

CMPT 733 – Big Data Programming II

Explainable Machine Learning

Instructor Steven Bergner

Course website <https://sfu-db.github.io/bigdata-cmpt733/>

Slides by: Xiaoying Wang and Jiannan Wang

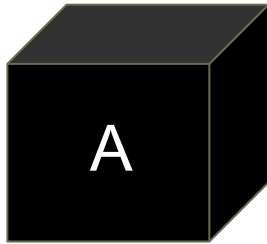
Outline

- Motivation: Why Explainable ML matters?
- Big Picture: Taxonomy State-of-the-art Techniques

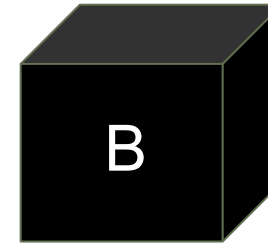
Outline

- Motivation: Why Explainable ML matters?
- Big Picture: Taxonomy State-of-the-art Techniques

Evaluation



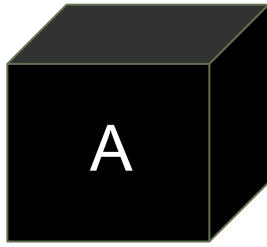
Bird: 99.0%



Bird: 99.9%

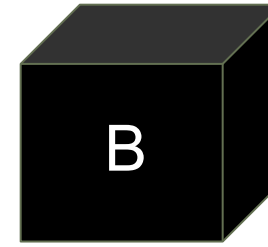
Which model are you going to choose?

Evaluation



Because it has
wings and a
beak

Bird: 99.0%



Because it is white
and the background
is blue

Bird: 99.9%

Which model are you going to choose?

Debugging

What's wrong?



Q: How symmetrical are the white bricks on either side of the building?

A: very

Q: How **asymmetrical** are the white bricks on either side of the building?

A: very

Q: How **fast** are the bricks **speaking** on either side of the building?

A: very

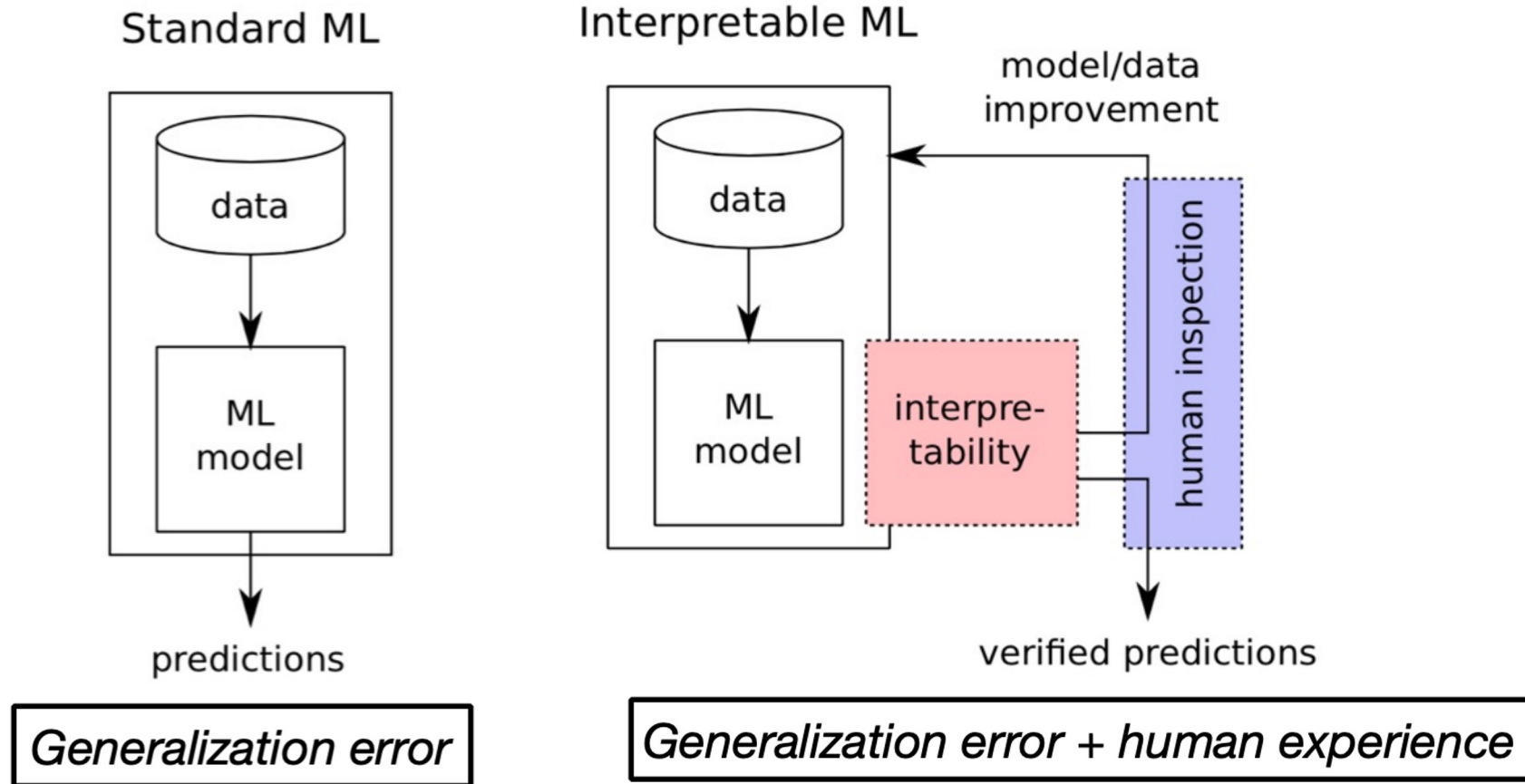
Debugging



How symmetrical **are** the **white** **bricks** on
either side of the building?

