UNIVERSITAT DE BARCELONA

MSc in Fundamental Principles of Data Science

Ethical Data Science

Transparency & Explainability

Jordi Vitrià

2020-2021

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

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

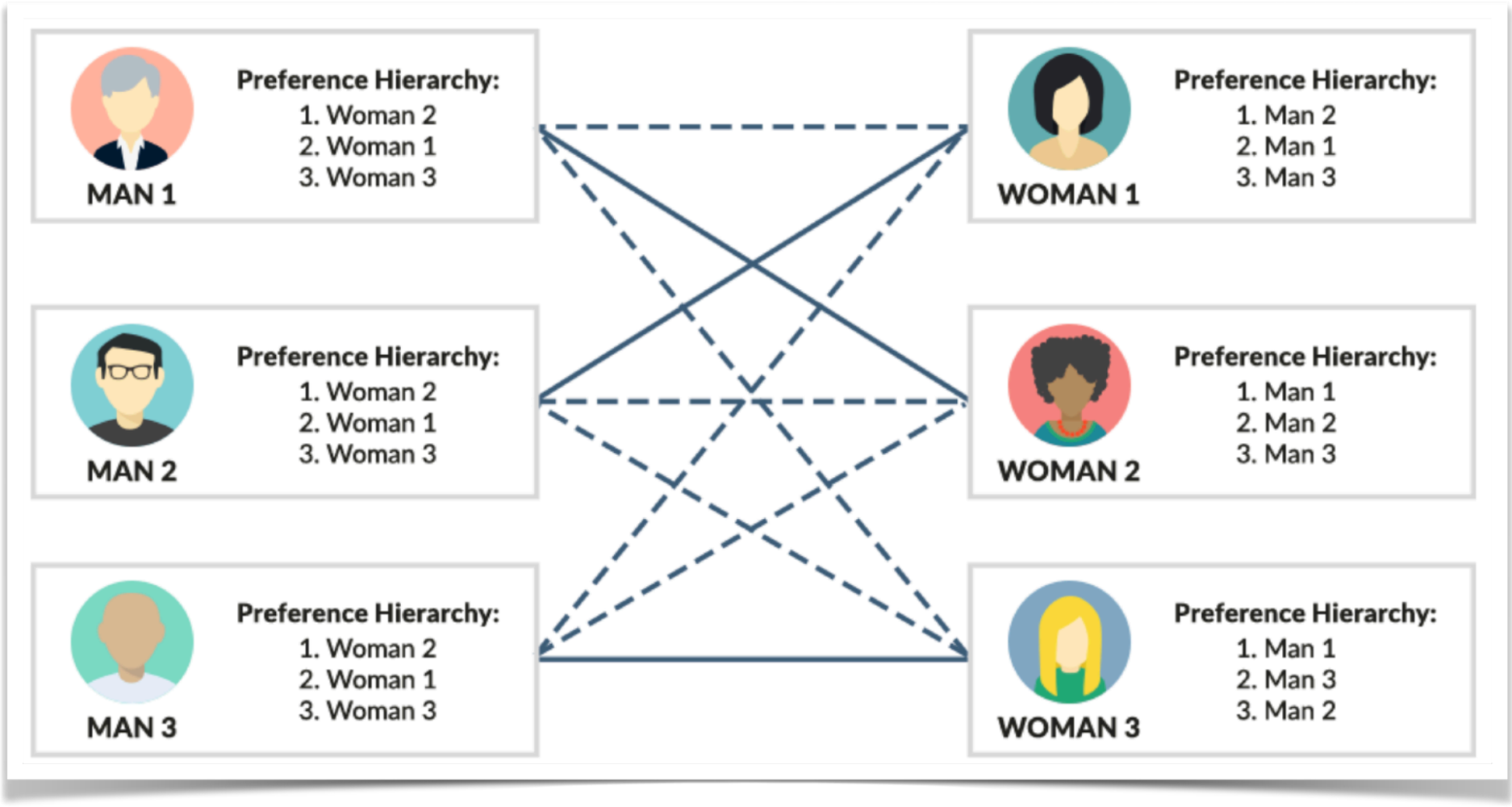
La répartition des étudiants entre les universités et les filières est un problème complexe puisqu'elle s'effectue sur base d'un conflit massif entre l'offre et la demande : on dénombre plus de 880.000 candidats pour un total (à raison de 10 vœux possibles par candidat) de quelques 7.000.000 de vœux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

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 Parcoursup, we first defined the EXPLANATION and then we designed the ALGORITHM.

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

Training Data →

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?

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**.

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, 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 .

1721

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.

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.

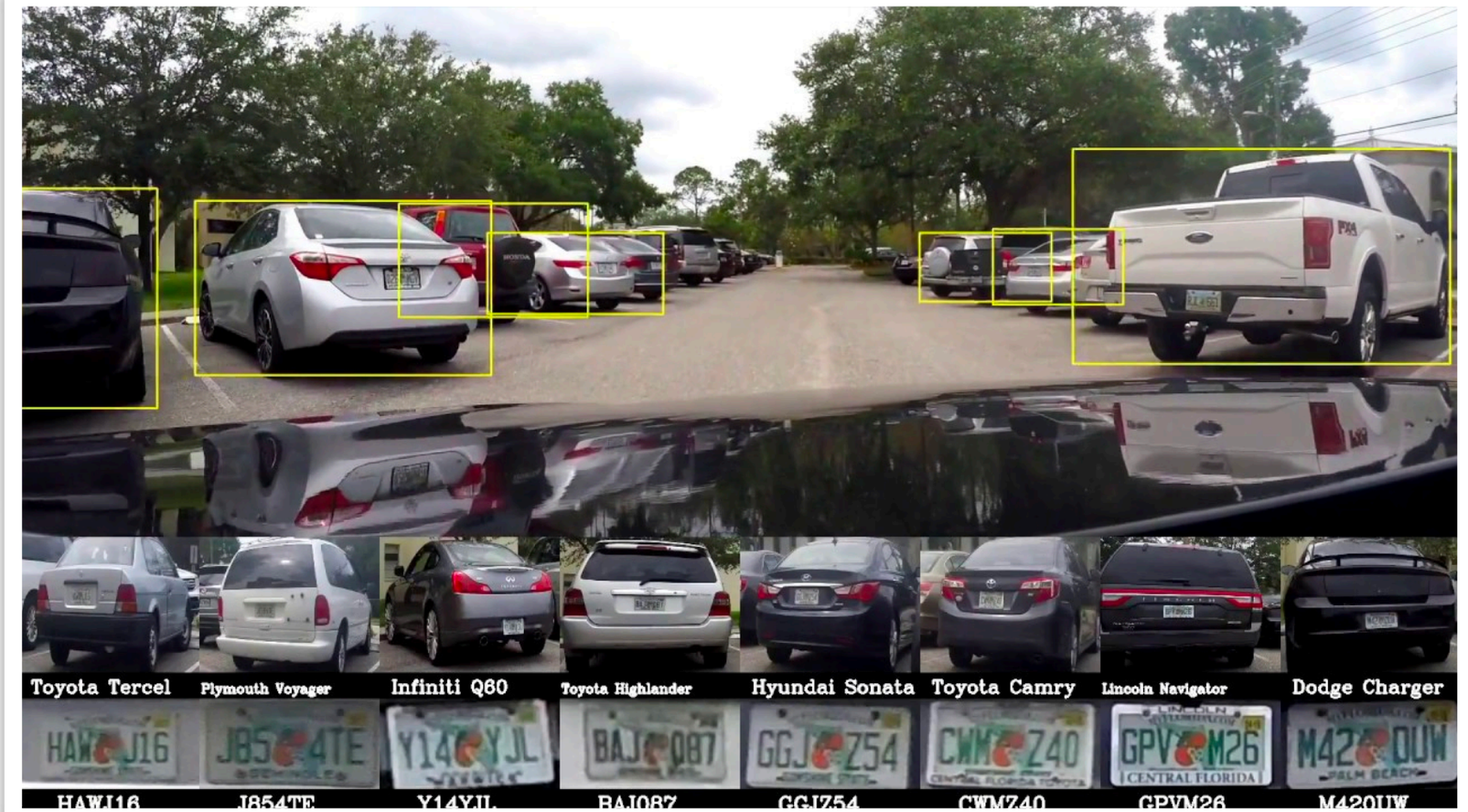
ACM 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2788613 .

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

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

WATCH THIS VIDEO!

Richard Feynman about “explanations” and “why questions”.

https://www.youtube.com/watch?v=Q1lL-hX027Q

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.

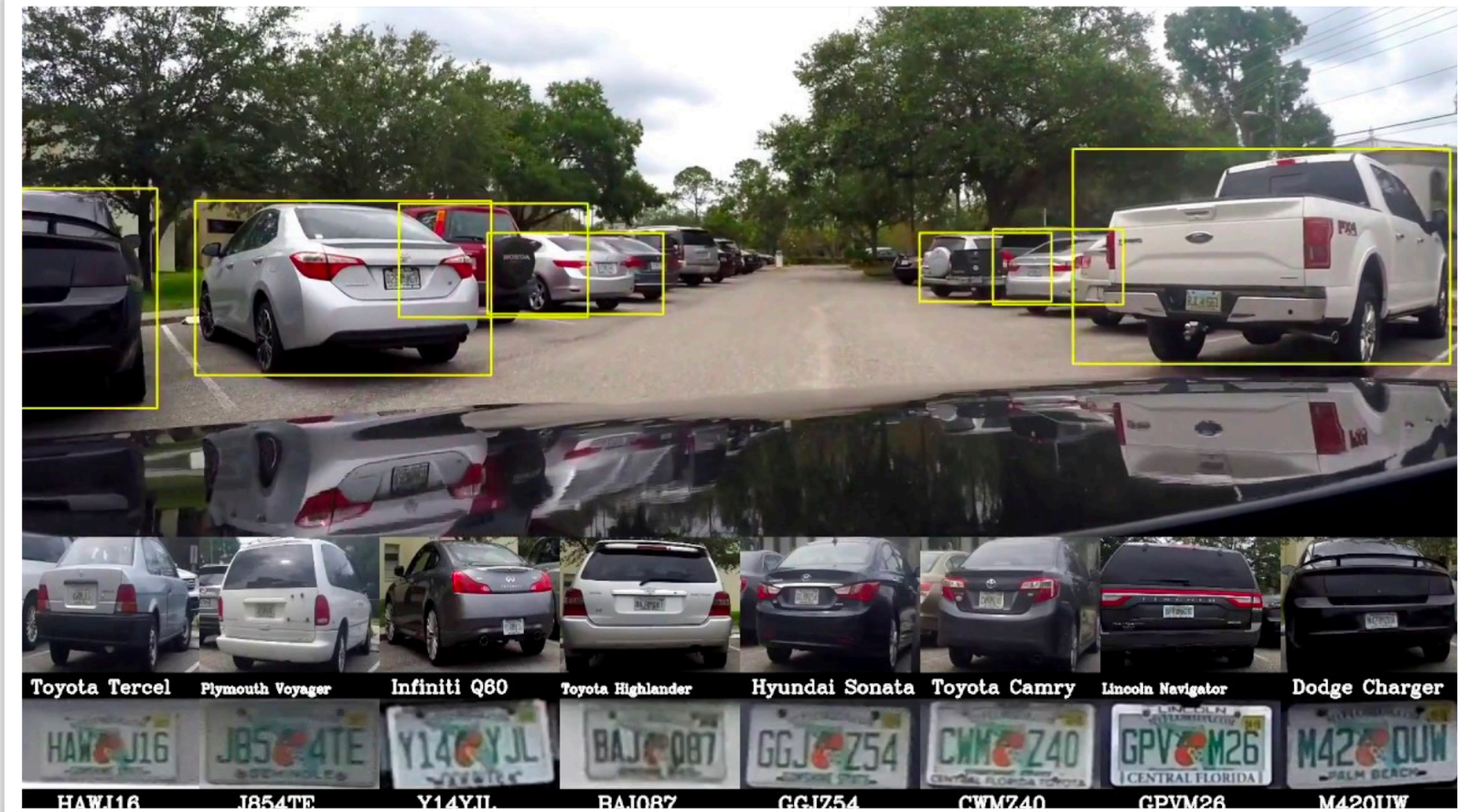
- The goal of science is **to gain knowledge**, but many problems are solved with big datasets and black box machine learning models. The model itself becomes the source of knowledge instead of the data. Interpretability makes it possible to extract this additional knowledge captured by the model.

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 **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

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.



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 can be used to manipulate

Paper: Langer, E., Blank, A., Chanowitz, B.: *The mindlessness of ostensibly thoughtful action: The role of "placebic" information in interpersonal interaction.* Journal of personality and social psychology, 36 (6), 635–642 (1978)

Authors of the paper arranged for researchers to try to jump in line to make a few photocopies at a busy library copy machine.

The researcher either (i) gave no explanation, asking simply "May I use the xerox machine?"; (ii) provided an 'empty' explanation: "May I use the xerox machine, because I have to make copies?"; or (iii) provided a 'real' explanation: "May I use the xerox machine, because I'm in a rush?"

The respective success rates were: (i) 60%; (ii) 93%; and (iii) 94%.

The startling conclusion was that saying "because something" seemed to work effectively to attain compliance, even if the 'something' had zero information content. Hence, a possible worry is that a deployer might provide an empty explanation as a psychological tool to soothe users.

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 a moral 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 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:

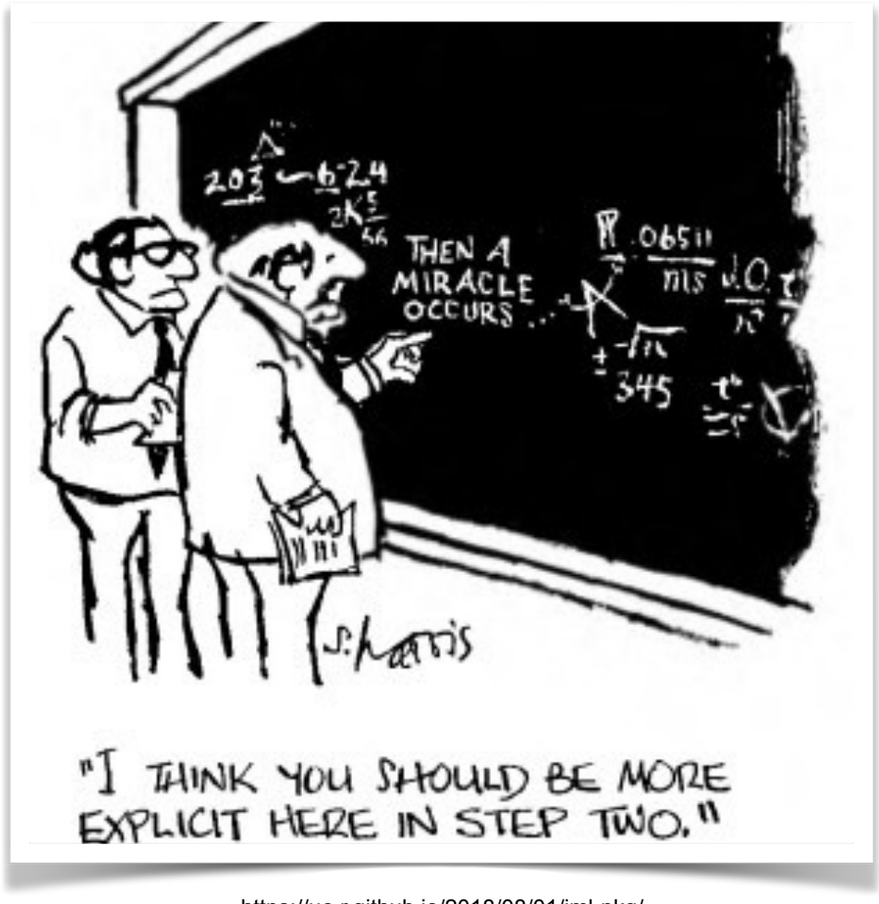
- To take the function $f(\mathbf{X})$ at its face value and ask which variable x_i has a big **impact** in $f(\mathbf{X})$.

  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 (properly normalized) 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!

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 (when all other feature values remain fixed) 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 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 perfecly 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.

$y = \beta_0 + \hat{\beta}_1 x_1$ is 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 **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 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.

Explanations & Fairness

The New York Times

YOUR MONEY ADVISER

In California, Gender Can No Longer Be Considered in Setting Car Insurance Rates

By Ann Carrns

Jan. 18, 2019

Minh Uong/The New York Times

California joined about a half-dozen states this month in banning the use of a person's gender when assessing risk factors for car insurance, a change that could potentially alter rates for scores of drivers across the state.

