UNIVERSITY of IOANNINA

Universität der Bundeswehr München

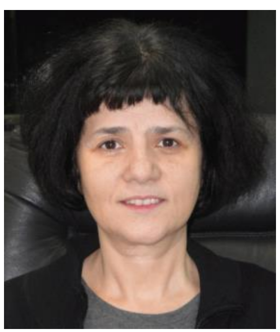
CODE FI

ARCHIMEDES

Tampere University

Fairness and Explainability in AI

Models, Measures, and Mitigation Strategies



Evaggelia Pitoura

Panayiotis Tsaparas

Eirini Ntoutsi

Kostas Stefanidis

ESSAI 2024, Athens (July 15 – July 19, 2024)

Course overview

Lecture 1 - Bias and discrimination in AI systems: Sources of bias, definitions and models of fairness

- Motivation and application examples of algorithms exhibiting biased behaviour
- Different types of bias and their cause
- Definitions of fairness

Lecture 2. Bias mitigation

- Pre-, In- and Post-processing approaches to fairness-aware learning
- End-to-end approaches to fairness-aware learning

Lecture 3. Solutions for mitigating unfairness in concrete contexts

- Fairness in rankings and recommendations, entity resolution, graphs

Lecture 4 - Explainable AI: Models and methods

- Introduction to explainable AI (XAI)
- Overview of post-hoc explanations
- LIME, Shapley values, counterfactual explanations

Lecture 5 - Connections between fairness and explanations

- Counterfactual explanation of unfairness
- Actionable recourse
- Shapley-based and data-based explanations of unfairness
- Fairness of explanations

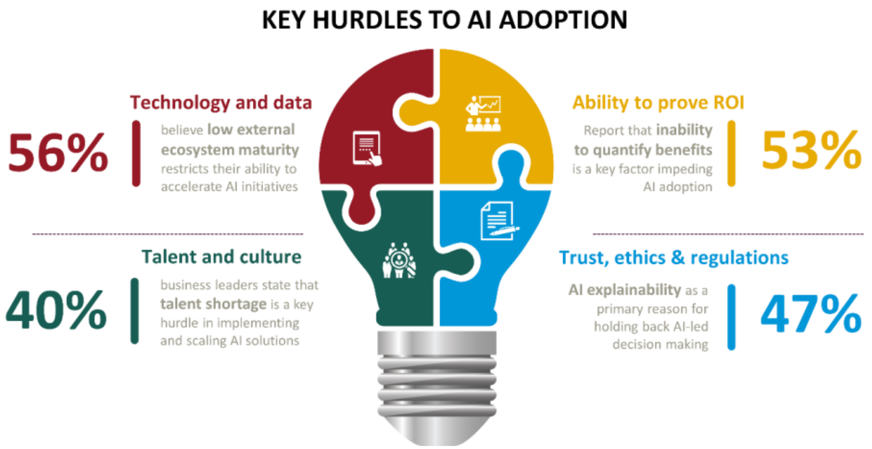
Outline

- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Outline

- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Impediments to AI adoption



Technology and data

- Low external ecosystem maturity (56%)
- Low digitization (53%)
- Inadequate training data (36%)
- Disparate datasets (34%)

Ability to quantify ROI

- Inability to objectively quantify benefits (53%)
- Inadequate number of use cases (44%)

Talent and culture

- Talent Shortage (40%)
- Cultural or behavioral impediments (38%)
- Workforce Displacement (33%)

Trust, ethics and regulations

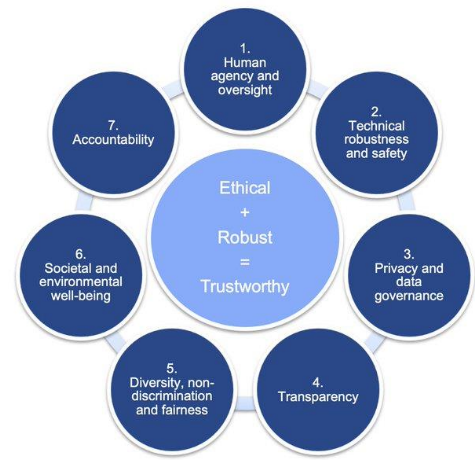
- AI Explainability (47%)
- Unintended consequence of AI decisions (38%)
- Regulations & Compliance (36%)
- Bias (20%)
- Ethics (18%)

Source: Link

Responsible/Trustworthy AI: Key principles and requirements

- A growing interest in principles, tools, and best practices for deploying AI ethically and responsibly.
- **4 Ethical Principles**
  - Respect for human autonomy
  - Prevention of harm
  - Fairness
  - Explicability — capable of being explained
- **7 Key Requirements**
  - Human agency and oversight
  - Technical robustness and safety
  - Privacy and data governance
  - Transparency
  - Diversity, non-discrimination and fairness
  - Societal and environmental wellbeing
  - Accountability

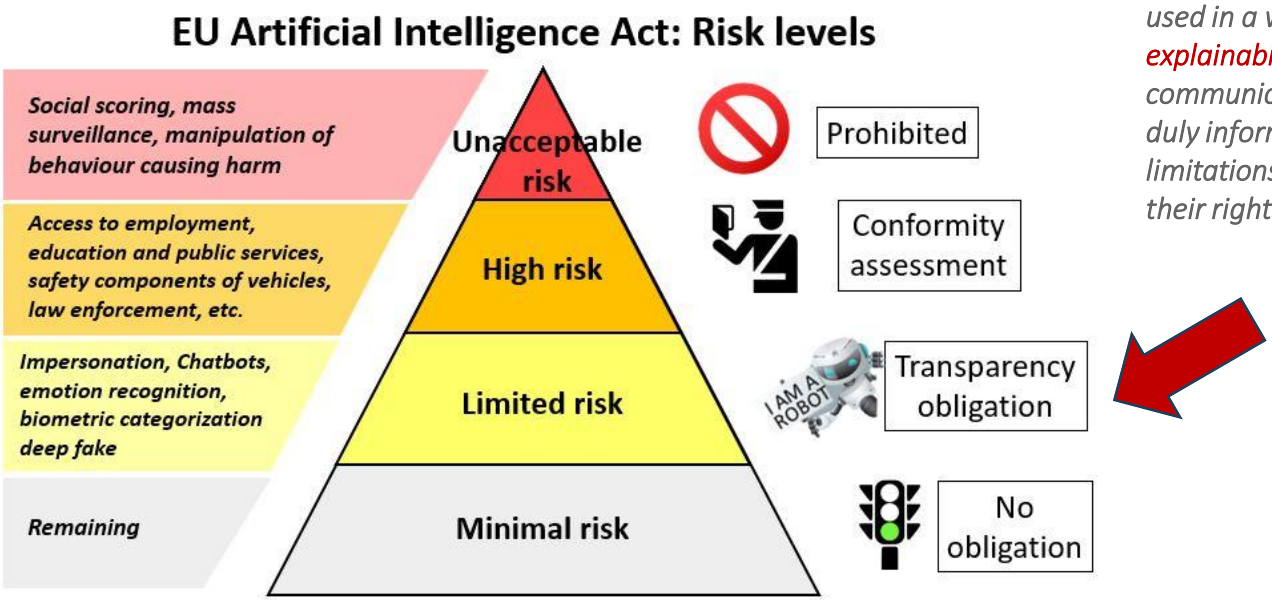
Seven key requirements



Source: https://ec.europa.eu/futurium/en/ai-alliance-consultation.1.html

The AI Act

- The AI Act is a proposed European law on artificial intelligence (AI) – the first law on AI by a major regulator anywhere. The law assigns applications of AI to three risk categories.



Transparency means that AI systems are developed and used in a way that allows appropriate traceability and explainability, while making humans aware that they communicate or interact with an AI system, as well as duly informing deployers of the capabilities and limitations of that AI system and affected persons about their rights.

Source: Link

Source: Image

Growing global AI regulations

SR 11-7: Guidance on Model Risk Management

BOARD OF GOVERNORS
OF THE FEDERAL RESERVE SYSTEM
WASHINGTON, D.C. 20551

What's driving Stress Testing and Model Risk Management efforts?

Regulatory efforts

SR 11-7 says "Banks benefit from **conducting model stress testing** to check performance over a wide range of inputs and parameter values, including extreme values, **to verify that the model is robust**"

In fact, **SR14-03** explicitly calls for **all models used for Dodd-Frank Act Company-Run Stress Tests must fall under the purview of Model Risk Management.**

In addition **SR12-07** calls for **incorporating validation or other type of independent review of the stress testing framework to ensure the integrity of stress testing processes and results.**

GDPR

Article 22. Automated individual decision making, including profiling

1. The data subject shall have the right not to be subject to a decision based solely on automated processing, including profiling, which produces legal effects concerning him or her or similarly significantly affects him or her.
2. Paragraph 1 shall not apply if the decision:
   (a) is necessary for entering into, or performance of, a contract between the data subject and a data controller;
   (b) is authorised by Union or Member State law to which the controller is subject and which also lays down suitable measures to safeguard the data subject's rights and freedoms and legitimate interests; or
   (c) is based on the data subject's explicit consent.
3. In the cases referred to in points (a) and (c) of paragraph 2, 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.
4. Decisions referred to in paragraph 2 shall not be based on special categories of personal data referred to in Article 9(1), unless point (a) or (g) of Article 9(2) apply and suitable measures to safeguard the data subject's rights and freedoms and legitimate interests are in place.

CALIFORNIA
CONSUMER
PRIVACY
ACT OF 2018

Credit: Lecueet al., Tutorial on XAI. AAAI 2020. https://xaitutorial2020.github.io/

Growing global AI regulations

- GDPR “right to explanation”: Article 22 empowers individuals with the right to demand an explanation of how an automated system made a decision that affects them.
- Algorithmic Accountability Act 2019: Requires companies to provide an assessment of the risks posed by the automated decision system to the privacy or security and the risks that contribute to inaccurate, unfair, biased, or discriminatory decisions impacting consumers
- California Consumer Privacy Act: Requires companies to rethink their approach to capturing, storing, and sharing personal data to align with the new requirements by January 1, 2020.
- Washington Bill 1655: Establishes guidelines for the use of automated decision systems to protect consumers, improve transparency, and create more market predictability.
- Massachusetts Bill H.2701: Establishes a commission on automated decision-making, transparency, fairness, and individual rights.
- Illinois House Bill 3415:States predictive data analytics determining creditworthiness or hiring decisions may not include information that correlates with the applicant race or zip code.

Credit: Lecueet al., Tutorial on XAI. AAAI 2020. https://xaitutorial2020.github.io/

XAI as a key requirement

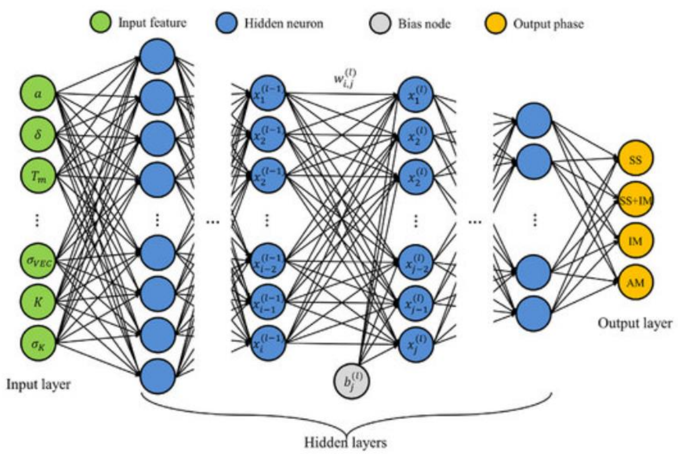
- Early phases of AI adoption
  - Ok to not fully understand how the model predicts, as long as the accuracy is high
- Shifting focus
  - Recognition of the importance of understanding the decision-making processes of AI systems.
  - Emphasis on building human interpretable models.
- Why it becomes important?
  - Trust: XAI helps us build trust in AI systems by explaining their decisions.
  - Transparency: XAI helps in understanding potential biases, limitations and risks in AI systems.
  - Accountability: It can help us hold AI systems accountable for their decisions.