red: high attribution
blue: negative attribution
gray: near-zero attribution

Improvement



Learning insights



"It's not a human move. I've never seen a human play this move"

"So beautiful."

- Fan Hui

Legal Concerns

SR 11-7: Guidance on Model Risk Management



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

DIVISION OF BANKING
SUPERVISION AND REGULATION

SR 11-7
April 4, 2011

**TO THE OFFICER IN CHARGE OF SUPERVISION AND APPROPRIATE SUPERVISORY AND EXAMINATION
STAFF AT EACH FEDERAL RESERVE BANK**

SUBJECT: Guidance on Model Risk Management



Art. 22 GDPR

**Automated individual decision-
making, including profiling**

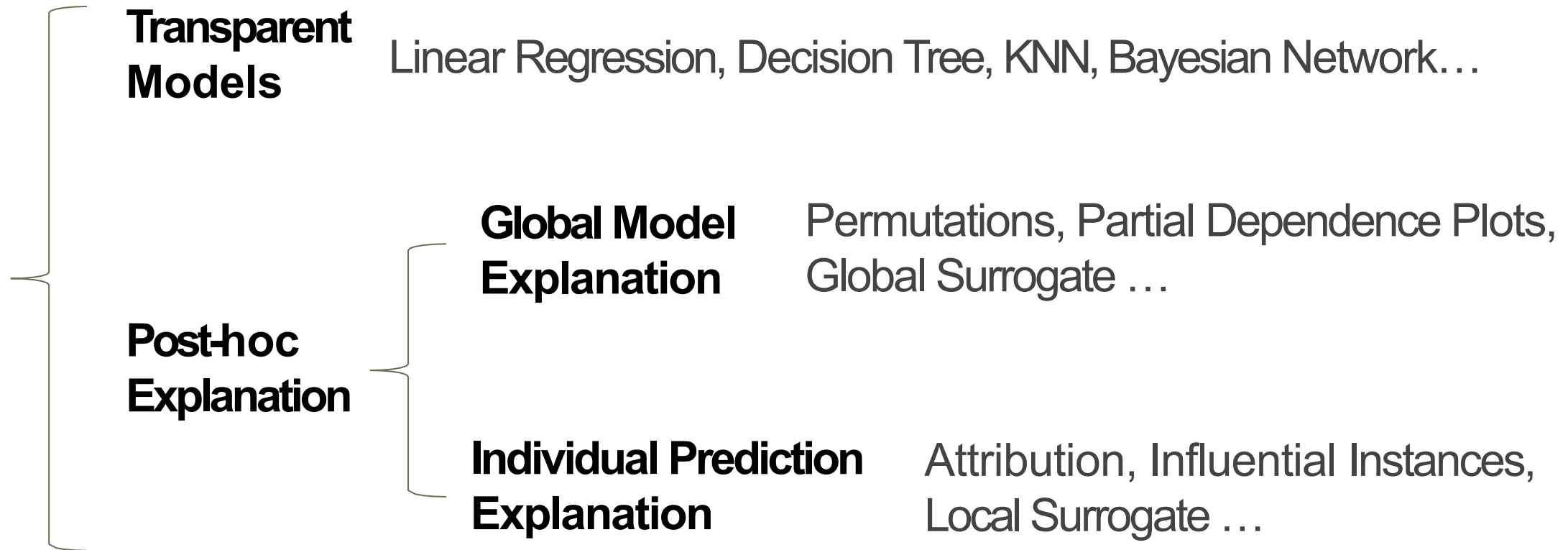
Outline

Motivation: Why Explainable ML matters?

