



MSc in Fundamental Principles of Data Science

# 4 Ethical Data Science

## Transparency & Explainability

# Algorithmic decision-making

**There was once a time when humans made important decisions about other humans.**

You went to your bank manager, in person, to ask for a loan.  
A human hiring committee decided which candidate would get a job.  
And judges even determined what a guilty offender's sentence would be!

Now, many of those decisions are made by AIs.

There are a **couple of problems** with AIs making these decisions. First of all, the algorithms the machines use are often **proprietary**, meaning they aren't available to the general public or other computer scientists to examine. Second, and perhaps more troubling, **AIs often interpret data in ways more complex than even their programmers can understand**. In those cases, nobody could explain how the "black box" in an AI works, even if they wanted to.

CBC Radio · January 19

<https://www.cbc.ca/radio/spark/422-1.4982026/asking-why-instead-of-how-could-better-explain-ai-decisions-1.4982038>

Big data enables algorithmic decisions

# Algorithmic decision-making

WIRED

## Spain is about to shatter the gig economy's algorithmic black box

Riders for companies like Deliveroo and Glovo are on the brink of being told how the algorithms ranking them really work

By MARIA ALEMANY ORTIZ  
1 day ago

[f](#) [t](#) [e](#)



Credit: Getty Images

It's lunch-time on a Saturday in Barcelona, and José Maracucho is waiting outside an Italian restaurant in the city centre. His large, squarish yellow Glovo backpack lies next to him, but the order he is about to deliver is from UberEats. For the past few months, Maracucho has been working for both food-delivery apps, because the money he earns from the few hours he is allowed to work for Glovo isn't enough to live on. Delivery riders like Maracucho have been clamouring for years for better regulations to protect their labour rights. Now regulation is finally coming, even if the change it will bring about is fraught with uncertainty.

In the coming months, Spain is **set to approve** a trailblazing new law enshrining the labour rights of Spain's estimated 8,000 delivery riders. In what could set a precedent for other European countries, the law will declare that gig economy riders are wage labourers, and **force delivery app companies to open up their algorithmic black boxes to explain to their riders what variables they are using to determine their in-app scores**. The labour unions fighting for Spain's overworked and poorly treated delivery riders see this as a victory.

# Algorithmic decision-making

## L'Obs

[POLITIQUE](#)[MONDE](#)[ÉCONOMIE](#)[CULTURE](#)[OPINIONS](#)[DÉBATS](#)[TENDANCES](#)[VIDÉOS](#)[PHOTOS](#)[M'identifier](#)[Je m'abonne](#)

L'Obs > Education

## Derrière l'algorithme de Parcoursup, un choix idéologique

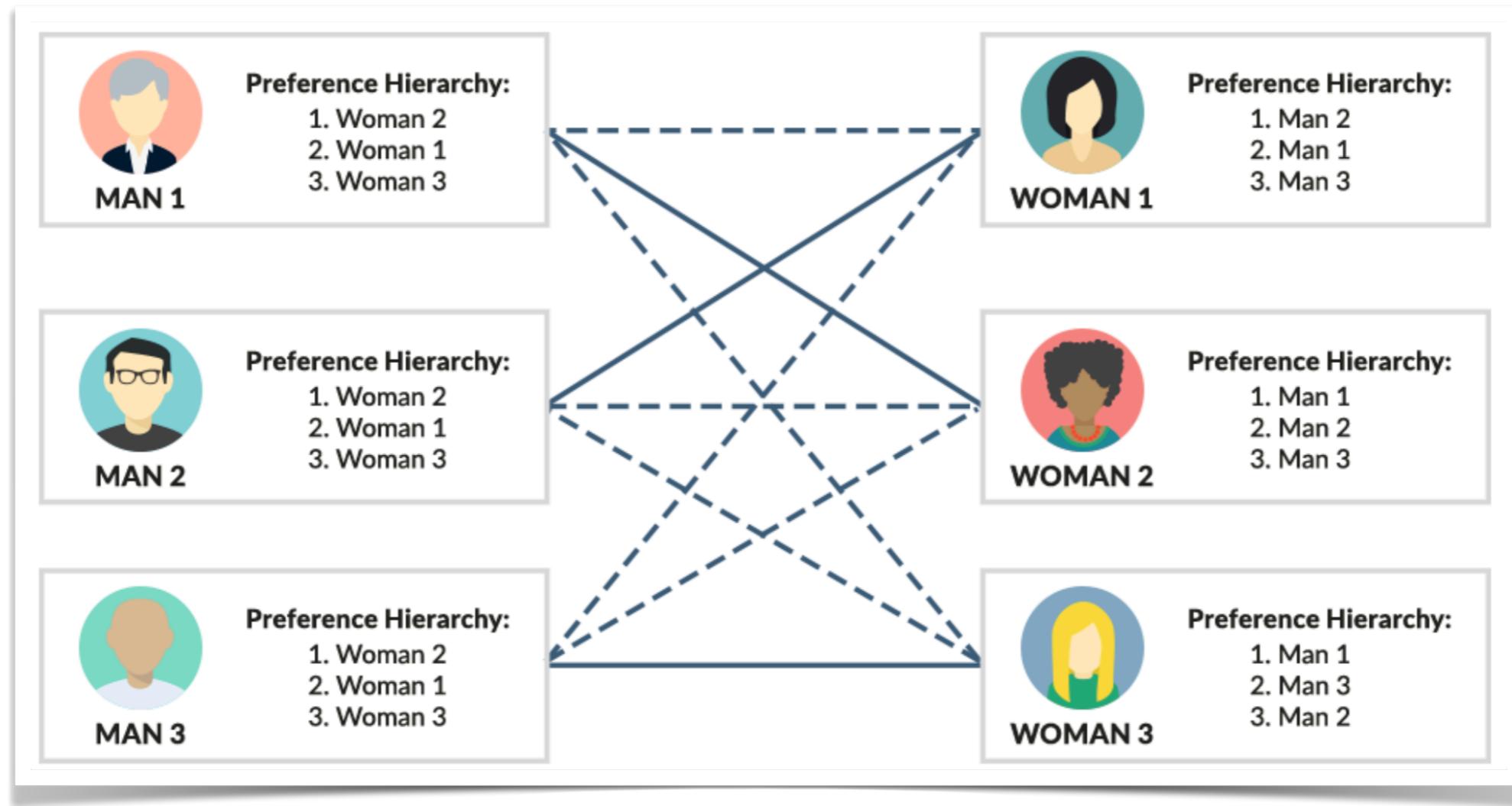
La répartition des étudiants entre les universités et les filières est un problème complexe puisqu'elle s'effectue sur base d'un conflit massif entre l'offre et la demande : on dénombre plus de 880.000 candidats pour un total (à raison de 10 vœux possibles par candidat) de quelques 7.000.000 de vœux de formation *[810.000 ont finalement validé leurs vœux, 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  
**EXPLANATION** and then  
we designed the  
**ALGORITHM**.

# Algorithmic decision-making

ML reverses one important aspect:

In the case of France, we  
first defined the  
**EXPLANATION** and then  
we designed the  
**ALGORITHM**.

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

# Algorithmic decision-making

## Transparency -----



# 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, naïve-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<sup>2</sup>M<sub>s</sub>) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy.<sup>1</sup> 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 [3]. 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 regression 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 "HasAsthma(x)  $\Rightarrow$  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 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 —

1SVMs 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.  
Copyright is held by the owner/author(s). Publication rights licensed to ACM.  
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.

