Explainable AI for Sensing Applications

Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, Molnar, 2019
Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda, Abdul et al., ACM CHI 2018

A Tesla Model S driven by Joshua Brown crashed into the side of a semi truck in May 2016.

https://www.latimes.com/business/autos/la-fi-hy-tesla-nhtsa-20190214-story.html

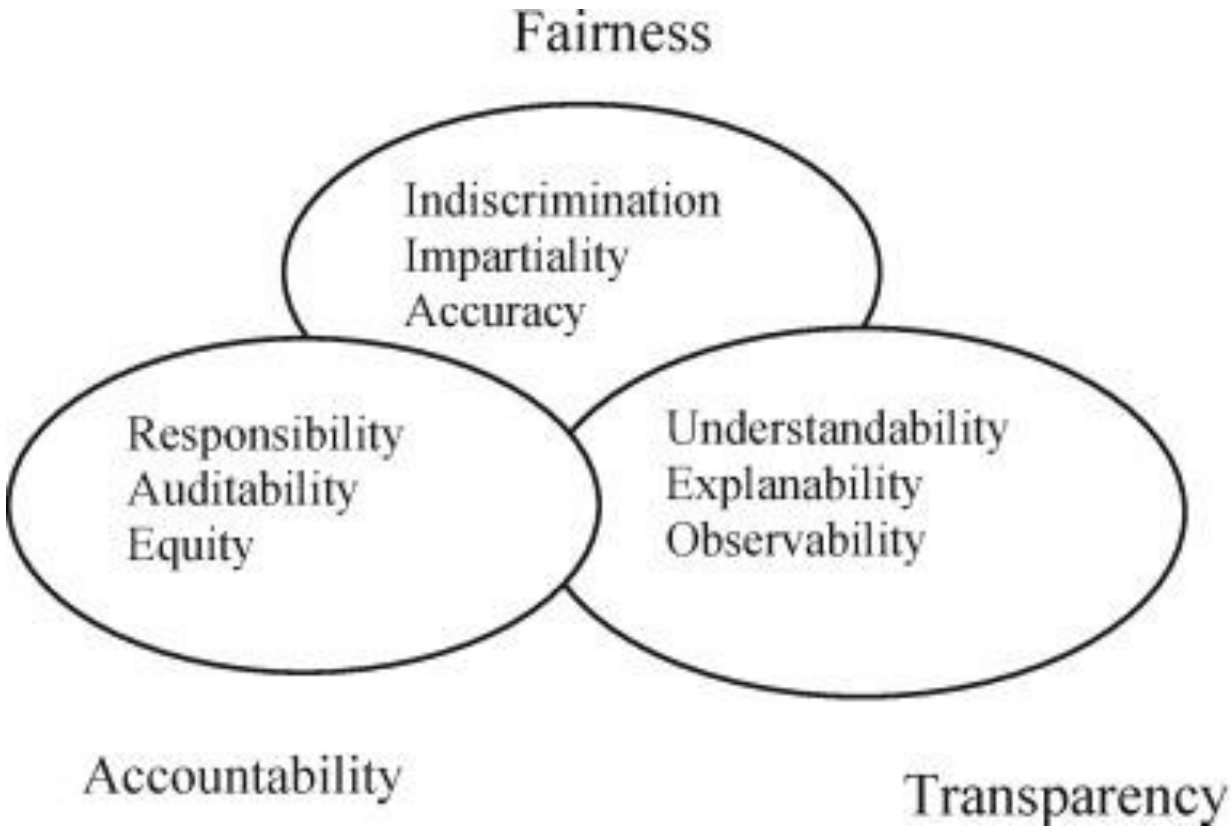
The vacuum cleaner got stuck. As an explanation for the accident, it said that it needs to be on an even surface.