Big Picture: Taxonomy

State-of-the-art Techniques

Taxonomy



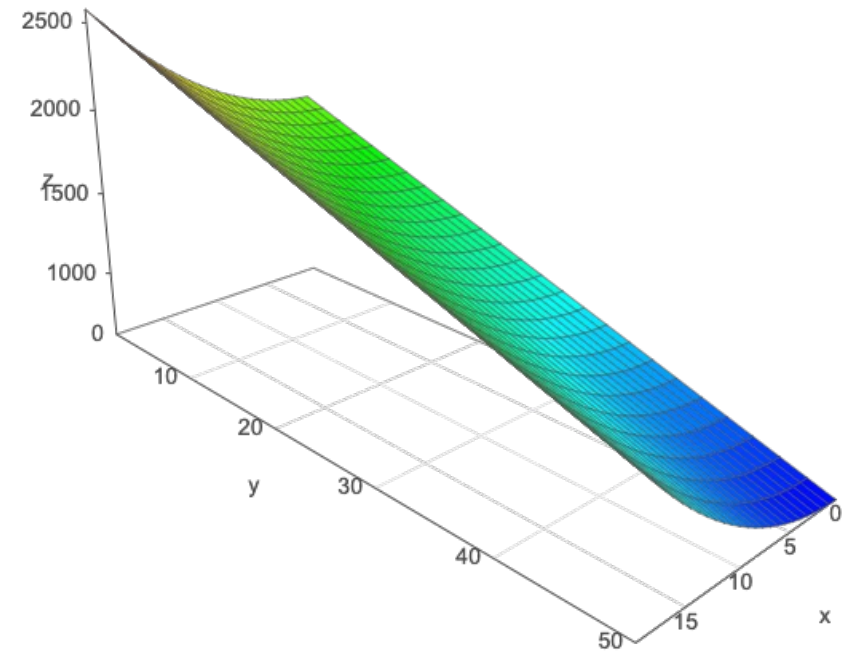
Taxonomy



Linear Regression

House rent (z) with respect to its area (x)
and distance from SFU (y)

$$z = 2.1x - 2.4y + 1800$$

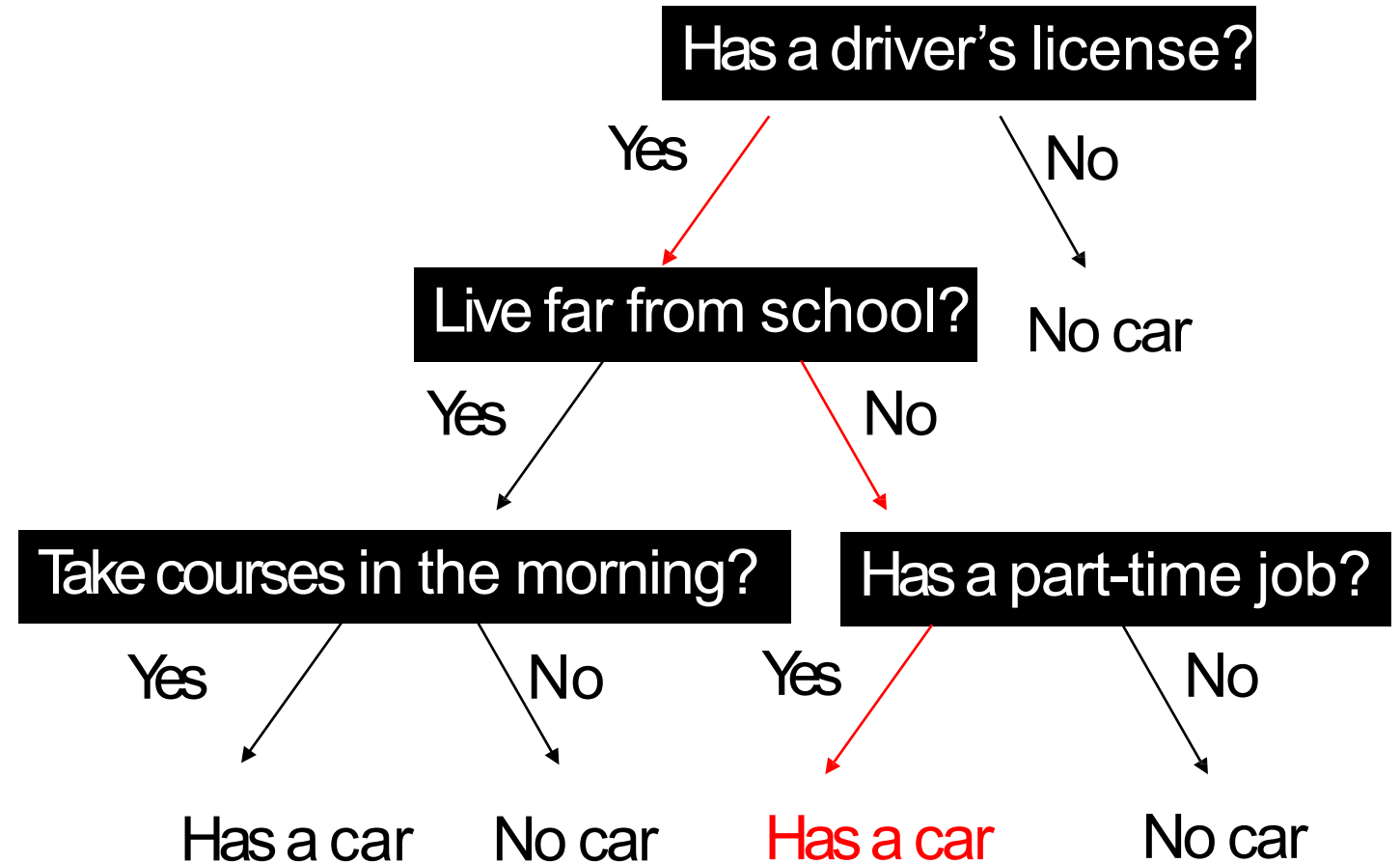


How do area and distance affect the house rent?

