Explainability

Jordi Vitrià

Algorithmic decision-making

L'OBS

Derrière l'algorithme de Parcoursup, un choix idéologique

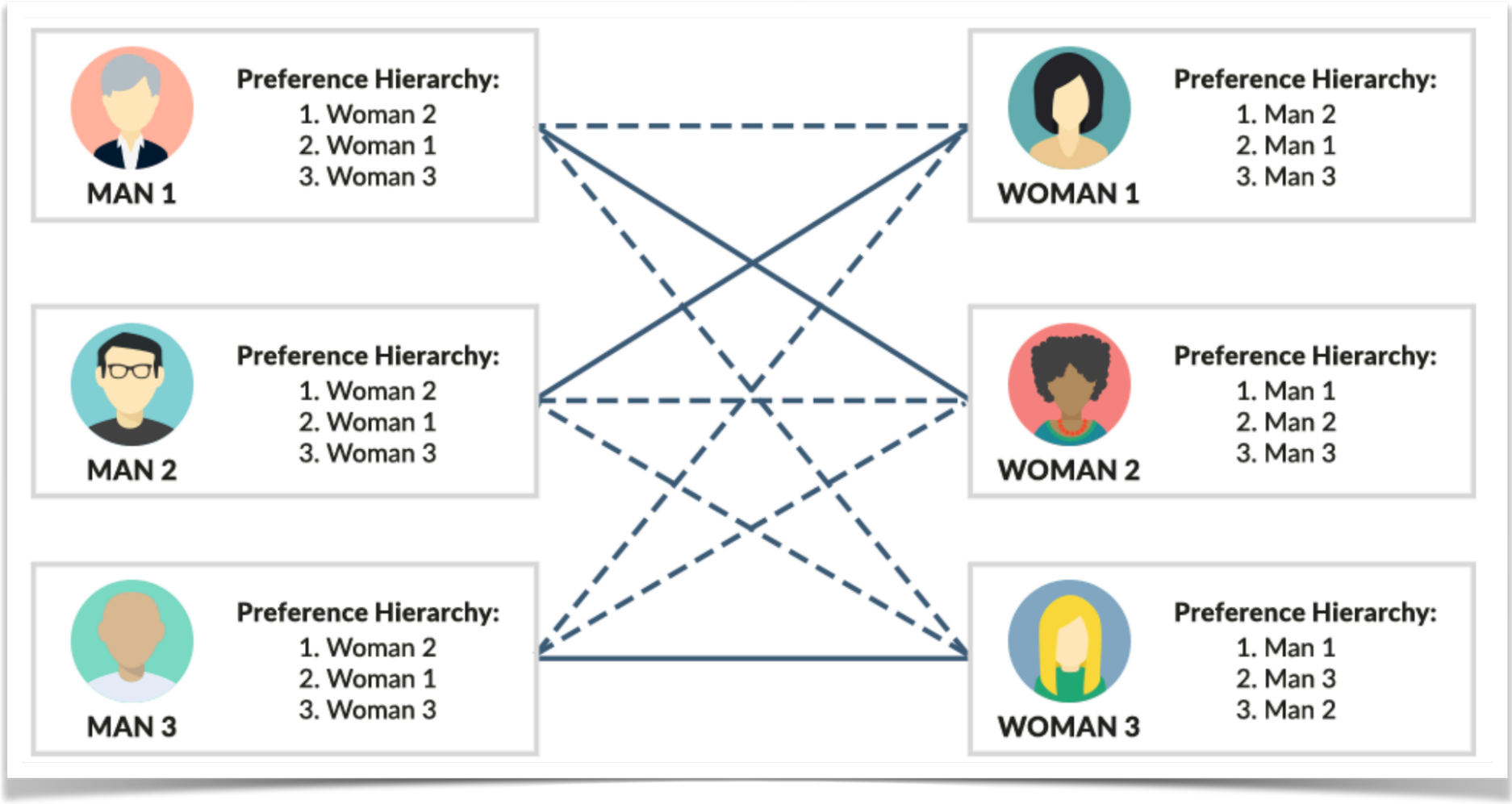
La répartition des étudiants entre les universités et les filières est un problème complexe puisqu'elle s'effectue sur base d'un conflit massif entre l'offre et la demande : on dénombre plus de 880.000 candidats pour un total (à raison de 10 vœux possibles par candidat) de quelques 7.000.000 de vœux de formation *[810.000 ont in fine validé leurs voeux, NDLR]*. La résolution d'un tel conflit n'est plus sérieusement envisageable humainement. Dès lors qu'un algorithme travaille à cette mise en relation n'est pas à remettre en question. La vraie question est celle de l'objectif assigné à l'algorithme et des choix qu'il doit exécuter.

Cette décision politique et idéologique se lit dans la formule algorithmique même de Parcoursup. Cet algorithme, dont l'objectif est de mettre en relation deux objets, d'un côté des établissements, de l'autre des étudiants, est en effet inspiré par le célèbre **algorithme de Gale et Shapley**, repris par Alvin Roth, prix Nobel d'économie en 2012. Il relève au fond d'un vieux problème économique que l'on appelle l'appariement stable.

https://www.nouvelobs.com/education/20180713.OBS9643/derriere-l-algorithme-de-parcoursup-un-choix-ideologique.html

Algorithmic decisions are not new.

Algorithmic decision-making



The **stable marriage problem** has been stated as follows:

> Given *n* men and *n* women, where each person has ranked all members of the opposite sex in order of preference, **marry the men and women together such that there are no two people of opposite sex who would both rather have each other than their current partners**. When there are no such pairs of people, the set of marriages is deemed **stable**.

Algorithmic decision-making

WIRED BACKCHANNEL BUSINESS CULTURE GEAR IDEAS SCIENCE SECURITY SIGN IN SUBSCRIBE

AMIT KATWALA, WIRED UK BUSINESS 08.15.2020 10:00 AM

An Algorithm Determined UK Students' Grades. Chaos Ensued

This year's A-Levels, the high-stakes exams taken in high school, were canceled due to the pandemic. The alternative only exacerbated existing inequities.

PHOTOGRAPHY: TOLGA AKMEN/AFP/GETTY IMAGES

Algorithmic decision-making

ML reverses one important aspect:

In the case of Parcoursup, we first defined the CRITERION (EXPLANATION) and then we designed the ALGORITHM (unique solution).

Algorithmic decision-making

ML reverses one important aspect:

In the case of Parcoursup, we first defined the CRITERION (EXPLANATION) and then we designed t ALGORITHM (uniq solution).

In the case of MACHINE LEARNING, we select an ALGORITHM that learns from data and then we ask for an EXPLANATION.

Algorithmic decision-making

Transparency

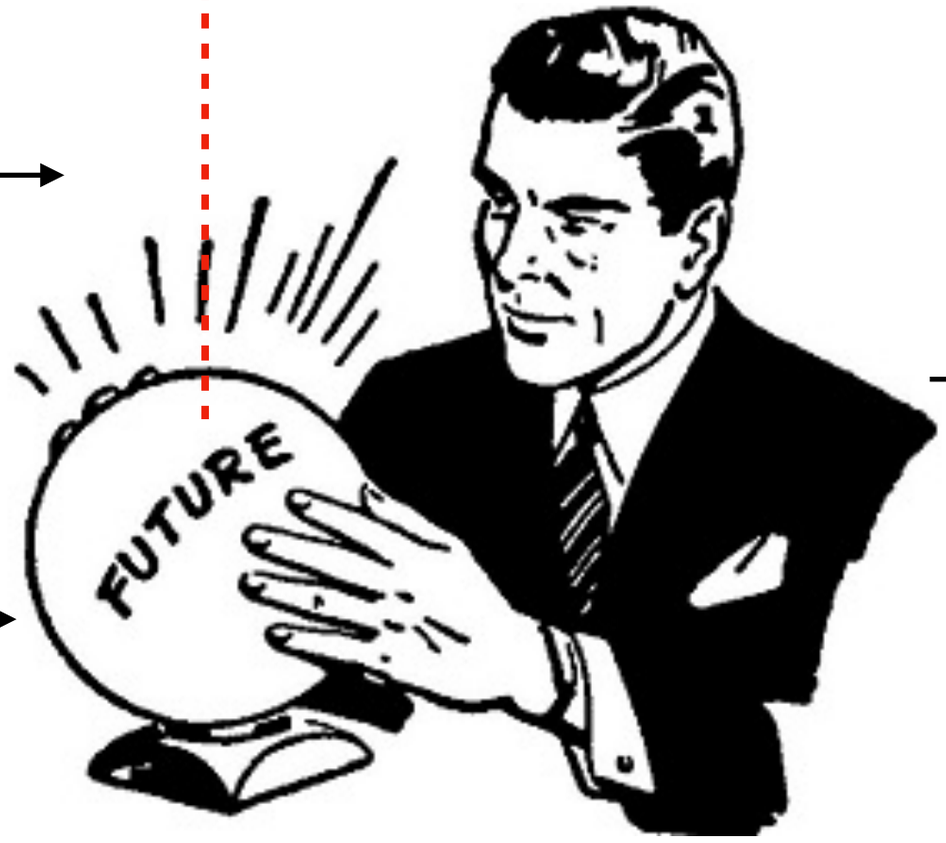
What are the factors/values that influence the decisions made by algorithms?

They should be visible, or transparent, to the people who use, regulate, and are impacted by systems that employ those algorithms

Can we understand the reasoning behind each decision?
Can we assure that all relevant knowledge is reflected in the model?

What is the certainty behind decisions?

