and also that you should avoid just showing something to the user without allowing a recourse. So maybe the explanation that you present does not resonate with your audience with the person to whom you show them. So you should possibly enable it within an interactive dialogue or an interactive visualization where the user can try different things. Look around and see whether there's any other explanation that is most more suitable for this particular case. So for example, if we go back to counterfactuals, maybe you do not want the counterfactual that is conditioned on your salary. If it's a loan application scenario, So had you earn 20 grand more a year, lower application to be accepted, maybe you want something else, maybe you don't want to change a job. So then you could possibly interact with the explainer to get an insight that is suitable for you. And that will help you in your particular case. Another important development was around 2019, when Cynthia Rudin said that, we actually should not be using blackbox models and then try to explain them. Instead, we probably should focus on designing and developing inherently transparent models. And this kind of stems from the depiction of the trade off between transparency and predictive power. And this plot first appeared in the DARPA's x ai. Project. So they said, look, there are some models that are transparent. Others predict have really high predictive power. So ideally, we would want something in the top right corner. So something that is both transparent, and predictive. Malbec because of the strain of we may not, not get there. So instead, maybe it's more reasonable to stick with models that have high predictive power like deep neural networks, and then try to explain them. And what Cynthia argued is that this is not based on any real data or any real insight. So this trade off may not actually exist in the real life. And maybe you are able to actually develop models that are both transparent and have high predictive power.

24:20

But that requires effort, and I'll get into that in a moment. But if you first develop a model that performs really well, then you've achieved your goal, right? And if you have a method that then you can plug on top of that blackbox and explain it somehow, then that solves both of your problems, and you don't have to trade anything off. And that's where post hoc explainers come in because this methods basically, touch on top of your model, whatever it may be, and provide you with some insights into how it functions and even better if these methods are model agnostic, which means If you don't really care what your model is, you just throw it there. And it works. Well it kind of works. It gives you something that looks like an explanation. But of course, there are no universal things. So I really like to compare that. That's scenario to the no free lunch theorem, which base is more or less states that no single solution can outperform all the others across the board. And so this is the case with poster explainers. So the explainers that you would touch on top of a pre existing model. And that's because what they usually do is that they try to approximate the blackbox decision boundary either locally or globally, in some manner. But if they don't catch the decision boundary quite that well, then you're not really explaining it, you're just making stuff up. So she argued that, instead of doing this, we probably should, first of all, come up with an explainability process that guides you through the development of a system that is transparent and explainable. And she compared that to similar processes for data mining, or in general data driven modeling. And then once you have this process, you probably should focus on actually putting the effort on putting the effort and engineering your features and designing your models specifically, for the use case for the problem that you're trying to solve, rather than trying to come up with a general solution that would address all of the problems but maybe not as good as a targeted solution. Unfortunately, while it's very still very early stages development in terms of explainability. So creating such a process is not possible just yet. But there were some attempts. So one attempt is ours, where we tried to design a taxonomy of explainability in AI. And we actually spent quite a lot of time discussing different properties that covers, but that's very abstract. And then there are also some more technical frameworks, but then again, fairly limited in scope. Right, so let's have a look at some of the properties that you may want to have when when explaining or when designing an explainer.

27:42

27:42

Even that we do really care about the person or the audience to whom we are explaining such a taxonomy has to cover both the technical aspects of your explainers, the systems, the algorithms, as well as how you communicate this information, who's your target audience, and towards the end, you're explaining? So we, we did go through a bunch of literature back in 2020 or so. And we collected that and we tried to split it into some reasonable categories. So we came up with five, so functional, which takes care of the algorithmic requirements of your explainers. Usability, which focuses on the user, what properties the user expects, and how can you make the users life easier operational, which then are concerned with what happens when you try to deploy the system when you try to provide these explanations to somebody? Safety? Are there any issues, when you add explanations on top of predictions? Does that compromise your model in any way or the whole the whole pipeline and then finally, how the explainability system was validated, verified whether it is suitable for the task.