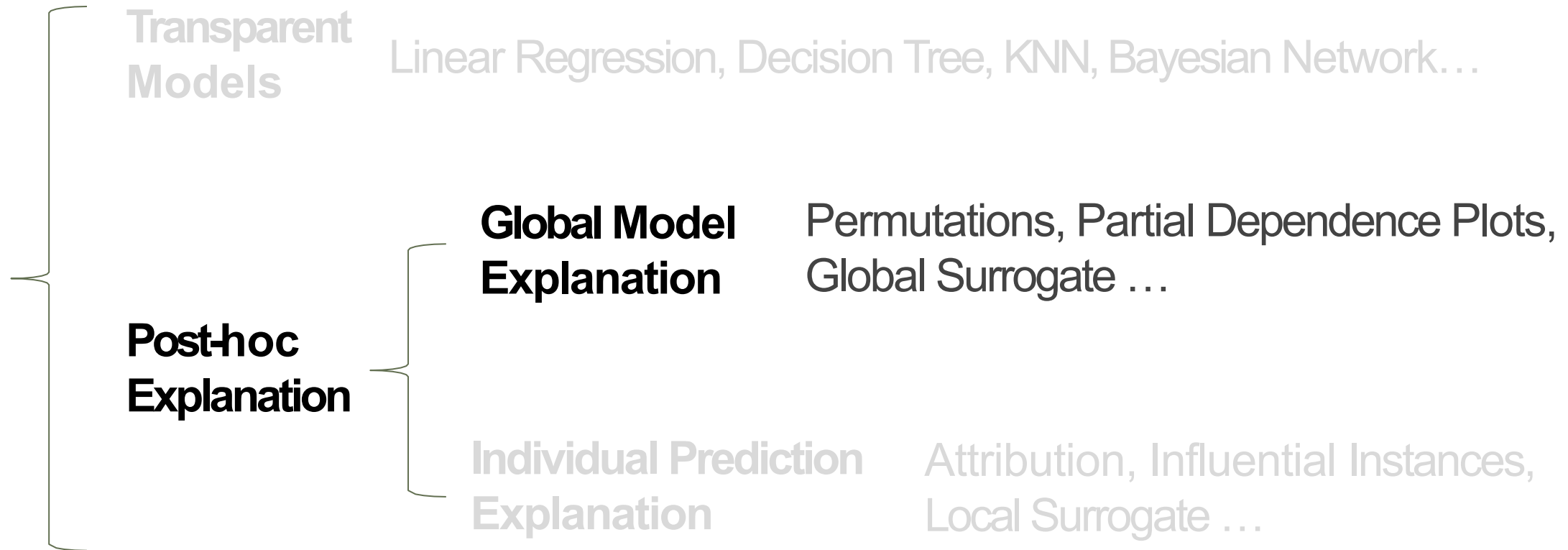
Decision Tree

Does a student own a car?

Why does the model predict student A **has a car** ?



Taxonomy



Permutations

Main idea: measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature

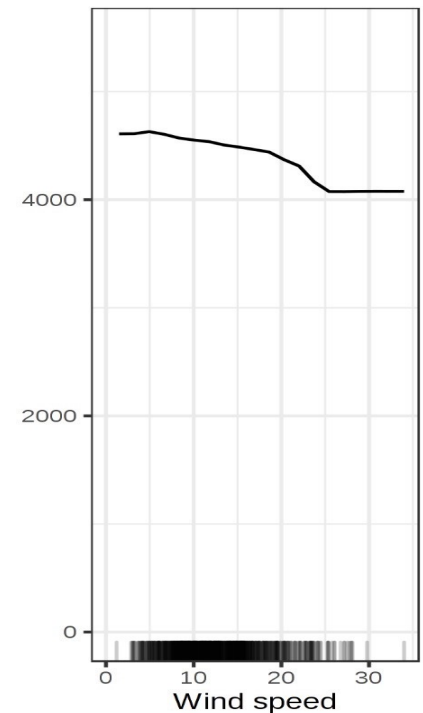
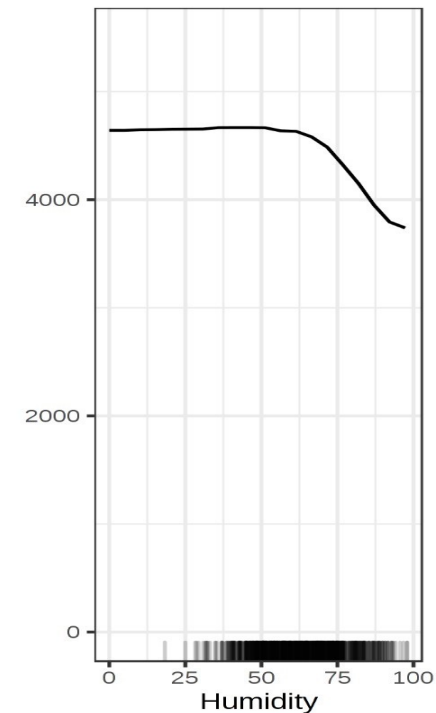
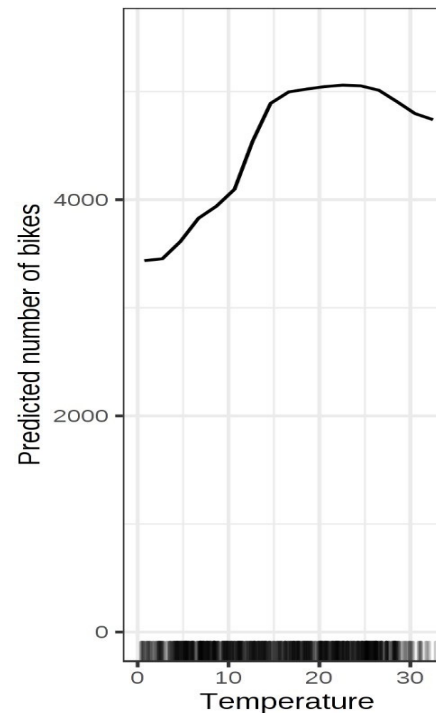
ID	Distance from SFU	# bathroom	Area	Closest bus stop	...
1	5.0km	1	670 ft^2	0.30km	...
2	8.2km	2	920 ft^2	0.12km	...
3	2.3km	2	880 ft^2	1.20km	...
...	...	...	...	...	...
9999	10km	1	680 ft^2	0.05km	...
10000	7.8km	1	730 ft^2	0.23km	...