Training Data →



New Data →

→ **Decision**

↕

Human

The need for explainability

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu

ABSTRACT

In machine learning often a tradeoff must be made between accuracy and intelligibility. More accurate models such as boosted trees, random forests, and neural nets usually are not intelligible, but more intelligible models such as logistic regression, naive-Bayes, and single decision trees often have significantly worse accuracy. This tradeoff sometimes limits the accuracy of models that can be applied in mission-critical applications such as healthcare where being able to understand, validate, edit, and trust a learned model is important. We present two case studies where high-performance generalized additive models with pairwise interactions (GA^2Ms) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy. In the pneumonia risk prediction case study, the intelligible model uncovers surprising patterns in the data that previously had prevented complex learned models from being fielded in this domain, but because it is intelligible and modular allows these patterns to be recognized and removed. In the 30-day hospital readmission case study, we show that the same methods scale to large datasets containing hundreds of thousands of patients and thousands of attributes while remaining intelligible and providing accuracy comparable to the best (unintelligible) machine learning methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning—*Induction*

Keywords

intelligibility; classification; interaction detection; additive models; logistic regression; healthcare; risk prediction

1. MOTIVATION

In the mid 90's, a large multi-institutional project was funded by Cost-Effective HealthCare (CEHC) to evaluate the application of machine learning to important problems in healthcare such as predicting pneumonia risk. In the study, the goal was to predict the probability of death (POD) for patients with pneumonia so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients. In the study [3, 2], the most accurate models that could be trained were multitask neural nets[1]. On one dataset the neural nets outperformed traditional methods such as logistic regression by wide margin (the neural net had AUC=0.86 compared to 0.77 for logistic regression), and on the other dataset used in this paper outperformed logistic regression by about 0.02 (see Table 2). Although the neural nets were the most accurate models, after careful consideration they were considered too risky for use on real patients and logistic resgression was used instead. Why?

One of the methods being evaluated was rule-based learning [1]. Although models based on rules were not as accurate as the neural net models, they were *intelligible*, i.e., interpretable by humans. On one of the pneumonia datasets, the rule-based system learned the rule "HasAsthama(x) ⇒ LowerRisk(x)", i.e., that patients with pneumonia who have a history of asthma have lower risk of dying from pneumonia than the general population. Needless to say, this rule is counterintuitive. But it reflected a true pattern in the training data: patients with a history of asthma who presented with pneumonia usually were admitted not only to the hospital but directly to the ICU (Intensive Care Unit). The good news is that the aggressive care received by asthmatic pneumonia patients was so effective that it lowered their risk of dying from pneumonia compared to the general population. The bad news is that because the prognosis for these patients is better than average, models trained on the data incorrectly learn that asthma lowers risk, when in fact asthmatics have much higher risk (if not hospitalized).

One of the goals of the study was to perform a clinical trial to determine if machine learning could be used to predict risk prior to hospitalization so that a more informed decision about hospitalization could to be made. The ultimate goal was to reduce healthcare cost by reducing hospital admissions, while maintaining (or even improving) outcomes by more accurately identifying patients that need hospitalization. As the most accurate models, neural nets were a strong candidate for clinical trial. Deploying neural net models that could not be understood, however, was deemed too risky —

[1]SVMs and boosted trees were not in common use yet, and Random Forests had not yet been invented.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788613 .

A classical example is a study carried out in the nineties using rule-based learning and neural networks to decide **which pneumonia cases should be admitted to hospital or treated at home**.

Two ML models (rules, NN) were trained on patients' recovery in historical cases (from a hospital).

The need for explainability

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu

ABSTRACT

In machine learning often a tradeoff must be made between accuracy and intelligibility. More accurate models such as boosted trees, random forests, and neural nets usually are not intelligible, but more intelligible models such as logistic regression, naive-Bayes, and single decision trees often have significantly worse accuracy. This tradeoff sometimes limits the accuracy of models that can be applied in mission-critical applications such as healthcare where being able to understand, validate, edit, and trust a learned model is important. We present two case studies where high-performance generalized additive models with pairwise interactions (GA^2Ms) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy. In the pneumonia risk prediction case study, the intelligible model uncovers surprising patterns in the data that previously had prevented complex learned models from being fielded in this domain, but because it is intelligible and modular allows these patterns to be recognized and removed. In the 30-day hospital readmission case study, we show that the same methods scale to large datasets containing hundreds of thousands of patients and thousands of attributes while remaining intelligible and providing accuracy comparable to the best (unintelligible) machine learning methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning—*Induction*

Keywords

intelligibility; classification; interaction detection; additive models; logistic regression; healthcare; risk prediction

1. MOTIVATION

In the mid 90's, a large multi-institutional project was funded by Cost-Effective HealthCare (CEHC) to evaluate the application of machine learning to important problems in healthcare such as predicting pneumonia risk. In the study, the goal was to predict the probability of death (POD) for patients with pneumonia so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients. In the study [3, 2], the most accurate models that could be trained were multitask neural nets[1]. On one dataset the neural nets outperformed traditional methods such as logistic regression by wide margin (the neural net had AUC=0.86 compared to 0.77 for logistic regression), and on the other dataset used in this paper outperformed logistic regression by about 0.02 (see Table 2). Although the neural nets were the most accurate models, after careful consideration they were considered too risky for use on real patients and logistic resgression was used instead. Why?

One of the methods being evaluated was rule-based learning [1]. Although models based on rules were not as accurate as the neural net models, they were *intelligible*, i.e., interpretable by humans. On one of the pneumonia datasets, the rule-based system learned the rule "HasAsthama(x) ⇒ LowerRisk(x)", i.e., that patients with pneumonia who have a history of asthma have lower risk of dying from pneumonia than the general population. Needless to say, this rule is counterintuitive. But it reflected a true pattern in the training data: patients with a history of asthma who presented with pneumonia usually were admitted not only to the hospital but directly to the ICU (Intensive Care Unit). The good news is that the aggressive care received by asthmatic pneumonia patients was so effective that it lowered their risk of dying from pneumonia compared to the general population. The bad news is that because the prognosis for these patients is better than average, models trained on the data incorrectly learn that asthma lowers risk, when in fact asthmatics have much higher risk (if not hospitalized).

One of the goals of the study was to perform a clinical trial to determine if machine learning could be used to predict risk prior to hospitalization so that a more informed decision about hospitalization could to be made. The ultimate goal was to reduce healthcare cost by reducing hospital admissions, while maintaining (or even improving) outcomes by more accurately identifying patients that need hospitalization. As the most accurate models, neural nets were a strong candidate for clinical trial. Deploying neural net models that could not be understood, however, was deemed too risky —

[1]SVMs and boosted trees were not in common use yet, and Random Forests had not yet been invented.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788613 .

Both models predicted patient recovery with high accuracy with the neural network found to be the most accurate.

The need for explainability

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu

ABSTRACT

In machine learning often a tradeoff must be made between accuracy and intelligibility. More accurate models such as boosted trees, random forests, and neural nets usually are not intelligible, but more intelligible models such as logistic regression, naive-Bayes, and single decision trees often have significantly worse accuracy. This tradeoff sometimes limits the accuracy of models that can be applied in mission-critical applications such as healthcare where being able to understand, validate, edit, and trust a learned model is important. We present two case studies where high-performance generalized additive models with pairwise interactions (GA^2Ms) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy. In the pneumonia risk prediction case study, the intelligible model uncovers surprising patterns in the data that previously had prevented complex learned models from being fielded in this domain, but because it is intelligible and modular allows these patterns to be recognized and removed. In the 30-day hospital readmission case study, we show that the same methods scale to large datasets containing hundreds of thousands of patients and thousands of attributes while remaining intelligible and providing accuracy comparable to the best (unintelligible) machine learning methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning—*Induction*

Keywords

intelligibility; classification; interaction detection; additive models; logistic regression; healthcare; risk prediction

1. MOTIVATION

In the mid 90's, a large multi-institutional project was funded by Cost-Effective HealthCare (CEHC) to evaluate the application of machine learning to important problems in healthcare such as predicting pneumonia risk. In the study, the goal was to predict the probability of death (POD) for patients with pneumonia so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients. In the study [3, 2], the most accurate models that could be trained were multitask neural nets[1]. On one dataset the neural nets outperformed traditional methods such as logistic regression by wide margin (the neural net had AUC=0.86 compared to 0.77 for logistic regression), and on the other dataset used in this paper outperformed logistic regression by about 0.02 (see Table 2). Although the neural nets were the most accurate models, after careful consideration they were considered too risky for use on real patients and logistic resgression was used instead. Why?

One of the methods being evaluated was rule-based learning [1]. Although models based on rules were not as accurate as the neural net models, they were *intelligible*, i.e., interpretable by humans. On one of the pneumonia datasets, the rule-based system learned the rule "HasAsthama(x) ⇒ LowerRisk(x)", i.e., that patients with pneumonia who have a history of asthma have lower risk of dying from pneumonia than the general population. Needless to say, this rule is counterintuitive. But it reflected a true pattern in the training data: patients with a history of asthma who presented with pneumonia usually were admitted not only to the hospital but directly to the ICU (Intensive Care Unit). The good news is that the aggressive care received by asthmatic pneumonia patients was so effective that it lowered their risk of dying from pneumonia compared to the general population. The bad news is that because the prognosis for these patients is better than average, models trained on the data incorrectly learn that asthma lowers risk, when in fact asthmatics have much higher risk (if not hospitalized).

One of the goals of the study was to perform a clinical trial to determine if machine learning could be used to predict risk prior to hospitalization so that a more informed decision about hospitalization could to be made. The ultimate goal was to reduce healthcare cost by reducing hospital admissions, while maintaining (or even improving) outcomes by more accurately identifying patients that need hospitalization. As the most accurate models, neural nets were a strong candidate for clinical trial. Deploying neural net models that could not be understood, however, was deemed too risky —

[1]SVMs and boosted trees were not in common use yet, and Random Forests had not yet been invented.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788613 .

Both models predicted that pneumonia patients with asthma shouldn't be admitted because they had a lower risk of dying!