Outline

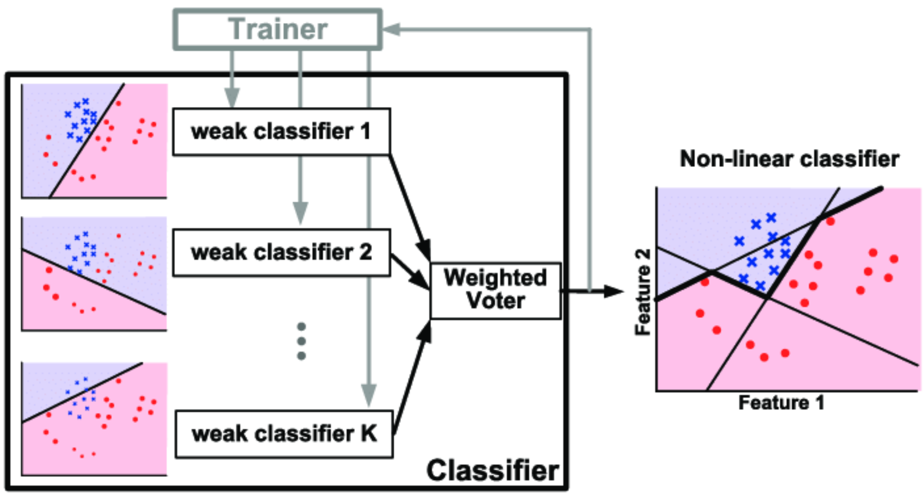
- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Black – vs white box models

- A Black Box model is a system that does not reveal its internal mechanisms.
  - In machine learning, "black box" describes models that cannot be understood by looking at their parameters
  - Examples of black-box models: neural networks, ensembles, SVMs, …



Source: link

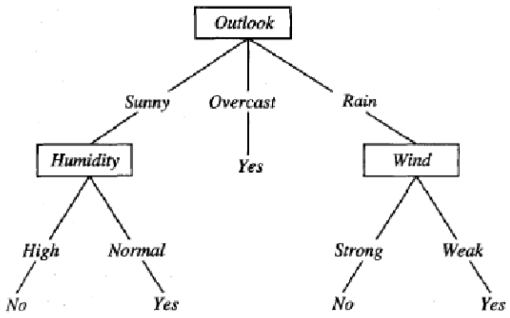


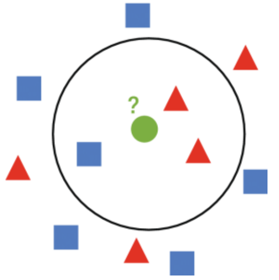
Black – vs white box models

- The opposite of a black box is sometimes referred to as White Box (or, interpretable model).
  - Linear regression, logistic regression and the decision tree are commonly used interpretable models.

Algorithm	Linear	Monotone	Interaction	Task
Linear regression	Yes	Yes	No	regr
Logistic regression	No	Yes	No	class
Decision trees	No	Some	Yes	class,regr
RuleFit	Yes	No	Yes	class,regr
Naive Bayes	No	Yes	No	class
k-nearest neighbors	No	No	No	class,regr

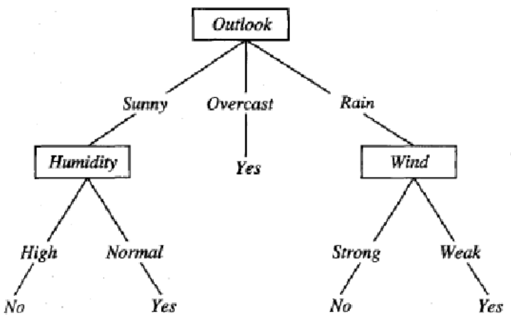
Source: Link

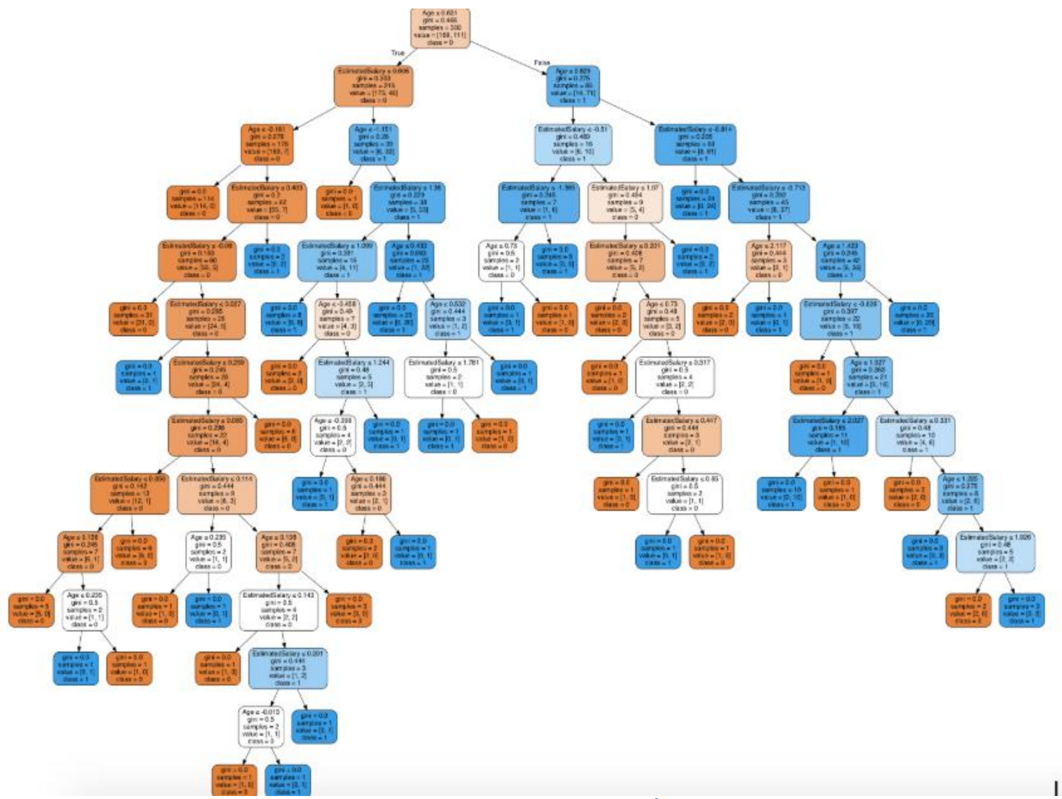




Black – vs white box models

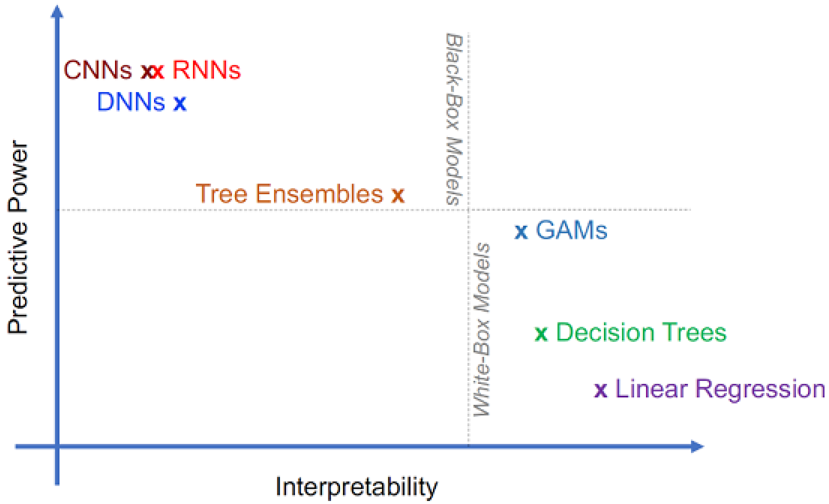
- We could argue whether such models are always interpretable (e.g., a very long decision tree)



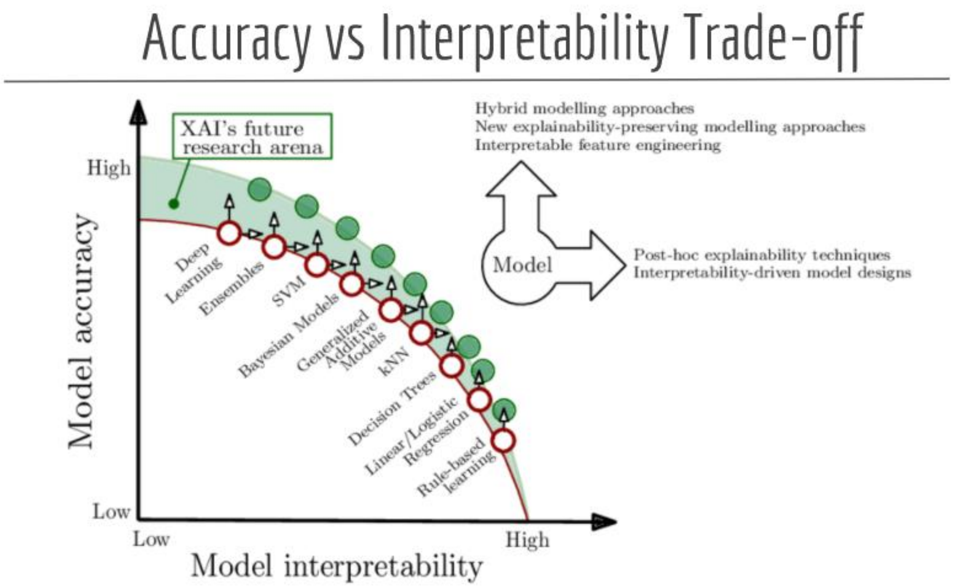


Source: [Link]()

Accuracy interpretability trade-off



Source: [Link]()



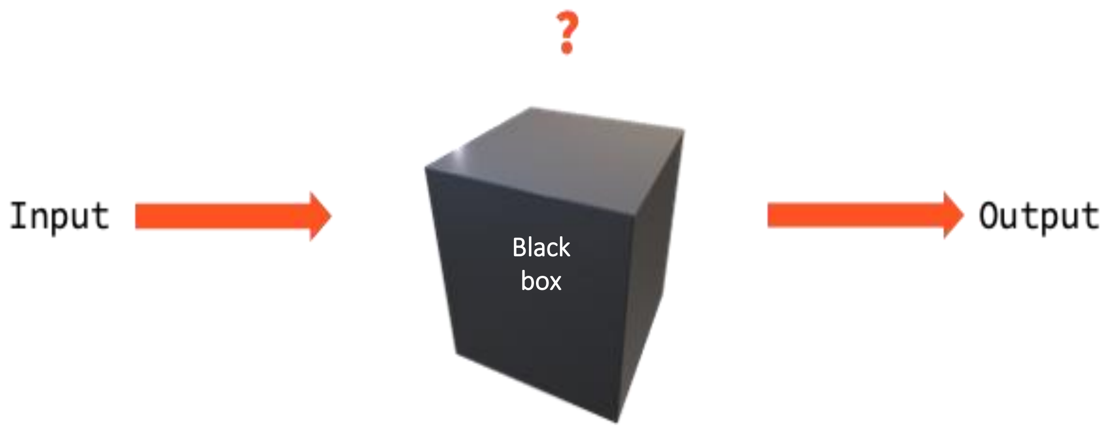
Black – vs white box models

- **2 directions**
  - Build inherently interpretable models
    - i.e., white models
  - Post-hoc explanations for black-box models
    - Assume black-box models and create a second (post-hoc) model to explain the first black-box model
    - Apply methods that analyze the model after training (post-hoc) (Carvalho et al., 2019)
- Advice:
  - If you can build an interpretable model which is also adequately accurate for your setting, do it!
  - Otherwise, post-hoc explanations come to the rescue.