Delip Rao
@deliprao

Seguint

Unless there's a finite laundry list of all factors the rate is based on *and* if they are all perfectly decorrelated (orthogonal) to the gender factor, this just boils down to a PR and a non-solution. Even worse, it will be a non-solution that appears like a solution. #stats101

Aaron Roth @Aaroth
California bans differential pricing for car insurance based on gender, to "ensure that auto insurance rates are based on factors within a driver's control, rather than personal characteristics over which drivers have no control." nytimes.com/2019/01/18/you… 1/4

Mostra el fil

Tradueix el tuit

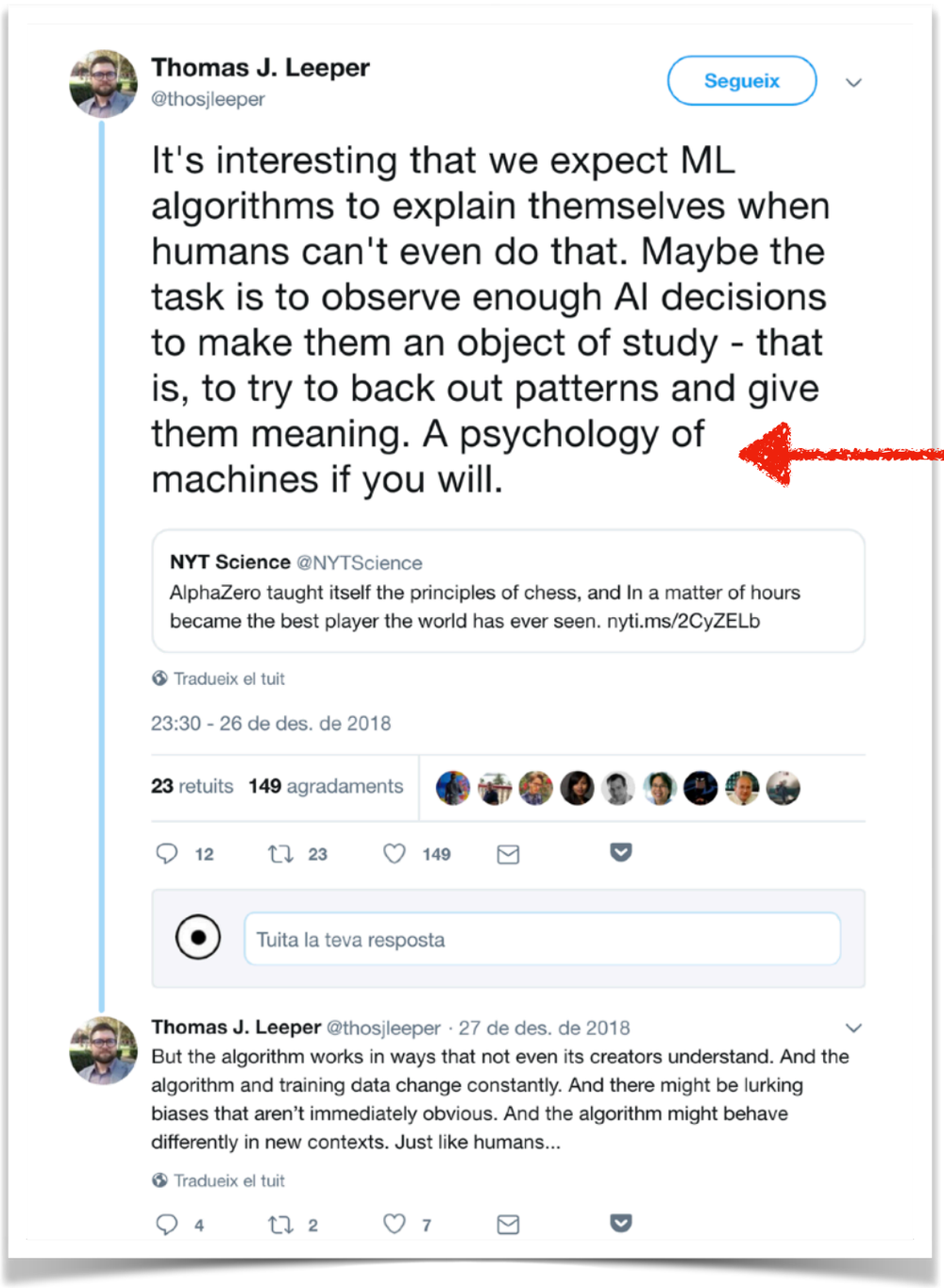
17:14 - 20 de gen. de 2019

3 retuits **12** agradaments

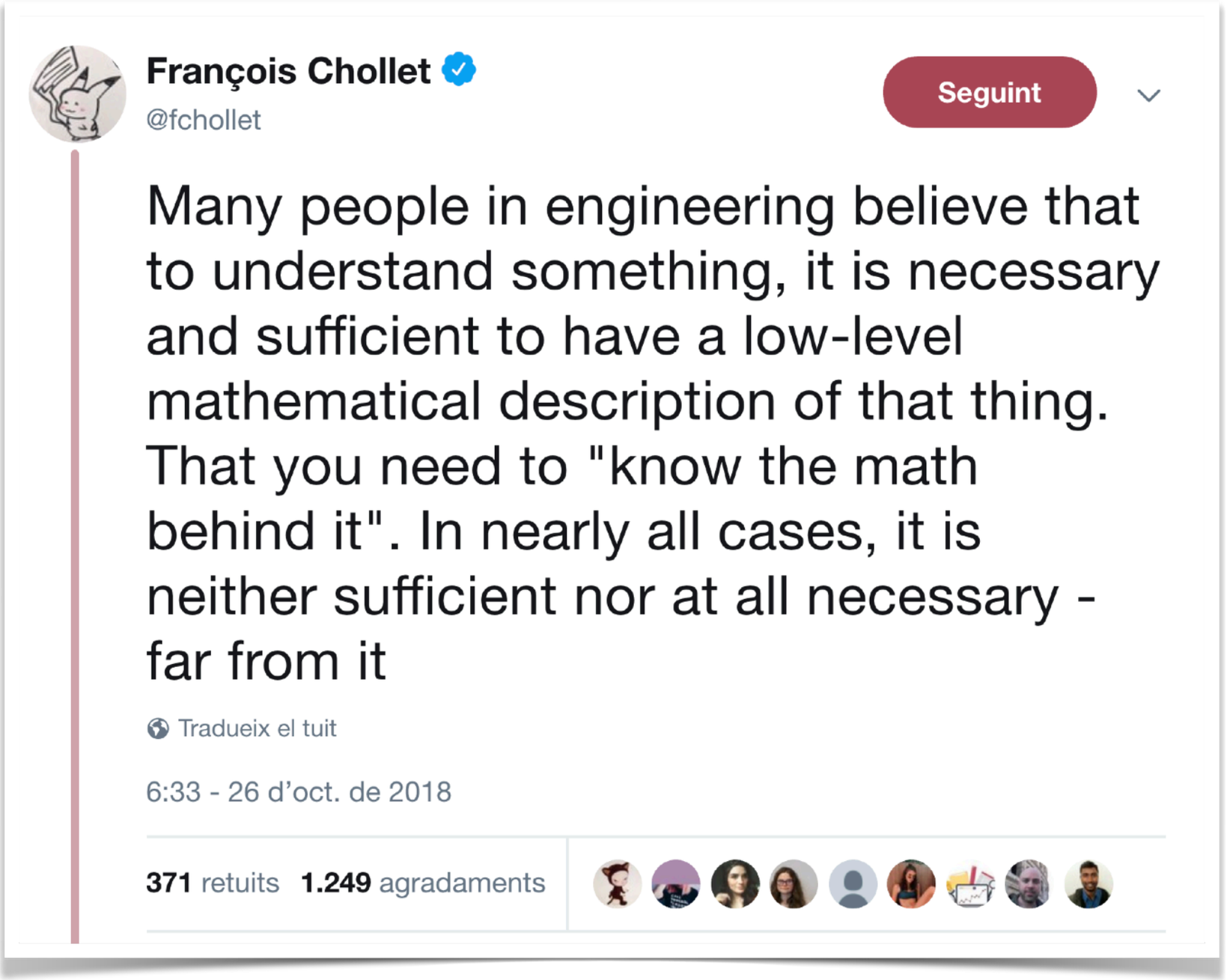
1 3 12

Explanations & Interpretable models

Explanations & Interpretable models



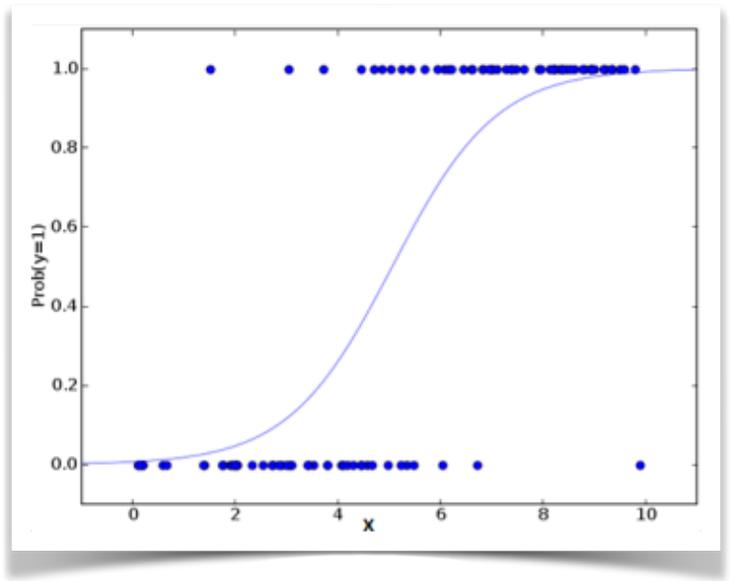
Explanations & Interpretable models



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 one feature by 1 unit changes de "odds" ratio by a factor of $\exp(\hat{\beta}_j)$.

Explanations & Interpretable models

Decision Trees

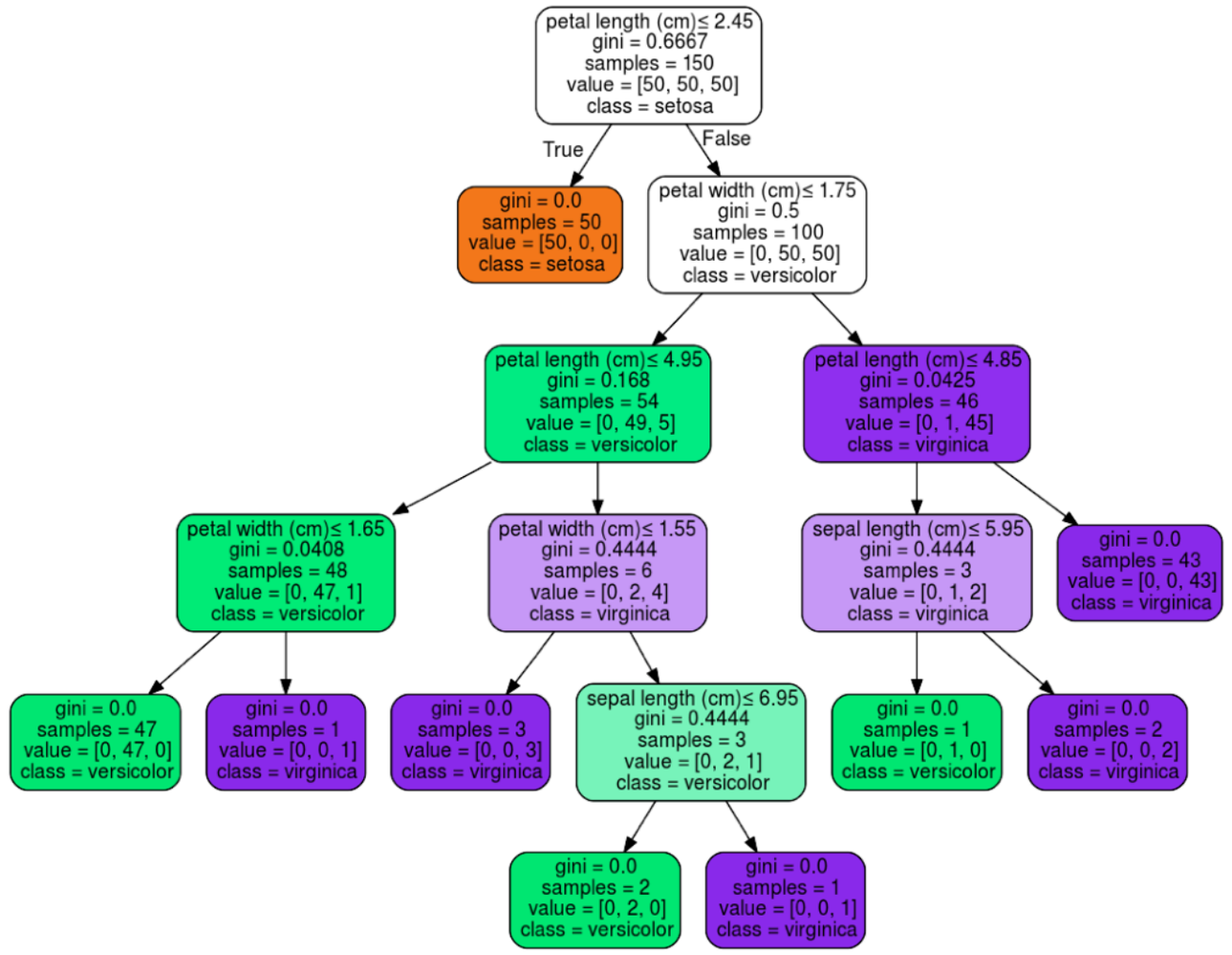
Gini index measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

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

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 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

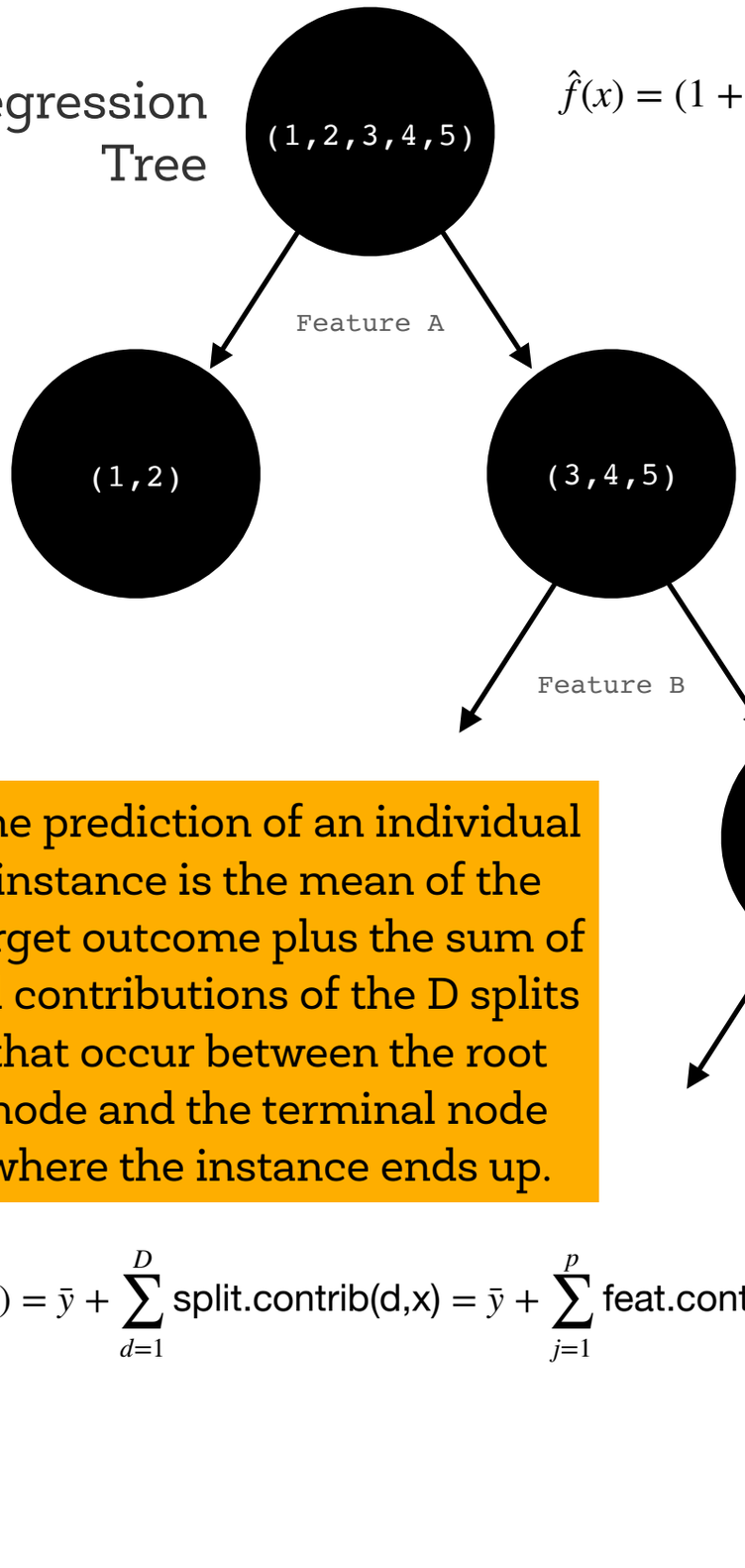
Decision Trees

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

Regression Tree



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

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

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.