The need for explainability

Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission

Rich Caruana
Microsoft Research
rcaruana@microsoft.com

Yin Lou
LinkedIn Corporation
ylou@linkedin.com

Johannes Gehrke
Microsoft
johannes@microsoft.com

Paul Koch
Microsoft Research
paulkoch@microsoft.com

Marc Sturm
NewYork-Presbyterian Hospital
mas9161@nyp.org

Noémie Elhadad
Columbia University
noemie.elhadad@columbia.edu

ABSTRACT

In machine learning often a tradeoff must be made between accuracy and intelligibility. More accurate models such as boosted trees, random forests, and neural nets usually are not intelligible, but more intelligible models such as logistic regression, naive-Bayes, and single decision trees often have significantly worse accuracy. This tradeoff sometimes limits the accuracy of models that can be applied in mission-critical applications such as healthcare where being able to understand, validate, edit, and trust a learned model is important. We present two case studies where high-performance generalized additive models with pairwise interactions (GA^2Ms) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy. In the pneumonia risk prediction case study, the intelligible model uncovers surprising patterns in the data that previously had prevented complex learned models from being fielded in this domain, but because it is intelligible and modular allows these patterns to be recognized and removed. In the 30-day hospital readmission case study, we show that the same methods scale to large datasets containing hundreds of thousands of patients and thousands of attributes while remaining intelligible and providing accuracy comparable to the best (unintelligible) machine learning methods.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning—*Induction*

Keywords

intelligibility; classification; interaction detection; additive models; logistic regression; healthcare; risk prediction

1. MOTIVATION

In the mid 90's, a large multi-institutional project was funded by Cost-Effective HealthCare (CEHC) to evaluate the application of machine learning to important problems in healthcare such as predicting pneumonia risk. In the study, the goal was to predict the probability of death (POD) for patients with pneumonia so that high-risk patients could be admitted to the hospital while low-risk patients were treated as outpatients. In the study [3, 2], the most accurate models that could be trained were multitask neural nets[1]. On one dataset the neural nets outperformed traditional methods such as logistic regression by wide margin (the neural net had AUC=0.86 compared to 0.77 for logistic regression), and on the other dataset used in this paper outperformed logistic regression by about 0.02 (see Table 2). Although the neural nets were the most accurate models, after careful consideration they were considered too risky for use on real patients and logistic resgression was used instead. Why?

One of the methods being evaluated was rule-based learning [1]. Although models based on rules were not as accurate as the neural net models, they were *intelligible*, i.e., interpretable by humans. On one of the pneumonia datasets, the rule-based system learned the rule "HasAsthama(x) ⇒ LowerRisk(x)", i.e., that patients with pneumonia who have a history of asthma have lower risk of dying from pneumonia than the general population. Needless to say, this rule is counterintuitive. But it reflected a true pattern in the training data: patients with a history of asthma who presented with pneumonia usually were admitted not only to the hospital but directly to the ICU (Intensive Care Unit). The good news is that the aggressive care received by asthmatic pneumonia patients was so effective that it lowered their risk of dying from pneumonia compared to the general population. The bad news is that because the prognosis for these patients is better than average, models trained on the data incorrectly learn that asthma lowers risk, when in fact asthmatics have much higher risk (if not hospitalized).

One of the goals of the study was to perform a clinical trial to determine if machine learning could be used to predict risk prior to hospitalization so that a more informed decision about hospitalization could to be made. The ultimate goal was to reduce healthcare cost by reducing hospital admissions, while maintaining (or even improving) outcomes by more accurately identifying patients that need hospitalization. As the most accurate models, neural nets were a strong candidate for clinical trial. Deploying neural net models that could not be understood, however, was deemed too risky —

[1]SVMs and boosted trees were not in common use yet, and Random Forests had not yet been invented.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788613 .

In fact, pneumonia patients were at high risk, but they were routinely admitted directly to the intensive care unit, treated aggressively, and as a consequence had a high survival rate.

Because the rule-based model was interpretable, it was possible to see that the model had learnt '*if the patient has asthma, they are at lower risk*'.

The need for explainability

There are different reasons that drive the demand for interpretability and explanations:

- The goal of science is to **gain knowledge**, but many problems are solved with big datasets and black box machine learning models. The model itself becomes the source of knowledge instead of the data. Interpretability makes it possible to extract this additional knowledge captured by the model.

The need for explainability

- Machine learning models take on real-world tasks that require **safety** measures and testing (f.e. self-driving cars).
- By default, machine learning models pick up biases from the training data. This can turn your machine learning models into racists that discriminate against protected groups. Interpretability can be a useful **debugging** tool.
- The process of integrating machines and algorithms into our daily lives requires interpretability to increase social acceptance and **trust**.

The need for explainability

The "**explainability**" concept is somewhat ambiguous, and can mean different things to different people in different contexts.

Each meaning requires a different sort of explanation, requiring different measures of efficacy:

- For a **developer**, to understand how their system is working, aiming to **debug** or improve it: to see what is working well or badly, and get a sense for why.
- For a **user**, to provide a sense for what the system is doing and why, to enable prediction of what it might do in unforeseen circumstances and build a sense of **trust** in the technology.
- For **society** broadly to understand and become comfortable with the **strengths and limitations** of the system, overcoming a reasonable fear of the unknown.
- For a **user to understand why one particular prediction or decision** was reached, to allow a check that the system worked appropriately and to enable meaningful challenge (e.g. credit approval or criminal sentencing).
- To provide an expert (perhaps a **regulator**) the ability to audit a prediction or decision trail in detail, particularly if something goes wrong (e.g. a crash by an autonomous car).
- Etc.

Explanations: the social science perspective

An explanation is the **answer to a why-question** (Miller 2017).

- Why did not the treatment work on the patient?
- Why was my loan rejected?

Miller, Tim. 2017. “Explanation in Artificial Intelligence: Insights from the Social Sciences.” arXiv Preprint arXiv:1706.07269.

Explanations: the social science perspective

A good explanation is:

- **Contrastive**. Humans usually do not ask why a certain prediction was made, but **why this prediction was made *instead of another prediction***. The solution for the automated creation of contrastive explanations might also involve finding prototypes or archetypes in the data.

- **Selected**. People do not expect explanations that cover the actual and complete list of causes of an event. We are used to selecting **one or two causes** from a variety of possible **causes** as **THE** explanation.

- **Social**. The social context determines the content and nature of the explanations. Getting the social part of the machine learning model right depends entirely on your specific application.

Miller, Tim. 2017. "Explanation in Artificial Intelligence: Insights from the Social Sciences." arXiv Preprint arXiv:1706.07269.

Explanations: the social science perspective

A good explanation is:

- **Focused on the abnormal**. People focus more on causes that had a small probability but nevertheless happened.
- **Consistent** with prior beliefs of the one who receives the explanation.
- **General and probable**. A cause that can explain many events is very general and could be considered a good explanation. Generality can easily be measured by the feature's support, which is the number of instances to which the explanation applies divided by the total number of instances.
- (...)

Miller, Tim. 2017. "Explanation in Artificial Intelligence: Insights from the Social Sciences." arXiv Preprint arXiv:1706.07269.

Explanations & ML

What is an ML-explanation?

Given a ML system $y = f(\mathbf{X})$ where $f: \mathbb{R}^n \rightarrow \mathbb{R}$ or $f: \mathbb{R}^n \rightarrow \{0,1\}$, one of the most commonly asked questions is about the **importance** of a component x_i of $\mathbf{X}$:

"*Annual income is the main factor for denying a load application*"

What is an ML-explanation?

Given a ML system $y = f(\mathbf{X})$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ or $f : \mathbb{R}^n \rightarrow \{0,1\}$, one of the most commonly asked questions is about the **importance** of a component x_i of $\mathbf{X}$:

"*Annual income is the main factor for denying a load application*"

As an alternative, we can also look for **contrastive** explanations:

"*Your application was denied because your annual income is $30,000 and your current balance is $200. If your income had instead been $35,000 and your current balance had been $400, your application would have been approved.*"

Variable Importance

There are at least 3 notions of variable importance:

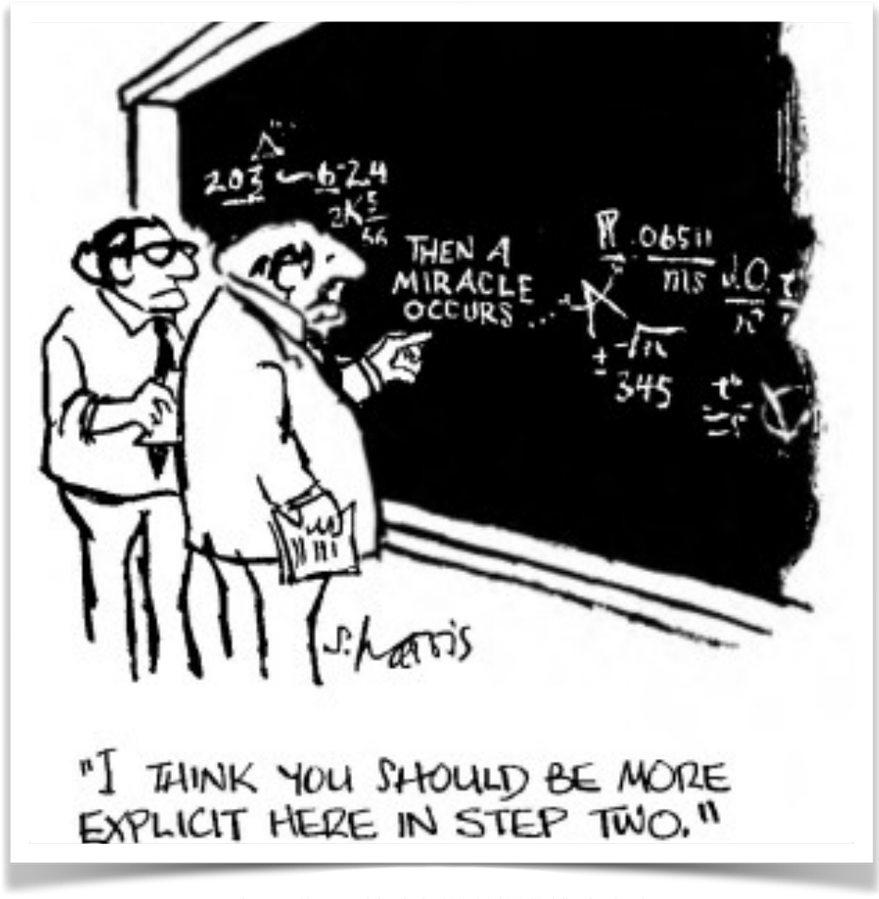
- To take the function $f(\mathbf{X})$ at its face value and ask which variable x_i has a big **impact** in $f(\mathbf{X})$.

  If $f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{n} \beta_j x_j$ is a linear model, β_j can be used to measure the importance of x_j (given it is properly normalized).

- To measure the importance of x_j by its **contribution** to predictive accuracy.

- To measure the **causal effect** of an intervention on x_j.