The models 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, naïve-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<sup>2</sup>M<sub>s</sub>) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy.<sup>1</sup> 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 [4]. 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 regression 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 "HasAsthma(x)  $\Rightarrow$  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 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 —

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

1721

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, naïve-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<sup>2</sup>M<sub>s</sub>) are applied to real healthcare problems yielding intelligible models with state-of-the-art accuracy.<sup>1</sup> 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 [3]. 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 regression 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 "HasAsthma(x)  $\Rightarrow$  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 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 —

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

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

Paul Koch  
Microsoft Research  
paulkoch@microsoft.com

Yin Lou  
LinkedIn Corporation  
ylou@linkedin.com

Marc Sturm  
NewYork-Presbyterian Hospital  
mas916@nyp.org

Johannes Gehrke  
Microsoft  
johannes@microsoft.com

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<sup>2</sup>Ms) 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

Persons to make difficult health care decisions of all types of this work for the general public. The project was not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

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

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3664-2/15/08 \$15.00.

DOI: <http://dx.doi.org/10.1145/2783258.2788613>.

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 to use on real patients and logistic regression 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 “HasAsthma(x)  $\Rightarrow$  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 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 —

<sup>1</sup>SVMs and boosted trees were not in common use yet, and Random Forests had not yet been invented.

1721

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’.

# What is an explanation?

An explanation is a **statement** that provides an account of the **reasons** for something.

When someone asks you for an explanation, they are looking for an answer to **why** something happened or occurred.

But there are limits...

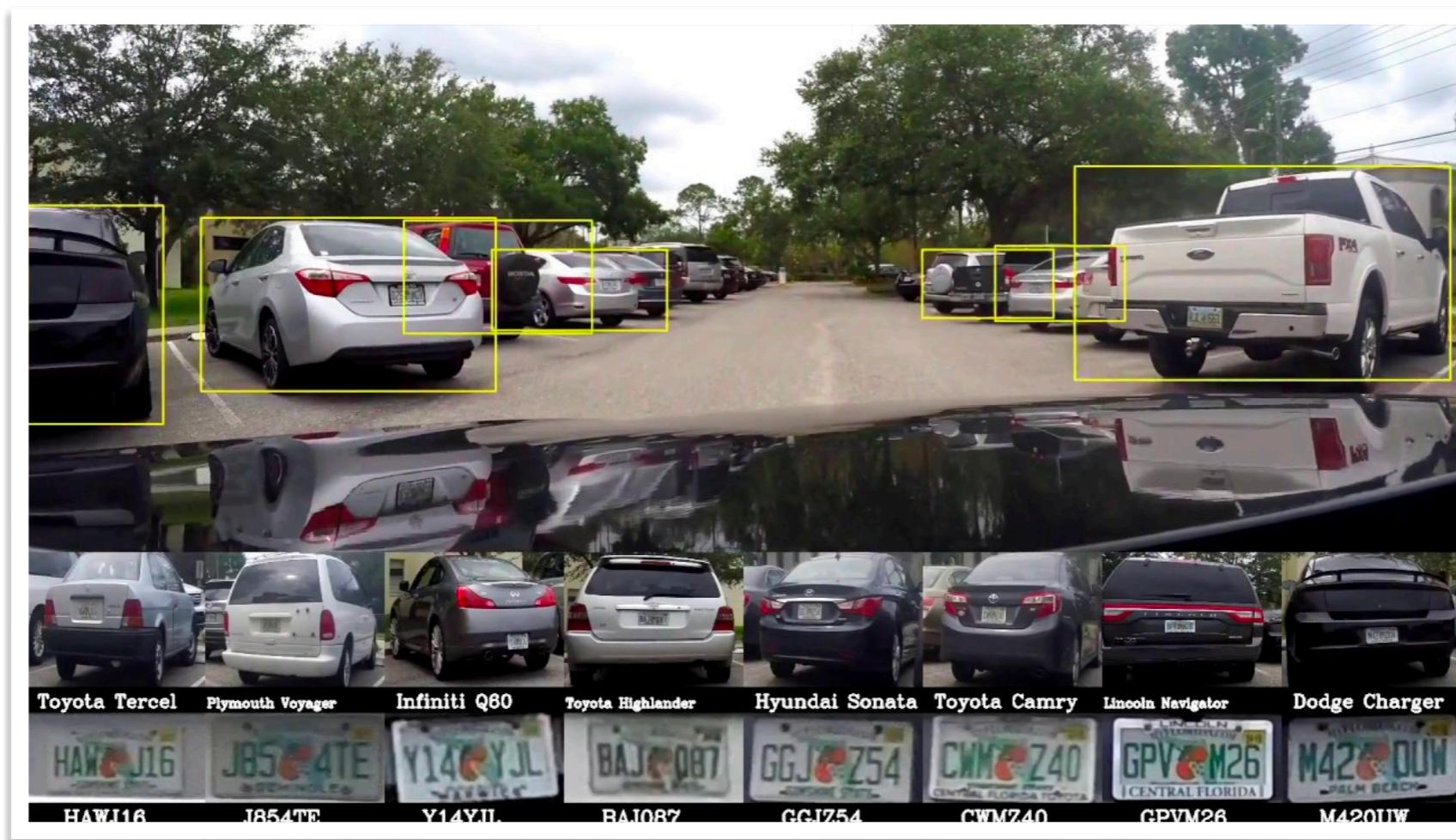
**WATCH THIS  
VIDEO!**



Richard Feynman about “explanations” and “why questions”.  
<https://www.youtube.com/watch?v=Q1lL-hXO27Q>

# The need for explainability

We don't need interpretability if the model has no significant impact, the problem is well studied (f.e. OCR) or if this enable people to manipulate or to game a critical system.



# The human factor

# The need for explainability

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

- **Human curiosity and learning:** Humans have a mental model of their environment that is updated when something unexpected happens. This update is performed by finding an explanation for the unexpected event.
- 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 **ethical debugging tool**.
- The process of integrating machines and algorithms into our daily lives requires interpretability to increase **social acceptance and trust**.

# Explanations...

This 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...

This 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.

**RELIABILITY, GLOBAL/LOCAL INTERPRETABILITY**

- 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.

**TRUST, GLOBAL INTERPRETABILITY**

- For society broadly to understand and become comfortable with the strengths and limitations of the system, overcoming a reasonable fear of the unknown.

**TRUST, GLOBAL INTERPRETABILITY**

- 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).

**TRUST, LOCAL INTERPRETABILITY**

- 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).

**RELIABILITY, GLOBAL/LOCAL INTERPRETABILITY**

- Etc.

GLOBAL: general understanding of how an overall system works  
LOCAL: an explanation of a particular prediction or decision

# Transparency

A system with explainability capabilities is said to be more **transparent**.

As we have seen, there are many types of transparency with different motivations. Each case may need a different way to measure it.

Actors with misaligned interests can **abuse transparency as a manipulation channel**, or inappropriately use information gained.

More transparency can also lead to **less efficiency**.

# 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.

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. This is difficult to integrate into machine learning!
- **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: the ethical perspective

**It is **unfair** that we can receive (from a person/algorithm) a low credit score, end up on a police watch list, get higher prison sentences, etc. without explanation about the considerations that led to those decisions.**

Getting algorithms to provide us with explanations about how a particular decision was made allows us to keep ‘meaningful human control’ over the decision.

**A principle of explicability**, then, is an ethical principle that should help bring us closer to **acceptable uses of algorithms**.

# Explanations: the ethical perspective

European General Data Protection Regulation (GDPR) includes an indirect ‘right to explanation’ when fully automated decisions significantly affect someone:

“the data controller shall implement suitable measures to safeguard the data subject's rights and freedoms and legitimate interests, at least the right to obtain human intervention on the part of the controller, to express his or her point of view and **to contest the decision**”.

But this is a high level vision...