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

We are measuring the importance of X_j by its **impact** in the outcome.

Explanations & Interpretable models

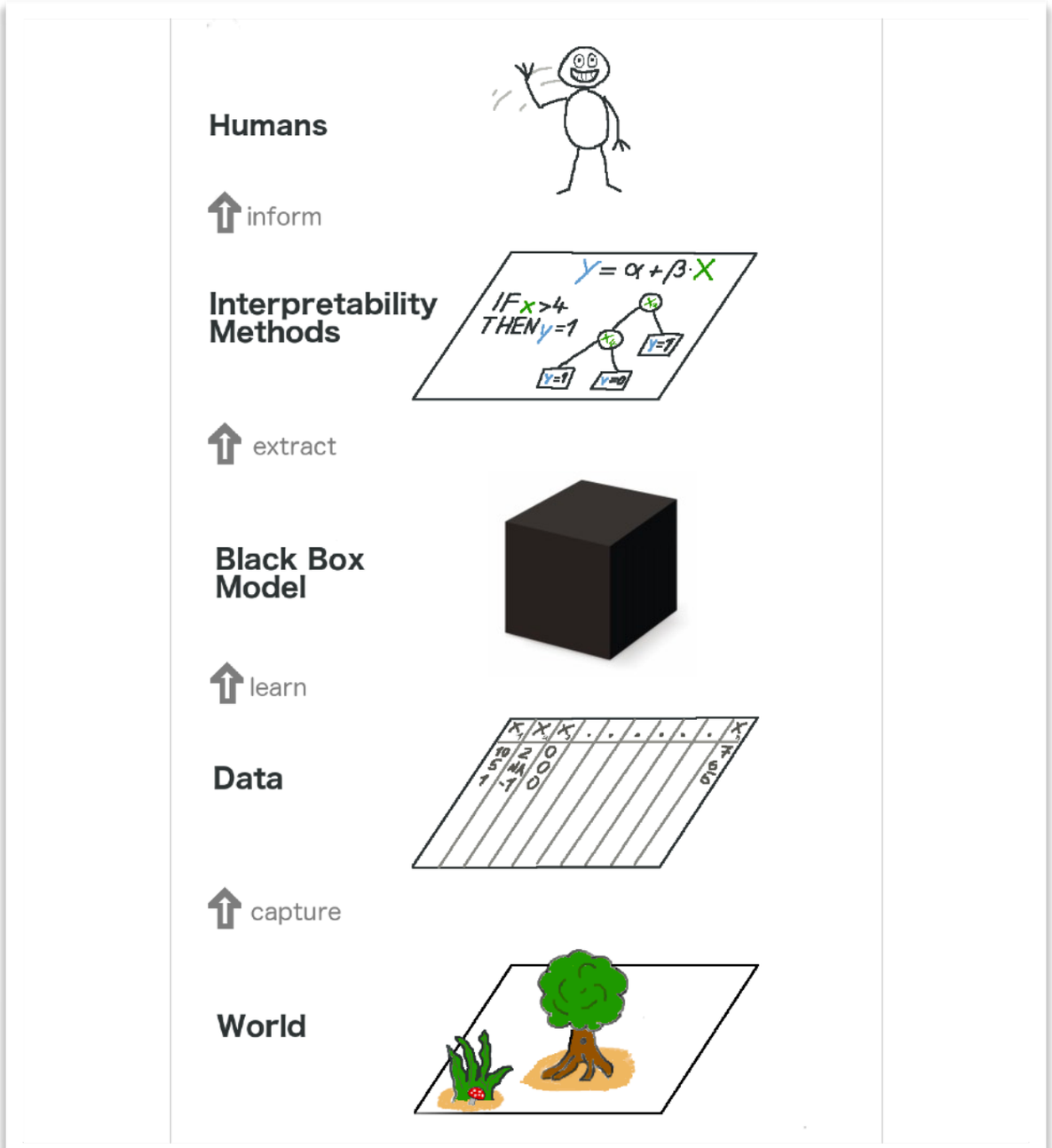
We can design this kind of explanations to other popular models:

- 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?

Model-agnostic explanation models



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

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:

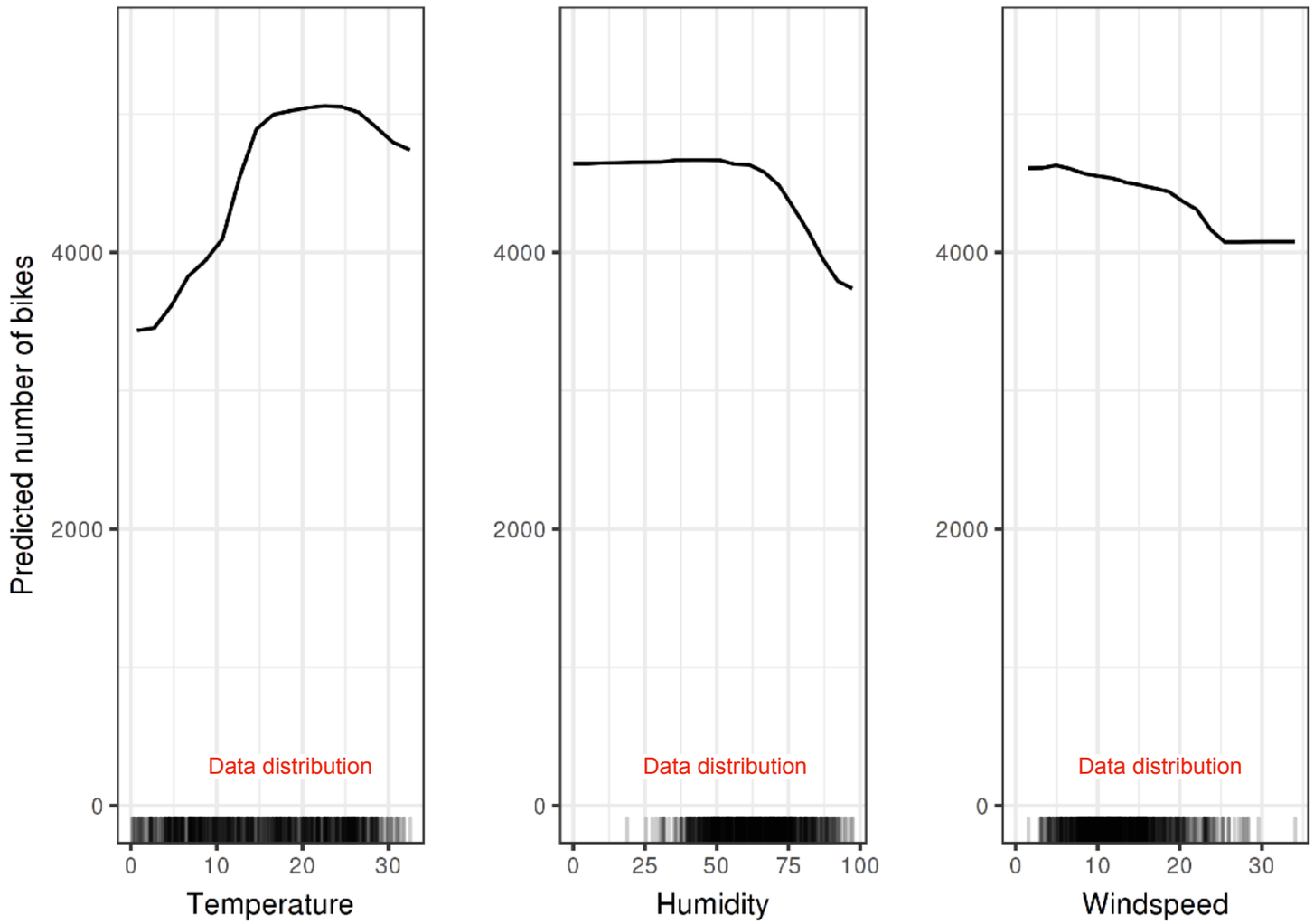
A	B	C	Y
a1	b1	c1	yA1
a1	b2	c2	
a1	b3	c3	
a2	b1	c1	yA2
a2	b2	c2	
a2	b3	c3	
a3	b1	c1	yA3
a3	b2	c2	
a3	b3	c3	

Finally, it plots out the average predictions.

X	A1	A2	A3
Y	yA1	yA2	yA3

Model-agnostic explanation models

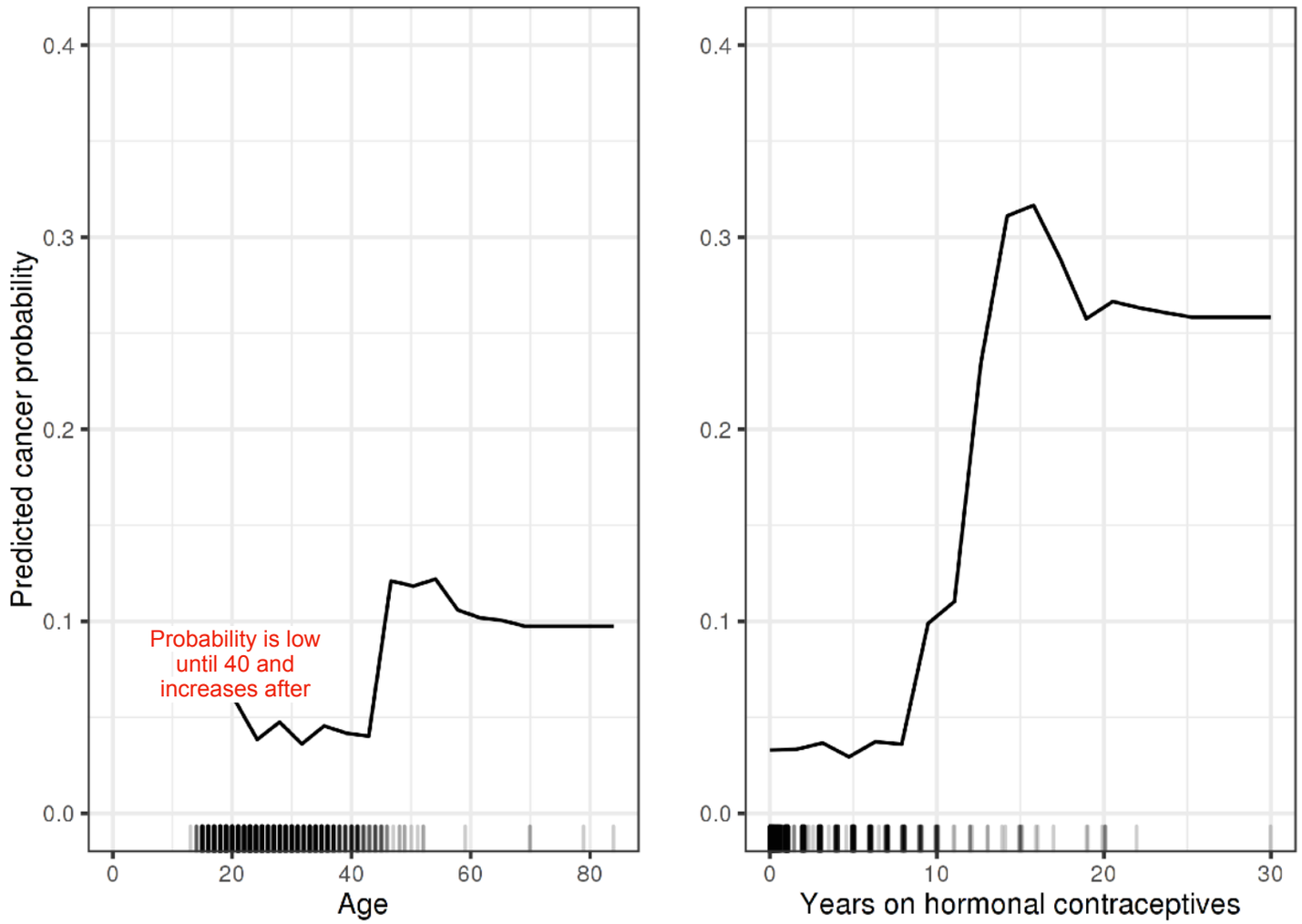
Problem: predict the number of bikes that will be rented on a given day. The influence of the weather features on the predicted bike counts is visualized in the following figure.



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

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

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

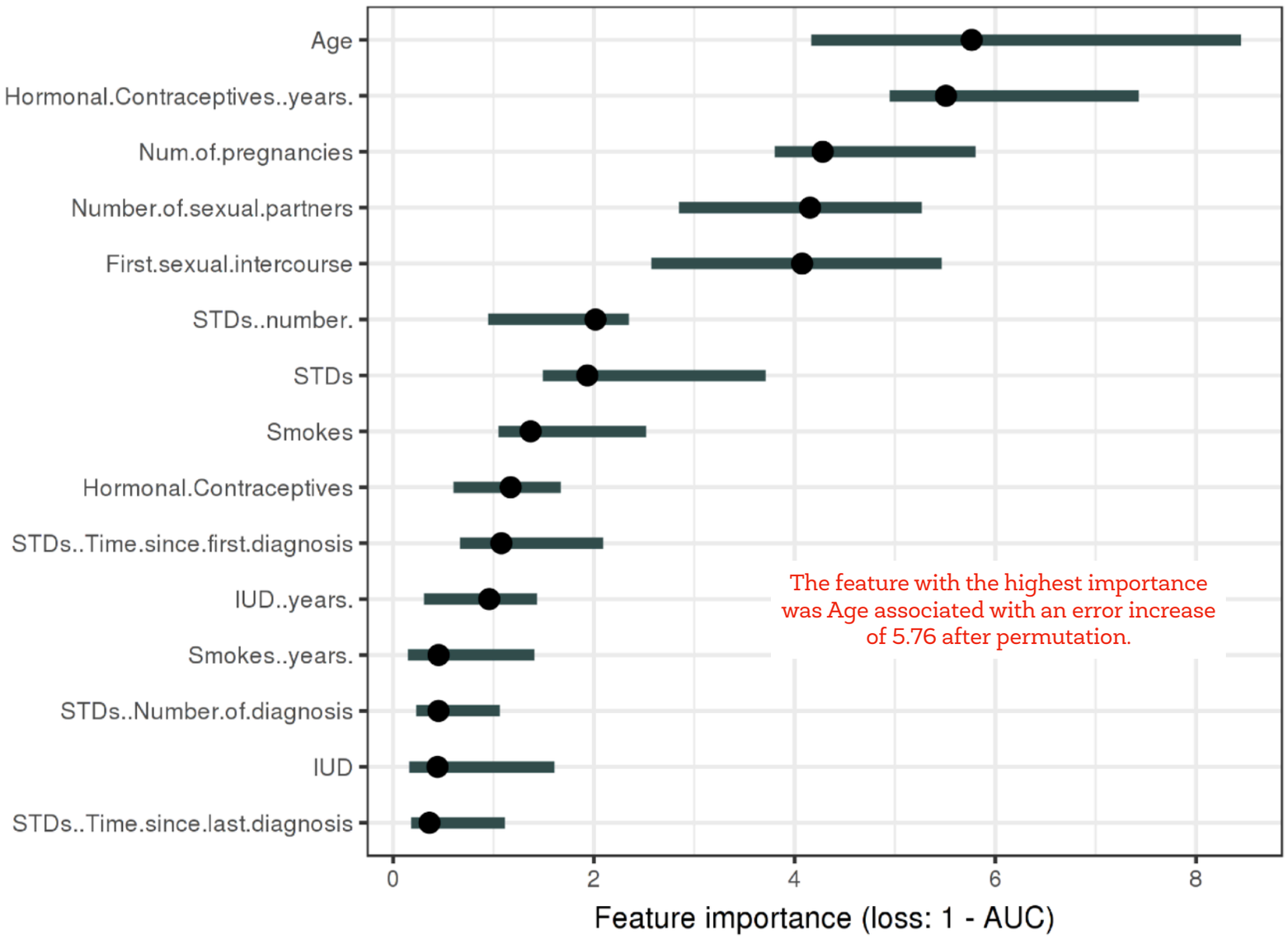
Input: Trained model f, feature matrix X, target vector y, error measure L(y,f).