Explanations & Interpretable models



https://uc-r.github.io/2018/08/01/iml-pkg/

Interpretable models are models who "explain themselves", such as decision trees, logistic regression, etc.

The easiest way to achieve interpretability is to use only a subset of algorithms that create **interpretable** models.

Explanations & Interpretable models

A **linear regression model** predicts the target as a weighted sum of the feature inputs:

$$y = \beta_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_n x_n + \epsilon, \quad \epsilon \sim N(0,\sigma^2)\,.$$

Estimated weights can come with confidence intervals, such as standard error values.

We have to measure the uncertainty of the parameter!

P.e. We can compute the **SE** of every parameter by by using **Bootstrapping**

Explanations & Interpretable models

We can interpret a linear regression model by considering the following observation:

An increase of feature x_i (<u>when all other feature values remain fixed</u>) by one unit increases the value of the outcome y by β_i units.

Then, we can measure the importance of x_i by this statistic:

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

The importance of a feature increases with increasing weight. The more variance the estimated weight has (or the less certain we are about the correct value), the less important the feature is.

Explanations & Interpretable models

The main limitation of this kind of explanations is due to the **multicoliniarity effect** in linear models:

Given a data point $(x_1, \ldots, x_p)$***, the value of*** x_i ***depends on (possibly) all other features.***

In other words, some features can be predicted by the other.

Extreme case: When representing a person, "$x_1 = person\ weight$" and "$x_2 = person\ height$" are correlated. If x_1 is correlated with the outcome, y, the model can assign a zero weight to x_2 because it is not necessary.

$y = \beta_0 + \hat{\beta}_1 x_1$ can be a solution as good as $y = \beta_0 + \hat{\beta}_2 x_2$

Explanations & Interpretable models

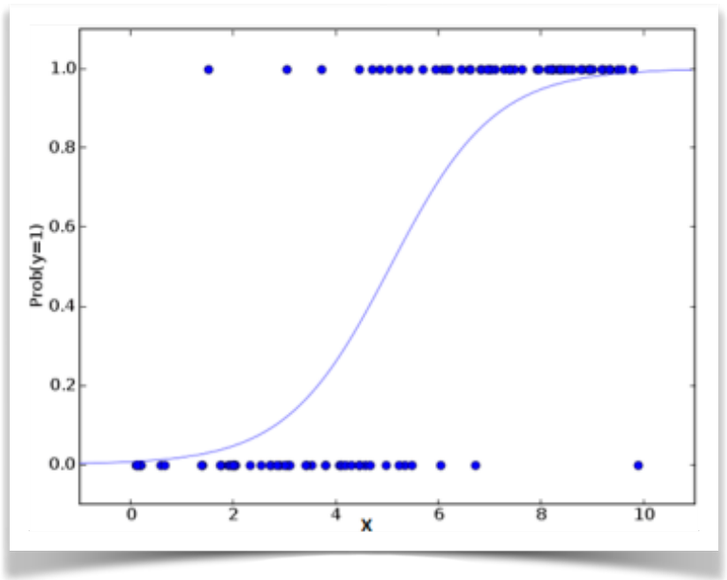
This interpretation of the linear model is "correct" from the point of view of the **predicting behavior of the model**, but it is "not correct" from the point of the view of the phenomena we are explaining! This is called a **mechanistic explanation**.

- The model is too dependent of **variable correlations**.
- The **probing strategy** ("An increase of feature, when all other feature values remain fixed, by one unit increases the value of the outcome by units") can produce **impossible** data points because it does not take into account the causal structure of the data generating process.

Explanations & Interpretable models

A **logistic regression model** predicts the target as:

$$P(y^{(i)} = 1) = \frac{1}{1 + exp(-(\hat{\beta}_0 + \hat{\beta}_1 x_1^{(i)} + \ldots + \hat{\beta}_p x_p^{(i)}))}$$



Then, it can be shown that a change in feature x_j by 1 unit changes de "odds" ratio by a factor of $\exp(\hat{\beta}_j)$.

Explanations & Interpretable models

Decision Trees

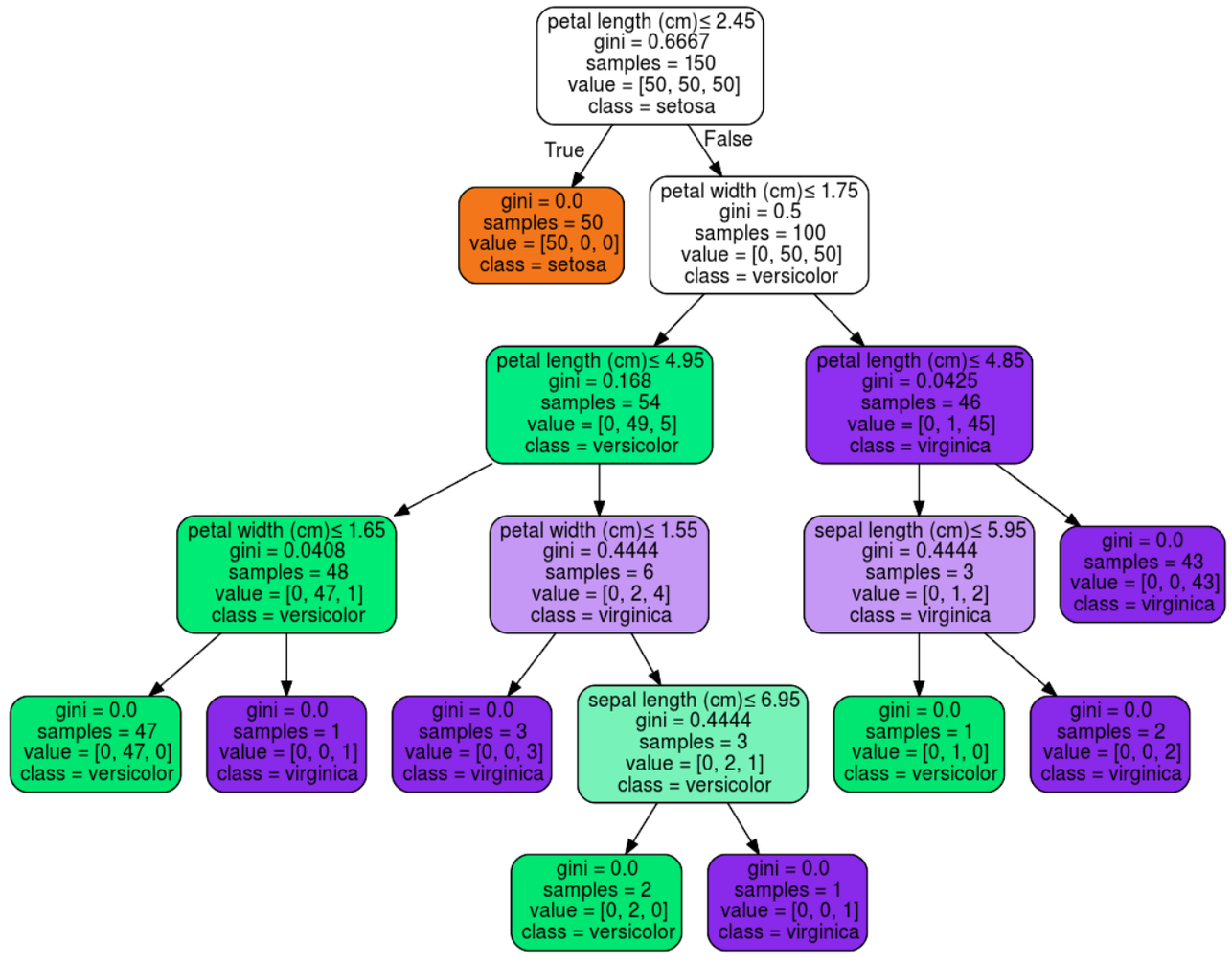
Gini index measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

$$G = 1 - \sum_{i=1}^{n} (p_i)^2$$

where p_i is the probability of an example being classified to a particular class.

The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes. A Gini Index of 0.5 denotes equally distributed elements into some classes.

In the case of regression tasks, we would use variance instead of Gini.



https://scikit-learn.org/stable/modules/tree.html

Explanations & Interpretable models

Decision Trees

The **overall importance of a feature** in a decision tree can be computed in the following way:

1. Go through all the splits for which the feature was used and measure how much it has reduced the variance/Gini index compared to the parent node.
2. The sum of all importances is scaled to 100. This means that each importance can be interpreted as share of the overall model importance.