D. V. Carvalho, E. M. Pereira, & Jaime S. Cardoso (2019). Machine Learning Interpretability: A Survey on Methods and Metrics. Electronics, 8, 832.

Why we need XAI?

- Many AI systems nowadays are black boxes.
  - As an example, ChatGPT 4 has 1.76 trillion parameters
- Post-hoc explanations are therefore necessary



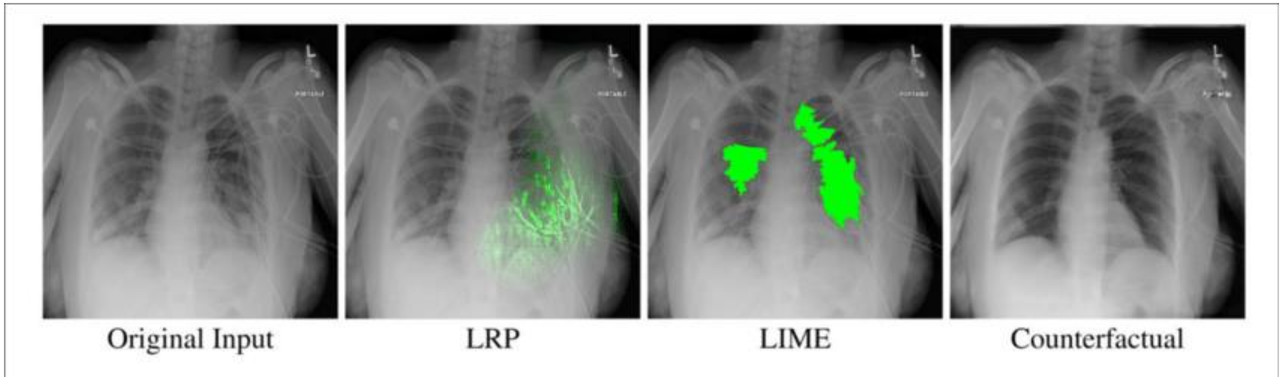
Explainability is a versatile tool for different types of users 1/3

- For end users, that "consume" the technology, to understand how a certain decision was made (GDPR "right to explanation")
  - In healthcare: "*Why was I classified as a high-risk patient for COVID?*"
  - In credit scoring: "*Why was my credit application rejected?*"
  - In predictive policing: "*Why was I selected for police inspection?*"
- And moreover:
  - "*Am I being treated fairly?*"
  - "*Can I contest the decision?*"
  - "*What could I do differently to get a positive outcome?*"
    - In credit scoring:"*What should I change in my application to get a loan?*"

Based on Fosca Giannoti (2022) keynote, ECMLPKDD ([link]() to a previous version of the slides)

Explainability is a versatile tool for different types of users 2/3

- For professionals that make decisions with (the help of) AI, to ensure that decisions are correct and in accordance with legal and societal standards (e.g., no discrimination)
  - E.g., An example x-ray image classified as Pneumonia, as well as the different XAI visualizations



Source: Link

Based on Fosca Giannoti (2022) keynote, ECMLPKDD (link to a previous version of the slides)

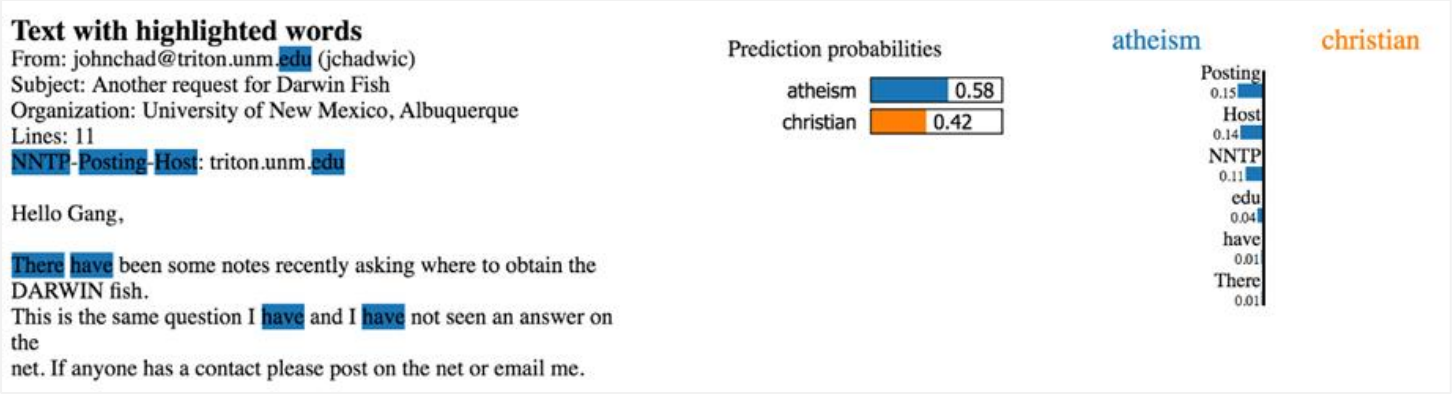
Explainability is a versatile tool for different types of users 3/3

- For AI technology developers, as an inspection/debugging tool, to ensure that the technology is robust *"Is my system working as designed?"*
  - Right decisions for the right reasons
  - Insights on how to improve model performance

Based on Fosca Ciannoti (2022) keynote, ECMLPKDD (link to a previous version of the slides)

XAI as an inspection/debugging tool

- Explaining a text classification: text is classified correctly but for the wrong reasons.
  - Actionable insights: The explanation reveals that the model focuses on html tags, common words,…

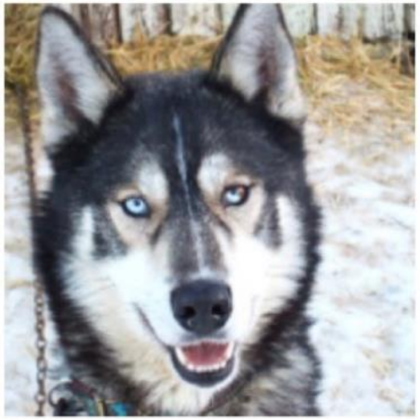


<u>Your ideas</u>:
What could have gone wrong during training?
How can we improve the model?

Source: Ribeiro et al, 2016

XAI as an inspection/debugging tool

- Explaining an image: the image is wrongly classified as a wolf
  - Actionable insights: The explanation reveals that the model focuses on the snow in the background.



(a) a Husky misclassified as a Wolf

(b) The Explanation shows the classifier only concentrate on the background

Your ideas:
What could have gone wrong during training?
How can we improve the model?

Source: [Ribeiro et al, 2016]

XAI for bias detection

- Explaining a text classification: text is classified wrongly as hate speech
  - Actionable insights: the explanation reveals that the model is oversensitive to group identifiers and unable to identify the context in which these words are used (Kennedy et al, 2020).

"[F]or many Africans, the most threatening kind of ethnic hatred is black against black." - *New York Times*

"There is a great discrepancy between whites and blacks in SA. It is … [because] blacks will always be the most backward race in the world." Anonymous user, *Gab.com*

Two documents which are classified as hate speech by a fine-tuned BERT classifier. Group identifiers are underlined.

muslim jew jews white islam blacks muslims women whites gay black democat islamic allah jewish lesbian transgender race brown woman mexican religion homosexual homosexuality africans

List of identity terms for bias detection

Your ideas:
What could have gone wrong during training?
How can we improve the model?

Outline

- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Overview of explanation methods

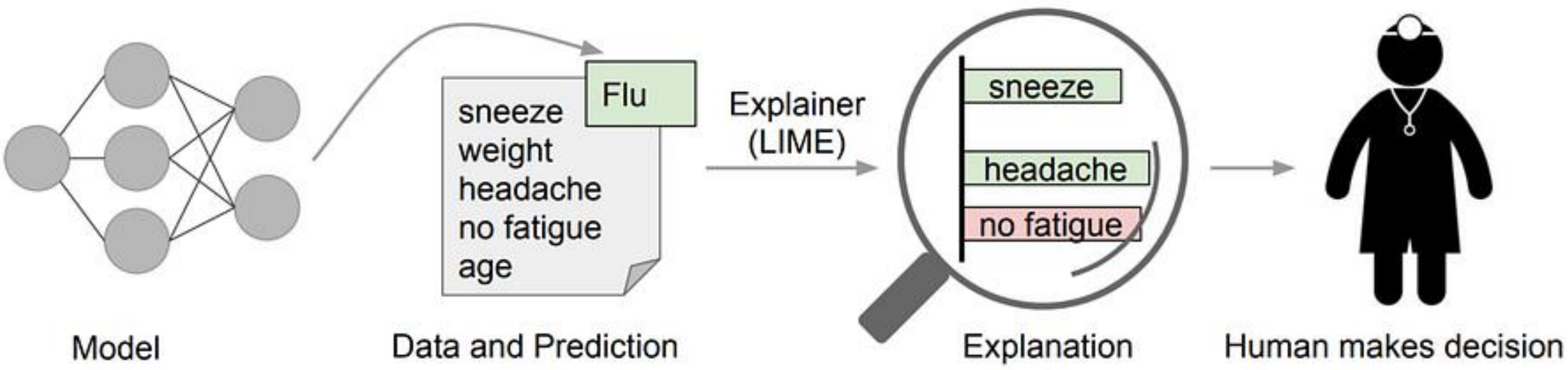
- Two general categories: Methods can explain a specific prediction (local), or the overall logic of the model (global)
- Local (or instance-based) explanations
  - Provide an explanation for a specific instance.
  - Focus on the decision-making process for a single instance rather than the entire model.
  - Representative methods:
    - Feature importance/attribution methods (LIME, Shapley, …), Saliency maps, Prototype-/example-based, Counterfactual, …
- Global explanations
  - Explain the overall behavior of the model across the entire dataset.
  - Provide a holistic view of how the model makes decisions based on the overall patterns it has learned.
  - Representative methods
    - Global feature importance (aggregated Shapley values), Accumulated local effects (ALE), Model distillation/ Global surrogate model, Partial dependence plots (PDP)

Outline

- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Local explanation methods

- Explain predictions on a single instance.
- A motivating example: Consider a clinic using AI to diagnose patients' illnesses. In this scenario, the AI application processes a patient's symptoms and related information, utilizes an AI model, and concludes that the symptoms align with those of the flu. The doctor can subsequently examine the results and initiate appropriate treatment
  - It is important for the doctor to understand why the model predicted “flue” and what were the key factors for the prediction
    - LIME: Sneeze and headache are portrayed as contributing to the “flu” prediction, while “no fatigue” is evidence against it