Explainable, Accountable, Intelligible

- Fair, Accountable, Transparent (FAT) Machine Learning
  - Algorithmic decision making should be fair (e.g., discrimination conscious by-design)
  - Algorithms should be accountable for their decisions (aka. algorithmic accountability) and explain their decisions; e.g., EU "right to explanation"
- Interpretable Machine Learning and Classifier Explainers
  - Explainable artificial intelligence (XAI): algorithmic and mathematical methods to explain the inner workings of machine learning models
- Intelligibility, Explanatory Debugging
  - *Intelligible* by informing users about "what they know, how they know it, and what they are doing with that information" (Bellotti and Edwards)
  - How end-users make sense of and control machine-learned programs, working towards intelligible and debuggable machine-learned programs

Trends and trajectories for explainable, accountable and intelligible systems:
An HCI research agenda, Abdul et al., ACM CHI 2018

FAT-ML: Fair, Accountable, Transparent



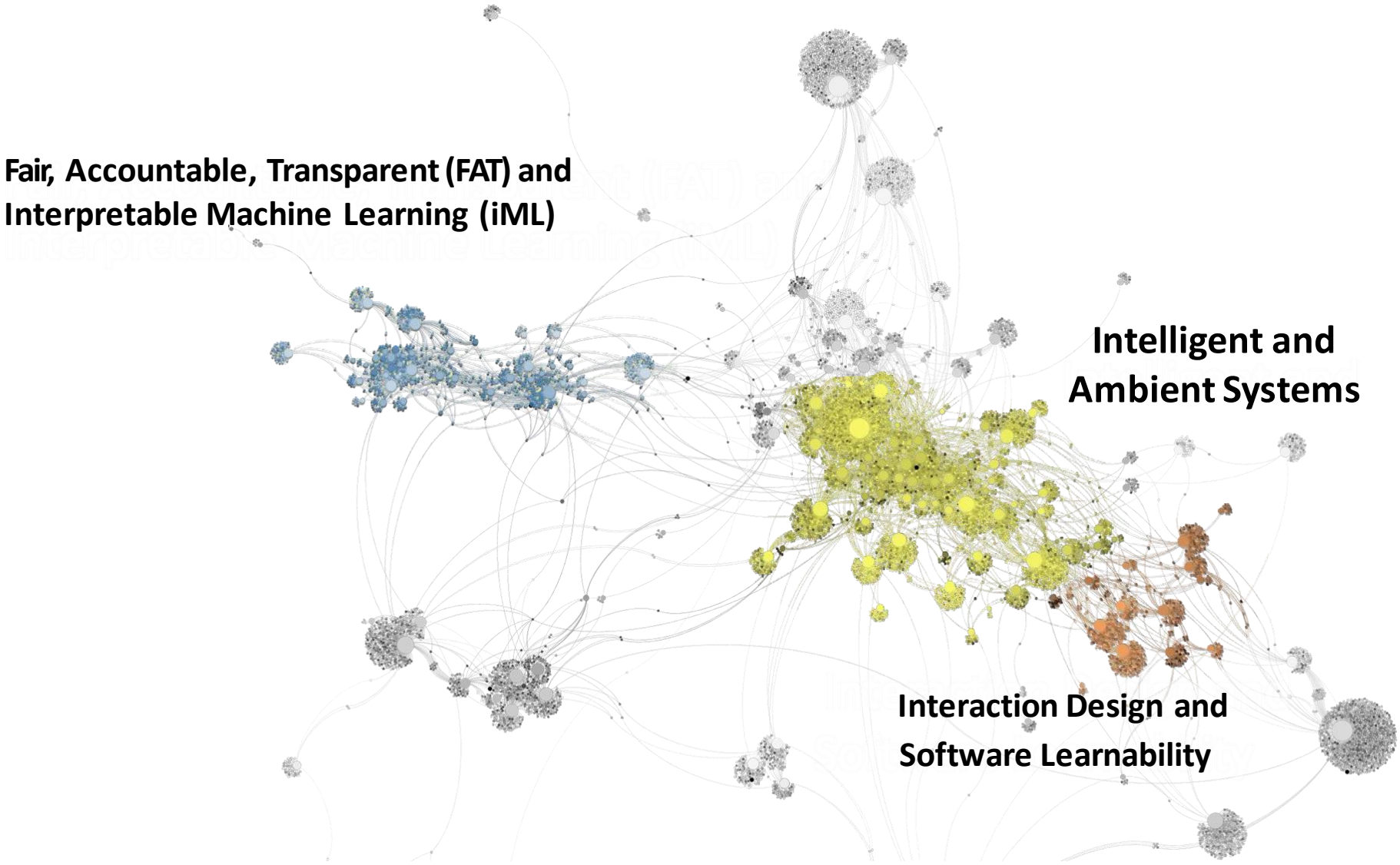
Role of fairness, accountability, and transparency in algorithmic affordance, Donghee Shin, Yong Jin Park, Computers in Human Behavior Volume 98, September 2019