# 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 most important factor when it comes to getting a loan.”*

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.”*

# Feature Importance

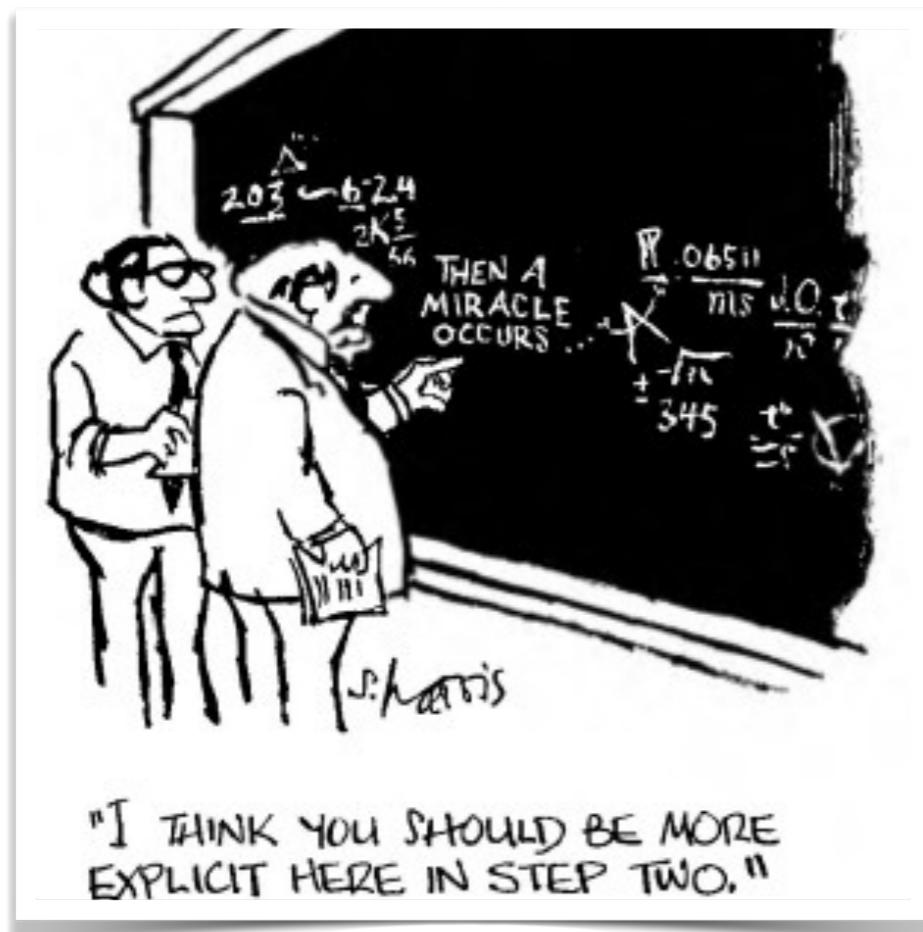
There are at least 3 notions of feature importance:

- 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,  $\beta_j$  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



**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 (properly normalized) feature inputs:

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

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

We have to measure the uncertainty of the parameter!

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

# Explanations & Interpretable models

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

**An increase of feature  $x_i$  (when all other feature values remain fixed) by one unit increases the value of the outcome  $y$  by  $\beta_i$  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 **multicollinearity effect** in linear models:

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

Example: years of education and annual income

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

**Extreme case:**

Suppose we have two variables,  $x_1, x_2$  that are perfectly 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.

If  $y = \beta_0 + \hat{\beta}_1 x_1$  is a good solution,  $y = \beta_0 + \hat{\beta}_2 x_2$  can also be a good solution.

# Explanations & Interpretable models

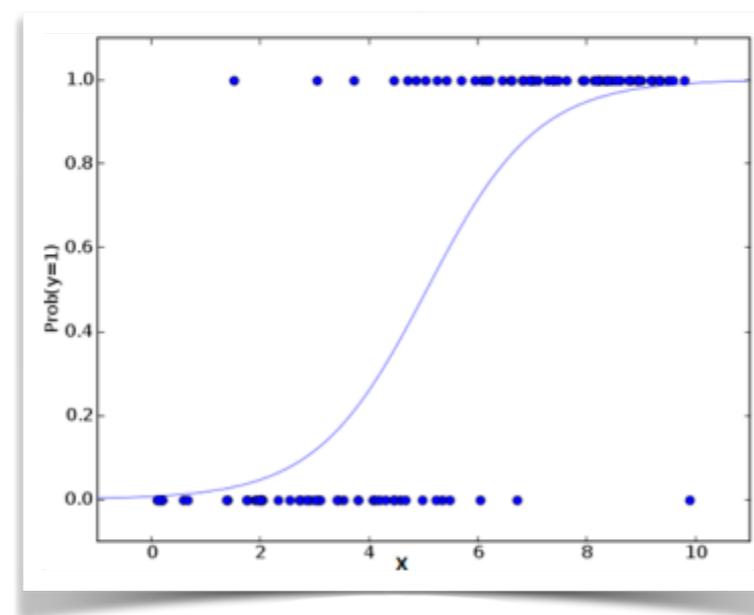
This interpretation of the linear model is “correct” from the point of view of the **behavior of the model** (designer POV), but it is “not intuitive” from the point of the view of the phenomena we are explaining (user POV)!

- The model is too dependent of **variable associations**.
- 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.