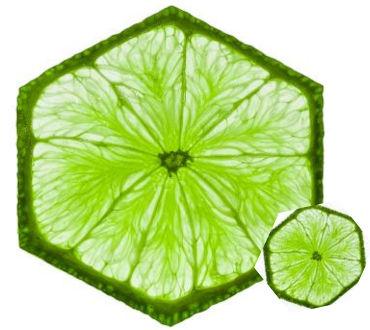


Source: [Link]

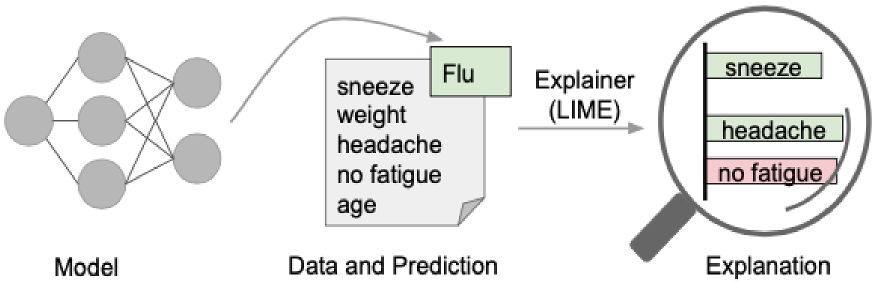
Outline

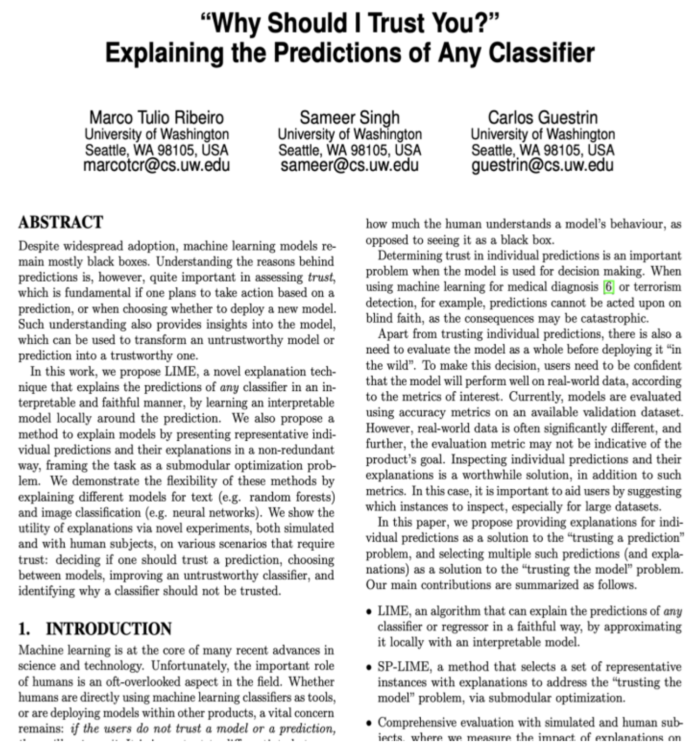
- Growing XAI requirements
- Key concepts
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

LIME (Ribeiro et al, 2016)



- LIME (**L**ocal **I**nterpretable **M**odel-agnostic **E**xplanations)
- One of the most popular methods in XA





"Why Should I Trust You?" Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and identifying why a classifier should not be trusted.

1. INTRODUCTION

Machine learning is at the core of many recent advances in science and technology. Unfortunately, the important role of humans is an oft-overlooked aspect in the field. Whether humans are directly using machine learning classifiers as tools, or are deploying models within other products, a vital concern remains: *if the users do not trust a model or a prediction,*

how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem. Our main contributions are summarized as follows.

- LIME, an algorithm that can explain the predictions of *any* classifier or regressor in a faithful way, by approximating it locally with an interpretable model.
- SP-LIME, a method that selects a set of representative instances with explanations to address the "trusting the model" problem, via submodular optimization.
- Comprehensive evaluation with simulated and human subjects, where we measure the impact of explanations on

Ribeiro, M. T., Singh, S., & Guestrin, C. [``Why should i trust you?" Explaining the predictions of any classifier](). KDD 2016.

LIME

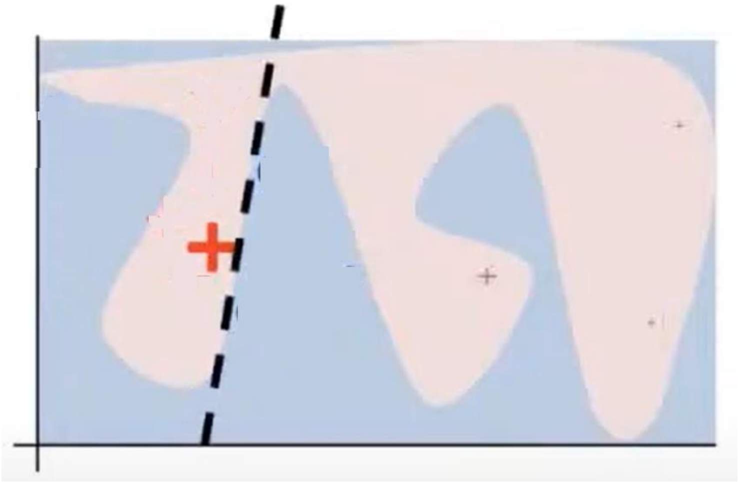
- LIME a technique that approximates any black box machine learning model with a local, interpretable model to explain individual instance predictions.

Local
Interpretable
Model-agnostic
Explanations

- **L**ocal: Replicates the model's behavior locally and can explain individual predictions.
- **I**nterpretable: Provides a qualitative understanding between the input variables and the response.
- **M**odel-agnostic: Treats the model as a black box.
- **E**xplanations: Uses locally weighted interpretable models

How it works?

- **Input:**
  - A black box model *f()*: for a given input *x* *(marked as +)* it can provide an output/prediction *f(x)*
  - The instance *x* to be explained
- **Goal:**
  - For the input instance *x*, explain the decision *f(x)* of the black box model *f()*
- **Key idea:**
  - Build a transparent surrogate model *g()* in the neighborhood of the instance *x* to simulate the local behavior of the black box f().

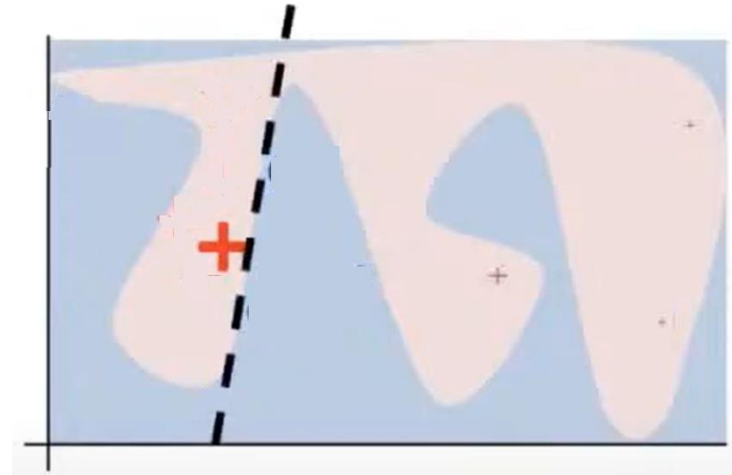


Black-box model:

- complex model (decision boundary shown in blue/pink background)
- Cannot be easily approximated by a linear model (dotted black line)

Key steps

- **Step 1:** Sample points around x
- **Step 2:** Use the black box model *f()* to predict their labels
- **Step 3:** Weight points based on their distance to *x*
- **Step 4:** Learn an interpretable model *g()* on the weighted samples



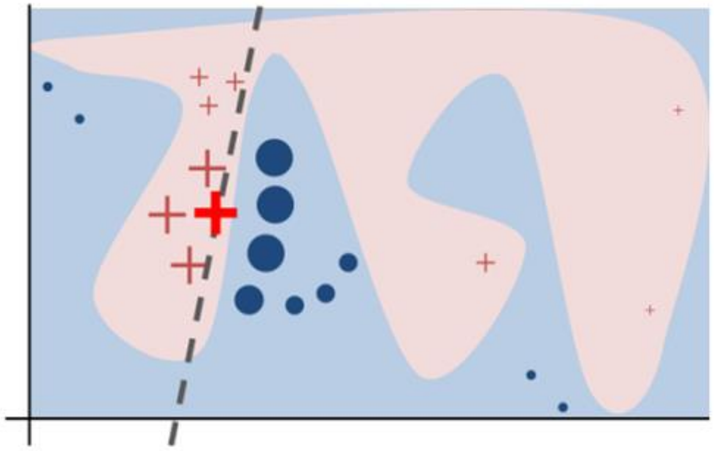
Black-box model:

- complex model (decision boundary shown in blue/pink background)
- Cannot be easily approximated by a linear model (denoted by the dotted black line)

Key steps: Step 1

Step 1: Sample points around *x*

- Create a neighborhood *N* of similar instances around *x*



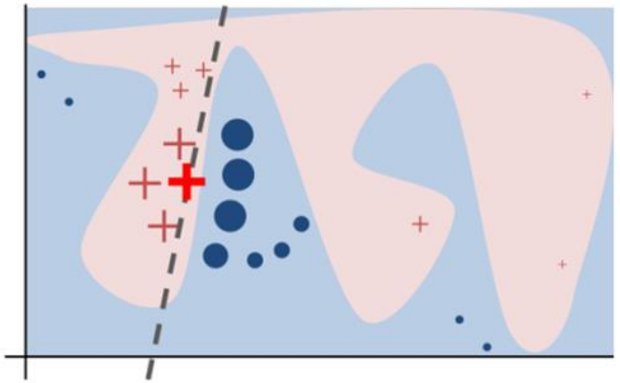
Ignore the colors, size and symbols of the instances for the moment

Key steps: Step 2

Step 1: Sample points around *x* → *local neighborhood N*

Step 2: Use the black box model *f()* to predict their labels

- For each instance *x'* in *N*, predict *f(x')* using the block box *f()*



Color and symbol indicates the class predicted by the black box.
Ignore the size of the instances for the moment

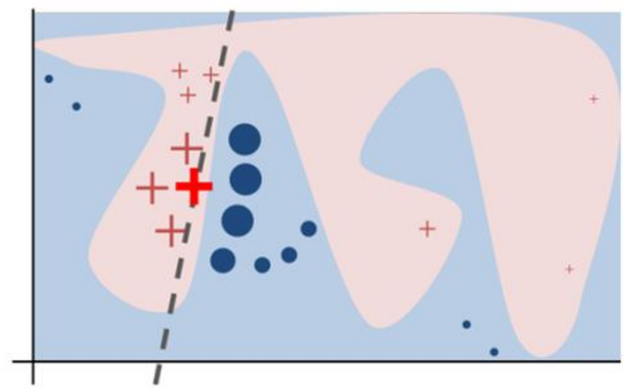
Key steps: Step 3

Step 1: Sample points around x → *local neighborhood N*

Step 2: Use the black box model *f()* to predict their labels → labeled local neighborhood *N*