Permutations

- Input: trained model and labeled dataset for evaluation
- Output: relative importance for each feature
- Method:
 - Apply the model on original dataset and get an estimation error E
 - For each feature:
 - Permute feature and apply the model again on the permuted data to get a new estimation error E'
 - The feature importance can be measured by $E' - E$ or E'/E

Partial Dependence Plots

Main idea: show the marginal effect one or two features have on the predicted outcome of a machine learning model



ID	Temperature	Humidity	Wind Speed	Rental#
1	20	30	20	3000
2	25	35	10	2500
3	22	25	15	3300
4	30	20	18	2000
..	..	...	...	...

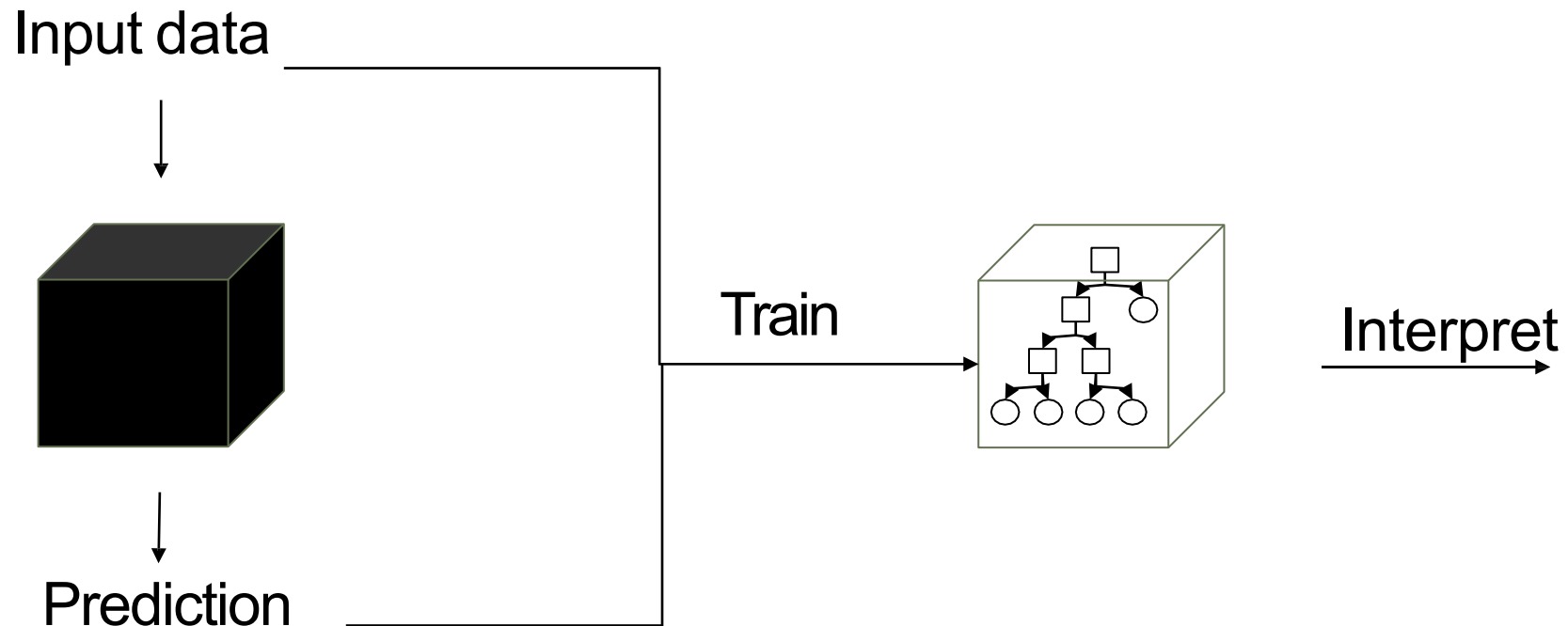
Partial Dependence Plots

Let x_s be the feature set ($|x_s| \in \{1,2\}$) we want to examine, and x_c be the rest of the features used in the model $\hat{f}$:

- Partial dependence function: $\hat{f}_{x_s}(x_c) = E_{x_c}[\hat{f}(x_s, x_c)] = \int \hat{f}(x_s, x_c) dP(x_c)$
- Can be estimated by: $\hat{f}_{x_s}(x_c) = \frac{1}{n} \sum_{i=1}^n \left(x_s, x_c^{(i)} \right)$

Global Surrogate

Main idea: train a transparent model to approximate the predictions of a black box model



Global Surrogate

Let $\hat{y}^{(i)}$ and $\hat{y}_*^{(i)}$ be the target model and surrogate model's prediction for the i th input data, we can use R-squared measure we can evaluate how good the surrogate model is in approximating the target model:

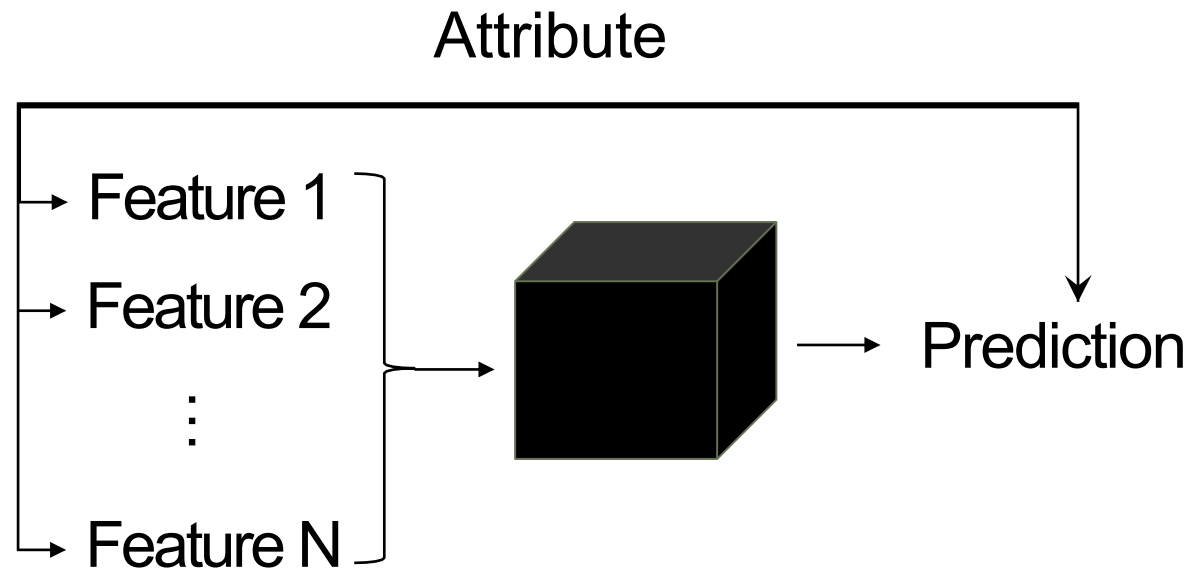
$$R^2 = 1 - \frac{\sum_{i=1}^n \left(\hat{y}_*^{(i)} - \hat{y}^{(i)} \right)^2}{\sum_{i=1}^n \left(\hat{y}^{(i)} - \hat{y}_{avg} \right)^2}$$

Taxonomy



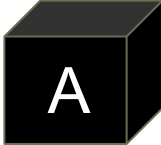
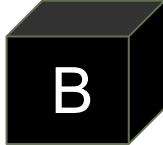
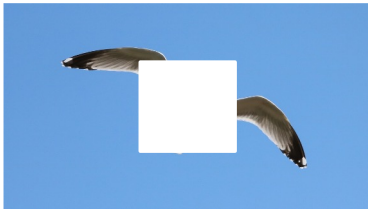
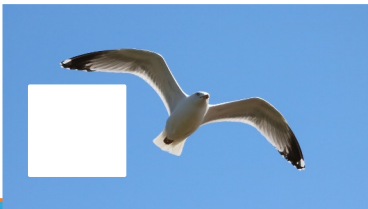
Attribution

- **Main idea:**
 - Attribute a model's prediction on a sample to its input features
- **Approaches**
 - Ablation
 - Shapely value
 - ...