# Explanations & Interpretable models

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

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



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

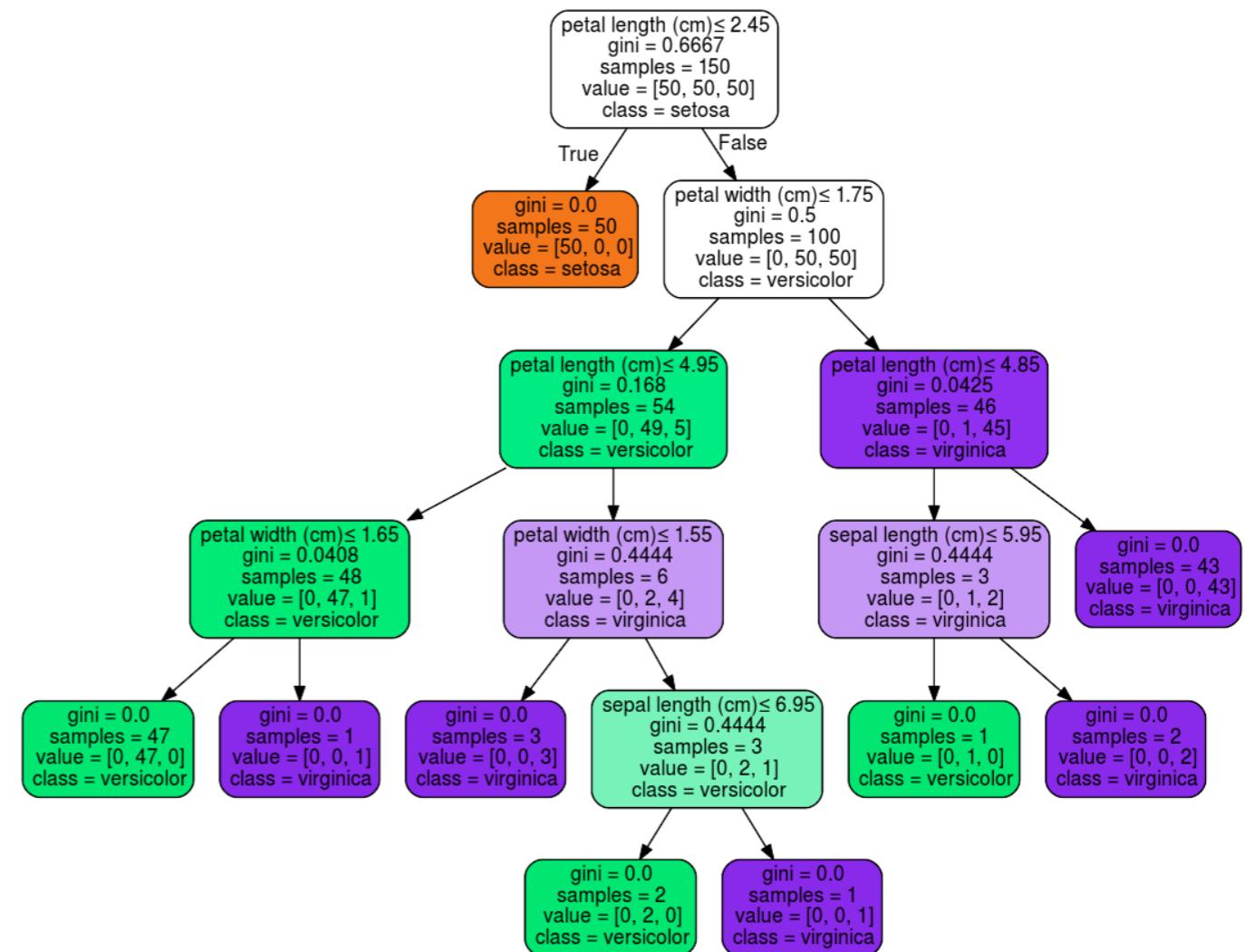
# Explanations & Interpretable models

## Decision Trees

**How do we measure the quality of a node?**

In the case of classification tasks, the degree of the Gini index varies between 0 and 1, where 0 expresses the purity of classification, and 1 indicates the random distribution of elements across various 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.

We are measuring the importance of  $X_j$  by its **contribution** to predictive accuracy.

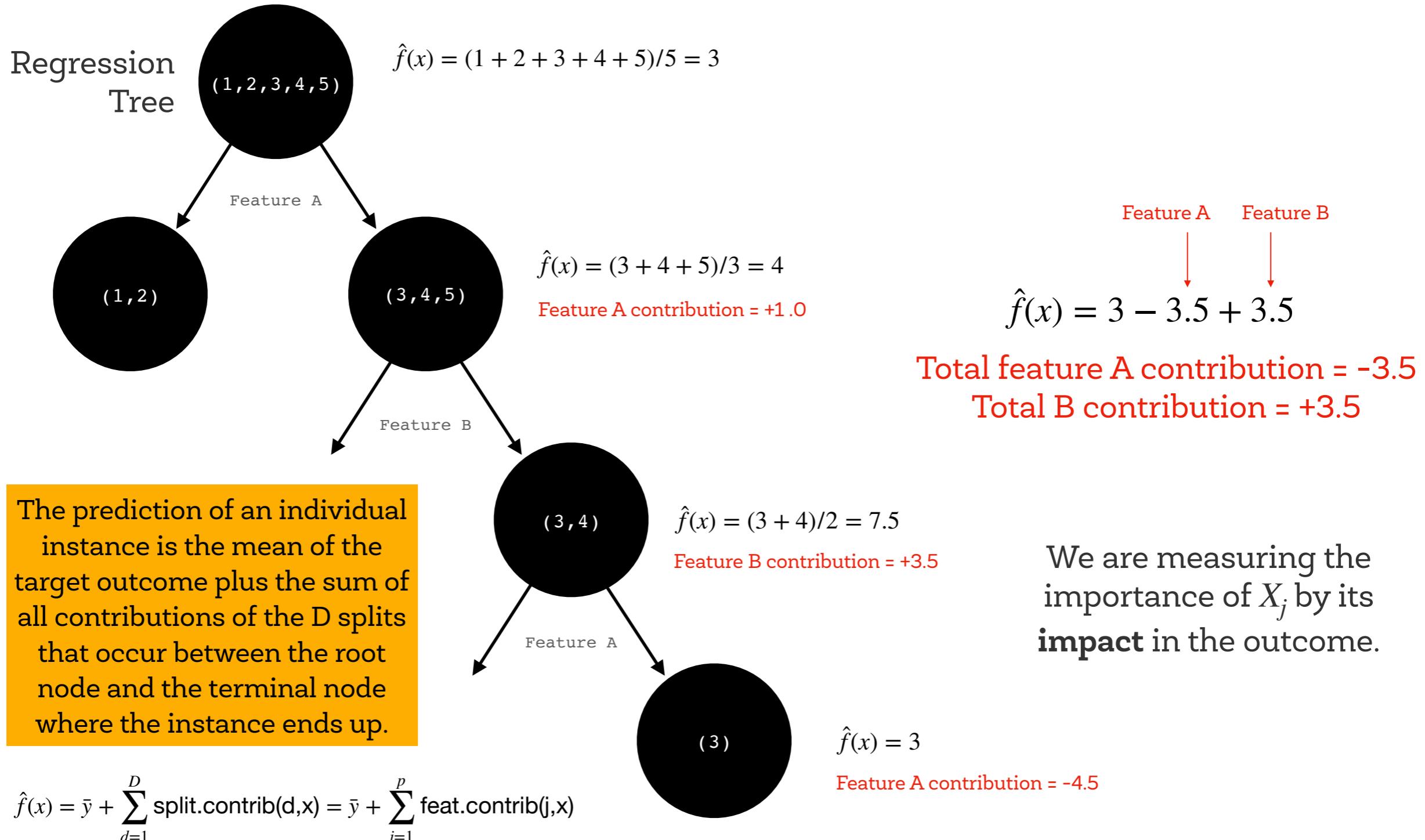
# Explanations & Interpretable models

## Decision Trees

**Individual predictions** of a decision tree can be explained by decomposing the decision path.

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

# Explanations & Interpretable models



# Explanations & Interpretable models

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

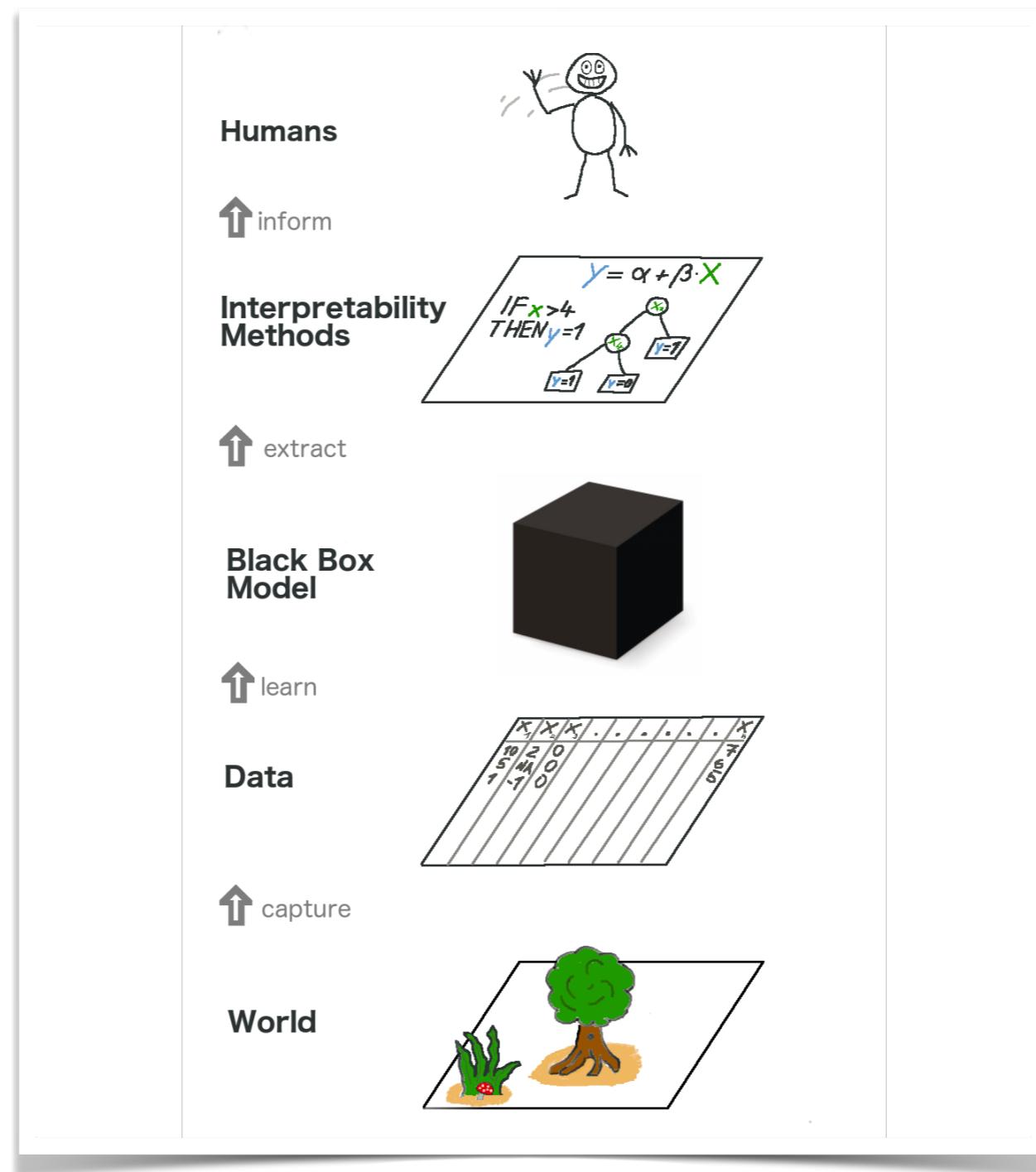
- Generalized Linear Models and Generalized Additive Models
- Naive Bayes
- k-nearest neighbors
- Etc.