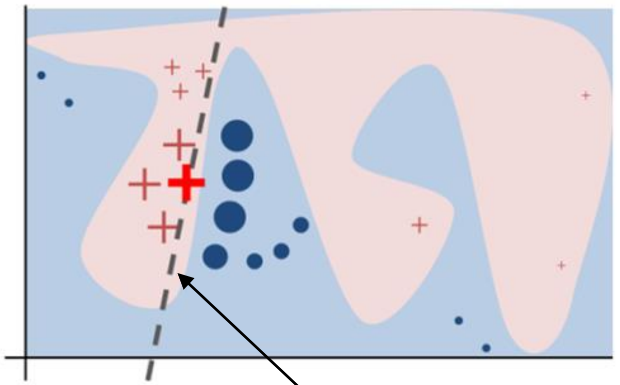
Step 3: Weight points based on their distance to x

- Higher weights for nearby instances
- Lower weights for far away instances



Color and symbol indicates the class predicted by the black box.
Size indicates the proximity to *x*

Key steps: Step 4



Interpretable model: in this case a linear classifier

Step 1: Sample points around *x* → *local neighborhood N*

Step 2: Use the black box model *f()* to predict their labels → labeled local neighborhood *N*

Step 3: Weight points based on their distance to *x* → *labeled weighted local neighborhood N*

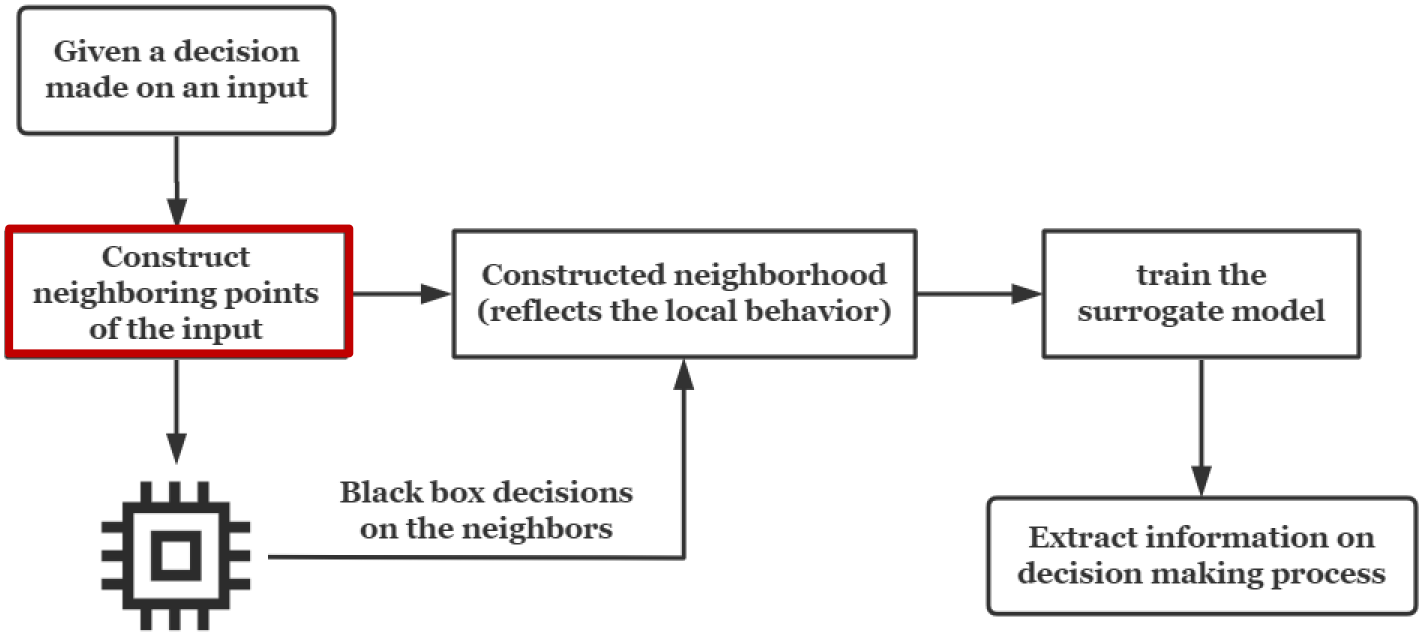
Step 4: Learn an interpretable model *g()* on the weighted samples

- Training set: the weighted samples.
- Choose from the class of interpretable models, e.g., a linear classifier
- The local model g() must correspond to how the model f() behaves in the vicinity of the instance being predicted (local fidelity)
- The complexity of *g()* can be further controlled to improve interpretability
  - For decisions trees, it can be the depth of the tree
  - For linear regression, it can be the number of features with non- zero weight
- Fidelity-Interpretability trade-off

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \boxed{\mathcal{L}(f, g, \pi_x)} + \boxed{\Omega(g)}$$

π_x is the neighborhood of x

LIME overview: reflection on key components



Neighborhood selection

- The definition of the neighborhood *N* around *x* is critical as it comprises the training set for the local classifier
- Recall that we don't have access to the training data of the black box model
- So how can we create a local neighborhood around *x*?
- In LIME, the neighborhood is created by *perturbing the input instance x*
- The perturbation depends on the data type (tabular, text, images)
  - For text and images: create new samples by turning single words or super-pixels on and off
  - For tabular data: create new samples by perturbing each feature individually, drawing from a normal distribution with mean and standard deviation taken from the feature.

Neighborhood selection: text data

- Source: Molnar Christoph
- Dataset: YouTube comments
- Model: a model that predicts whether a YouTube comment is spam or normal

	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

- The neighborhood of an instance is created by perturbing the instance (turning words on and off)

For	Christmas	Song	visit	my	channel!	;)	prob	weight
1	0	1	1	0	0	1	0.17	0.57
0	1	1	1	1	0	1	0.17	0.71
1	0	0	1	1	1	1	0.99	0.71
1	0	1	1	1	1	1	0.99	0.86
0	1	1	1	0	0	1	0.17	0.57

Your ideas:
What can go wrong with the pertubations?

Neighborhood selection: image data

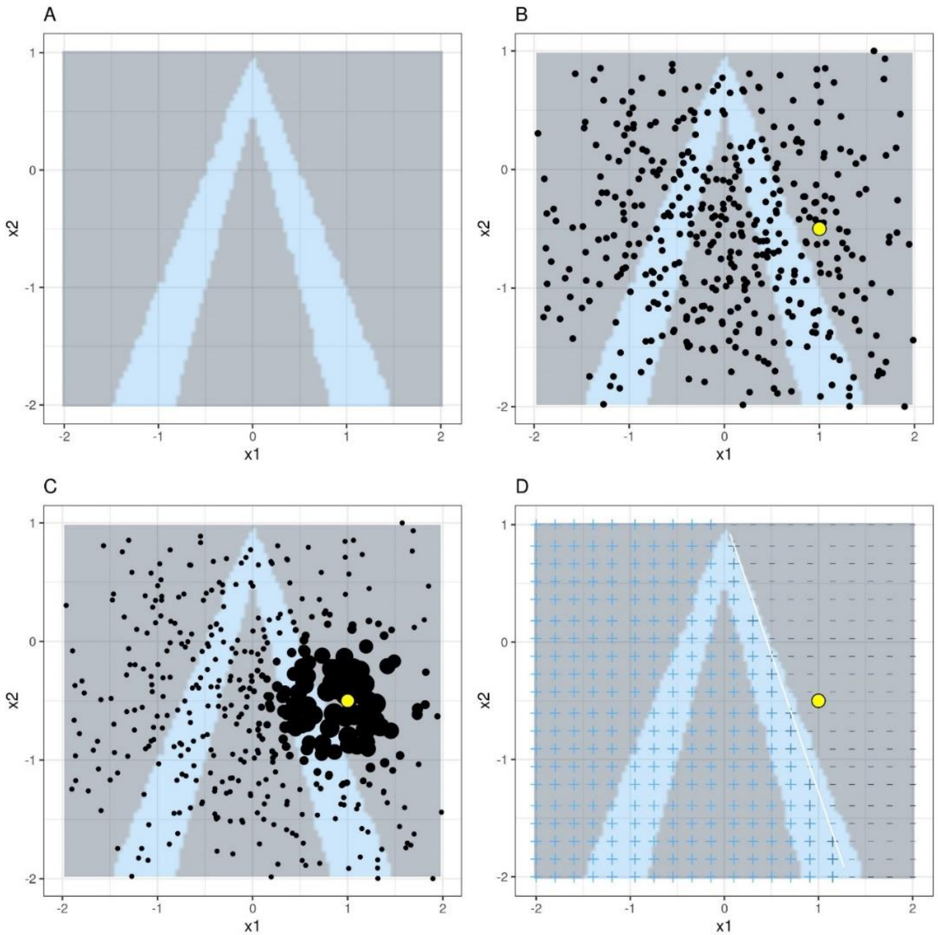
- Naïve idea: randomly change pixels
  - Likely won't affect the prediction much since >1 pixels contribute to a class.
  - Instead, create image variations by segmenting into "superpixels" and turning them on or off.
    - Superpixels are interconnected pixels with similar colors and can be turned off by replacing each pixel with a user-defined color such as gray.
    - The user can also specify a probability for turning off a superpixel in each permutation.

Neighborhood selection: image data

- What does LIME really see in images? Garreau & Mardaoui, 2021 ([paper]())

Figure 2. Sampling procedure of LIME for images. The image to explain, ξ, is first split into d superpixels (*lower left corner*, here $d = 72$). A replacement image $\bar{\xi}$ is computed, which is by default the mean of ξ on each superpixel (*top row*), see Eq. (1). This replacement image can also be filled uniformly with a pre-determined color (*bottom row:* replacement with the color black). Then, for each new generated example x_i with $1 \leq i \leq n$, the superpixels are randomly switched depending on the throw of d independent Bernoulli random variables with parameter $1/2$. Thus LIME creates n new images where key parts of ξ disappear at random.

Neighborhood selection: tabular data



Source: Molnar Christoph

FIGURE 9.5: LIME algorithm for tabular data.

A) Random forest predictions given features x1 and x2. Predicted classes: 1 (dark) or 0 (light).
B) Instance of interest (big dot) and data sampled from a normal distribution (small dots).
C) Assign higher weight to points near the instance of interest.
D) Signs of the grid show the classifications of the locally learned model from the weighted samples. The white line marks the decision boundary (P(class=1) = 0.5).

LIME: discussion

- **Advantages**
  - Model-agnostic
    - can explain the decisions of any ML model, regardless of its complexity. This makes it a versatile tool for XAI
  - Generates local explanations
    - useful in many practical situations
- **Limitations**
  - Sensitive to perturbations (for the local neighborhood of the instance)
    - Small changes in the instance might result in different explanations
  - The choice of the distance function to assess proximity between a point and the instance to be explained can affect the explanations.
    - Which function to use?
    - Challenges with high dimensional data, mixed-data types, …
    - Approaches exist that work on the latent-space, e.g., Cai et al, 2023, Lambridis et al, 2020

Outline

- Introduction - Growing XAI requirements
- Explanations in a nutshell
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

SHAP

- SHAP (SHapley Additive exPlanations) by Lundberg and Lee (2017).
- Another popular method in XAI

SHapley
Additive
ExPlanations

SHAP

A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
slund1@cs.washington.edu

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
suinlee@cs.washington.edu

Abstract

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between *accuracy* and *interpretability*. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

1 Introduction

The ability to correctly interpret a prediction model's output is extremely important. It engenders appropriate user trust, provides insight into how a model may be improved, and supports understanding of the process being modeled. In some applications, simple models (e.g., linear models) are often preferred for their ease of interpretation, even if they may be less accurate than complex ones. However, the growing availability of big data has increased the benefits of using complex models, so

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems (pp. 4768–4777).