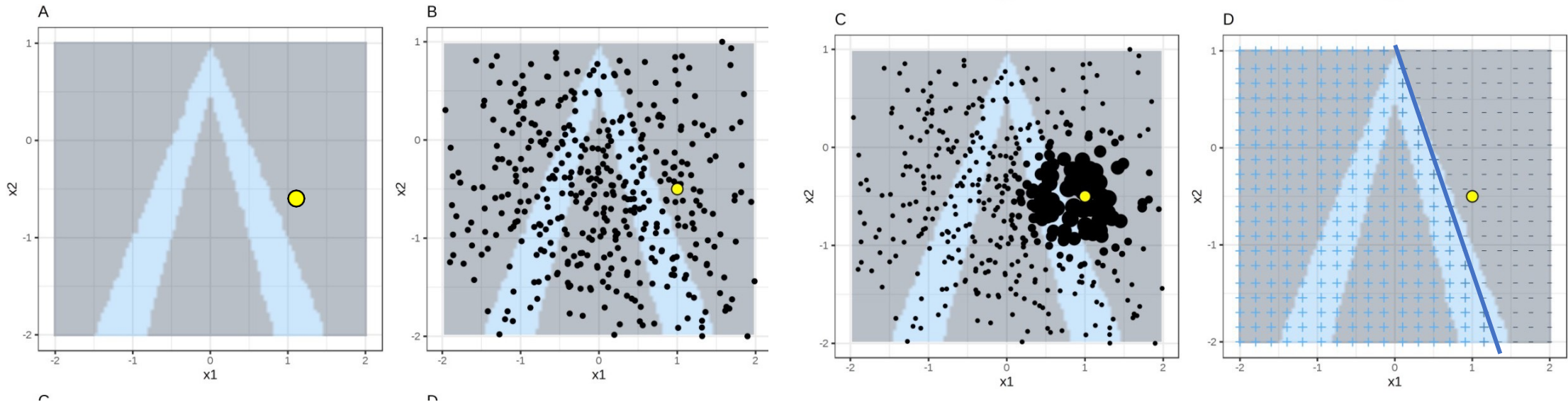
Attribution (Ablation)

Ablation: drop each feature and attribute the change in prediction to the feature

	 A	 B
	Bird (99%)	Bird (99%)
	Bird (20%)	Bird (98%)
	Bird (96%)	Bird (35%)

Local Surrogate (LIME)

Main idea: Test what happens to the prediction when give variations of data into the machine learning model

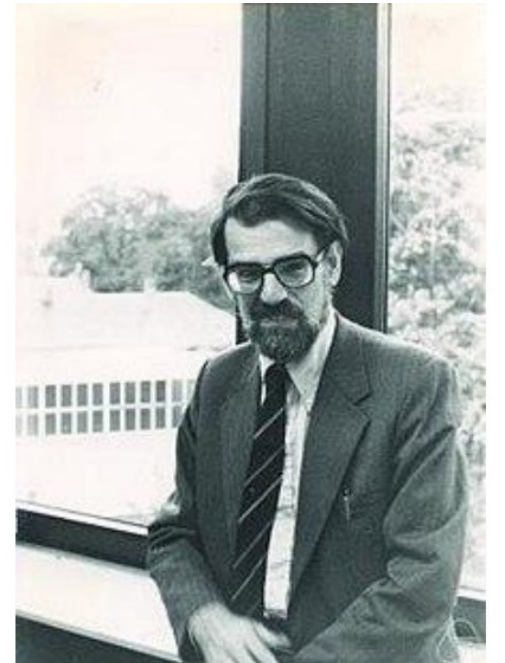


Local Surrogate (LIME)

- The local surrogate model is obtained by: $\operatorname{argmin}_{g \in G} L(f, g, \pi_x) + \Omega(g)$
 - f : target model, g : surrogate model, G : family of all possible g , π_x : neighborhood of target sample
 - L : measure fidelity, how the surrogate model approximate the target model
 - Ω : measure complexity of the surrogate model
- Get variation of data:
 - Text and image: turn single word or super-pixels on and off
 - Tabular data: create new samples by perturbing each feature individually

Shapley Value

- Classic result in game theory on distributing the total gain from a **cooperative game**
- Introduced by **Lloyd Shapley** in 1953 , who won the **Nobel Prize in Economics** in 2012
- Popular tool in studying cost-sharing, market analytics, voting power, and most recently **explaining ML models**



Lloyd Shapley in 1980

"A Value for n-person Games". Contributions to the Theory of Games 2.28 (1953): 307-317

Attribution (Shapely Value)

- Shapely value: derive from game theory on distributing gain in a coalition game
- Coalition game: players collaborating to generate some gain, function $val(S)$ represents the gain for any subset S of players
 - Game: prediction task
 - Players: input features
 - Gain: marginalized actual prediction minus average prediction $val_x(S) = \int \hat{f}(x_1, x_2, \dots, x_p) dP_{x \notin S} - E(\hat{f}(X))$
- Marginal contribution of a feature i to a subset of other features: $val_x(S \cup \{x_i\}) - val_x(S)$

Attribution (Shapely Value)

- Shapely value of a feature i on sample x : weighted aggregation of its marginal contribution over all possible combinations of subsets of other features

$$\sum_{S \subseteq \{x_1, x_2, \dots, x_p\} \setminus \{x_i\}} \frac{|S|! (p - |S| - 1)!}{p!} (val_x(S \cup \{x_i\}) - val_x(S))$$

- Intuition: The feature values enter a room in random order. All feature values in the room participate in the game (= contribute to the prediction). The Shapley value of a feature value is the average change in the prediction that the coalition already in the room receives when the feature value joins them.

Example

- A company with two employees **Alice** and **Bob**
 - No employees, **0** profit
 - Alice alone makes **20** units of profit
 - Bob alone makes **10** units of profit
 - Alice and Bob make total **50** units of profit
- What should be the bonuses be?