29:25

This were designed with three different groups of people in mind. So first of all, researchers people who create explainability algorithms, and then they can operationalize this taxonomy as worksheets. So you go through these properties as you design or develop your system. And these are supposed to prompt you to think about different aspects. Have I considered this is it important? Is the explainer suitable here? Are there are Are there any issues? Practitioners so users of explainability all grips on? Are such as engineers and data scientists, and they may use this taxonomy as fact sheets. So you, you want to deploy an explainability algorithm. And then you need to assess whether it's suitable for your particular use case. And then possibly compare and contrast one method against the others in order to pick the best one. Finally, there are also evaluators. So people who come in to evaluate your system. And they may use explainability, as a way to peek inside your model your data driven system, and assess its compliance or try to certify it. And they then may use it as a checklist. So they come with a bunch of properties that they want to check whether your system has, and they can go through all this data and see, oh, that complies, that fails, maybe need some work here or there. Because explainability algorithms, the implementations may be different than the algorithmic design or the theory that these are built upon, you probably should consider all three with your factsheets. So are the taxonomy. Because, for example, you may come up with a really nice property when you're developing the algorithm. But then there is an assumption there, like for example, the block box has to be a linear model. And then you have all of these nice properties. So even though you may report these properties, with respect to the the theory that you're you're proposing, it may not carry through to the design or the implementation, so you may lose some of the properties. Therefore, you should probably cover both three aspects. So that people know and then obviously, there may be alternative implementations of the same method that make different traders. Us as I'll go through some of these properties, I want you to have this example in mind. So this is a very simple justice stress and non causal counterfactual. So it says, as you've been 10 years younger, your loved one will be accepted. So that's in form of text, but you can also have contrastive statements as images. So here, if you remove this part of the image, the prediction changes from tennis ball to Golden Retriever. And that will refer to the to this example, in general to counterfactuals as I go through the properties, right, so this is a summary of functional requirements. See that, that some how somewhat corresponds to properties that you'd expect to have a machine learning algorithm more or less. So for example, an explainer may be designed for a particular supervision level of your data driven model. So it may be only suitable for unsupervised models semi supervised more than reinforcement learning supervised models. It also may deal with different problem types. And at the end of this long list of properties, I'll show you a few examples where this is actually important. So then it may only be suitable for classification, whether probabilistic binary multivariable regression or clustering, then there is applicable model class. So whether your explained there is agnostic of the predictive model that you're trying to explain whether it's only suitable for a certain class of models. So for example, it may only work with differentiable models, or it can be model specific. So then it may rely on some parameters being extracted from the model, and then somehow transform that into an explanation. Then there's also relation to the predictive system. So you can either have an explanation that is based directly on the model. So or the model may be transparent or interpretable itself, like the example of a decision tree or a rule list that you've seen before, or it may be post hoc. So that's the one that you attach on top of a pre existing model. And it doesn't really care what the model is, it operates around the model. You may also care about computational complexity. So whether it's possible to use the algorithm in real time or whether it needs a lot of computation in the background, to produce an explanation, depending on which may be important depending on In your deployment use case, if you want to deploy it in real life system, real time system, then that's important. And then what sort of features it's compatible with? If you require all of your features to be numerical? Or does the world work with categorical features, maybe it allows you to one hot encoder features and get around the delimitation of categorical features. This all needs to be clarified. And then obviously, any assumptions that the explainer makes, like, for example, I just mentioned linearity of your black box, or of your underlying model that you're trying to explain. Ah, then what is the target of your explanation? Are we trying to explain the predictions like it is with counterfactuals. Or maybe you're trying to explain the entire model. Or maybe you're just dealing with the data and you don't really, you're not at that stage. So you're trying to extract some insights from raw data, then your explanation can be local. That is with respect to a single instance, a single data point and a single prediction, a collection of instances or a subspace of your model. Or you may want to explain summarize the entire model, if that's helpful, right, for usability requirements. This deals with how the explanations are generated and how they correspond to the model. So for example, soundness reports on how truthful the explanation is with respect to that model, believe it or not, but you may generate explanations that are actually incorrect with respect to the underlying predictive model, for example, just to give you one, if the model is constantly revised, and you choose a counterfactual that is close to the decision boundary, if the decision boundary shifts, your explanation is not valid anymore, it may be the case. But counterfactuals usually are at least at the time of generation, because they rely explicitly on finding the decision boundary. And going just across it. Complete, this tells you how well your explanation generalizes. So whether you the explanation that you've got, can be applied to any other similar instance. And in terms of counterfactuals, it can because it is generated very specifically for that instance. So you can't transfers the the insights that you get to some other examples. That's when context fulness comes into play. So you may actually provide a context for explanation, which then allows provides people with limits or with some guidelines on when the explanations work, how they can be generalized to other cases, and when it's safe to use them. Then obviously, when you generate explanations, you want them to be short, because you don't want to overwhelm your user. And again, counterfactuals achieve that, because you usually rely on a very small number of changes to the feature space, that result in a different prediction.