SHAP

- SHAP is a technique that computes the contribution of each attribute to the final prediction(s).

SHapleyl

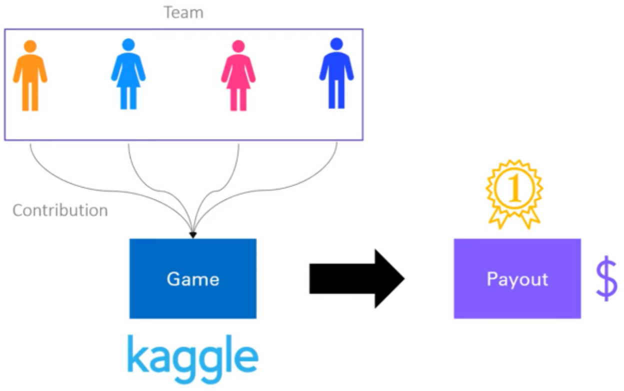
Additive

Ex**P**anations

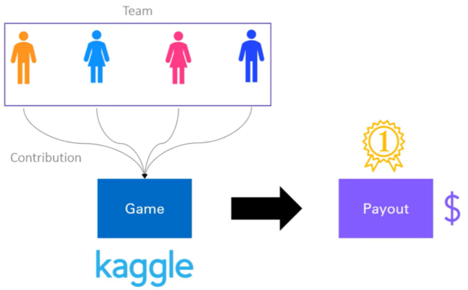
- SHapley: Based on Shapley values from game theory
- Additive: the contribution of each feature to the final prediction can be computed independently and then summed up.
- ExPlanations

Motivation

- Idea behind SHAP comes from cooperative game theory
- Cooperative games: In a cooperative game, “players” have the possibility to forge coalitions to achieve a common goal. After the game is over, the coalition gets a certain payout/benefit/gain for the results.
- Key question: How should the money be distributed among the team?
- Example: a team of data scientists, cooperate in a Kaggle competition and won the first prize. How the prize should be distributed among the team members?



Motivation



- Key question: How should the money be distributed among the team?
- One idea: Equal distribution among the players. Is this a good idea?
- Key intuition:
  - Some players may contribute more to the coalition than others (for example, an ML expert in the Kaggle team) or may possess different bargaining power (for example, threatening to destroy the whole surplus)
- Rephrased questions:
  - How important is each player to the overall cooperation, and what payoff can he or she anticipate as a result?
  - How interactions between players should be considered?
- One possible answer: Shapley values (term coined by Shapley (1953))
  - Shapley won the Nobel Memorial Prize in Economic Sciences for it in 2012.


Lloyd Shapley in 2012

Shapley, Lloyd S. "A value for n-person games." Contributions to the Theory of Games 2.28 (1953): 307-317.

Shapley values: Notation

- Assume a coalition of N players (grand coalition).
  - For example, the 4 team members in the Kaggle competition
- S ⊆ N is a subset of participants of the grand coalition N (partial coalition).
- v is a value function that maps subsets of players S to a real number

$$v(S) \equiv \text{"}\mathit{the\ revenue\ of\ the\ coalition}\ S\text{"}$$

  - v(N) is the value function of the grand coalition. In our example, the value generated by all players is 100 credits: v(N)=100K.
- When a player i joins a set of players S, the marginal contribution of player i to S is:

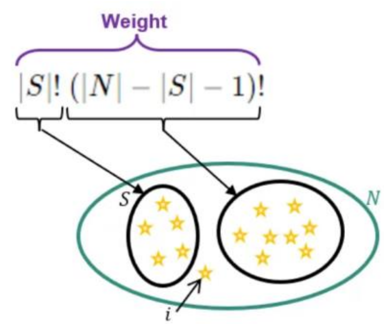
$$v(S \cup \{i\}) - v(S)$$

  - The marginal contribution measures the value that player i added when (s)he joined the group of players S. This contribution can be zero, positive or even … negative.
- The Shapley value of player i tells us the average contribution of player i to the payout v(N)
  - Average over all possible ways to form a coalition

More formally

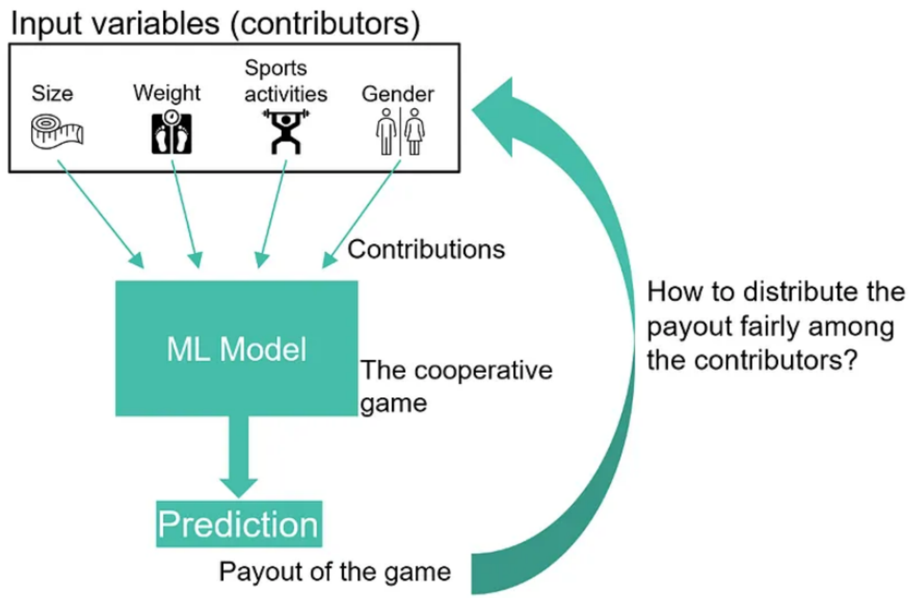
- Given a set of players N and a value function *v()*, the Shapley value of player *i* is a weighted average of the marginal contributions of *i* over all possible coalitions *S* of N.

$$\phi_i(N,v) = \boxed{\frac{1}{|N|!}} \sum_{S \subseteq N \setminus \{i\}} \boxed{|S|!\,(|N|-|S|-1)!} \boxed{[v(S \cup \{i\}) - v(S)]}$$



- Average: average the marginal contributions over all possible ways to form a coalition.
  - |N|! is the number of ways to arrange the grand coalition N.
- Weight: ensures that each marginal contribution is fairly averaged across all possible permutations and is the product of the number of ways to arrange coalition S (|S|!) and the number of ways to arrange the remaining players excluding i ((|N| - |S| -1)!).
- Marginal contribution: the marginal contribution of player i to the subset S

From games to XAI



- SHAP explanations: an XAI technique based on Shapley values used to determine how input variables contribute to output predictions.
- A prediction can be explained by assuming that each feature is a "player" in a game where the prediction is the payout.
  - Game: the prediction problem
  - Players: the features
  - Payout: the prediction for the instance
- So, Shapley values tell us how to distribute the payout/prediction among the features.
  - In other words, what are the feature contributions to model predictions

Shapley values

- The Shapley value of a feature is its contribution to the payout, weighted and summed over all possible feature combinations:

$$\phi_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{j\}\right) - val(S)\right)$$

- j: the feature of interest
- S: a subset of the features used in the model
- p: the number of features.

How to calculate the Shapley values: 2 key challenges

$$\phi_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{j\}\right) - val(S)\right)$$

- **Challenge 1:** The Shapley value is based on evaluating all possible combinations of players.
  - For a large number of features (e.g., pixels in an image, words in a document etc), calculating individual feature contributions becomes impractical as the number of coalitions exponentially increases as more features are added.
  - Key idea: use approximation
- **Challenge 2:** How to exclude a feature from a ML model?
  - We cannot just remove a feature, will affect the representation
  - Key idea: instead of removing a feature, set its value to a random value

Challenge 1: How to calculate the values?

- SHAP does not attempt to calculate the actual Shapley values.
- It uses sampling and approximations to calculate the SHAP values.
- Strumbelj & Kononenko, 2014 propose an approximation with Monte-Carlo sampling
  - *M*: number of iterations

$$\hat{\phi}_j = \frac{1}{M} \sum_{m=1}^{M} \left(\hat{f}\left(x^m_{+j}\right) - \hat{f}\left(x^m_{-j}\right) \right)$$

the prediction for x, but with a random number of feature values replaced by feature values from a random data point z, except for the respective value of feature j.

Almost identical to x_{+j}, but the value x_j is taken from z.

Pseudocode: Approximate Shapley estimation (Strumbelj & Kononenko, 2014)

Data Matrix X: where the samples should come from? Often implemented as a background dataset of instances from the domain

The procedure has to be repeated for each of the features to get all Shapley values.

- Output: Shapley value for the j^{th} feature
- Input: Instance *x*, feature *j*, data matrix *X*, ML model *f()*, number of iterations *M*
- For all *m=1 … M*
  - Draw random instance *z* from the data matrix *X*
  - Choose a random permutation of the feature values
  - Order instance x: $x_0=(x_1, \ldots, x_j, \ldots . x_p)$
  - Order instance z: $z_0=(z_1, \ldots, z_j, \ldots . z_p)$
  - Construct two new instances
    - With feature *j*: $x_{+j}=(x_1, \ldots, x_{j-1}, x_j, z_{j+1} \ldots . z_p)$
    - Without feature *j*: $x_{-j}=(x_1, \ldots, x_{j-1}, z_j, z_{j+1} \ldots . z_p)$
  - Computer marginal distribution of feature *j*:

$$\phi_j^m = \hat{f}\left(x_{+j}^m\right) - \hat{f}\left(x_{-j}^m\right)$$

- Compute Shapley value as the average:

$$\phi_j(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_j^m$$

//For each iteration, a random instance z is selected from the data and a random order of the features is generated

//Two new instances are created by combining values from the instance of interest x and the sample z.

//The instance x_{+j} is the instance of interest, but all values in the order after feature j are replaced by feature values from the sample z.

//The instance x_{-j} is the same as x_{+j}, but in addition has feature j replaced by the value for feature j from the sample z

//The difference in the prediction from the black box is computed:

SHAP: discussion

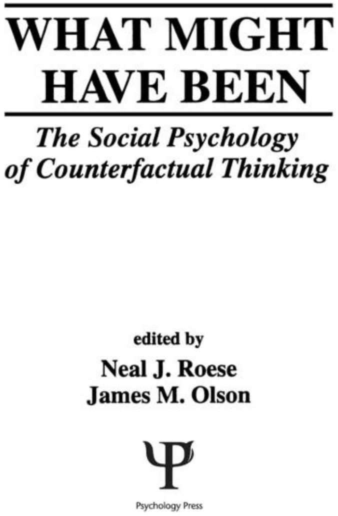
- **Advantages**
  - Grounded on game theory
  - Model-agnostic
    - can explain the decisions of any ML model, regardless of its complexity. This makes it a versatile tool for XAI
  - Generates local and global explanations
    - can provide both local explanations (for individual instances) and global explanations (aggregating feature importances across all instances).
- **Limitations**
  - Computational cost
  - Approximation necessity
  - The need for a background dataset
  - ….