But this is **mechanistic explanation** of the behavior of the model. Is this the kind of 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?

# Black-Box Explanations

# Model-agnostic explanation models



# Model-agnostic explanation models

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 models

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 models

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 models

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

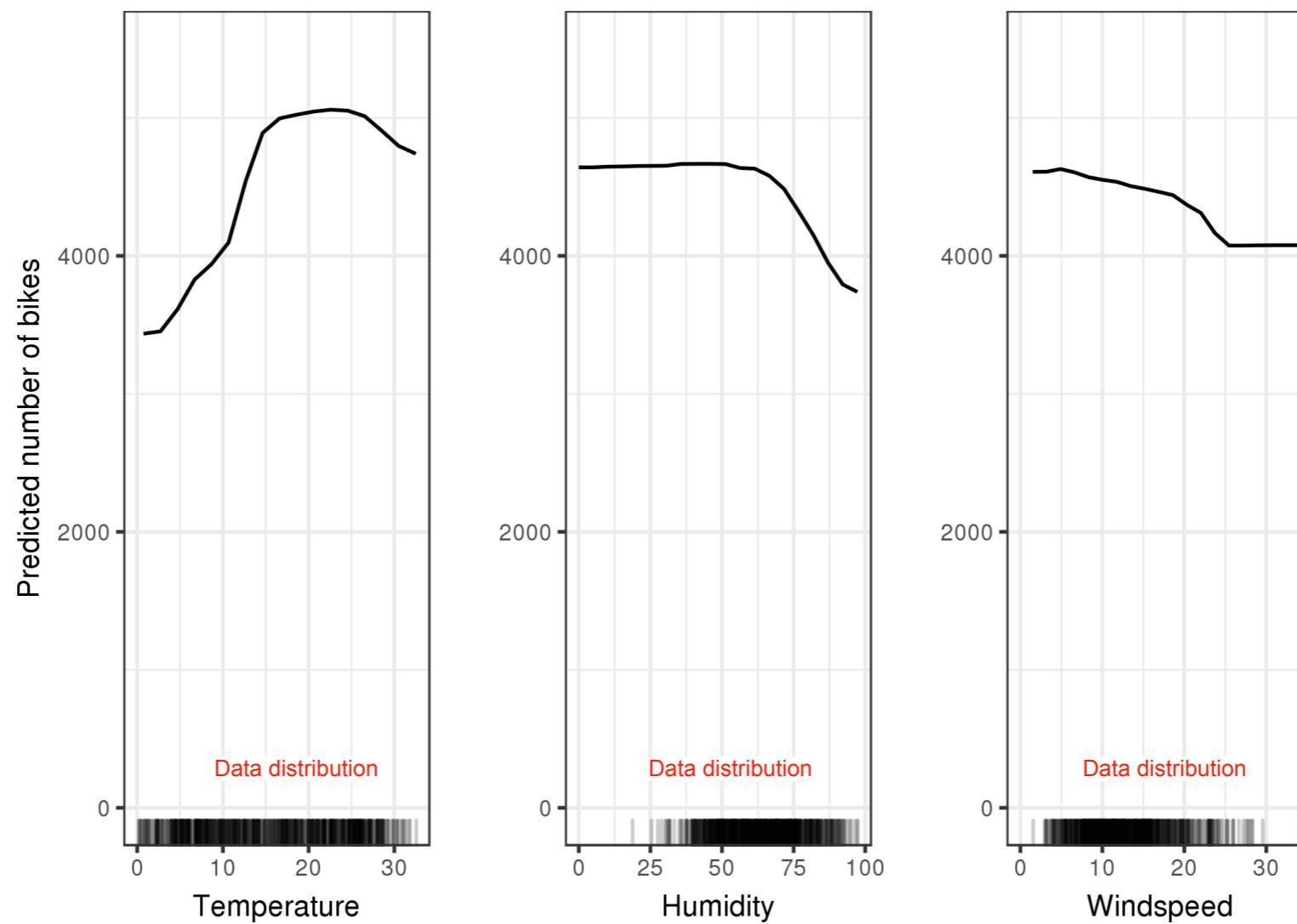
A	B	C	Y
a1	b1	c1	yA1
a1	b2	c2	
a1	b3	c3	
a2	b1	c1	yA2
a2	b2	c2	
a2	b3	c3	
a3	b1	c1	yA3
a3	b2	c2	
a3	b3	c3	

Finally, it plots out the average predictions.