All Possible Orders	Marginal for Alice	Marginal for Bob
Alice, Bob		
Bob, Alice		
Shapley Value		

Example

- A company with two employees **Alice** and **Bob**
 - No employees, **0** profit
 - Alice alone makes **20** units of profit
 - Bob alone makes **10** units of profit
 - Alice and Bob make total **50** units of profit
- What should be the bonuses be?

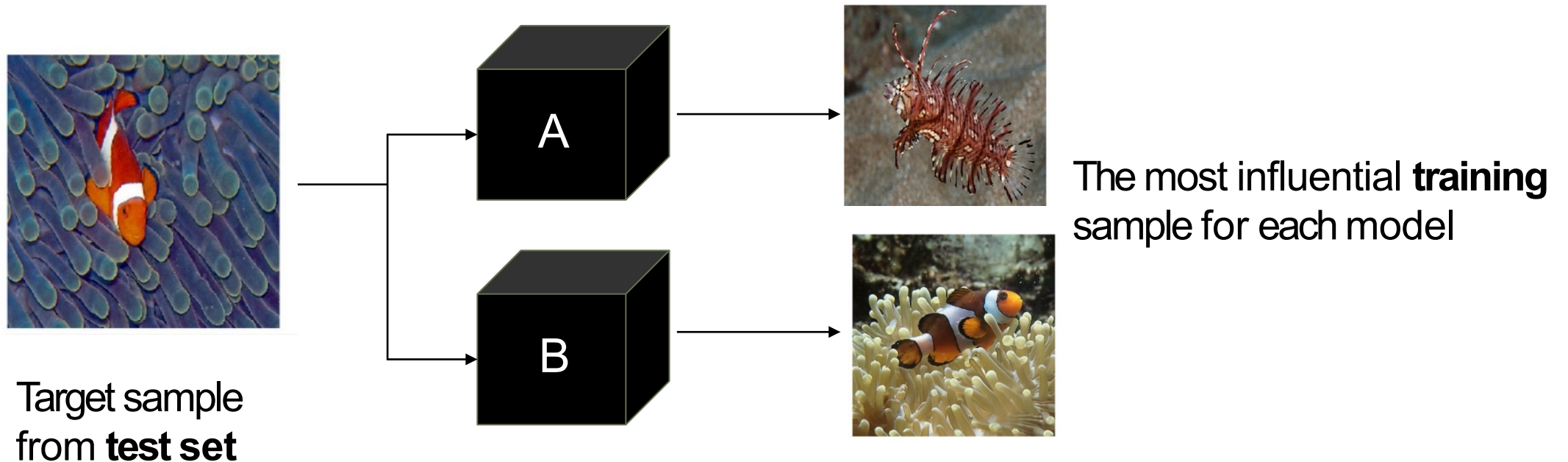
All Possible Orders	Marginal for Alice	Marginal for Bob
Alice, Bob	20	30
Bob, Alice	40	10
Shapley Value	30	20

Attribution (Shapely Value)

- Two challenges when computing shapely value:
 - Exponential time since the permutation
 - Cannot inference on models when some features are not provided
- SHAP (SHapley Additive exPlanations) provide solutions for these two challenges:
 - KernelSHAP: an approximation solution for all models:
 - Sample a subset of feature orders
 - Filling missing features with background dataset provided by user

Influential Instances

Main idea: debug machine learning model by identifying influential training instances (a training instance is influential when its deletion from training data considerably changes the model's prediction)



Influential Instances

- **Naïve approach: deletion diagnostics**
 - Train a model on all data instances, predict on test data and choose a target sample, for example: an incorrectly predicted sample with high confidence
 - For each training data, remove the data and retrain a model, predict on target sample and calculate the differences between the prediction and original prediction
 - Get the most influential top K instances (very likely to be mislabeled in this scenario)
 - Train a transparent model to find out what distinguishes the influential instances from the non-influential instances by analyzing their features (optional, for better understand the model)

Evaluation

- Human review: which method that human can get more insight of the model?
- Fidelity: how well does the method approximate the black box model?
- Stability: how much does an explanation differ for similar instances?
- Complexity: computational complexity of the method
- Coverage: the types of models that the method can explain
- ...

Available Tools

- LIME <https://github.com/ankurtaly/Integrated-Gradients>
- SHAP implementation in Python <https://github.com/slundberg/shap>
- Captum: PyTorch model interpretability tool <https://github.com/pytorch/captum>
- Skater: a Python Library for Model Interpretation/Explanations
<https://oracle.github.io/Skater/overview.html>
- ELI5: a library for debugging/inspecting machine learning classifiers and explaining their predictions
<https://eli5.readthedocs.io/en/latest/>
- Influence function implementation in Python <https://github.com/kohpangwei/influence-release>

References

- Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2021. <https://christophm.github.io/interpretable-ml-book/>.
- Anon. KDD'19 Explainable AI Tutorial. Retrieved September 13, 2019 from <https://sites.google.com/view/kdd19-explainable-ai-tutorial>
- Anon. ICCV'19 Tutorial on Interpretable Machine Learning in Computer Vision. Retrieved September 20, 2019 from <https://interpretablevision.github.io/>

Summary