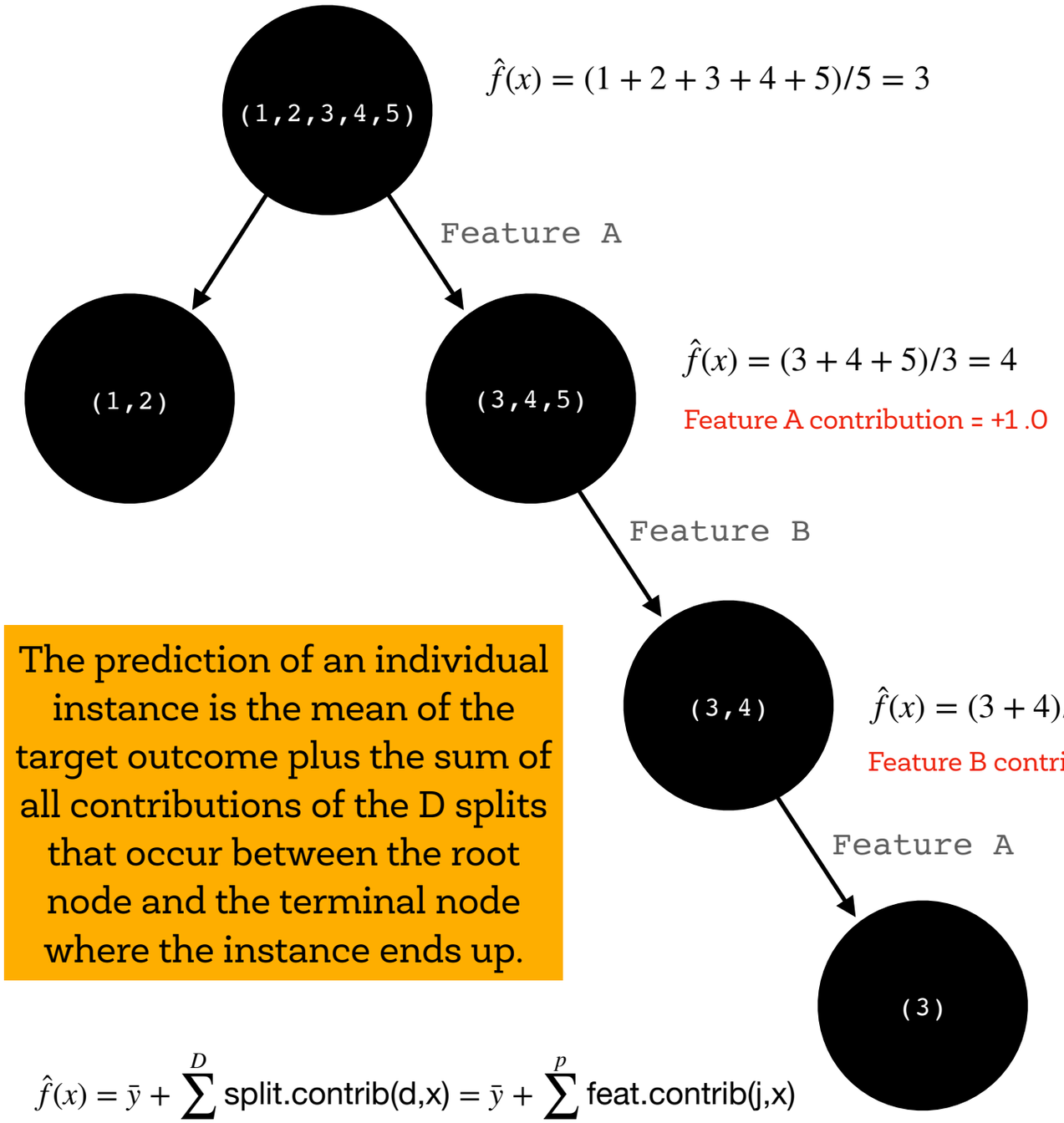
Explanations & Interpretable models

Decision Trees

Individual predictions of a decision tree can be explained by decomposing the decision path into one component per feature.

We can track a decision through the tree and explain a prediction by the contributions added at each decision node.

Explanations & Interpretable models



$\hat{f}(x) = (1+2+3+4+5)/5 = 3$

$\hat{f}(x) = (3+4+5)/3 = 4$

Feature A contribution = +1 .0

$\hat{f}(x) = (3+4)/2 = 7.5$

Feature B contribution = +3.5

$\hat{f}(x) = 3$

Feature A contribution = -4.5

Feature A Feature B

$$\hat{f}(x) = 3 - 3.5 + 3.5$$

Total feature A contribution = -3.5
Total B contribution = +3.5

The prediction of an individual instance is the mean of the target outcome plus the sum of all contributions of the D splits that occur between the root node and the terminal node where the instance ends up.

$$\hat{f}(x) = \bar{y} + \sum_{d=1}^{D} \text{split.contrib(d,x)} = \bar{y} + \sum_{j=1}^{p} \text{feat.contrib(j,x)}$$

Explanations & Interpretable models

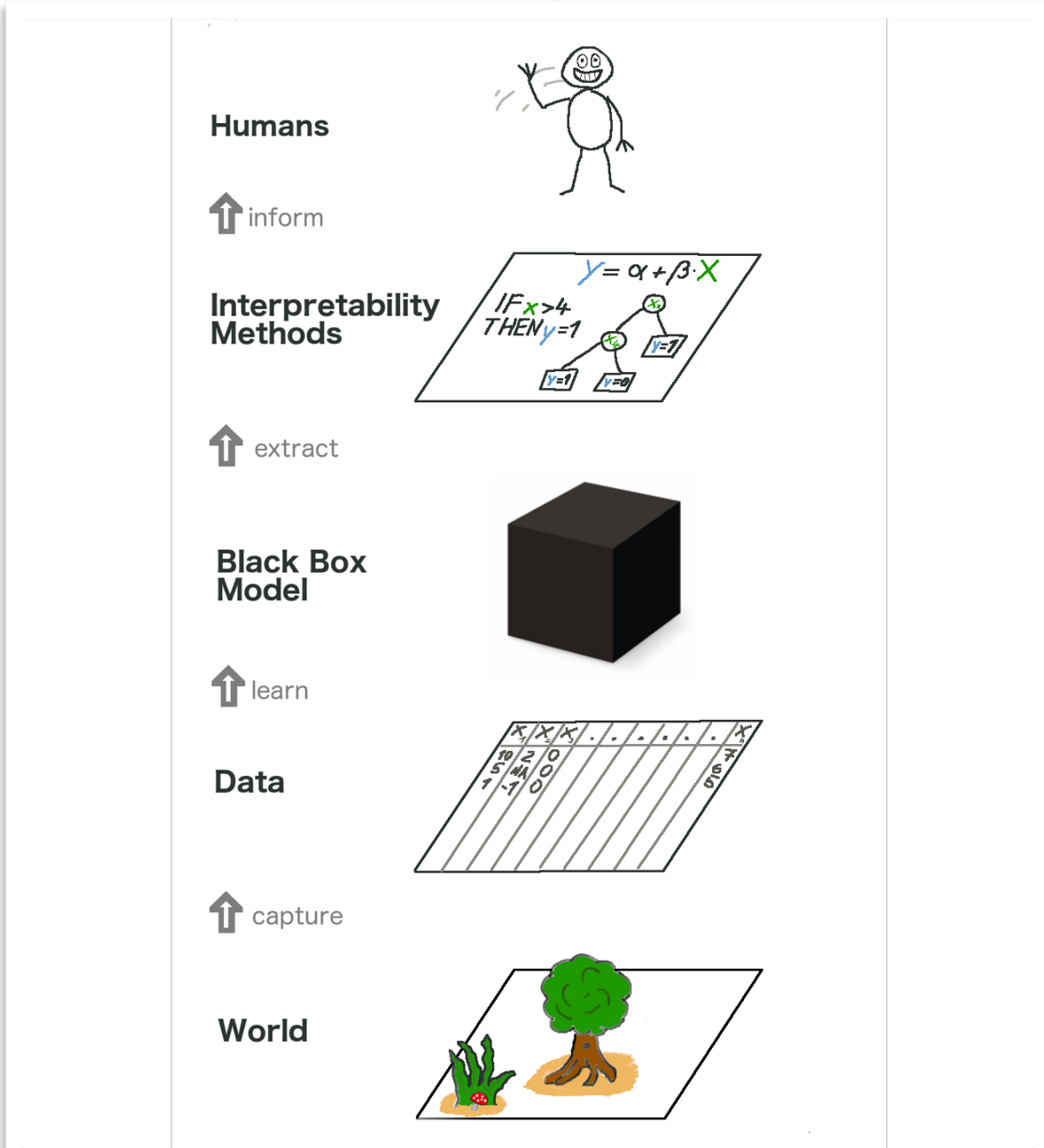
We can design this kind of explanations to other popular models:

- Generalized Linear Models and Generalized Additive Models
- Naive Bayes
- k-nearest neighbors
- Etc.

But this is mechanistic explanation of the behavior of the model. Is this the explanation we are looking for?

- Does it make sense to increase the value of a feature without taking into consideration other features?
- What about explanations that require complex combinations of features?

Model-agnostic explanation



https://christophm.github.io/interpretable-ml-book/

Model-agnostic explanation

Model-agnostic methods are methods you can use for any machine learning model, from support vector machines to neural networks.

Moreover, these are the **most interesting methods when the objective of explanations is to understand, discuss, and potentially contest decisions, not to explain how a specific model works**.

It is the **only alternative when models are too complex to be understood**.

Model-agnostic explanation

The **partial dependence plot** (PDP) method shows the marginal effect one (or two features) have on the predicted outcome of a machine learning model.

For example, let's assume a data set that only contains three data points and three features (A, B, C) as shown below.

A	B	C	Y
a1	b1	c1	**y1**
a2	b2	c2	**y2**
a3	b3	c3	**y3**

Model-agnostic explanation

If we want to see how feature A is influencing the prediction Y, what PDP does is to **generate a new data set** as follows and **do prediction as usual**.

A	B	C	Y
a1	b1	c1	**y1**
a2	b2	c2	**y2**
a3	b3	c3	**y3**

A	B	C	Y
a1	b1	c1	**y11**
a1	b2	c2	**y21**
a1	b3	c3	**y31**
a2	b1	c1	**y12**
a2	b2	c2	**y22**
a2	b3	c3	**y32**
a3	b1	c1	**y13**
a3	b2	c2	**y23**
a3	b3	c3	**y33**

This method can produce unlikely data instances when two or more features are correlated.