38:21

People tend to prefer insights, or explanations that take into account time. So if you have multiple events, that all can be provided as an explanation, you probably want to rely on the most recent one, just because users prefer that. He also don't want to break any natural laws, or you don't want to say something that is outright incorrect. Whether it's the user's mental model or natural laws here, you need to be a bit careful, because if the user is wrong, if the user has wrong beliefs, then you obviously want to correct face. And then this then the next one novelty, that that is somehow related to brevity. So you don't really want to state the obvious, you don't want to tell the users what they already know, because you're wasting their time. And if you don't show all the statements, then you can save the space and you can put there are more important insights, which allows you to generate shorter explanations. And then you want to tune the complexity for the audience that you're dealing with. So for example, if you're dealing with patients, you will provide different insights than if you're dealing with medical professionals, nurses, etc. Each audience has some preference and has some background knowledge that allows them to comprehend the explanation to a certain extent. So we want to take that it into consideration, as well as the operational context as a whole. Then explanations should be actionable, such that, for example, in the loan application, if the user actually wants to reapply for the loan, because she was so she was denied it, then the insights should allow that person to take an action that leads to a different outcome. So for example, the the counterfactual that I showed you was conditioned on age. So that's not optional, you can change your age, but say if it were conditioned on salary, then possibly you can get a different job. And that provides you certain a path towards achieving the goal. Again, ideally, you'd want to enable this whole explanatory process is a dialogue between the user and the explainers such that the user can guide the the product, the explanatory process, and arrive at some insights are useful for the person. And that, again, enables personalization. So you could possibly through interaction, generate insights that are suitable for you and for a particular scenario within which you are placed. Right, let's move on to the operational requirements. So this describe the explanation. So for example, you may have different types of explanations. And these are associations between certain features or properties of your instances of your data of your model. And how this affects the modeling process. You might have contrast in differences. That's, for example, counterfactuals, it tells you two contrasting cases that lead to different outcomes. Or then you may rely on causal models and causal mechanisms to generate your explanations. You may then convey this this insight, either a statistical numerical summarization, so a bunch of numbers, maybe if you're dealing with an engineer, that's what this person expects, because it allows for fine tuning the model, maybe you want to visualize it, because it better communicates the message for the US text of paragraph. Or maybe you want to actually put it within the formal argumentation process. You may also want to combine a bunch of these. So for example, you might provide a figure with a caption, if that allows you to better communicate the message. Then again, the system can either be static, so you just put the expression out. And that's it. Or it can be interactive in some ways, which I'll also discuss a bit more in the moment. Then, how do you generate this and what does it rely on. So you can either take the original data domain or the original parameters from your to generate the explanation, or you can transform it somehow, if it's not intelligible, and then present it as such. Then in the your particular use case, you may require the data to be transparent. So for example, if you're dealing with tabular data