1. Estimate the original model error e^{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 models

Permutation Test
Problem: Predict cervical cancer.



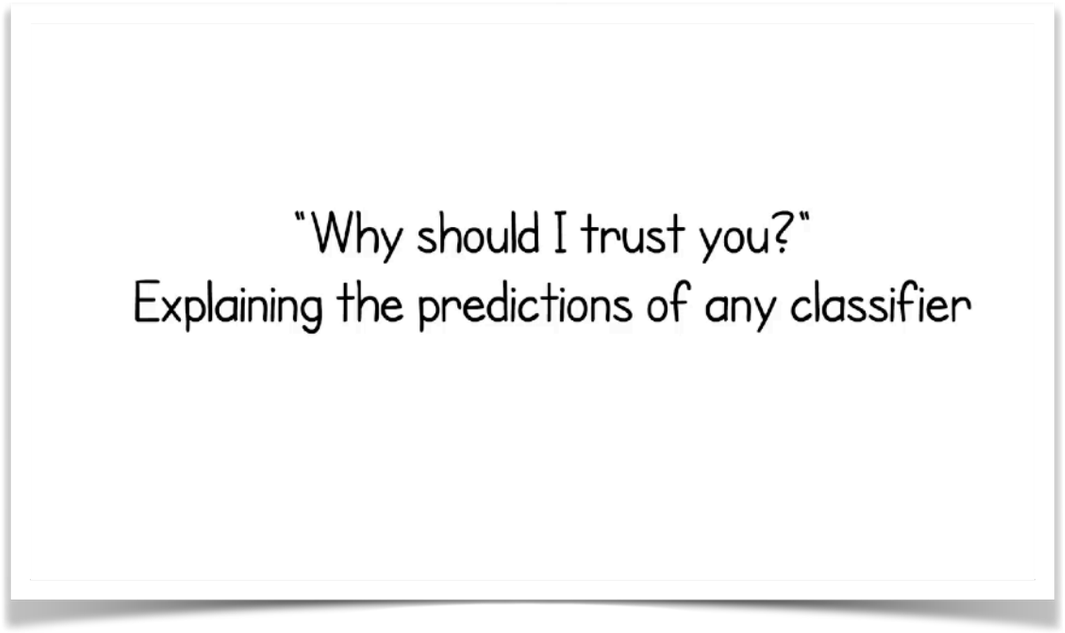
Model-agnostic explanation models

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.

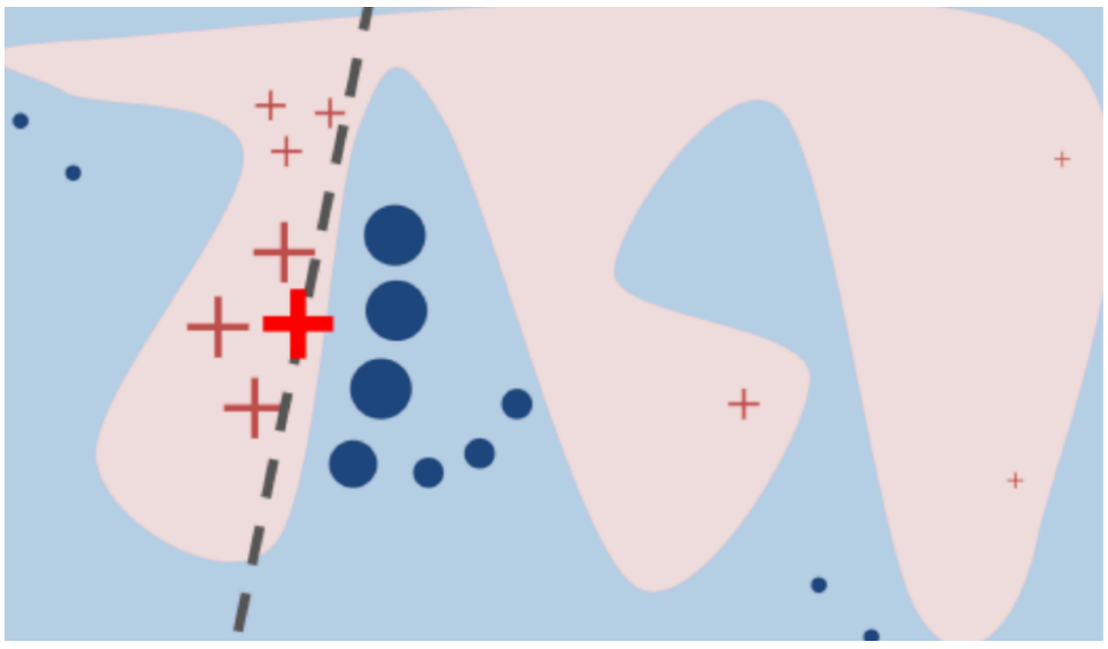
The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background.

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

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.

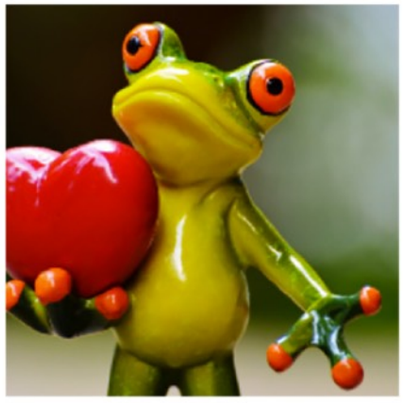
Model-agnostic explanation models

LIME: Local interpretable model-agnostic explanations.

Local Surrogate (LIME) for images

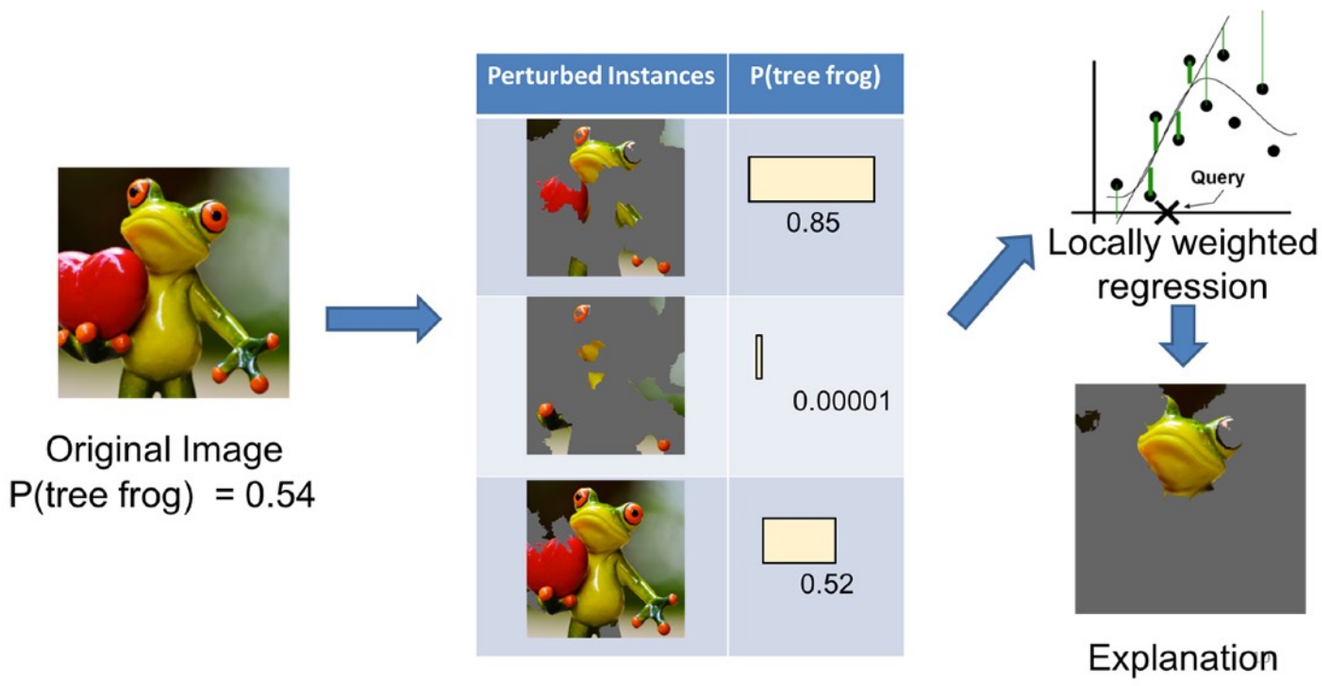
Intuitively, it would not make much sense to perturb individual pixels, since many more than one pixel contribute to one class. Randomly changing individual pixels would probably not change the predictions by much. Therefore, variations of the images are created by segmenting the image into “superpixels”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

Model-agnostic explanation models



Original Image

Interpretable Components



https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime

Model-agnostic explanation models

Shapley Values: Game Theory Attribution

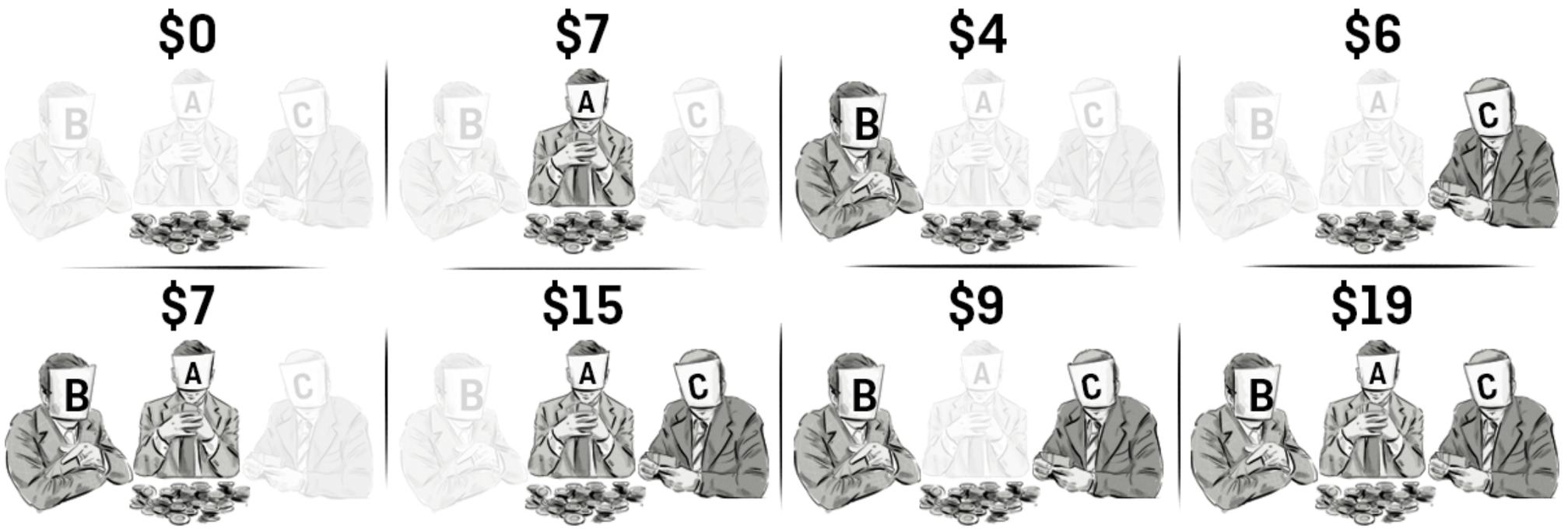
In cooperative situations, something known as
the Shapley value is used to fairly distribute credit or value
to each individual player/participant.

https://clearcode.cc/blog/game-theory-attribution/

Model-agnostic explanation models

Shapley Values: Game Theory Attribution

You first start by identifying each player's contribution when they play individually, when 2 play together, and when all 3 play together.

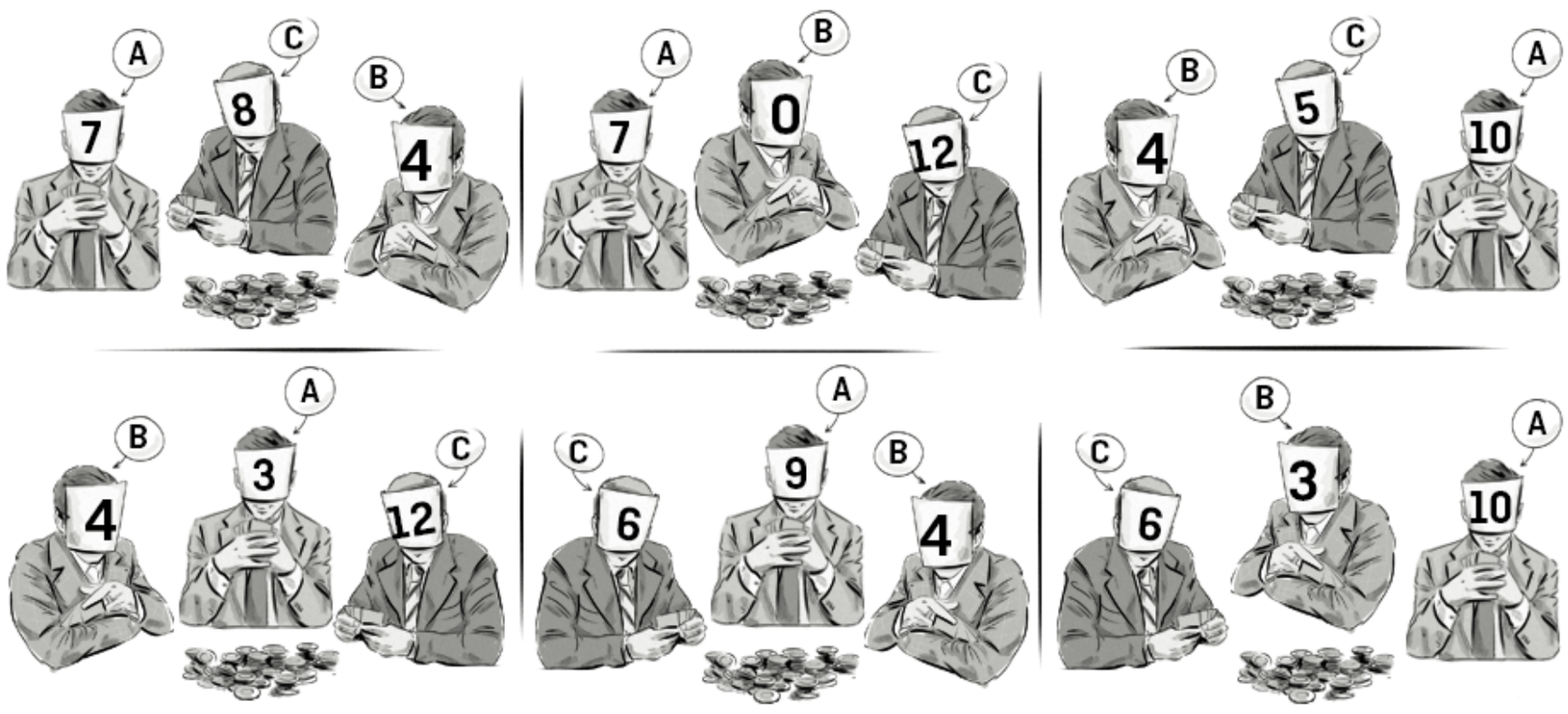


https://clearcode.cc/blog/game-theory-attribution/

Model-agnostic explanation models

Shapley Values: Game Theory Attribution

Then, you need to consider all possible orders and calculate their marginal value – e.g. what value does each player add when player A enters the game first, followed by player B, and then player C.