X	A1	A2	A3
Y	yA1	yA2	yA3

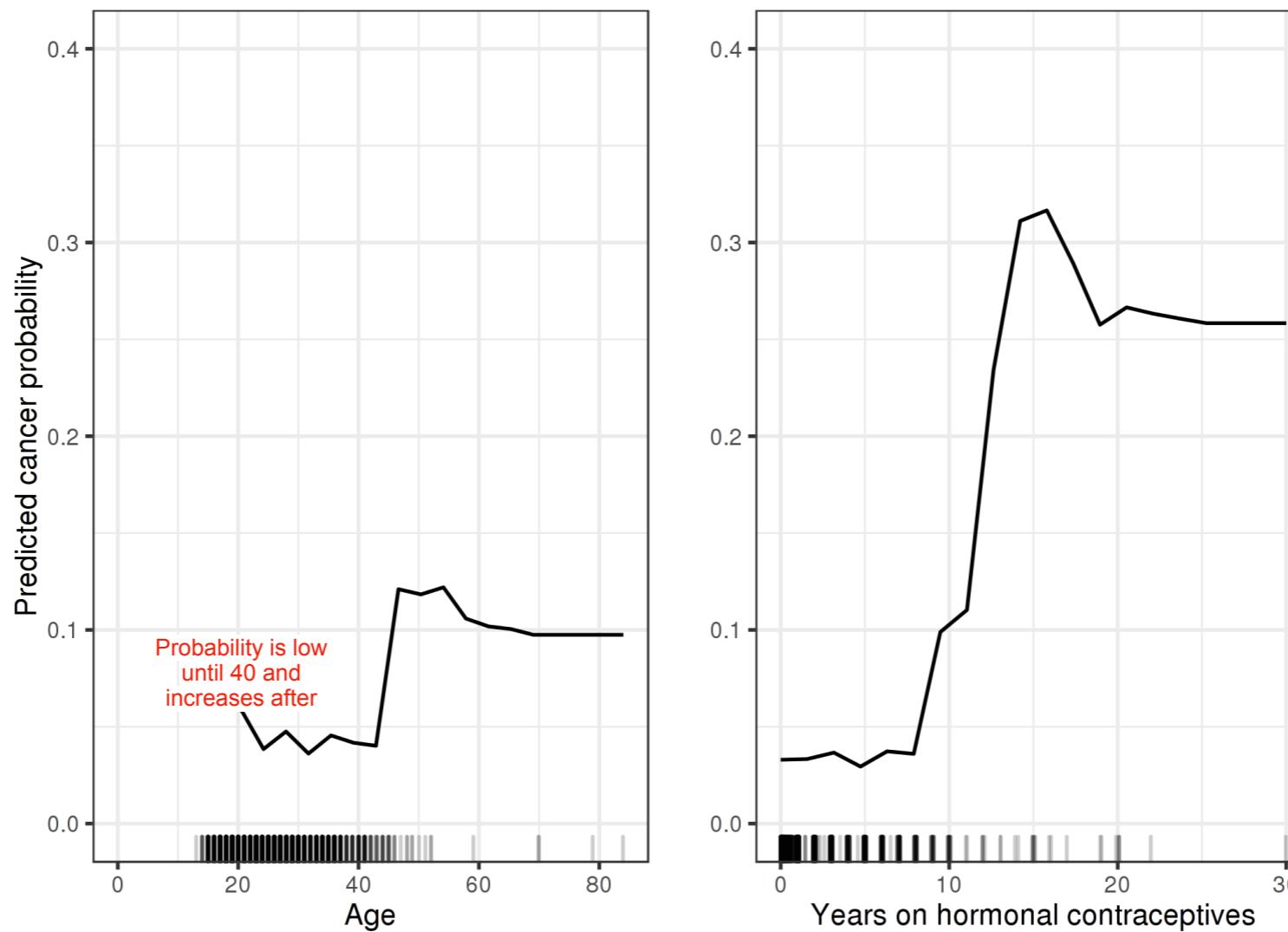
# Model-agnostic explanation models

**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.



# Model-agnostic explanation models

**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 models

**Solve this problem!**



Partial Dependence Plots  
Notebook

# Model-agnostic explanation models

## 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 models

## Permutation Test

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

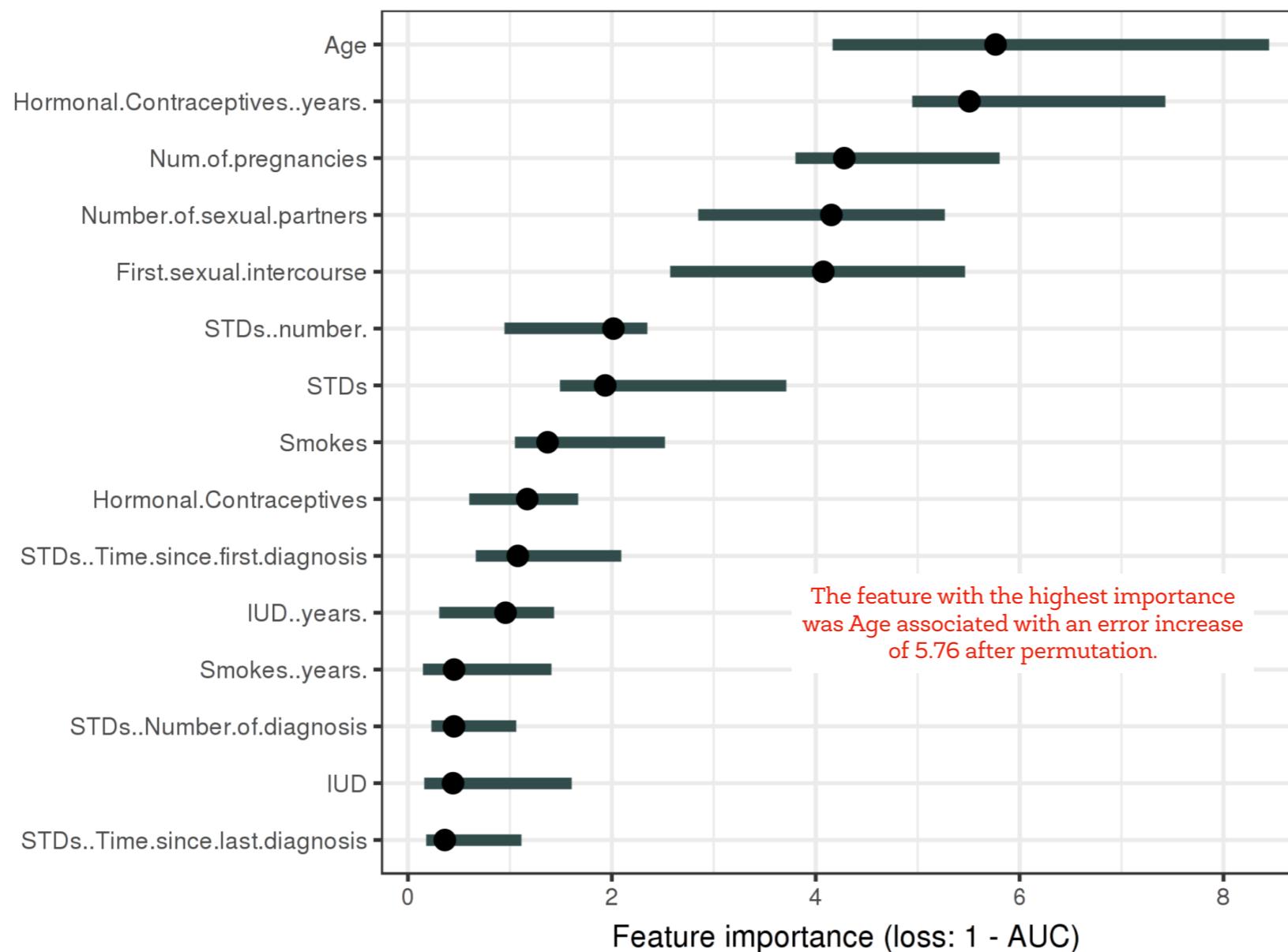
1. Estimate the original model error  $e^{\text{orig}} = L(y, f(X))$  (e.g. mean squared error)
2. For each feature  $j = 1, \dots, p$  do:
  - Generate feature matrix  $X^{\text{perm}}$  by permuting feature  $j$  in the data  $X$ . This breaks the association between feature  $j$  and true outcome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

## Permutation Test

**Problem:** Predict cervical cancer.



# Model-agnostic explanation models

Solve this problem!