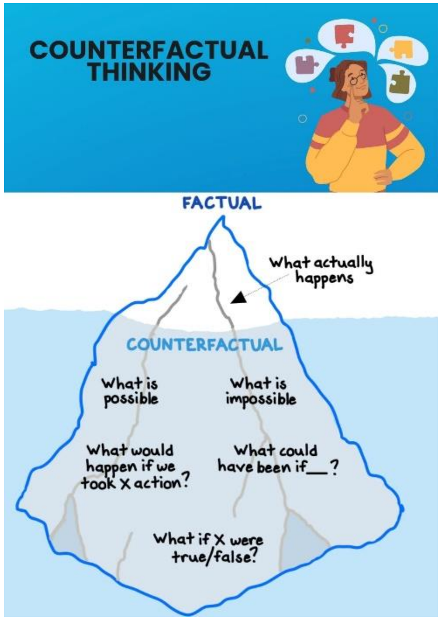
Outline

- Introduction - Growing XAI requirements
- Explanations in a nutshell
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Motivation for counterfactual explanations

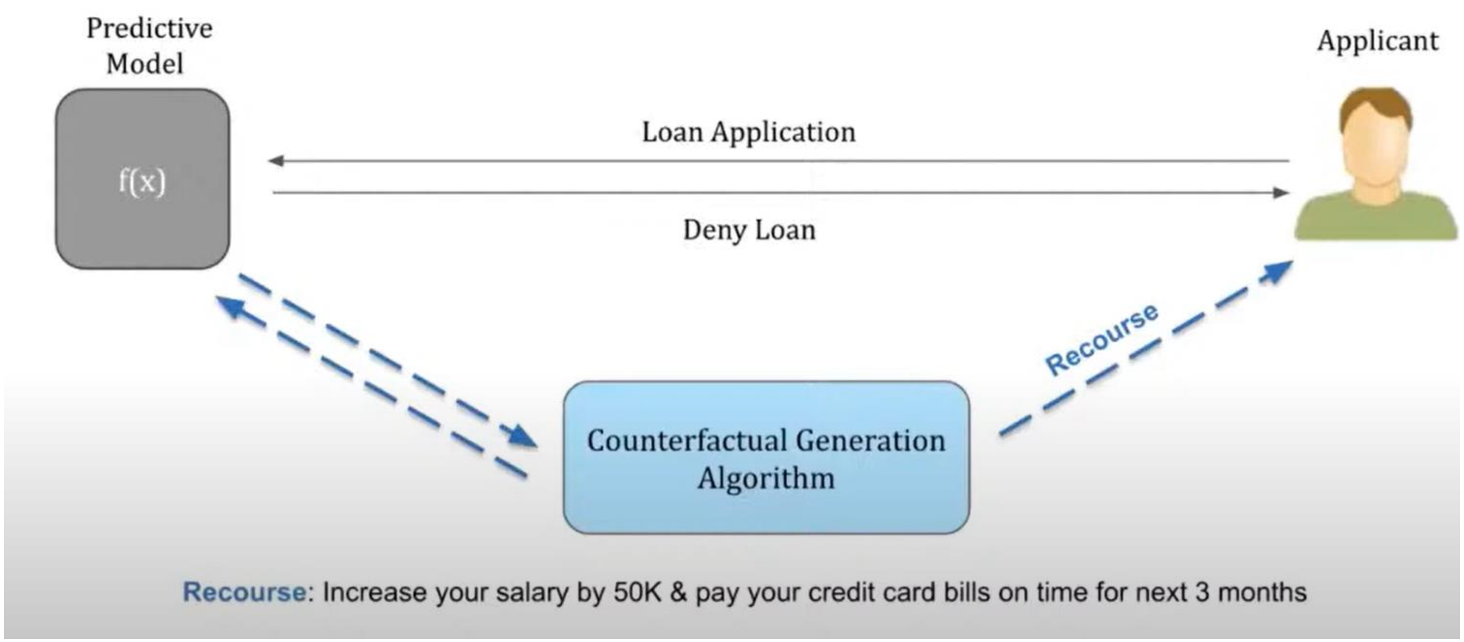
- Counterfactual thinking is a psychological term that refers to the human tendency to imagine alternative outcomes or scenarios that might have occurred in the past, present, or future, but did not actually happen.
- It involves mentally exploring "what if" scenarios and considering how things might be different under different circumstances.





Counterfactual explanations

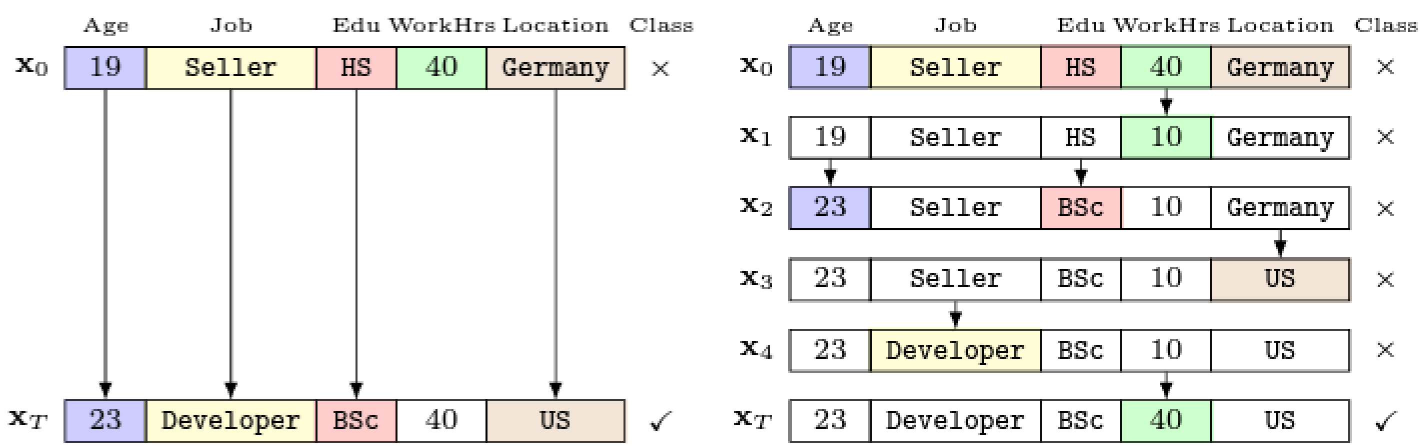
- What features need to be changed to flip the decision of a model? (Verma et al, 2020) → Counterfactual explanations (CFs)



Source: (Joshi et al, 2021)

Why are CFs useful?

- Counterfactuals are particularly useful because they offer both an explanation and actionable changes that can be applied to achieve a desired outcome.
- Example: How to attain a higher salary?



(a) Traditional counterfactual (b) Sequential counterfactual

Source: [Naumann and Ntoutsi, 2021]

What are counterfactual explanations (CFs)?

- CFs aim to determine the changes needed in the given input *x* to transform it into x' in order to alter the prediction outcome *f(x')* (Wachter et al., 2017)

$$\underbrace{x'}_{\text{Counterfactual}} = \underbrace{x}_{\text{Original input}} + \underbrace{\delta}_{\text{Explanation}} \; ; \quad x, x', \delta \in \mathcal{X} \subseteq \mathbb{R}^n$$

$$\text{s.t. } f(x') = y' \neq y = f(x); \quad y, y' \in \mathcal{Y}, \; f : \mathcal{X} \to \mathcal{Y}$$

- These changes (δ) are the explanation of the original prediction

- There are many possible *x'* …..

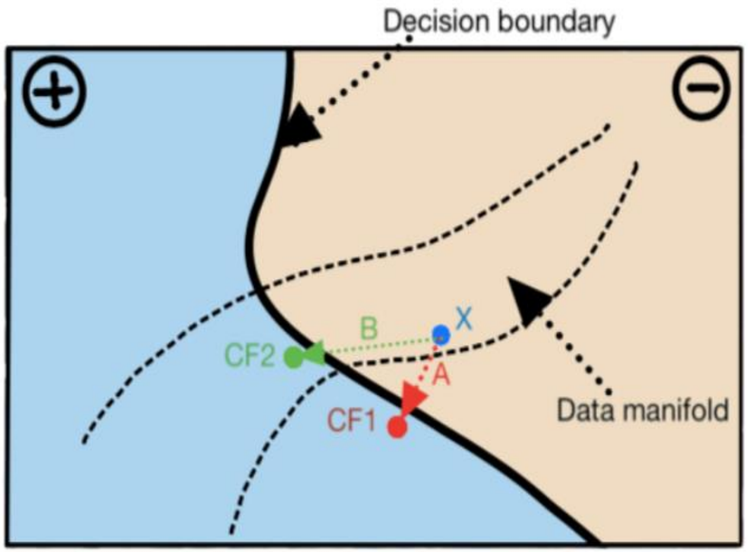
Wachter, Sandra, Brent Mittelstadt, and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." Harv. JL & Tech. 31 (2017): 841.

Design principles for counterfactual explanations

$$\underbrace{x'}_{\text{Counterfactual}} = \underbrace{x}_{\text{Original input}} + \underbrace{\delta}_{\text{Explanation}} \quad ; \quad x, x', \delta \in \mathcal{X} \subseteq \mathbb{R}^n$$

$$\text{s.t. } f(x') = y' \neq y = f(x); \quad y, y' \in \mathcal{Y}, \ f : \mathcal{X} \to \mathcal{Y}$$

- Desiderata for CFs ([Dandl et al, 2020]):
  - Closest possible world/ Proximity:
    - *x'* should be close to *x*, e.g., L2 norm
  - Sparsity:
    - change only a few features
  - Plausibility/ Feasibility:
    - *x'* should come from the data distribution
  - Actionability:
    - ARs should only recommend changes to the features that are actionable (e.g., do not change immutable features)
  - Causality:
    - Adhere to problem-specific causal constraints (e.g., age cannot decrease)
  - ....



Source: Verma et al, 2010

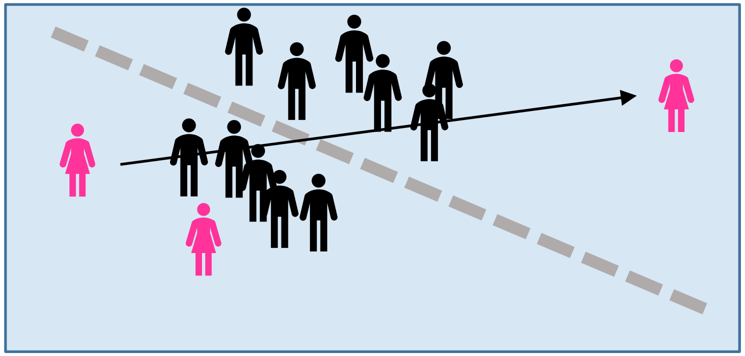
Verma, Sahil, John Dickerson, and Keegan Hines. "[Counterfactual explanations for machine learning: Challenges revisited.]" *arXiv preprint arXiv:2106.07756* (2021).

Methods for generating CFs

- Naïve approach to CF generation
- Single-objective optimization (Wachter et al., 2017)
  - single objective (proximity)
  - requires access to model gradients
- Single-objective (Tolomei et al, 2017)
  - single objective (proximity)
  - Requires access to a trained Random Forest model
- Multi-objective optimization (Dandl et al, 2020)
  - multiple objectives
- Single-objective, diverse CFs (Mothilal et al., 2020)
  - diversity objective (Determinental Point Process)
- Sequential CFs (Naumann and Ntoutsi, 2021)
  - consider the order in which changes in features (actions) are applied
- Amortized (scalable) CFs (Verma et al, 2021)
  - Learn a policy to generate CFs, e.g., with RL (Panagiotou and Ntoutsi, 2023)

Naïve approach to CF generation

- Why not select an existing instance from the target class?