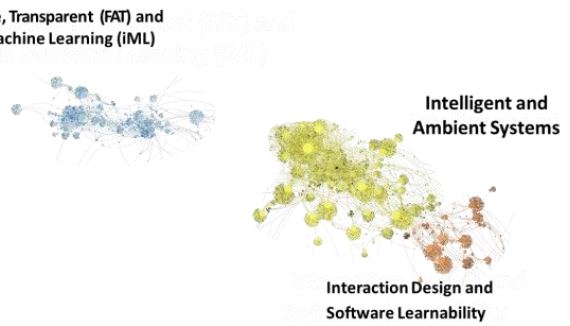
Ubicomp: Intelligibility & Accountability

- Intelligibility and accountability for context-aware systems
  - **Intelligibility:**
    - Context-aware systems seek to act upon what they infer about the context must be able to represent to their users what they know, how they know it, and what they are doing about it
  - **Accountability:**
    - Context-aware systems mediate between people (and the world); e.g., sharing files, redirecting phone calls
    - Context-aware systems must enforce **user accountability (i.e., systems + users)** when, based on their inferences about the social context, they seek to mediate user actions that impact others

Intelligibility and Accountability: Human Considerations in Context-Aware Systems, Victoria Bellotti and Keith Edwards, Xerox Palo Alto Research Center, HUMAN-COMPUTER INTERACTION, 2001, Volume 16, pp. 193–212