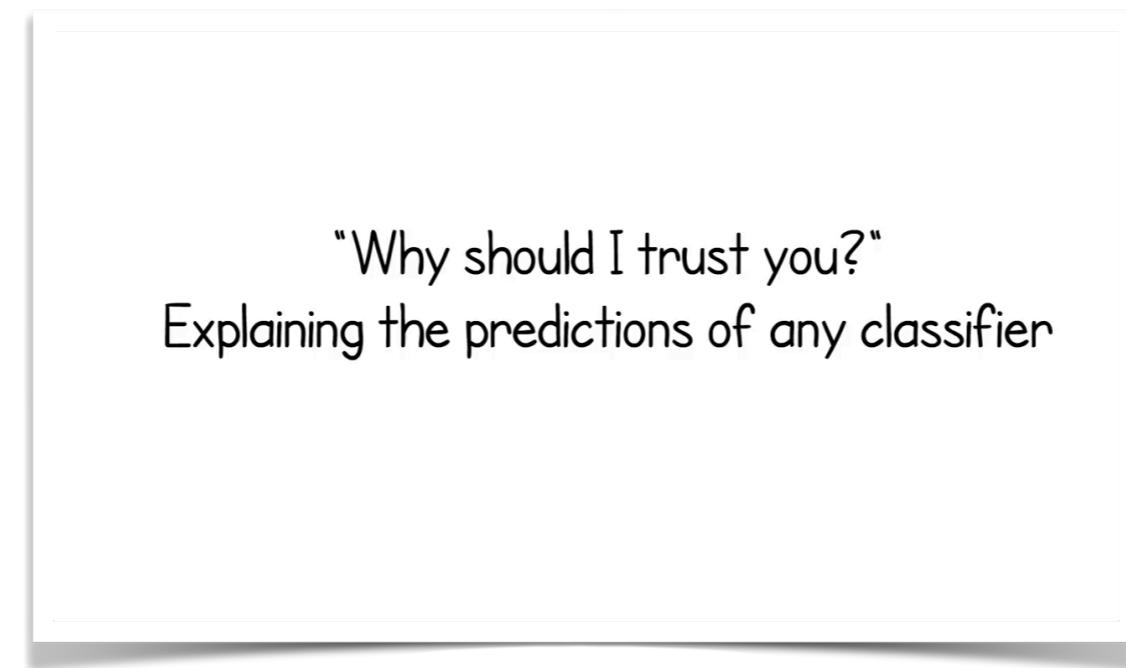


Permutation Test Notebook

# Model-agnostic explanation models

**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 models

**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 models

**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 models

**LIME:** Local interpretable model-agnostic explanations.

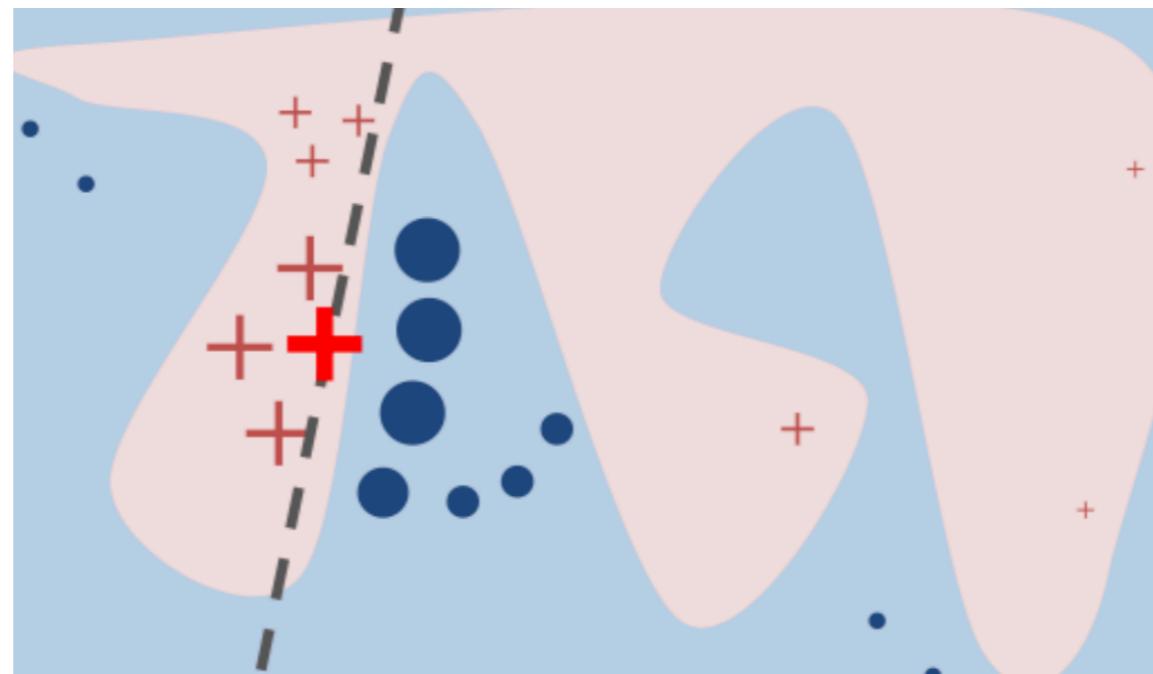
The black-box model's complex decision function  $f$  (unknown to LIME) is represented by the blue/pink background.

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 models



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

# Model-agnostic explanation models

**LIME:** Local interpretable model-agnostic explanations.

Local Surrogate (LIME) for **text**

Starting from the original text, new texts are created by randomly removing words from the original text.

The dataset is represented with binary features for each word. A feature is 1 if the corresponding word is included and 0 if it has been removed.

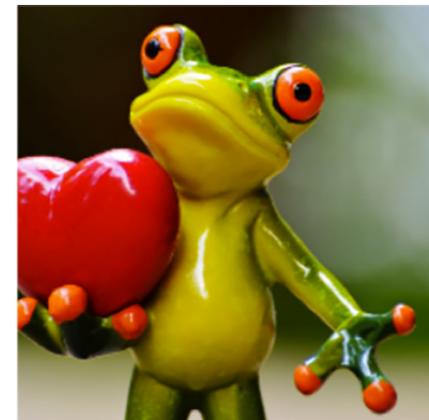
# Model-agnostic explanation models

**LIME:** Local interpretable model-agnostic explanations.

Local Surrogate (LIME) for images

Intuitively, it would not make much sense to perturb individual pixels, since many more than one pixel contribute to one class. Randomly changing individual pixels would probably not change the predictions by much. Therefore, variations of the images are created by segmenting the image into “superpixels” and turning superpixels off or on.

# Model-agnostic explanation models



Original Image

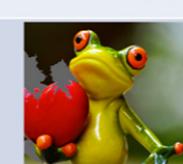


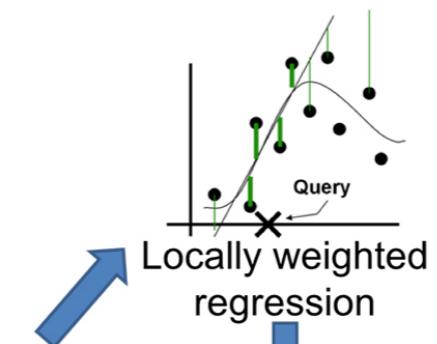
Interpretable Components



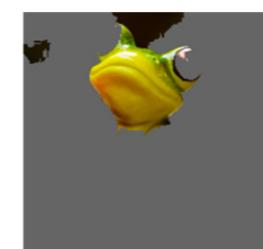
Original Image  
 $P(\text{tree frog}) = 0.54$



Perturbed Instances	$P(\text{tree frog})$
	0.85
	0.00001
	0.52



Locally weighted regression



Explanation

# Actionable Explanations

# GDPR Scenario

GDPR grants the data subject the right to know  
“the purposes of the processing for which the  
personal data are intended (...), the existence of  
automated decision-making, including  
profiling, (...) and, at least in those cases,  
**meaningful information about the logic  
involved**“.

**Proposal:** If a negative decision is made about an individual (denying a loan, rejecting an application for housing, and so on), she must be able to ask **how we might change circumstances** to get a favorable decision the next time.

**Counterfactual** explanations deal with the question: *how should the features change to obtain a different outcome?*

One could explain a credit rejection by saying:  
*‘Had you earned \$5,000 more, your request for credit would have been approved.’*

# Counterfactual Explanations

# Counterfactuals

How can we provide **meaningful** explanations to engender trust and to empower “users” in the algorithmic decision process?

How can we repair the **probing** mechanism to generate meaningful explanations?

Which information can we provide/use?

- Source data?
- Input data?
- Features?
- Uncertainty?
- Etc.