43:38

you may require for the features to be intelligible to the user. So for example, if you have some test results in direct this may be intelligible for doctors. And that's fine. And this may be a requirement for your explainer, because it relies directly on these features. Or maybe you transform it in a way such that it doesn't really matter. So for example, you can partition the numerical range into categories that are intelligible. Chelsea, Olga, in this case. And then again, who's your target audience? Are we dealing with domain experts? Is it the lay audience? Who do you want this extension to be meaningful to? Then what's the function of your explainability algorithm isn't just to get insights into a black box, or maybe you want to inspect fairness of predictions of your model. And you can for example, achieve that again, in a very crude way with counterfactuals because if you condition, a counterfactual explanation on race or gender, if you can find such controversial it means that the model is not fair. And then also, you may want to try and test the accountability robot Asness of your model. And again, you may you may somewhat see the similarities between counterfactual examples and adversarial examples where the difference, I suppose, is that with counterfactual examples, that difference between the fact and the foil is meaningful to humans. Whereas in adversarial examples, there's not that difference not that meaningful. So there's something that makes you surprised why, why is the model returning a different prediction for that change to this change is not really meaningful. And then you want to look at the causality and action ability of your insight. So again, counterfactual that I showed you, is not causal, but it looks causal. And you can't confuse the two. So you need to clearly state whether it just looks actionable, or whether it's actually causal. And then how it relates to predictive performance of your model. If you deploy this particular explainability method, does it hurt your model in terms of predictive performance? Or it doesn't affect it at all? Do you have to make any trade offs there? And then one final thing for operational properties is, where does the exponential actually come from? Do you rely on the behavior of your predictive model? So again, that's the case with counterfactuals. They tried to just find where the decision boundary is, and cross it. Or maybe it relies on the dataset training dataset, there are some additional datasets that you use to train your explainability algorithms. For example, you might want to build a sample of instances to explain certain region, and it relies on that. Or maybe you take this instances from the original training data. This is important because if you sample the instances may be out of distribution, so that again, may skew your explanation. Or maybe it relies on both the predictive model and the data set, or maybe some parameters or other properties. So again, you want to be clear where the explanation comes from. Safety requirements, that's just four of the software. First of all, if you're, if you're explaining, you're obviously providing additional information. And this may not really be what you want, especially when your model is proprietary, you don't want to reveal any trade secrets. So you really need to consider what sort of information you're providing to the users and how it affects the entire pipeline. So for example, counterfactuals give you very precise values for changing the decision boundary. And now if you think of an example, for, let's say, a decision tree, which uses axes parallel splits in the feature space, a counterfactual will probably be based on one of the splits, which means if you provide a counterfactual, you also provide a precise threshold that the decision tree is using. And then if you can gather enough of these counterfactuals, or for example, if the system is interactive, and you can query it, you can probably structure your query such that you are able to recover the entire model. So for example, you can reverse engineer either a part of the student boundary or the whole model. Again, depending on how you implement your explainer, it may rely on some stochastic procedure. So within the exact same scenario, if you run the if you extract the explanation twice, you may get different answers, ideally, not too far apart, or giving you the same sort of answered, but it may be the case as well, that you'll get opposing answers opposing explanations, which is not very good for the trust of your explanations. And then, again, what's the quality check like a general properties if you use counterfactuals, and you pick upon this just across the decision boundary, it may not come from a dense region of your data's, which means that this instance may be impossible in the real world. And also, you may want to consider the confidence of your model when dealing with explainability. So if you're explaining a point that is on the boundary, that's not very helpful. If you're explaining your point that your mother is very sure about, then the explanation will be worth a bit more will be a bit more informative. For validation. There are usually two approaches. One is you go to the users and you do a bunch of user studies and try to See whether the explanations are helpful whether they they help users solve a certain problem. You see what what issues arise when users are served the assumptions whether they misuse them, or generalize them to some other examples that are incorrect. Or you may run synthetic experiments. So you come up with a benchmark, you come up with a metric, and then you just compute that. Right? So let's go through a few examples. And I'll consider a researchers point of view, then a an engineers, and finally, somebody who assesses explainability algorithms. So here, let's say that your explainability algorithm can handle can explain numerical outputs. So now you should clarify whether you can apply that to both regression models, or probabilistic classifiers as well, because these are numbers derived. So if you get the numeric calls, but you can still deploy it in that case, but it may not necessarily work in the same way, or it may provide you with incorrect insights. So clarifying that is important. Then you also may only work with numerical features. And obviously, if you have a model that works with numerical features, you can learn how to encode them, or pre processed them in other ways. Does that compromise the integrity of your explainer? Will the insights that you get still be helpful, truthful, reliable? If your explainer is model agnostic, so you just plug it in with any model, regardless of what it is, does it actually work equally well, for all of the models, or maybe make certain assumptions that make it more appropriate for certain types of models that rely on some properties? Or have for example, access, parallel splits, etc? And then again, if you have some nice theoretical properties that you claim your explainer has, does that hold? First of all for the implementation? Like I said, but also, would it necessitate a particular blackbox model that you're explaining? Or does the property hold whatever the model is? Now let's look at an engineers perspective. So say that you have a music streaming service that provides some recommendations. And then you want to explain that. So first of all, you may want something that relies on listening habits of your users, and their interactions with the system. So you need to clarify what what was what are the properties that you're interested in? When explaining? Then if to deploy it? The if you want to deploy the explainer in real life? Will it work? Or does it require a lot of computation? So you can't do that, that allows you to assess whether a certain method is suitable.