Citation Network Analysis: 3 Clusters





Intelligent and Ambient Systems

Recommender Systems

Software Engineering

Intelligent Agents & Systems

Trust

Context-Aware Systems

Automation Bias

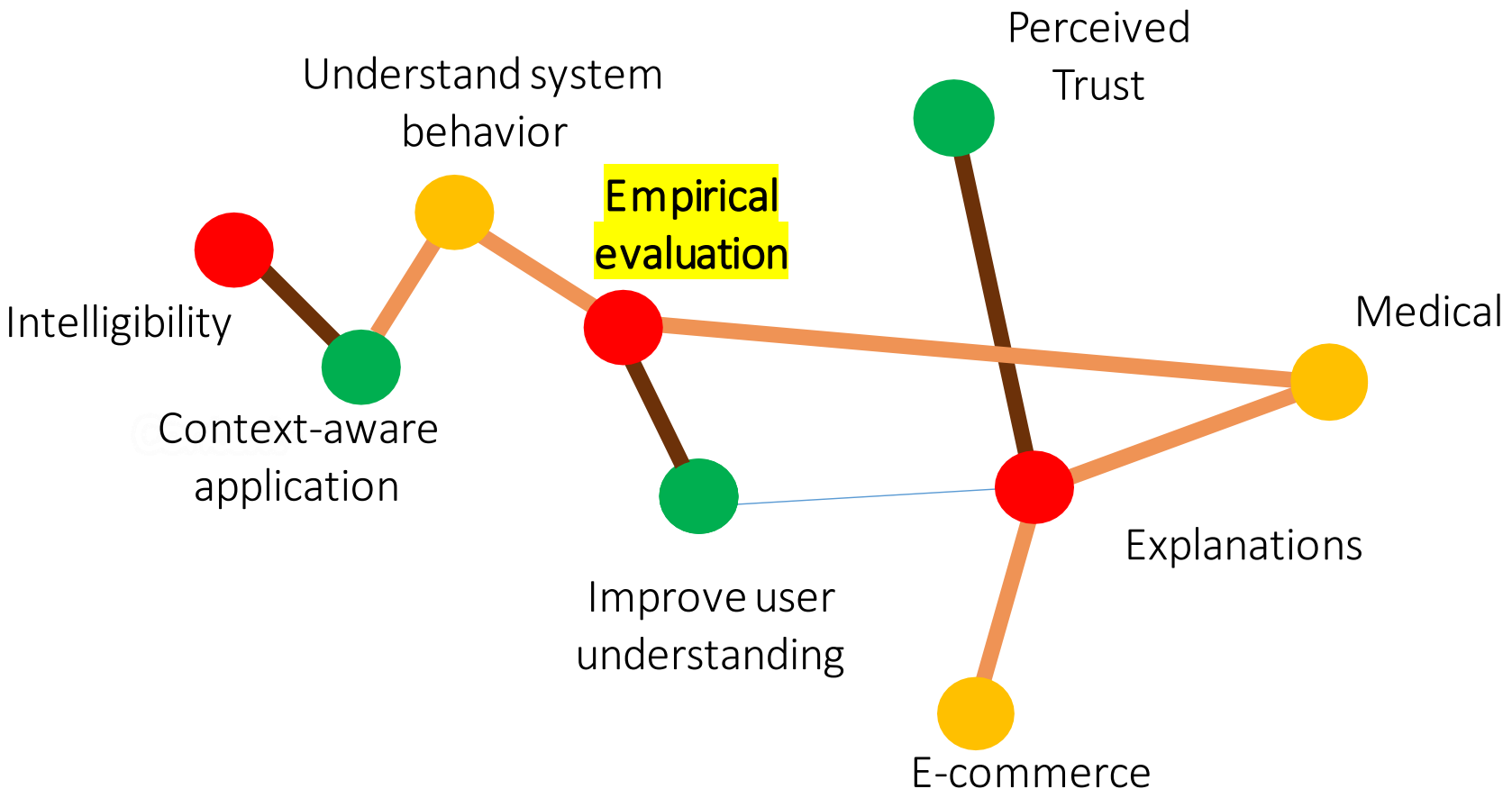
Sensing

Intelligent User Interfaces

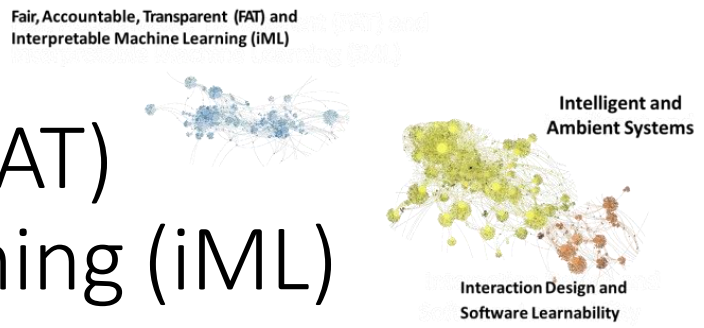