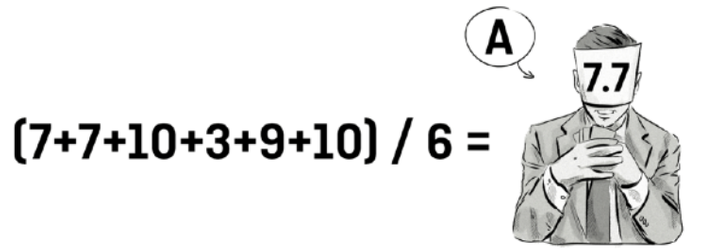


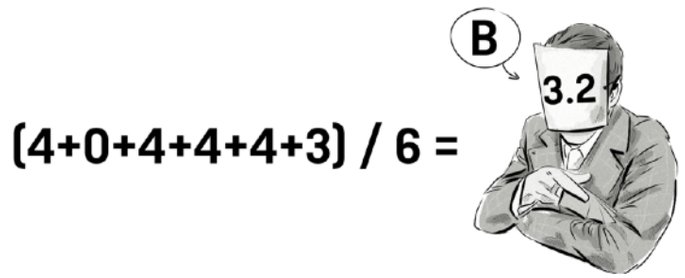
https://clearcode.cc/blog/game-theory-attribution/

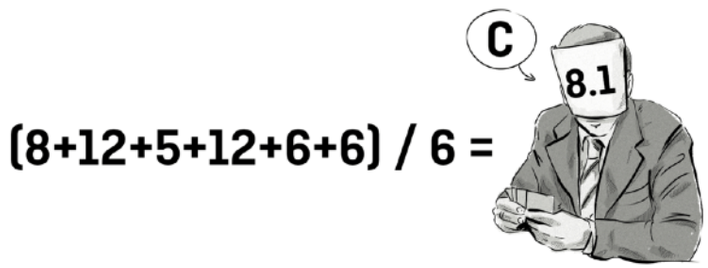
Model-agnostic explanation models

Shapley Values: Game Theory Attribution

Now that we have calculated each player's marginal value across all 6 possible order combinations, we now need to add them up and work out the Shapley value (i.e. the average) for each player.







https://clearcode.cc/blog/game-theory-attribution/

Model-agnostic explanation models

Shapley Values: Game Theory Attribution

A prediction can be explained by assuming that each feature value of the instance is a "player" in a game where the prediction is the payout and the Shapley value is the average marginal contribution of a feature value across all possible coalitions.

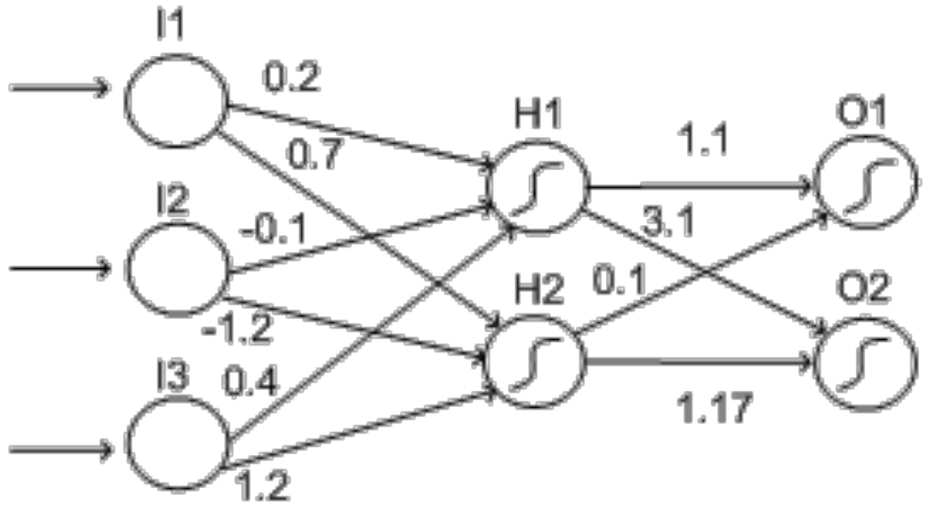
In a prediction problem, possible coalitions (sets) of feature values have to be evaluated with and without the j-th feature to calculate the exact Shapley value.

For more than a few features, the exact solution to this problem becomes problematic as the number of possible coalitions exponentially increases as more features are added and **several approximations** have been proposed.

Model-agnostic explanation models

SHAP Notebook

Interpretable Deep Learning



- What does the weights of each connection mean in terms of interpreting the result?
- Which is the set of weights that play the most important role in the final prediction?
- Does knowing the magnitude of the weights tells me anything about the importance of the input variables?

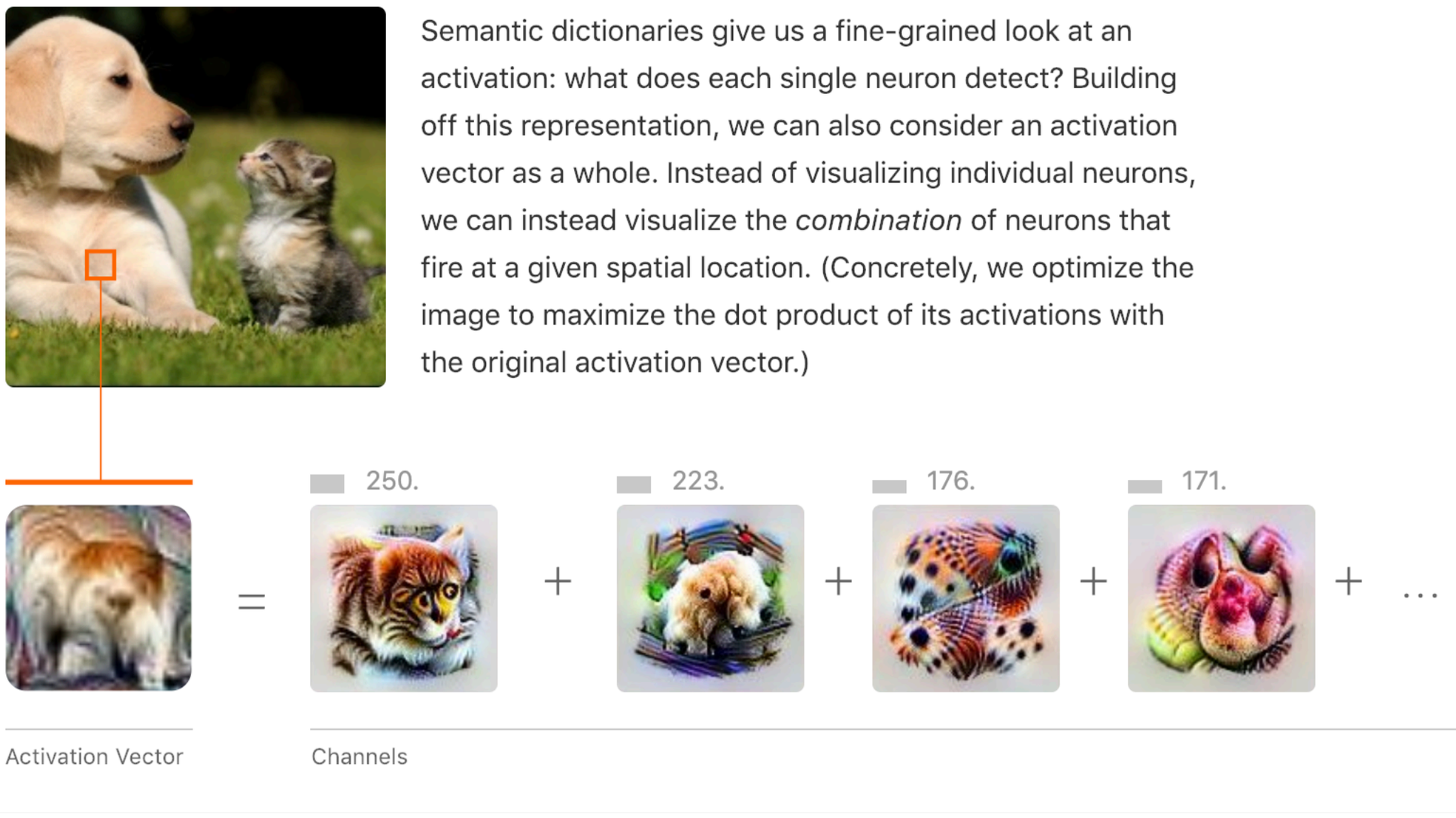
Interpretable Deep Learning

Methods for explaining (vision) neural networks generally falls within two broad categories: **feature visualization** and **pixel attribution methods**.

The firsts aim at **visualizing** what is going on inside the network and answering questions such as (1) which weights are being activated given some inputs? or (2) what regions of an image are being detected by a particular convolutional layer? .

Interpretable Deep Learning

What Does the Network See?

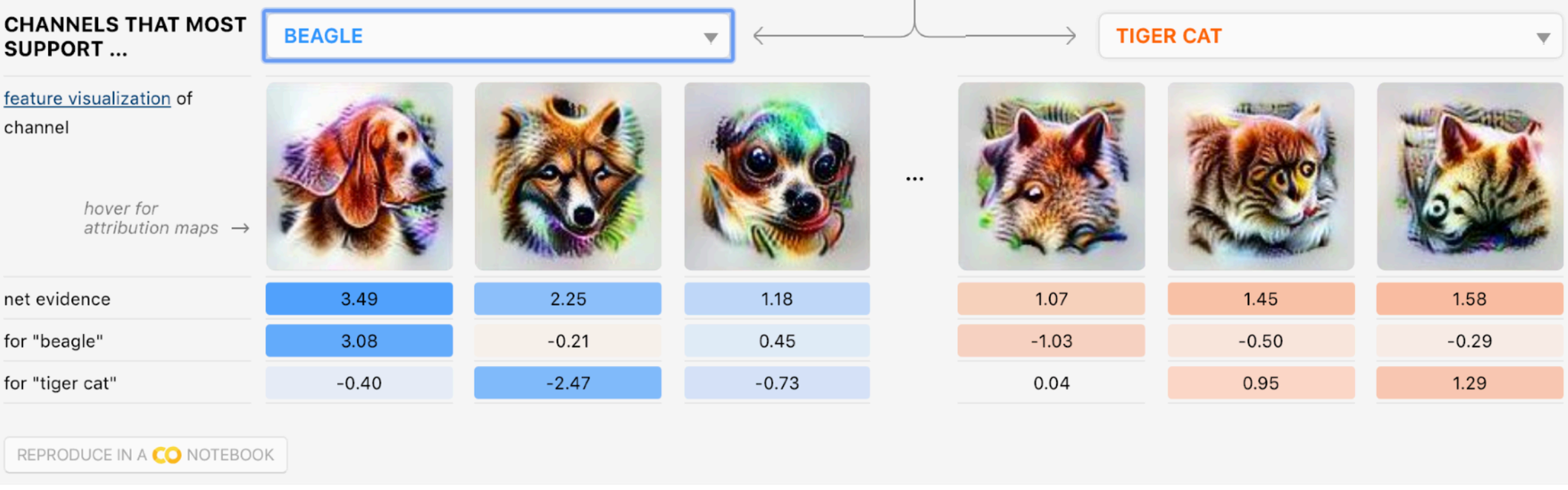


https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

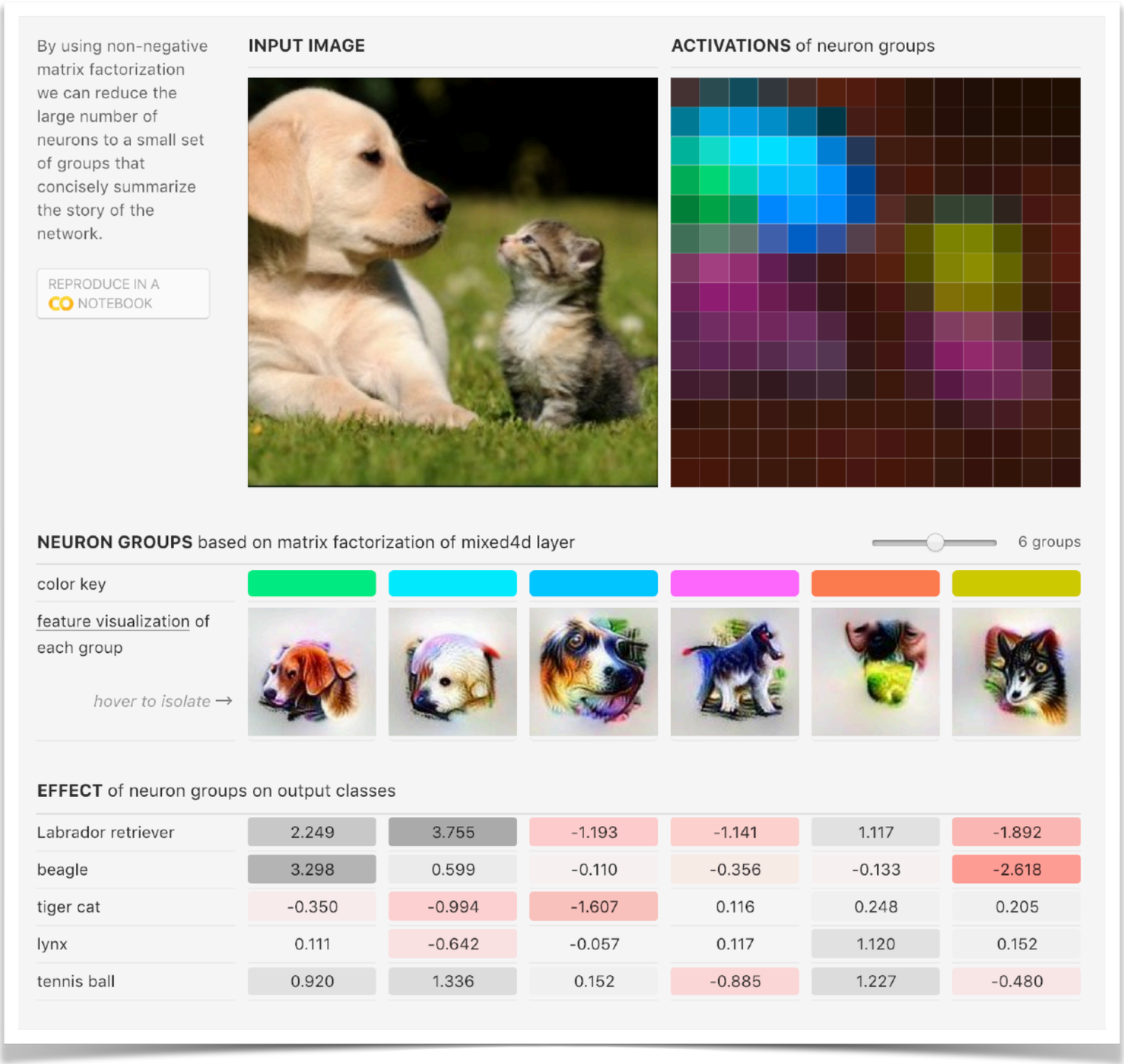
For instance, by combining feature visualization (*what is a neuron looking for?*) with attribution (*how does it affect the output?*), we can explore how the network decides between labels like **beagle** and **tiger cat**.

Several floppy ear detectors seem to be important when distinguishing dogs, whereas pointy ears are used to classify "tiger cat".



https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

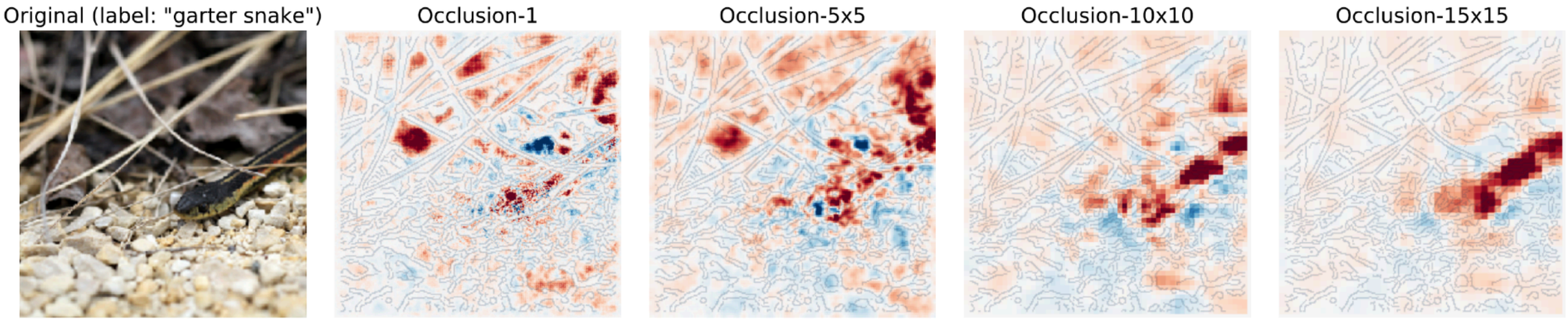


https://distill.pub/2018/building-blocks/

Interpretable Deep Learning

Pixel attribution methods.

Perturbation-based methods directly compute the attribution of an input feature by removing, masking or altering them, and running a forward pass on the new input, measuring the difference with the original output.