- Pros
  - Easy to implement (e.g., just choose closest neighbor)
- Cons
  - Exposing other users' real data
  - Some instances will not have a close target neighbor
    - This becomes more prominent with class imbalance/ other biases

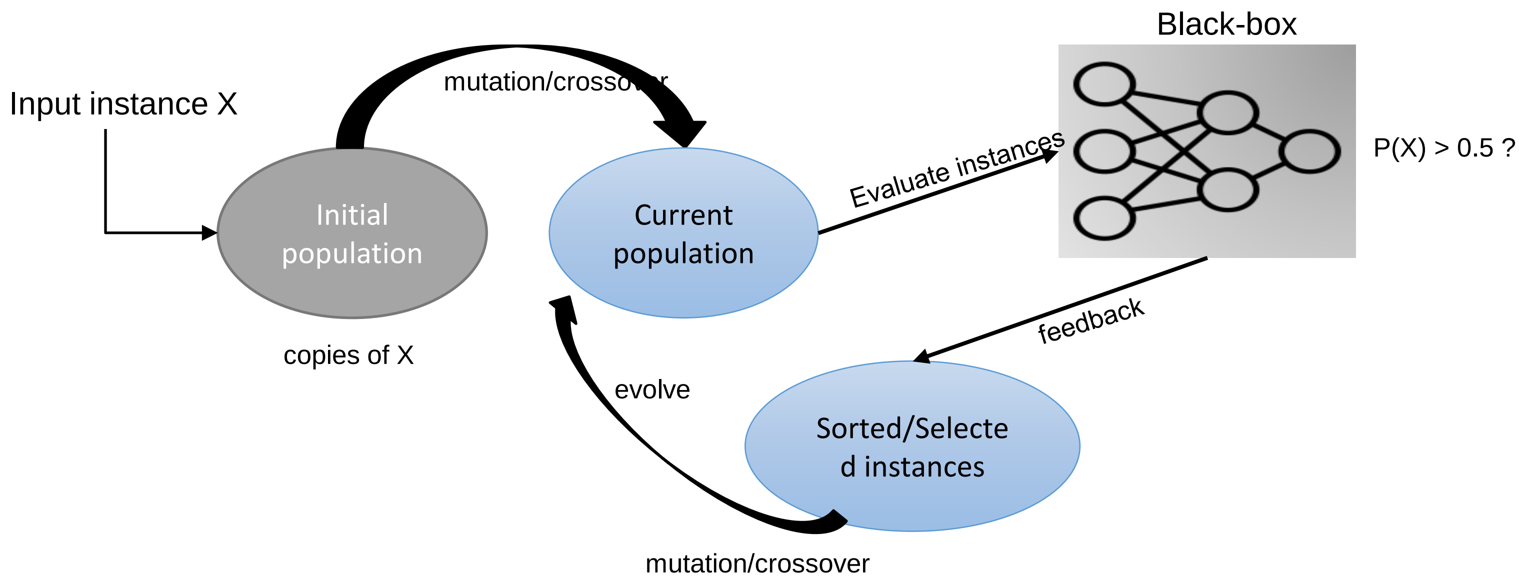
Multi-objective optimization approach (Dandl et al, 2020)

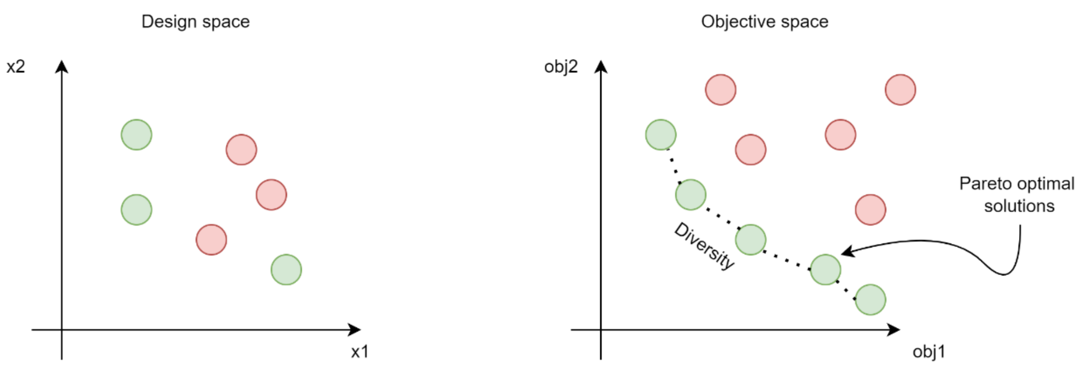
- A counterfactual explanation x for an observation x* is defined as a data point fulfilling proximity, sparsity, plausibility objectives
- Formulate the CF generation as a MOO problem

$$\min_{\mathbf{x}} \mathbf{o}(\mathbf{x}) := \min_{\mathbf{x}} \left(o_1(\hat{f}(\mathbf{x}), Y'),\ o_2(\mathbf{x}, \mathbf{x}^*), o_3(\mathbf{x}, \mathbf{x}^*), o_4(\mathbf{x}, \mathbf{X}^{obs})\right)$$

  - o1: the distance between the predicted class and the target class Y'
  - o2: the proximity between x and x*, measured using Gower distance to account for mixed features (proximity)
  - o3: the number of changed features (sparsity)
  - o4: KNN distance to ground truth data (plausibility)
- Balancing the four objectives is difficult since the objectives contradict each other, e.g., o1 becomes harder when we require o2
- They solve the problem using the Nondominated Sorting Genetic Algorithm (NSGA-II)

Multi-objective optimization approach (Dandl et al, 2020)





CFs: discussion

- Advantages
  - Nice concept close to counterfactual human thinking
  - Actionable insights: what to change in my instance to achieve a desired outcome?
- Limitations
  - Many possible worlds/ CFs, which one(s) to choose?
  - Typically based on desiderata
  - Various ways to evaluate the different desiderata objectives
  - Evaluation typically assesses the quality w.r.t. design desiderata

Outline

- Introduction - Growing XAI requirements
- Explanations in a nutshell
- Types of explanations
- Local-explanation methods
  - LIME
  - SHAP
  - Counterfactual explanations
- Reflections on XAI

Reflecting on explanations

- A versatile tool for different user groups
- Different explanation types
  - Feature attribution methods like SHAP, LIME, …
  - Counter factual explanations
  - Also for specific data types, e.g., timeseries, images, text ….

  …. and many more not covered in this course (see excellent surveys by Guidotti et al, 2022; Bodria et al, 2023; etc)

Reflecting on explanations

- Still many open questions and challenges
  - Which explanation?
  - One vs many explanations?
  - Can we trust the explanations?
    - Recall the many assumptions of LIME, for example
    - Explanations can be easily manipulated/attacked ([Yang et al, 2022]())
  - Computational aspects
    - E.g., optimizing for each instance or learning a policy for explanation generation
  - Evaluation!!!!
    - No ground truth
    - User studies
    - ...

Can We Really Trust Explanations? Evaluating the Stability of Feature Attribution Explanation Methods via Adversarial Attack

Zhao Yang[1,2], Yuanzhe Zhang[1,2], Zhongtao Jiang[1,2], Yiming Ju[1,2], Jun Zhao[1,2], Kang Liu[1,2,3*]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences / Beijing, 100049, China
[2]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences / Beijing, 100190, China
[3]Beijing Academy of Artificial Intelligence / Beijing, 100084, China
{zhao.yang, yzzhang, zhongtao.jiang}@nlpr.ia.ac.cn
{yiming.ju, jzhao, kliu}@nlpr.ia.ac.cn

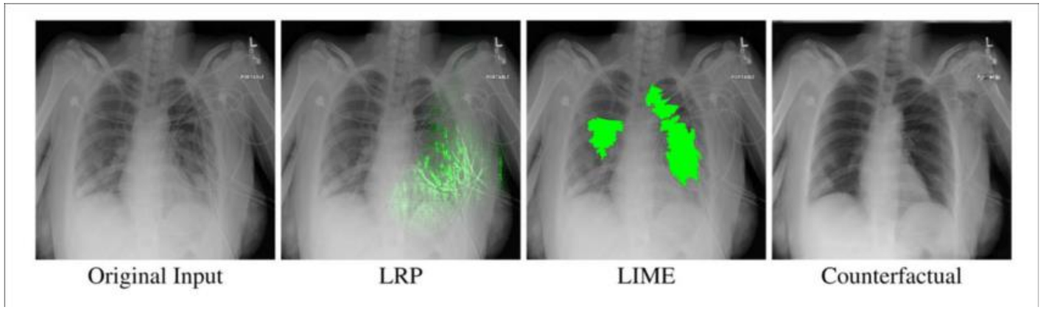
Abstract

Explanations can increase the transparency of neural networks and make them more trustworthy. However, can we really trust explanations generated by the existing explanation methods? If the explanation methods are not stable enough, the credibility of the explanation will be greatly reduced. Previous studies seldom considered such an important issue. To this end, this paper proposes a new evaluation frame to evaluate the *stability* of current typical feature attribution explanation methods via textual adversarial attack. Our frame could generate adversarial examples with similar textual semantics. Such adversarial examples will make the original models have the same outputs, but make most current explanation methods deduce completely different explanations. Under this frame, we test five classical explanation methods and show their performance on several stability-related metrics. Experimental results show our evaluation is effective and could reveal the *stability* performance of existing explanation methods.

1 Introduction

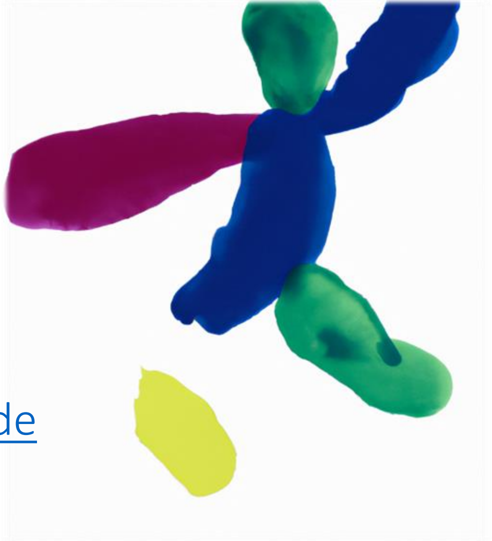
Fueled by recent rapid development in deep learning, NLP systems have obtained promising results in several fields, such as medical, law and commerce (Rudin, 2019; Bommasani et al., 2021). However, besides the predicted results, users concern more on how these results are generated (Lipton, 2018). To this end, lots of emphases have been set upon the explanation methods for neural networks (Ribeiro et al., 2016; Li et al., 2016; Simonyan et al., 2013; Bastings et al., 2019).

Although the current explanation methods have increased the transparency of the neural networks and provided explanations as supports for predicted results, most of them ignored important questions: *are these methods reliable and the generated explanations really trustful?* Besides the widely used focused properties of explanation methods, such as faithfulness, plausibility (Adebayo et al., 2018; Jacovi and Goldberg, 2020; Atanasova et al., 2020), readableness (Bastings et al., 2019) and compactness (Miller, 2019; Jiang et al., 2021), we believe *stability* is an important but often overlooked property (Robnik-



Source: [Link]()

Thank you for your attention!



- Contact data:
  - eirini.ntoutsi@unibw.de
  - @entoutsi
  - https://www.unibw.de/aiml
  - https://aiml-research.github.io/