Implicit Interaction

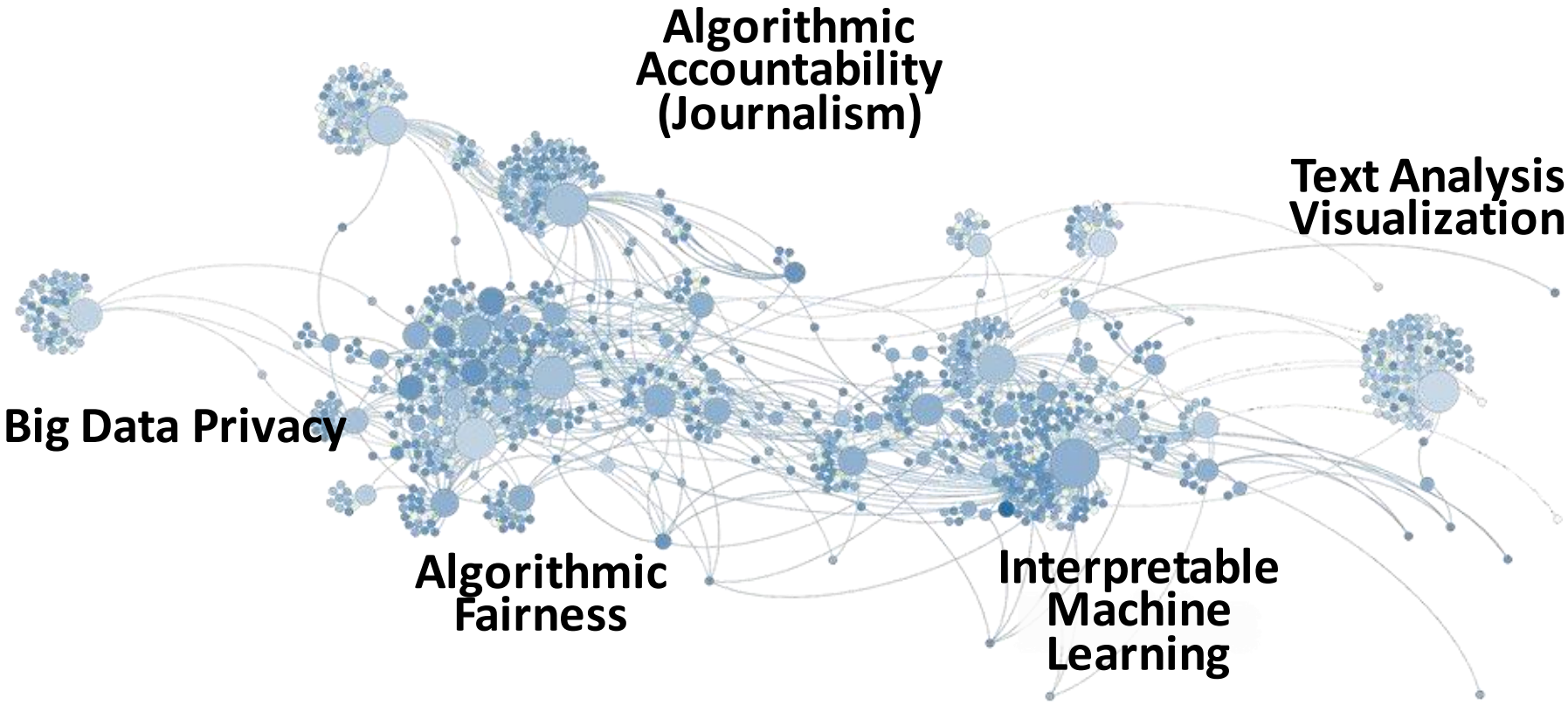
Topics in Intelligent and Ambient Systems

- Strong focus of validating explanations in real world settings



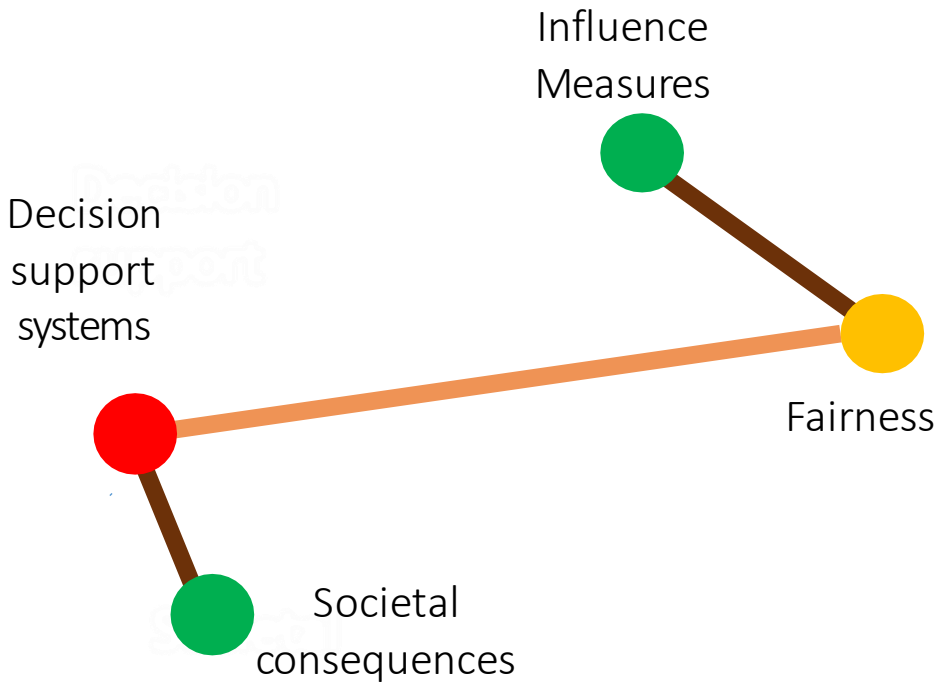
Fair, Accountable, Transparent (FAT) and Interpretable Machine Learning (iML)