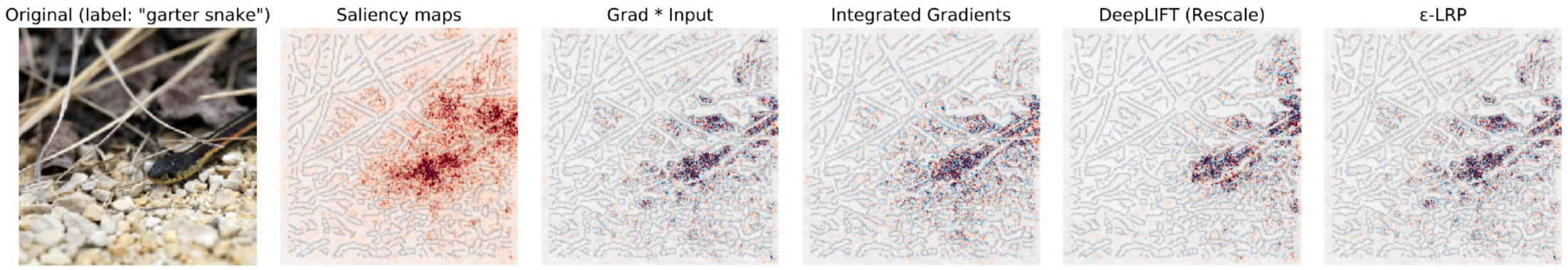


https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

Interpretable Deep Learning

Pixel attribution methods.

Gradient based methods constructs attributions by considering the partial derivative of the target output with respect to the input features (pixels).

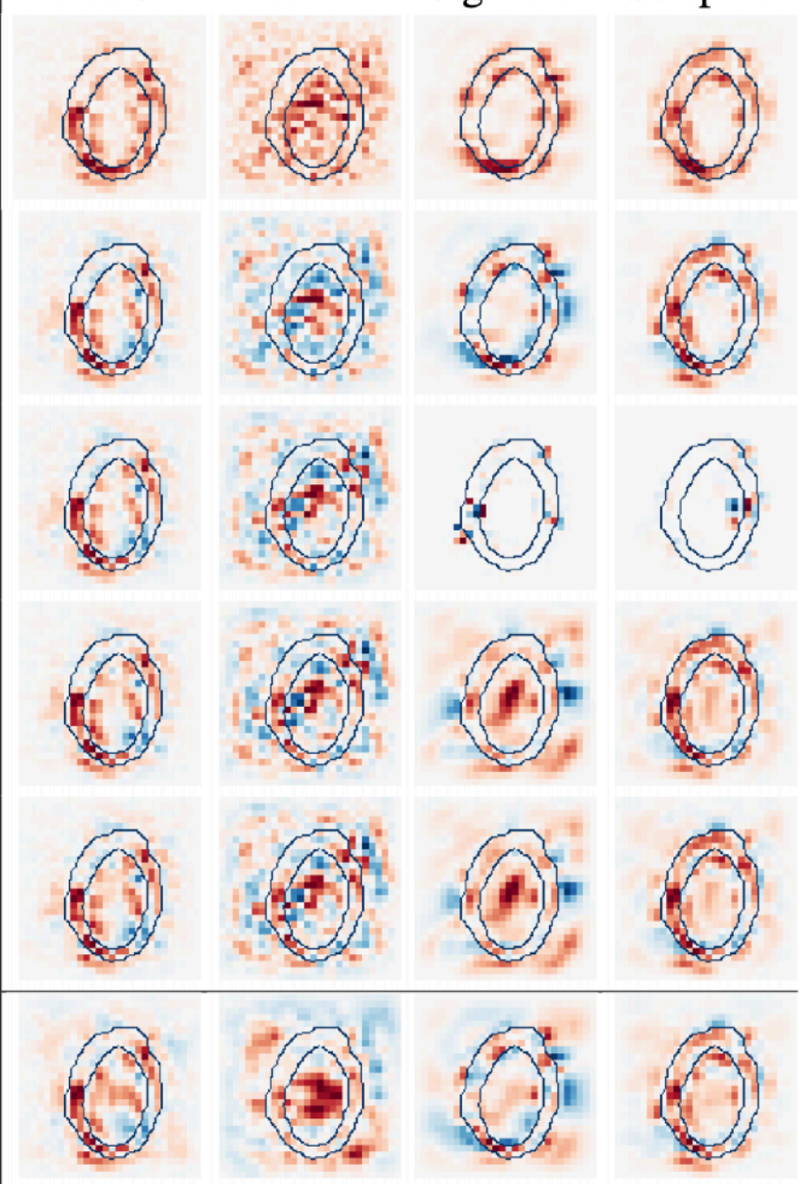


https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

Interpretable Deep Learning

Method	Attribution $R_i^c(x)$
Saliency Maps	$\left\lvert \frac{\partial S_c(x)}{\partial x_i} \right\rvert$
Gradient * Input	$x_i \cdot \frac{\partial S_c(x)}{\partial x_i}$
ϵ-LRP	$x_i \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z)}{z}$
DeepLIFT	$(x_i - \bar{x_i}) \cdot \frac{\partial^g S_c(x)}{\partial x_i}, \quad g = \frac{f(z) - f(\bar{z})}{z - \bar{z}}$
Integrated Gradient	$(x_i - \bar{x_i}) \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x_i})}\Big\rvert_{\tilde{x}=\bar{x}+\alpha(x-\bar{x})} d\alpha$
Occlusion-1	$x_i \cdot \int_{\alpha=0}^{1} \frac{\partial S_c(\tilde{x})}{\partial(\tilde{x_i})}\Big\rvert_{\tilde{x}=x_{[x_i=\alpha \cdot x_i]}} d\alpha$

Example of attributions on MNIST



https://pdfs.semanticscholar.org/7a56/72796aeca8605b2e370d8a756a7a311fd171.pdf

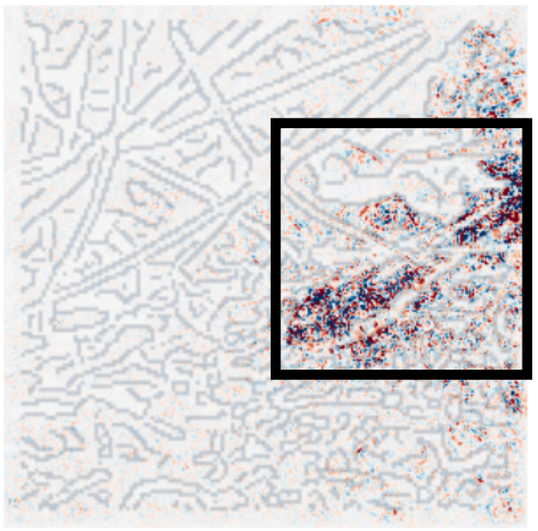
Interpretable Deep Learning

Original (label: "garter snake")

Explanation:

These pixels/region are the evidence of prediction.

DeepLIFT (Rescale)



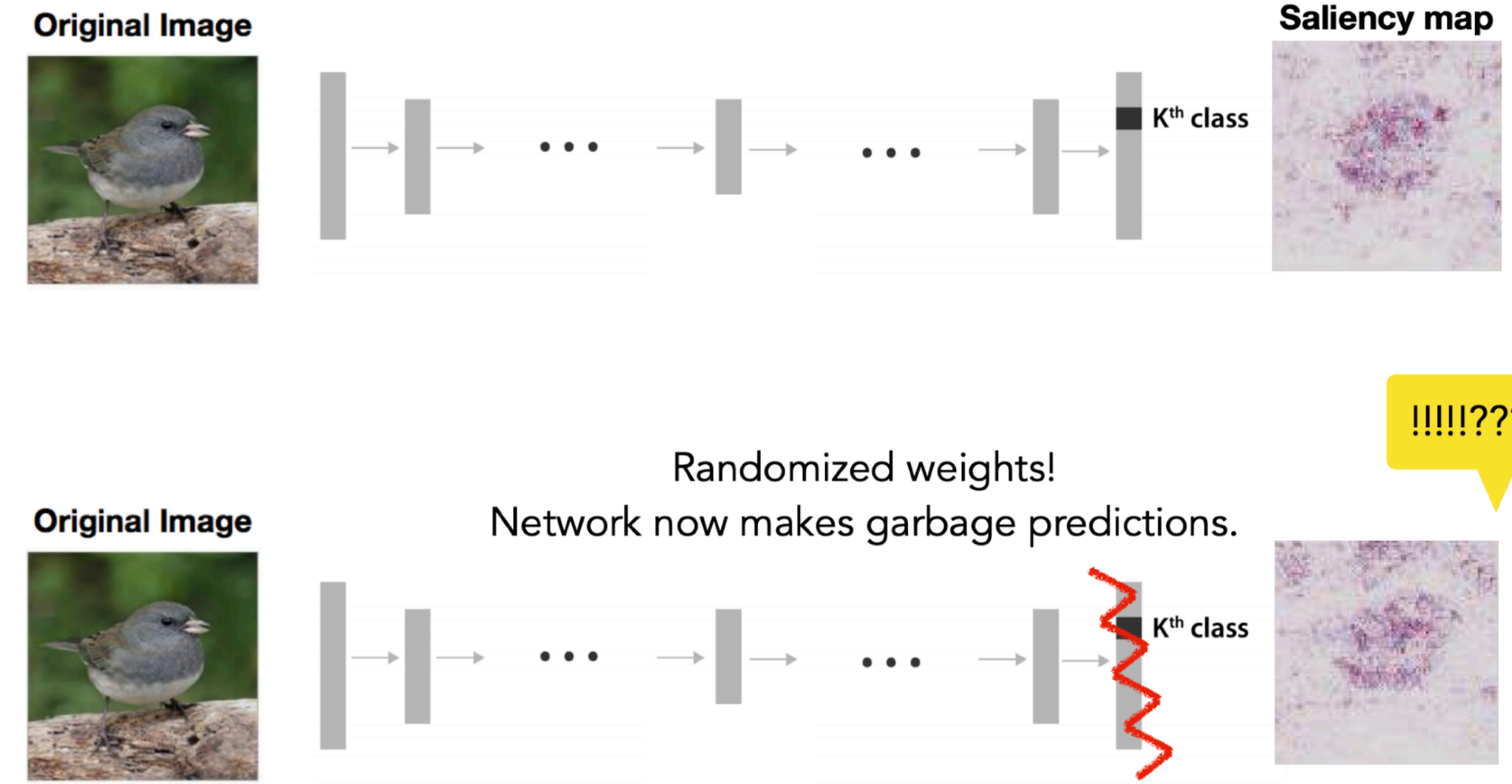
Interpretable Deep Learning

Pixel attribution methods.

If the assumption is correct, when prediction changes the explanation should change.

Extreme case: If prediction is random, the explanation should really change.

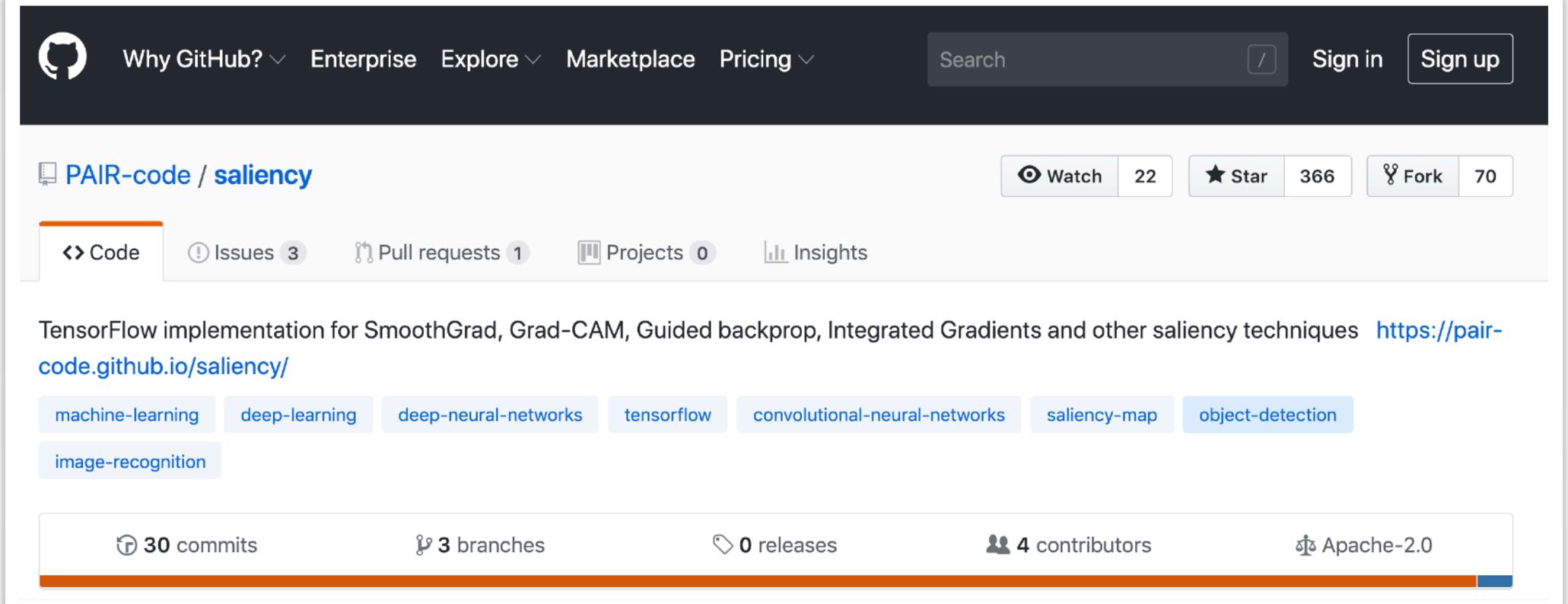
Interpretable Deep Learning



https://nips.cc/media/Slides/nips/2018/220e(05-09-45)-05-10-30-12640-Sanity_Checks_f.pdf

Interpretable Deep Learning

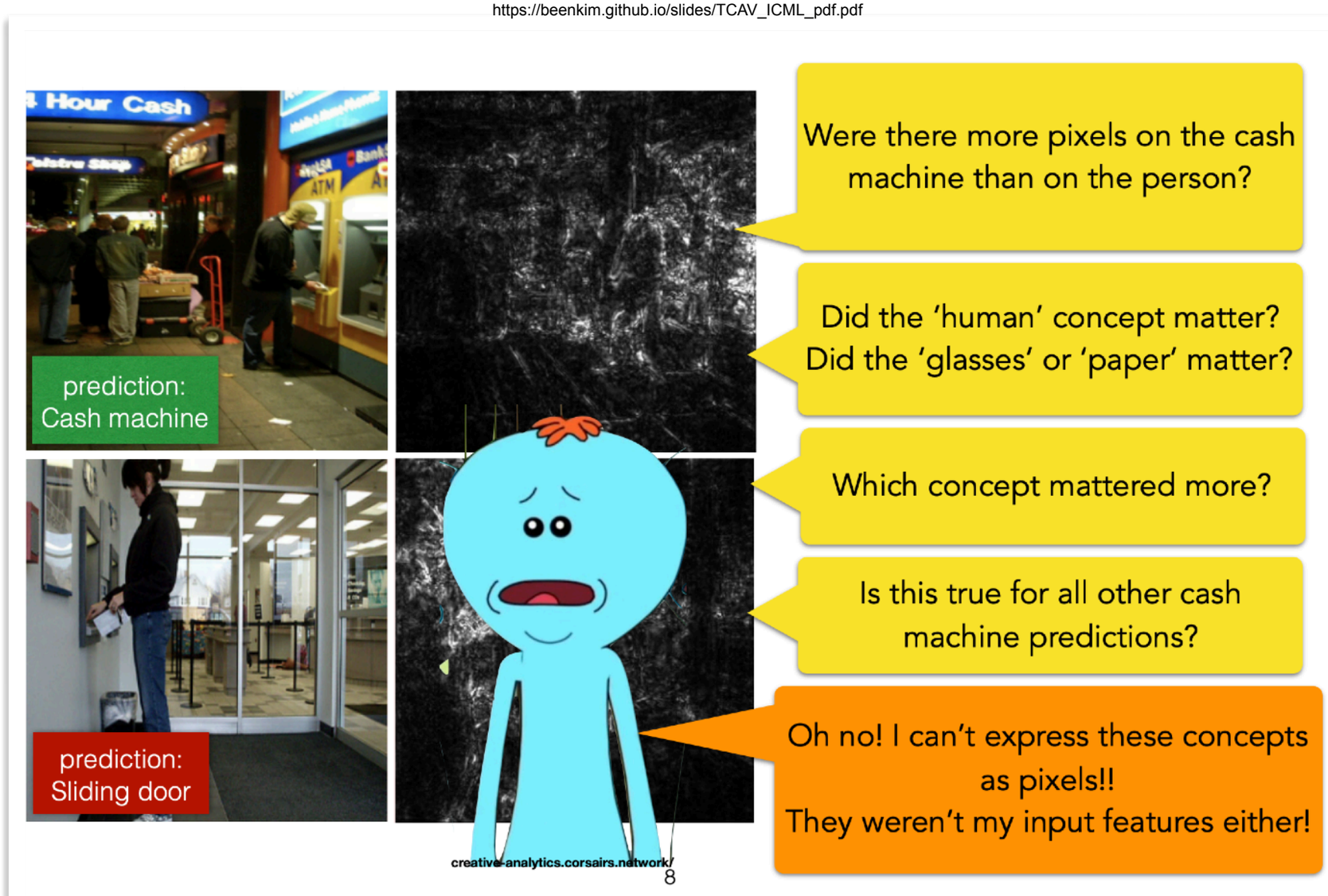
https://github.com/PAIR-code/saliency/blob/master/Examples.ipynb



Interpretable Deep Learning

In general, we cannot express explanations (for humans) as pixels. We need to explain decisions in terms of **concepts**, not pixels.