Model-agnostic explanation

Then, it averages the predictions for having a unique value of feature A:

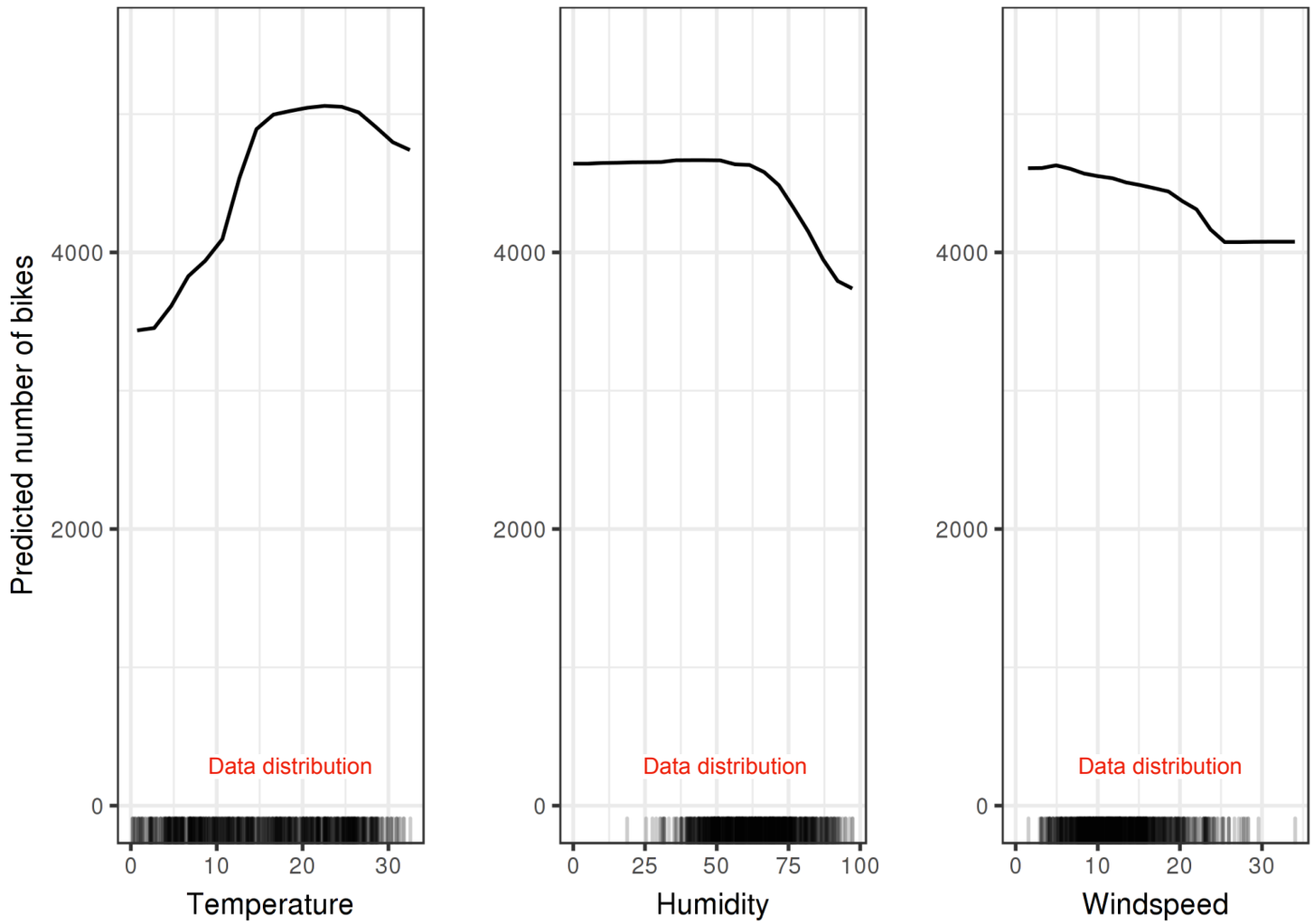
A	B	C	Y
a1	b1	c1	yA1
a1	b2	c2	
a1	b3	c3	
a2	b1	c1	yA2
a2	b2	c2	
a2	b3	c3	
a3	b1	c1	yA3
a3	b2	c2	
a3	b3	c3	

Finally, it plots out the average predictions.

X	A1	A2	A3
Y	yA1	yA2	yA3

Model-agnostic explanation

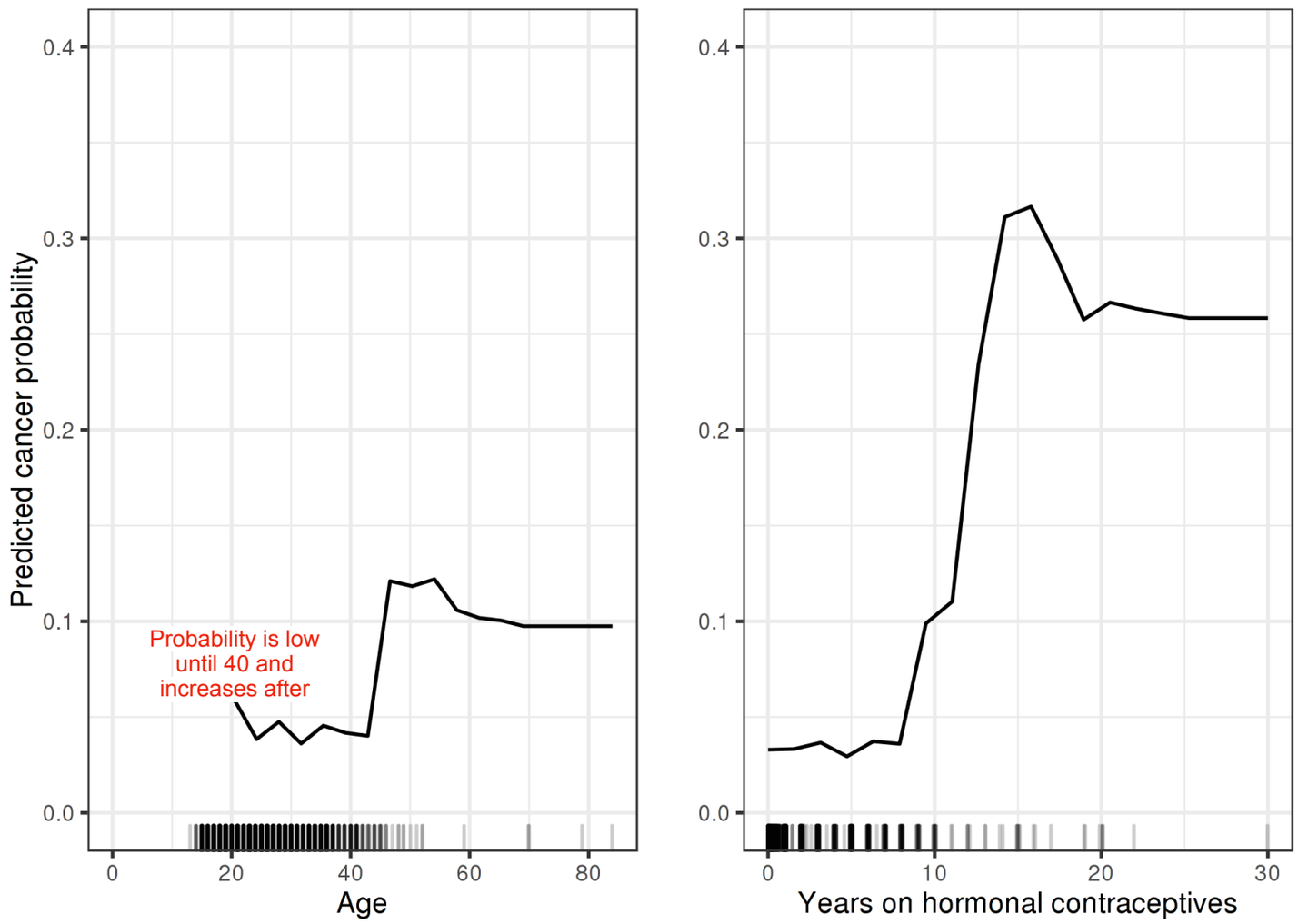
Problem: predict the number of bikes that will be rented on a given day. The influence of the weather features on the predicted bike counts is visualized in the following figure.



https://christophm.github.io/interpretable-ml-book/

Model-agnostic explanation

Problem: cervical cancer classification.



For both features not many data points with large values were available, so the PD estimates are less reliable in those regions.

https://christophm.github.io/interpretable-ml-book/

Model-agnostic explanation

Partial Dependence Plots Notebook

Model-agnostic explanation

Permutation Test

The importance of a feature is the increase in the prediction error of the model after we permuted the feature's values, which breaks the relationship between the feature and the true outcome.

A feature is "**important**" if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction.

A feature is "**unimportant**" if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.

Model-agnostic explanation

Permutation Test

Input: Trained model f, feature matrix X, target vector y, error measure L(y,f).

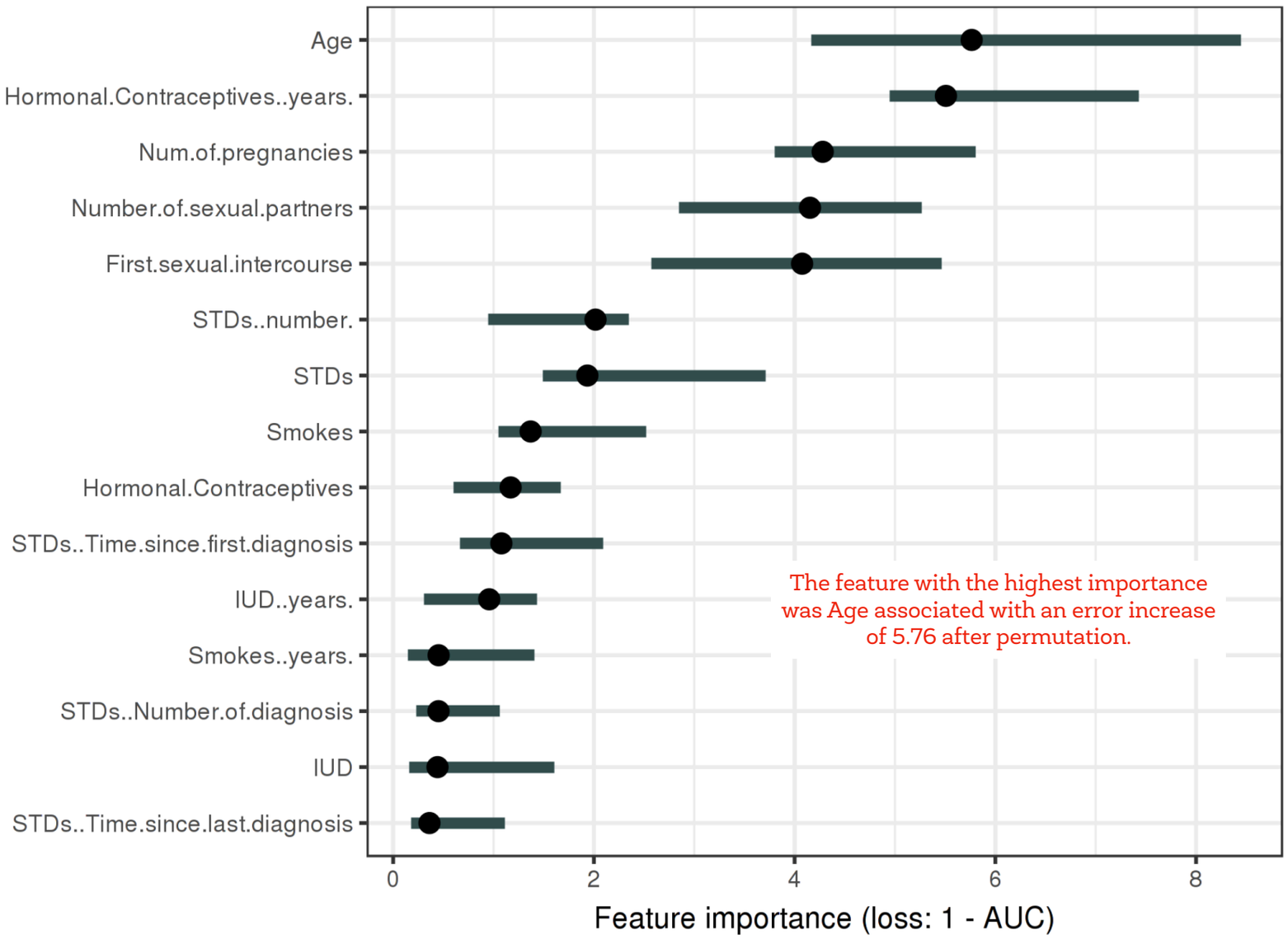
1. Estimate the original model error e^{orig} = L(y, f(X)) (e.g. mean squared error)
2. For each feature j = 1,...,p do:
    - Generate feature matrix X^{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
    - Estimate error e^{perm} = L(Y,f(X^{perm})) based on the predictions of the permuted data.
    - Calculate permutation feature importance FI^j= e^{perm}/e^{orig}. Alternatively, the difference can be used: FI^j = e^{perm} - e^{orig}
3. Sort features by descending FI.