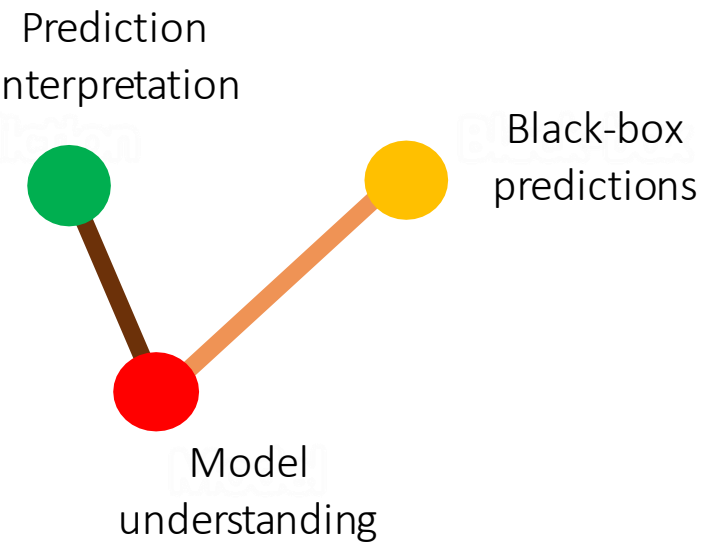


Topics in Fair, Accountable, Transparent (FAT) and Interpretable Machine Learning (iML)

- Both mathematical, but FAT focuses on society, while iML on models



FAT



iML

Why Explanation?

- Detect bias in machine learning models -- fairness
- Increase social acceptance (trust)
- Accountability
- Facilitate social interactions (i.e., interactive ML)
- Debug & audit
- Reliability/robustness
- Privacy

Taxonomy of Interpretability

- **Intrinsic or post hoc**
  - **Intrinsic interpretability** lies in interpretable models such as decision trees and linear models
  - **Post hoc interpretability** is achieved by using interpretation methods after model training (opt-out each feature)

- **Result of the interpretation method**
  - **Feature summary:** e.g., importance of each feature
  - **Model internals**: e.g., learned weights in decision trees
  - **Returning (similar) data point(s)** (or examples): e.g., what if: counterfactual inference, k-nearest neighbors, prototypes
  - **Intrinsically interpretable models**: e.g., linear models, trees

- **Model-specific vs. model-agnostic?**
  - **Model-agnostic** methods can be used on any machine learning methods

- **Local vs. global explanation**
  - **Local** – individual prediction vs. **global** – entire model behaviors