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf



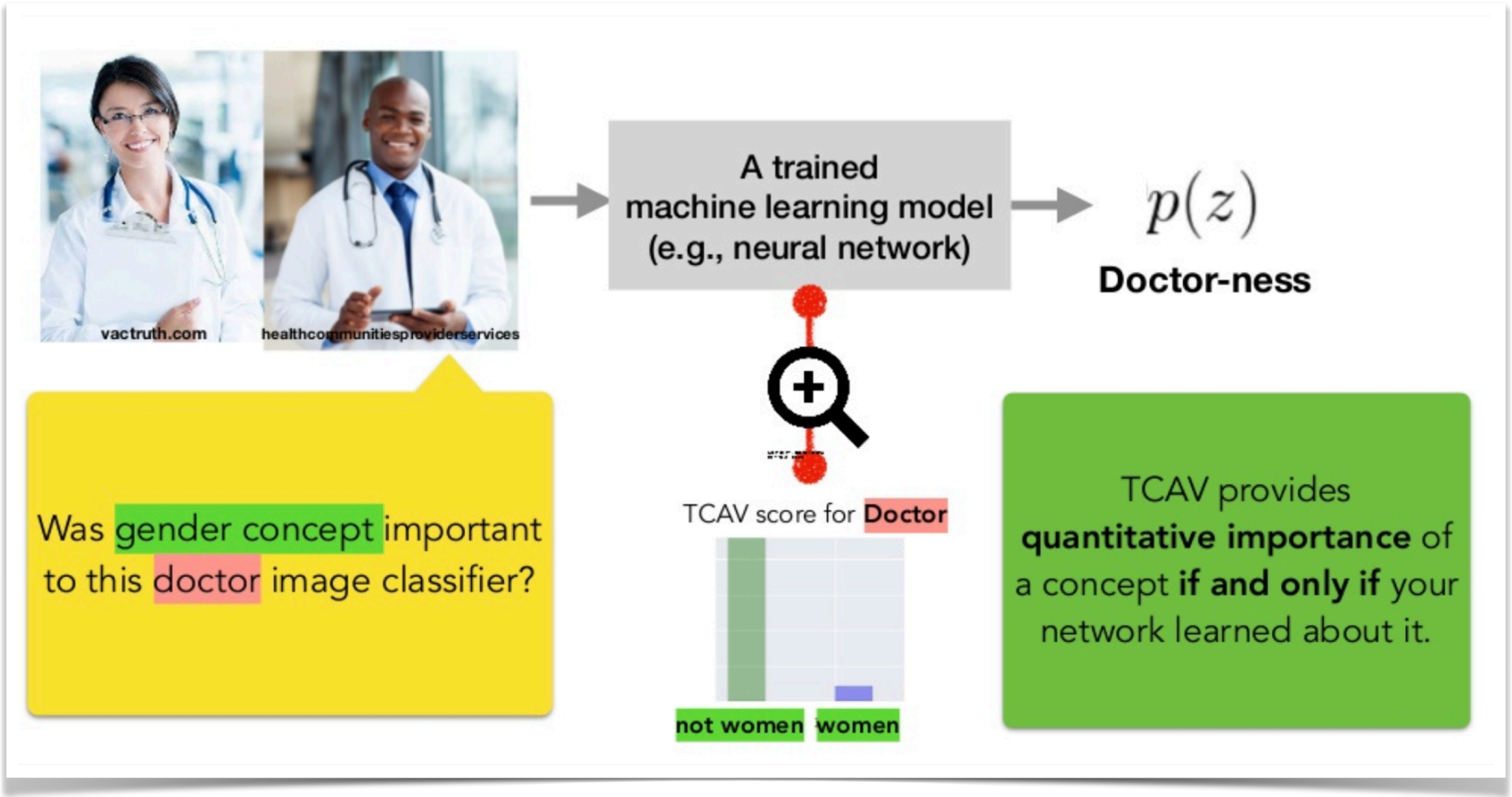
Interpretable Deep Learning

Instead of pixels, we would like to use a **quantitative explanation expressing how much a concept** (gender, visual class, etc.) was important for a decision, **even if the concept was not part of the training**!

This has been recently explored in a method called **Concept Activation Vectors**.

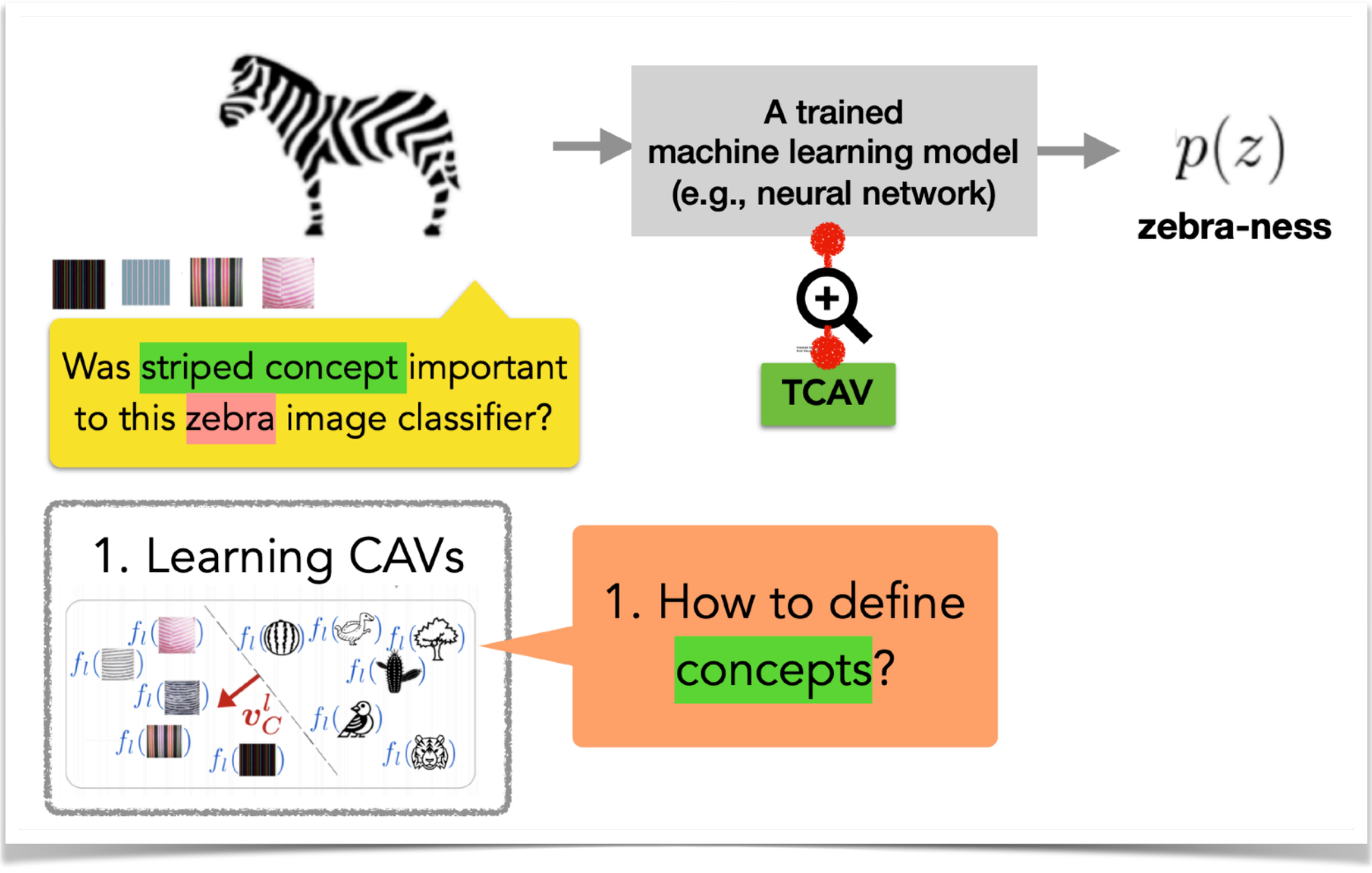
Kim, Been, et al. "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)." International Conference on Machine Learning. 2018.

Interpretable Deep Learning



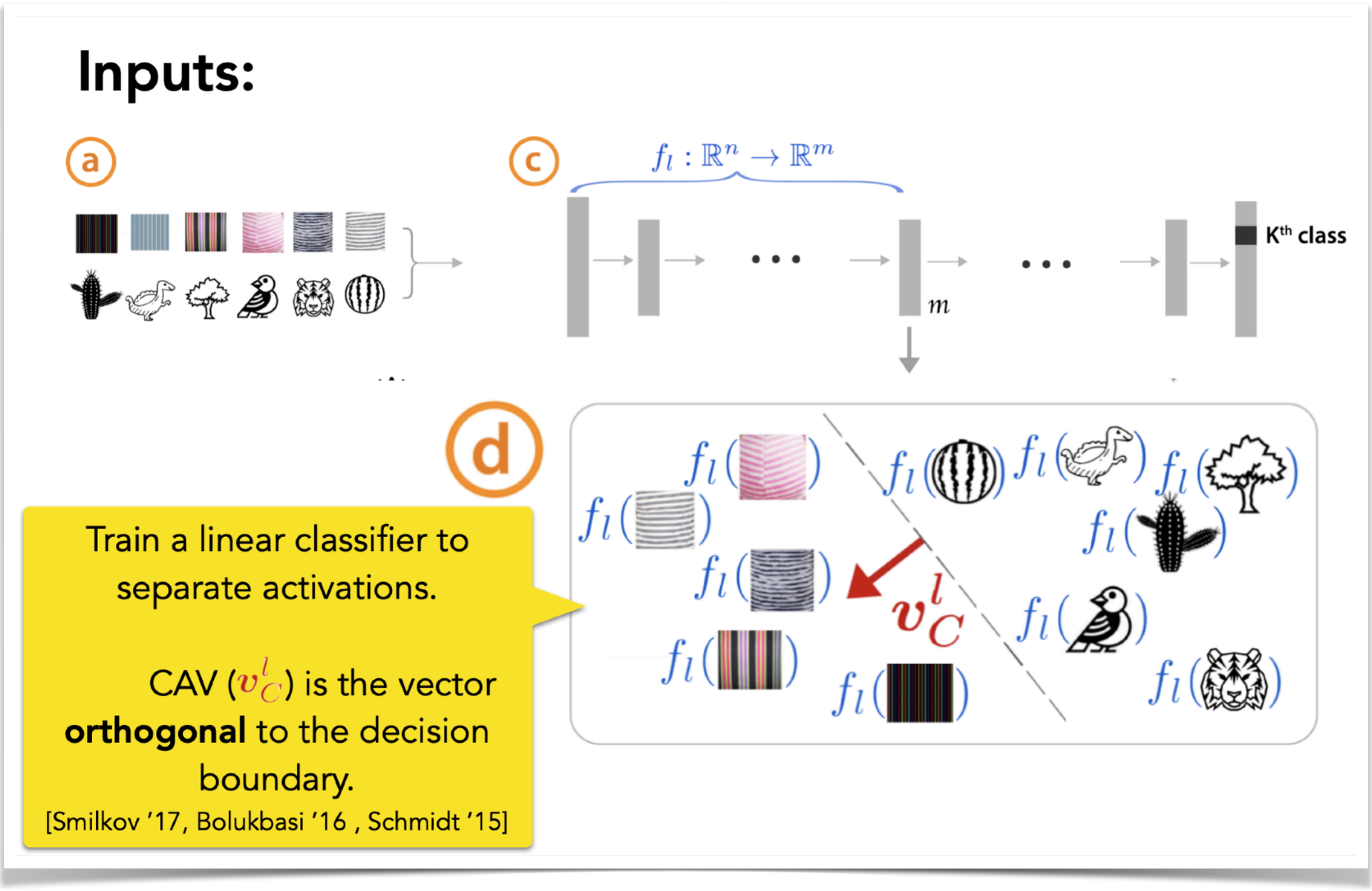
https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning



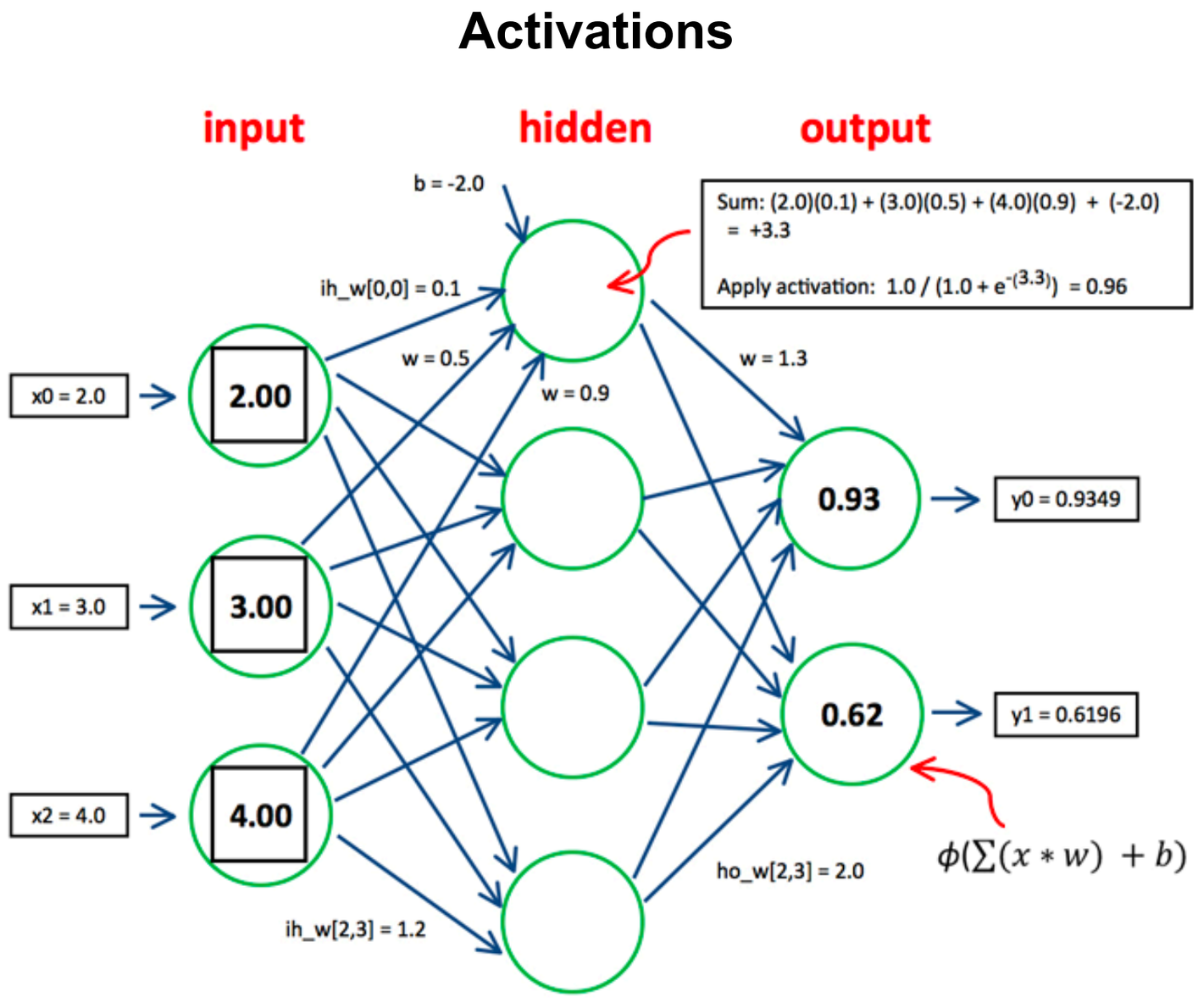
https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