53:30

Then who's your audience? If you're targeting the explanation towards the music listeners, then you should not really assume any background knowledge in machine learning, or computer science? And then what is the domain within which you're explaining so then maybe you want to rely on general music concepts that people can relate to? So you've been looking for a method that satisfies these properties? And how do you want to deliver the explanation? Do you want the pop up giving a piece a snippet of text saying, we recommended you this song? Because how long do you want the extension to be if it's just a pop up that shows up for say, five seconds, you do not really want to overload the user, maybe you want one important property that prompted this particular recommendation, and then a hyperlink to a more elaborate explanation. And then you just want one way communications, or you want the user to be able to interact, like for example, oh, maybe if you say edit your playlist, and then you can experiment with the playlist. This is how the recommendations will change. And then finally, an auditor may come and try to see whether the explanations are sound and complete. So whether they agree with a predictive model, and whether any particular explanation is also coherent with explanations for other instances, and whether there's any context provided it and that complex is useful. Then the stability of your explanations. If you don't change your playlist or don't change anything, are they stable? Will we will you still get the same explanations tomorrow? Are there any random effects that may affect the quality or the content of the explanations? And then did they leak any sensitive information. So first of all, counterfactuals already leak information, if you want them to be placed in a context, that context also leaks information, which then possibly allows you so for example, if you're, if you pick up that your model relies on round the numbers, like the threshold is that 25 grand, and you're earning 24, or 500, maybe next time we'll push you across the threshold. And then in terms of validation, if you're taking a method that was developed and tested for a music recommendation setting, and you're suddenly transferring it to say, a medical domain, it may not yield the same results, it may not be suitable. So you should also be very careful to check how it was validated. And whether the validation procedure that is reported corresponds to the setting in which you want to deploy the explainer. And we have summarized these properties for line, which is one of the very popular post to explainability methods, you can look it up, it's an appendix to the paper. But that obviously does not mean that the properties cover every single aspect of your explainer. So, you may still probe as we go, and we explore more of explainability there may be properties that you also need to consider so that the list that I showed you is neither exhaustive nor prescriptive. Also, some properties may be incompatible. So if you want to optimize for one, you may need to trade off the other. And again, this needs to be decided on a case by case basis. So there are no universal guidelines guidelines, and oh, you need to do that. And that's the only way to go. No, it depends very much on your context. And again, some of the properties are very vague, as you've seen, so they can't be answered uniquely. This, these are not defined mathematically. And then they require a discussion, you need to clearly state your assumptions, what's happening and why. And finally, even though you can list all these properties, and go through them, they do not really define explainability. Right? So I'll just go through one more section, and then we'll break. So how do we define explainability? Now that you've seen all these properties and things to consider, can we actually define it? Well, there isn't a universally accepted definition, at least not the some of the interesting ones are, for example, civil liability. And this is the user's ability to mimic that the specific process. So for example, if you have a model that predicts, and then you're able to replicate where this decision comes from, so for example, if you're given a decision tree, and you can follow the path and the decision trees, that that would deem the model explainable in terms of similar probability. But there's a very interesting counter argument to that, which is called the Chinese room theorem. So basically, this is a concept where you look up a person in the room, and this person, neither is neither speaks Chinese, or English, but then you provide that person with all the resources necessary to translate one language into the other. And then you provide a person who even though the English person takes however long to translate that and gives you back translation, even though the task has been completed. This does not mean that the person either understands English, Chinese, or the content of the note. So being able to complete a task is not the same as understanding what's happening, what's the task, and why you're completing it. An alternative definition is based on mental models. And two important mental models are functional and structural. Functional allows you to operate a concept without really understanding it in depth. So for example, if you want to turn the light on, you need to know that you need to flip the switch and that's it. You don't really need to know all the underlying physics or what's the mechanism governing this process. You just want to know what to do and what's the outcome, but that this knowledge may not necessarily generalize well to other scenarios, where a structural mental model provides you with in depth understanding that you can possibly apply to other related scenarios. So for example, if you understand the the electrical circuits, or the physics that governs a light switch, then you can take this knowledge and apply it to a different circuit or a different scenario. So, it allows you to generalize better, but either way, it may require you to know more. Given all of that, how would I define explainability, where it's a reasoning that is either algorithmic or carried out by human applied to transparent insights. So, these are the nuggets of information that you extract from your data driven process. And then you interpret this in a given context and with a pilot with a particular background knowledge. And all of these should lead to understanding why the reasoning can be algorithmic or are carried out by a person? Well, if you refer back to the example that I gave with the linear model, there, I did a lot of reasoning myself, right, because I explained oh, I need to plug this in, I need to multiply it and that's how I interpret it, but then again, you can do this partly algorithmically and then generate this figure. So, the reasoning can be on both sides. And however, you achieve that, it is important that the explaining walks away with understanding understanding that that is helpful to solve the problem for which the explainability was designed. Another important aspect is that none of these concepts are binary. So something isn't either transparent or opaque explainable or not explainable. Usually to choose a point on the spectrum that is against suitable for audience like I said with the medical example. certain pieces of information may be understandable and transparent to medical professionals, but not necessarily to patients or their families. So you need to choose what's suitable for your particular use case. Right? Single break here. We can resume five past I suppose.

1:02:33

Brilliant, amazing.