The problem is the same as with partial dependence plots: The permutation of features produces **unlikely** data instances when two or more features are correlated.

Model-agnostic explanation

Permutation Test

Problem: Predict cervical cancer.



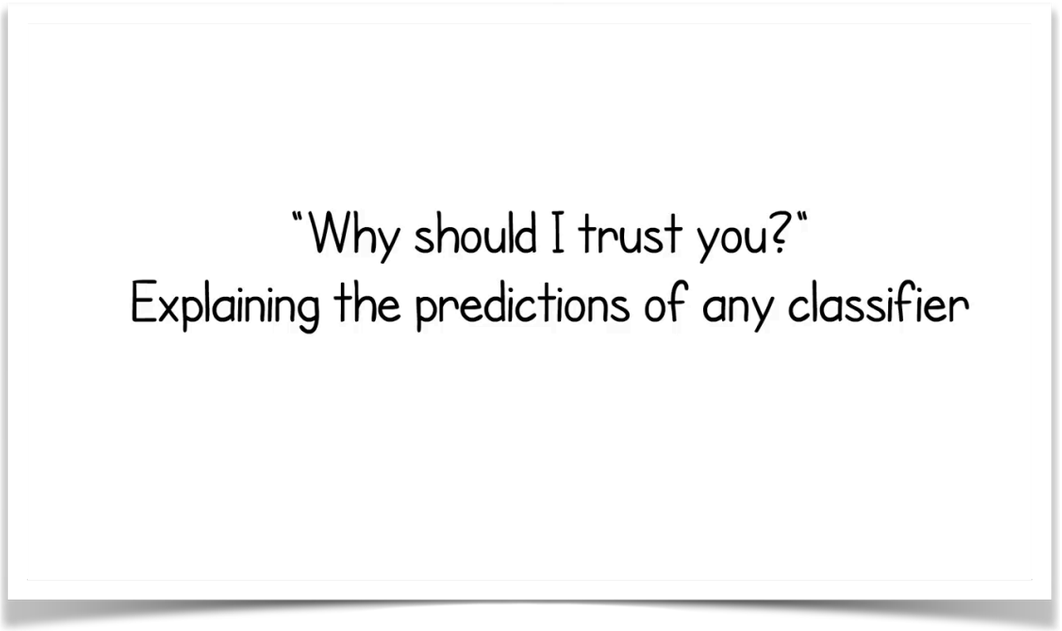
Model-agnostic explanation

Permutation Test Notebook

Model-agnostic explanation

LIME: Local interpretable model-agnostic explanations.

Local surrogate models are interpretable models that are used to explain **individual predictions** of black box machine learning models.



Surrogate models are trained to approximate the predictions of the underlying black box model. Instead of training a global surrogate model, LIME focuses on training local surrogate models to explain individual predictions.

Model-agnostic explanation

LIME: Local interpretable model-agnostic explanations.

Imagine you can probe the box as often as you want.

LIME generates a new dataset consisting of “perturbed” samples and the corresponding predictions of the black box model.

On this new dataset LIME then trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest.

Model-agnostic explanation

LIME: Local interpretable model-agnostic explanations.

The recipe for training local surrogate models:

- Select your instance of interest for which you want to have an explanation of its black box prediction.
- Perturb your dataset and get the black box predictions for these new points.
- Weight the new samples according to their proximity to the instance of interest.
- Train a weighted, interpretable model on the dataset with the variations.
- Explain the prediction by interpreting the local model.

Model-agnostic explanation

LIME: Local interpretable model-agnostic explanations.

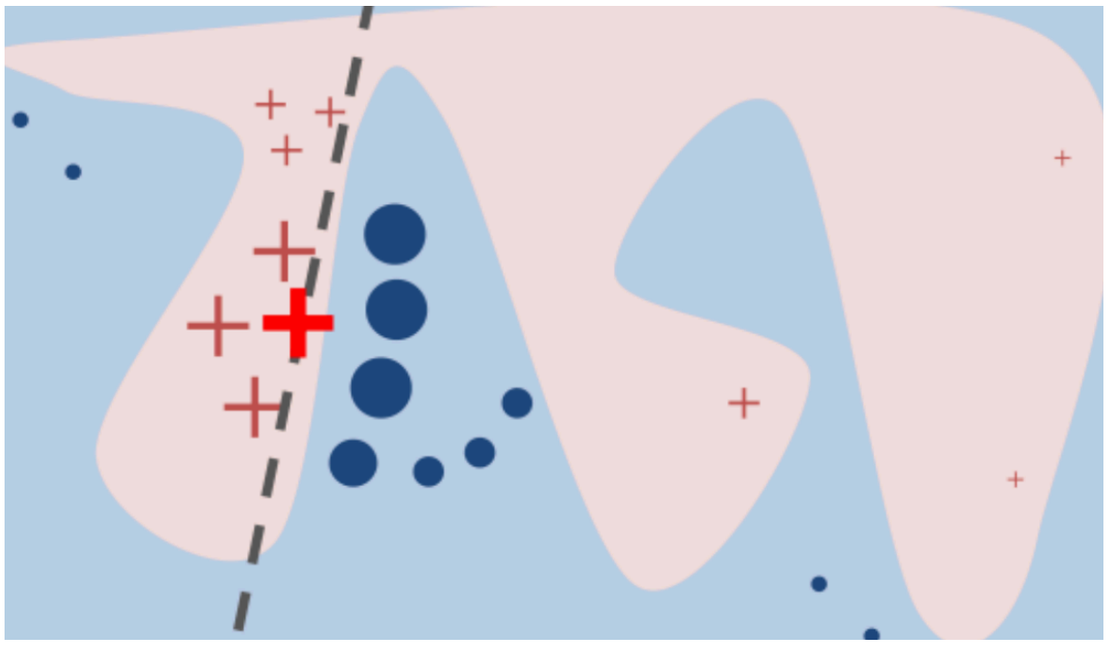
The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background.

The bright bold red cross is the instance being explained.

LIME samples instances, gets predictions using f, and weighs them by the proximity to the instance being explained (represented here by size).

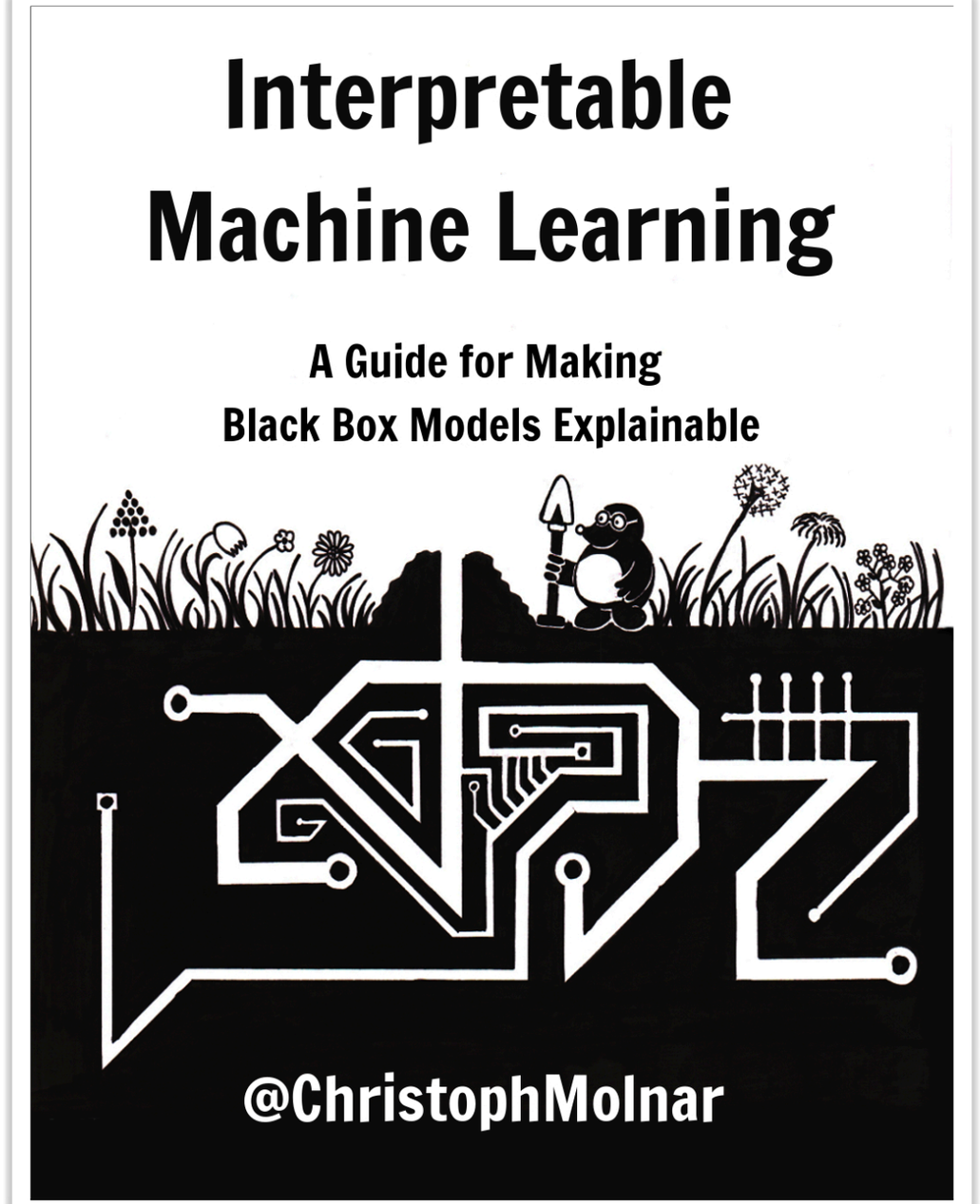
The dashed line is the learned explanation that is locally (but not globally) faithful.