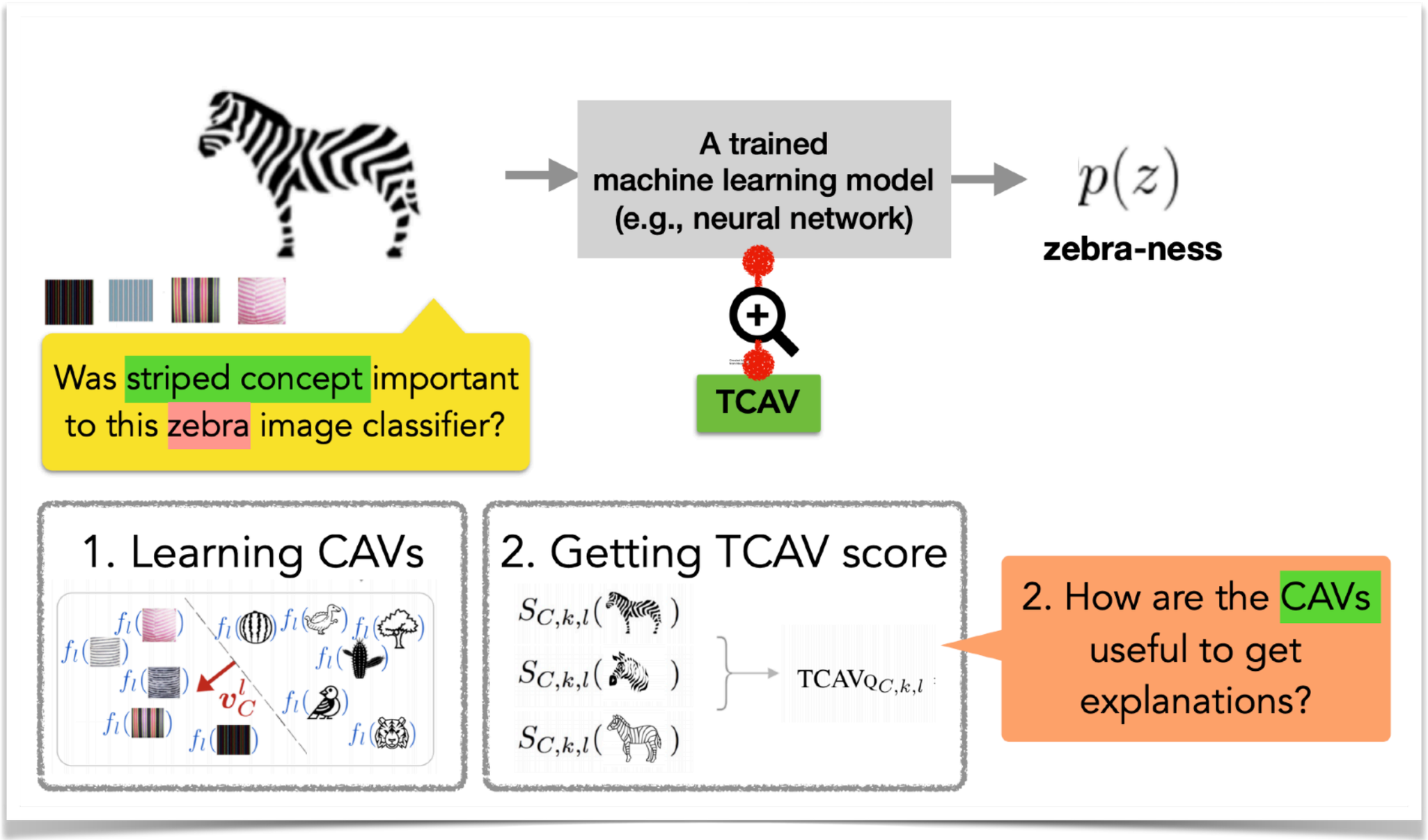


https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

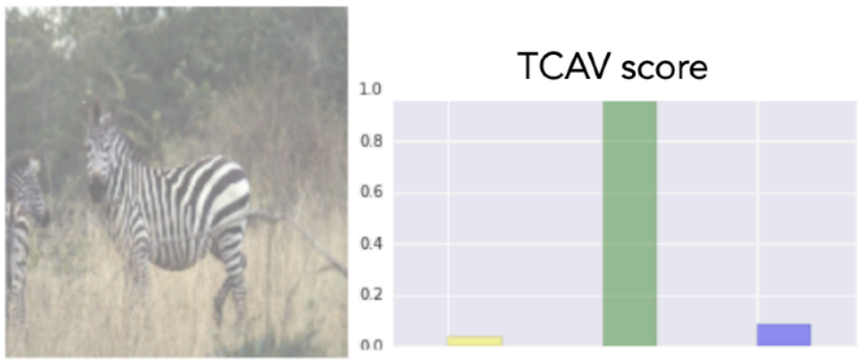


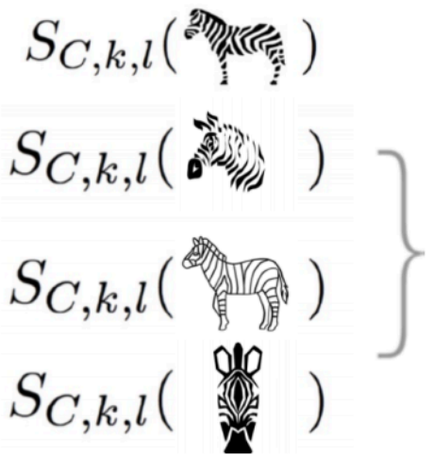
Interpretable Deep Learning



https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning





$$S_{C,k,l}(\quad)$$
$$S_{C,k,l}(\quad)$$
$$S_{C,k,l}(\quad)$$
$$S_{C,k,l}(\quad)$$

zebra-ness → $\dfrac{\partial p(z)}{\partial \boldsymbol{v}_C^l} = S_{C,k,l}(\boldsymbol{x})$

striped CAV →

$$\text{TCAVQ}_{C,k,l} = \frac{|\{\boldsymbol{x} \in X_k : S_{C,k,l}(\boldsymbol{x}) > 0\}|}{|X_k|}$$

Directional derivative with CAV

https://beenkim.github.io/slides/TCAV_ICML_pdf.pdf

Interpretable Deep Learning

stripes concept (score: 0.9) was important to **zebra** class for this trained network.

What we know: Recap

An explanation can be seen as a social conversation and interaction for transfer knowledge. A fruitful exchange implies that who explains must be able to recognize the mental model of who receives the information.

The final aim is in fact is to produce explanations that allow affected parties, regulators and more broadly non-insiders to understand, discuss, and potentially contest decisions provided by black-box models

The rise in machine learning-assisted decision-making has led to concerns about the fairness of the decisions and techniques to mitigate problems of discrimination.

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.

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.*'

What do we mean by "real pattern"?

Let's consider a credit lending model suggesting increasing income by $5,000.

One may act on this by either waiting to obtain a raise at their current job or taking up a new high-paying job.

Either of these actions would increase income but would also affect "length of employment", which may be another feature of the model. The unforeseen change to "length of employment" may adversely affect the prediction despite the increase in income.

Trust and Counterfactuals

Trust and Counterfactuals

How can we provide **meaningful** explanations to engender trust in the algorithmic decision process?

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

Which information can we provide/use?

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

Trust and 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

Trust and 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.

Trust and 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.

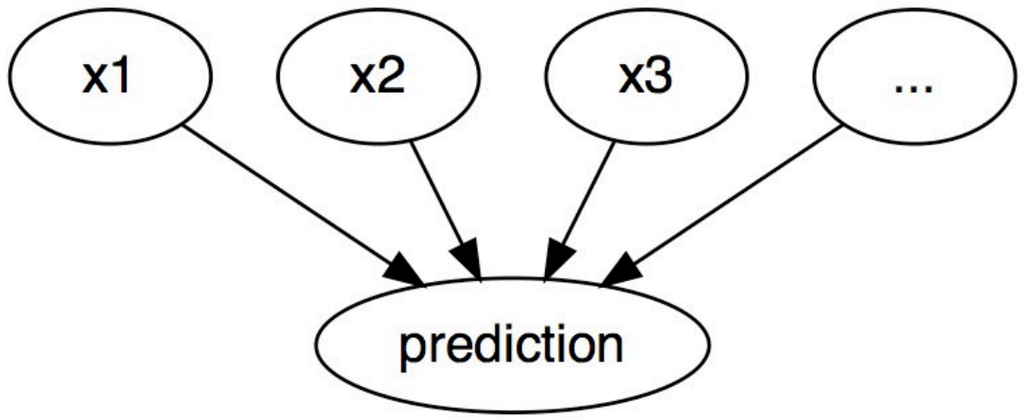
Japanese filmmaker Akira Kurosawa directed the 1950 film Rashomon, in which four different people provide contradictory accounts of a samurai's murder, despite all having witnessed the crime.

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*]

Trust and 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.

Trust and 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.

Trust and Counterfactuals

Counterfactuals in our life:

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

Let's support that we want to rent an apartment and we train a model with real data to predict a price.

After entering all the details about size, location, whether pets are allowed and so on, the model tells us that we can charge 900€.

How could we get (by doing an intervention) 1000€? We can play with the feature values of the apartment to see how we can improve the value of the apartment!

We find out that the apartment could be rented out for over 1000 Euro, if it were 15 m^2 larger. Interesting, but non-actionable knowledge, because we cannot enlarge the apartment.

Finally, by tweaking only the feature values under our control (built-in kitchen yes/no, pets allowed yes/no, type of floor, etc.), we find out that if we allow pets and install windows with better insulation, we can charge 1000€.

Trust and 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 $L_1, L_2 \ldots$

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 λ balances the distance in prediction (first term) against the distance in feature values (second term). The loss is solved for a given λ and returns a counterfactual. A higher value of λ 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 λ is very large, the instance with the prediction closest to y' will be selected, regardless how far it is away from x.

Trust and Counterfactuals

The proposed method has some disadvantages:

- It only takes the first and second criteria into account not the last two ("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**.

Trust and 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.

Trust and Counterfactuals

What do we mean by "real pattern"?

Let's consider a credit lending model suggesting increasing income by $5,000.

One may act on this by either waiting to obtain a raise at their current job or taking up a new high-paying job.

Either of these actions would increase income but would also affect "length of employment", which may be another feature of the model. The unforeseen change to "length of employment" may adversely affect the prediction despite the increase in income.

Trust and Causal Counterfactuals

Problem:

Suppose we want to sent a marketing promotion text message to all of our customers for whom we have a valid cell phone number and want to know the **causal effect** on the likelihood to make a purchase in the next month.

Phone number is an optional field for our customers, and we know that those who do not provide a phone number are less frequent shoppers on average. So, we have two different groups.

Thus, if we simply compare the two groups, the promotion will look more effective than it really was because it is being sent to generally more active customers.

Trust and Causal Counterfactuals

Data:

This is the observed outcome under an intervention

This is the observed outcome under no intervention

i	T: message	Y: observed purchases	Y(0)	Y(1)	Y(0) - Y(1)
1	0	2	2	?	?
2	1	12	?	12	?
3	1	3	?	3	?
4	0	4	4	?	?
5	0	11	11	?	?
6	1	9	?	9	?

This two variables are not the same!

Users under intervention and users under intervention are two different populations because the intervention depends on their characteristics.

Trust and Causal Counterfactuals

The question we are asking is about the full population under two different interventions!

i	T: message	Y: observed purchases	Y(0)	Y(1)	Y(0) - Y(1)
1	0	2	2	?	?
2	1	12	?	12	?
3	1	3	?	3	?
4	0	4	4	?	?
5	0	11	11	?	?
6	1	9	?	9	?

This is the **fundamental problem of causal inference**.

Introduction to Causal Inference

Let's now suppose that I want an answer to this question:

> *Given that I have a beard, and that I have a PhD degree, and everything else we know about me, with what probability would I have obtained a PhD degree, had I never grown a beard?*

This is not an interventional query. This is a **counterfactual query**!

A causal query talks about a randomly sampled individual, while a counterfactual query talks about a specific individual!

To get an answer to our question we have to step beyond causal graphs and introduce another concept: **structural equation models**.

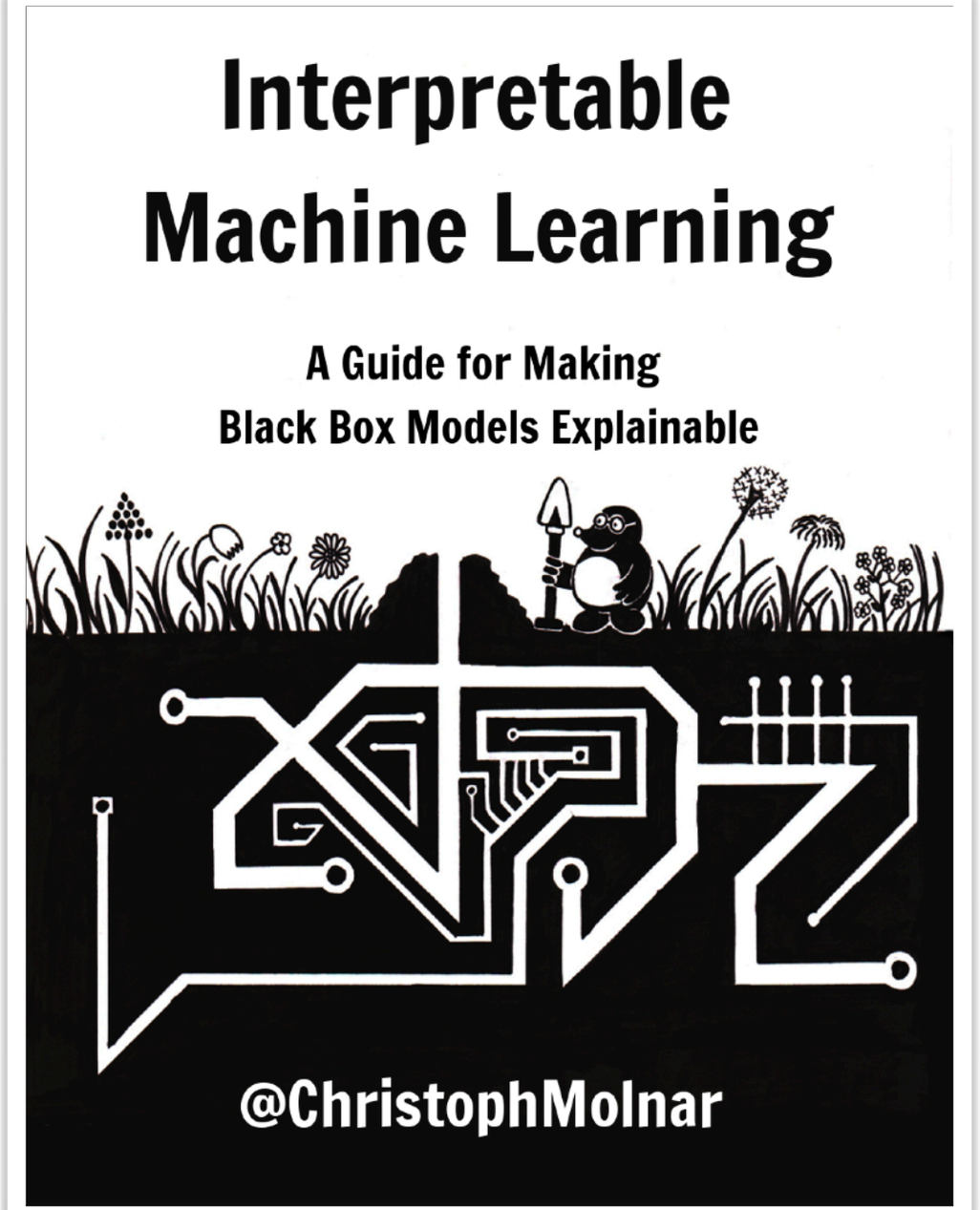
Introduction to Causal Inference

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 similar counterfactual that is both close to the actual datapoint and likely to occur in the real world.

Resources



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