# Counterfactuals

*Harvard Journal of Law & Technology*  
Volume 31, Number 2 Spring 2018

## COUNTERFACTUAL EXPLANATIONS WITHOUT OPENING THE BLACK BOX: AUTOMATED DECISIONS AND THE GDPR

*Sandra Wachter,\* Brent Mittelstadt, \*\* & Chris Russell\*\*\**

### TABLE OF CONTENTS

I. INTRODUCTION .....	842
II. COUNTERFACTUALS.....	844
A. <i>Historic Context and the Problem of Knowledge</i> .....	846
B. <i>Explanations in A.I. and Machine Learning</i> .....	849
C. <i>Adversarial Perturbations and Counterfactual Explanations</i> .....	851
D. <i>Causality and Fairness</i> .....	853
III. GENERATING COUNTERFACTUALS .....	854
A. <i>LSAT Dataset</i> .....	856
B. <i>Pima Diabetes Database</i> .....	859
C. <i>Causal Assumptions and Counterfactual Explanations</i> .....	859
IV. ADVANTAGES OF COUNTERFACTUAL EXPLANATIONS .....	860
V. COUNTERFACTUAL EXPLANATIONS AND THE GDPR .....	861

# Counterfactuals

A counterfactual explanation describes a causal situation in the form:

Score  $s$  was returned because some set of variables  $\mathbf{V} = \{V_i\}$  had values  $\{v_i\}$ . If  $\mathbf{V}$  instead had variables  $\{v'_i\}$ , score  $s'$  would have been returned.

Proposed counterfactual explanation in (Wachter, 2018):

Score  $s$  was returned because some set of variables  $\mathbf{V} = \{V_i\}$  had values  $\{v_i\}$ . If  $\mathbf{V}$  instead had variables  $\{v'_i\}$  and all other variables remain constant, score  $s'$  would have been returned.

# Counterfactuals

A counterfactual explanation describes a causal situation in the form:

Score  $s$  was returned because some set of variables  $\mathbf{V} = \{V_i\}$  had values  $\{v_i\}$ . If  $\mathbf{V}$  instead had variables  $\{v'_i\}$ , score  $s'$  would have been returned.

Proposed counterfactual explanation in (Wachter, 2018):

Score  $s$  was returned because some set of variables  $\mathbf{V} = \{V_i\}$  had values  $\{v_i\}$ . If  $\mathbf{V}$  instead had variables  $\{v'_i\}$  and all other variables remain constant, score  $s'$  would have been returned.

This is not a causal counterfactual!

# Counterfactuals



Counterfactuals are human-friendly explanations, because they are contrastive to the current instance and because they are selective, meaning they usually focus on a small number of feature changes.

But counterfactuals suffer from the 'Rashomon effect': there are usually multiple different counterfactual explanations.

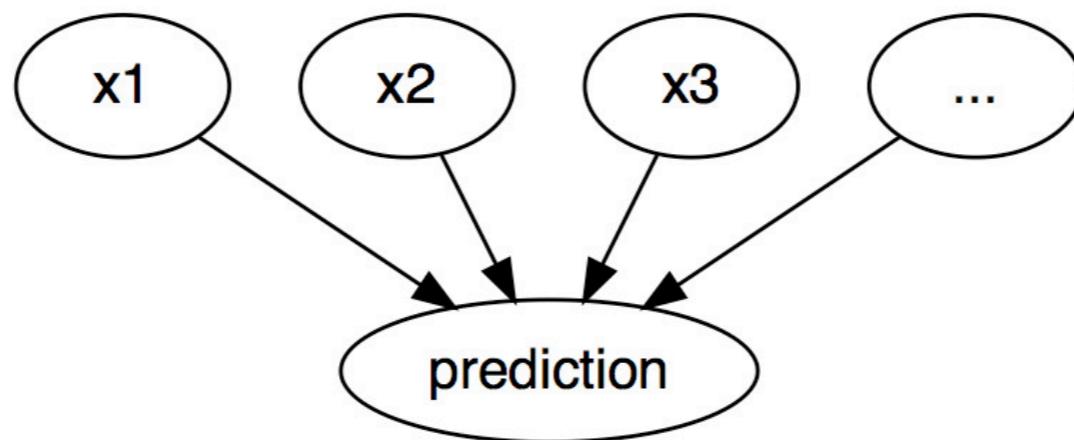
Japanese filmmaker Akira Kurosawa directed the 1950 film *Rashomon*, in which four different people provide contradictory accounts of a samurai's murder, despite all having witnessed the crime.

The *Rashomon* effect refers to an instance when the same event is described in significantly different (often contradictory) ways by different people who were involved.

Leo Breiman used this concept to describe the fact that there is often a multitude of different descriptions in a class of functions giving about the same minimum error rate. [L.Breiman, **Statistical Modeling: The Two Cultures**, Statistical Science 2001, Vol. 16, No. 3, 199–231]

# Counterfactuals

In machine learning, **counterfactual** explanations can be used to explain predictions of individual instances. The "event" is the predicted outcome of an instance, the "causes" are the particular feature values of this instance that were input to the model and "caused" a certain prediction.



<https://christophm.github.io/interpretable-ml-book/counterfactual.html>

Given this simple graph, it is easy to see how we can simulate counterfactuals for predictions of machine learning models: We simply change the feature values of an instance before making the predictions and we analyze how the prediction changes.

# Counterfactuals

Explanation process:

1. The user of a counterfactual explanation defines an alternative change in the prediction of an instance (“loan denied to loan approved”).
2. We look for a counterfactual instance (**minimal alternative set of input values**) that produce the defined outcome. In order to be a counterfactual, the new values should be as **similar** as possible to the original ones. Another important requirement is that a counterfactual instance should have feature values that are **likely**.
3. Sometimes it is useful to generate **multiple, diverse** counterfactuals.
4. We report a **local** explanation.

# Counterfactuals

Wachter (2018) suggest minimizing the following loss:

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

$(\hat{f}(x') - y')^2$  is the quadratic distance between the model prediction for the counterfactual  $x'$  and the desired outcome  $y'$ , which the user must define in advance.

$d(x, x')$  is the distance between the instance  $x$  to be explained and the counterfactual  $x'$ . We can consider different distances  $L_1, L_2 \dots$

The loss measures how far the predicted outcome of the counterfactual is from the predefined outcome and how far the counterfactual is from the instance of interest.

The parameter  $\lambda$  balances the distance in prediction (first term) against the distance in feature values (second term). The loss is solved for a given  $\lambda$  and returns a counterfactual. A higher value of  $\lambda$  means that we prefer counterfactuals with predictions close to the desired outcome, a lower value means that we prefer counterfactuals that are very similar to  $x$  in the feature values. If  $\lambda$  is very large, the instance with the prediction closest to  $y'$  will be selected, regardless how far it is away from  $x$ .

# Counterfactuals

The proposed method has some disadvantages:

- It does not take into account the criterion: "*produce counterfactuals with only a few feature changes and likely feature values*".
- The method does not handle **categorical** features with many different levels well.
- Counterfactuals are a causal concept, but the method is not based on causal inference. It can use **non-meaningful counterfactuals**.

# Causal Counterfactuals

One could explain a credit rejection by saying: ‘Had you earned \$5,000 more, your request for credit would have been approved.’

But explanations must capture **real** patterns in the input data.

The argument is that explanations are used not just to understand the model at hand but also to **extract relationships underlying the phenomena being modeled**.

This is especially true for scenarios where the goal is not just to predict different outcomes but also reveal the rules governing those outcomes.

# Causal Counterfactuals

Causal counterfactual queries are the base of good explanations!

*“If your data had looked like this, you would have given this score instead.”*

The problem is to find a (set of) similar (but diverse) counterfactual(s) that is(/are) both close to the actual datapoint and likely to occur in the real world.



# Resources

# Interpretable Machine Learning

A Guide for Making  
Black Box Models Explainable



@ChristophMolnar

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