Model-agnostic explanation



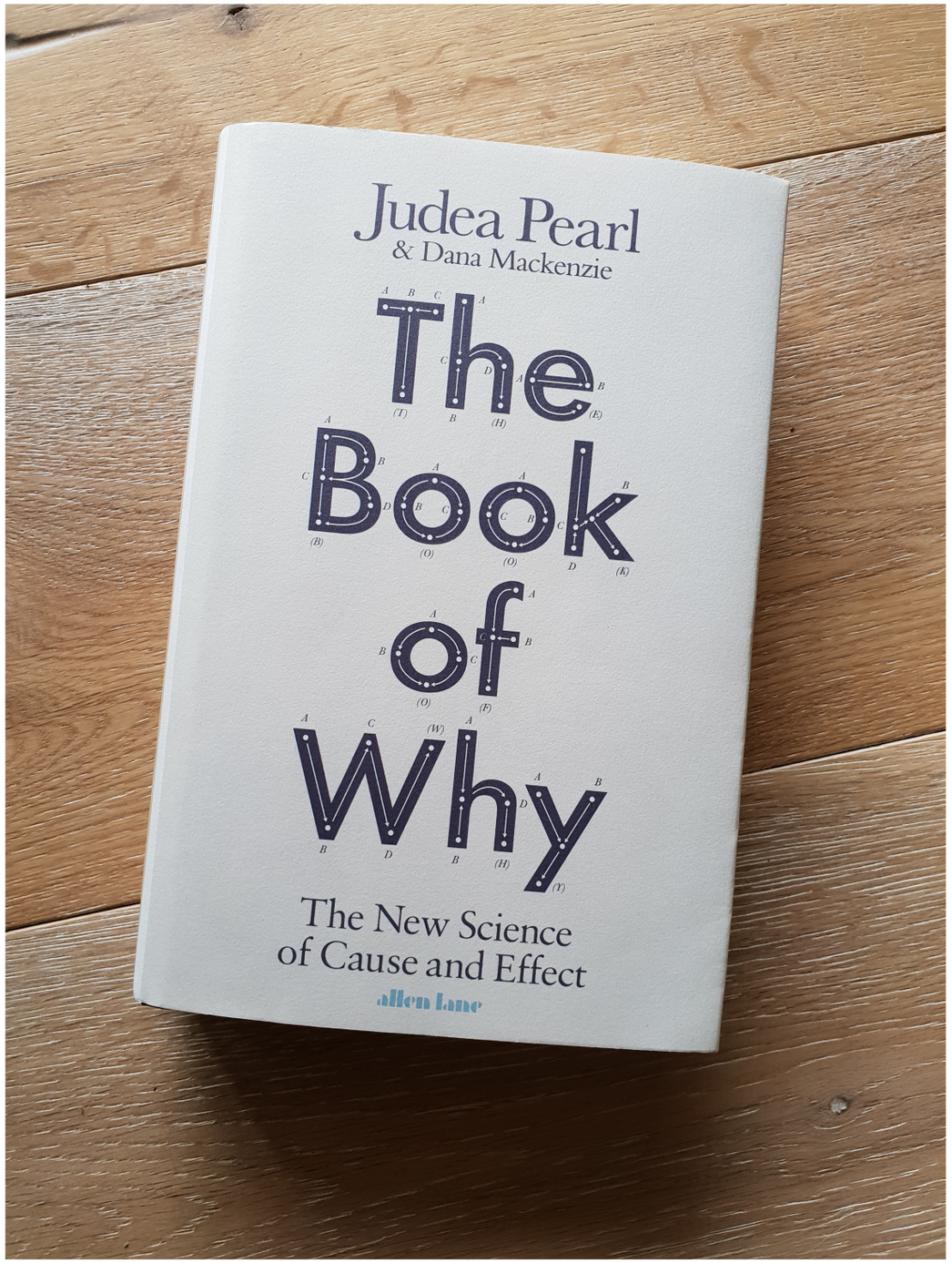
https://www.cc.gatech.edu/~alanwags/DLAI2016/(Gunning)%20IJCAI-16%20DLAI%20WS.pdf

Resources

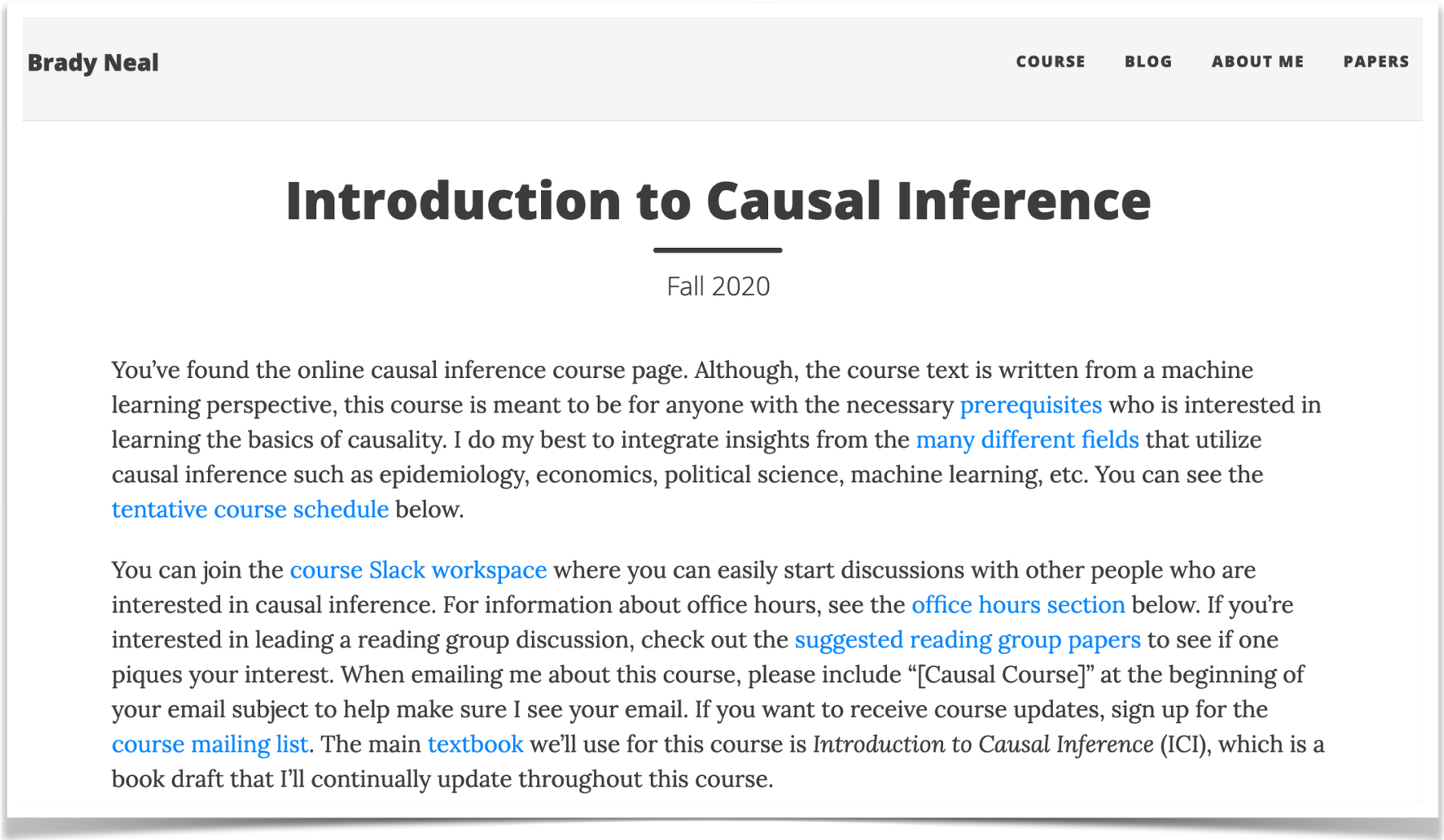


https://christophm.github.io/interpretable-ml-book/

Introduction to Causal Inference



Introduction to Causal Inference



Brady Neal COURSE BLOG ABOUT ME PAPERS

Introduction to Causal Inference

Fall 2020

You've found the online causal inference course page. Although, the course text is written from a machine learning perspective, this course is meant to be for anyone with the necessary prerequisites who is interested in learning the basics of causality. I do my best to integrate insights from the many different fields that utilize causal inference such as epidemiology, economics, political science, machine learning, etc. You can see the tentative course schedule below.

You can join the course Slack workspace where you can easily start discussions with other people who are interested in causal inference. For information about office hours, see the office hours section below. If you're interested in leading a reading group discussion, check out the suggested reading group papers to see if one piques your interest. When emailing me about this course, please include "[Causal Course]" at the beginning of your email subject to help make sure I see your email. If you want to receive course updates, sign up for the course mailing list. The main textbook we'll use for this course is *Introduction to Causal Inference* (ICI), which is a book draft that I'll continually update throughout this course.