Intrinsically Interpretable Models

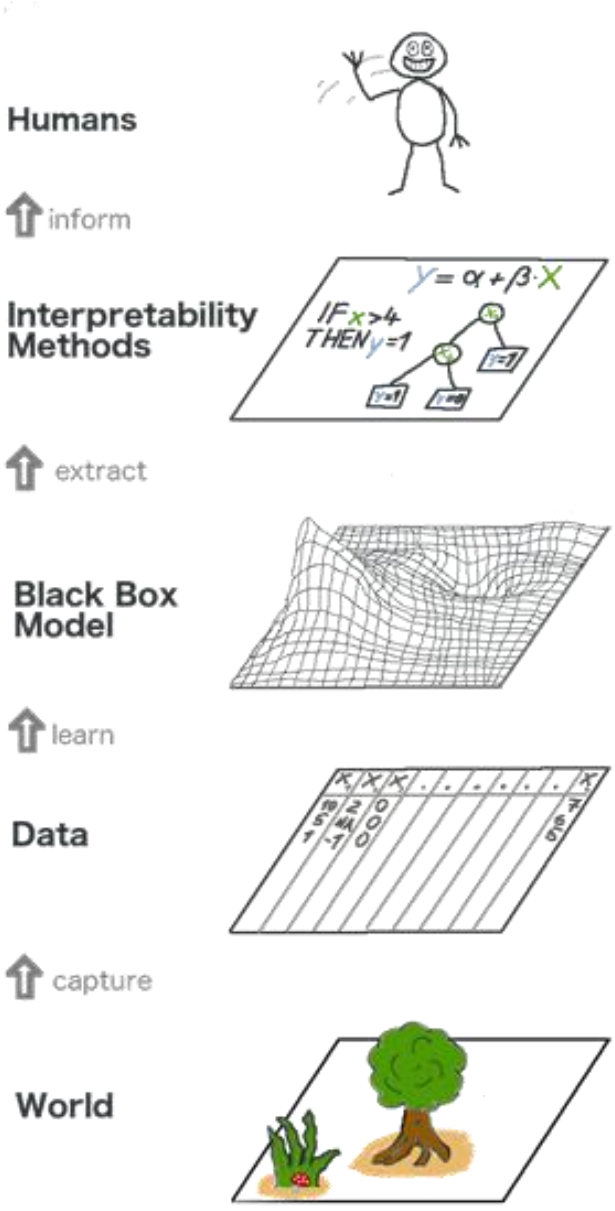
easily understand

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

classification

- A model is linear if the association between features and target is modelled linearly
- A model with monotonicity: an increase in the feature value either always leads to an increase or decrease in the target outcome
- Interactions between features to predict the target outcome

Model-agnostic Interpretation



Model-agnostic Methods

① • **Permutation feature importance** HTML
  - Ranking feature importance by calculating the increase in the model's prediction error after shuffling the feature (i.e., breaking association w/ dependent variable)

② • **Global surrogates vs. LIME (local surrogates)** HTML
  - Surrogate models are trained to approximate the predictions of the underlying black box model
  - Global surrogate models approximate the predictions of a black box model
  - LIME trains a local surrogate model (e.g., Lasso, decision tree) to explain "individual predictions" of the black box model
    - Model training using a new dataset with corresponding predictions (each sample is weighted based on the "instance" that needs to be explained)

- **SHAP (SHapley Additive exPlanations)** HTML
  - Sharpley value means average marginal contribution of a feature value (against feature combinations) for the prediction
  - SHAP combines LIME and Sharpley (i.e., weighting feature combinations)
  - Possible to get SHAP feature importance values

Model-agnostic Methods

- **Permutation feature importance**
  - Measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature
  - 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

$x_1 = (x_{1,1}, x_{1,2}, x_{1,3}, \ldots, x_{1,p})$
$x_2 = (x_{2,1}, x_{2,2}, x_{2,3}, \ldots, x_{2,p})$
$x_3 = (x_{3,1}, x_{3,2}, x_{3,3}, \ldots, x_{3,p})$
$\vdots$
$x_n = (x_{n,1}, x_{n,2}, x_{n,3}, \ldots, x_{n,p})$

Feature Matrix $X = [x_1, x_2, x_3, .., X_n]^T$
(# features = p)

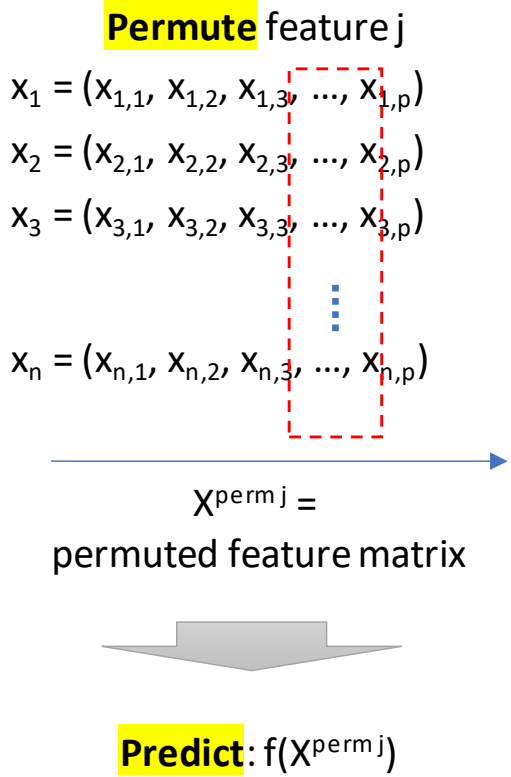
Shuffle feature #3

$x_1 = (x_{1,1}, x_{1,2}, x_{9,3}, .., x_{1,p})$
$x_2 = (x_{2,1}, x_{2,2}, x_{n,3}, .., x_{2,p})$
$x_3 = (x_{3,1}, x_{3,2}, x_{1,3}, .., x_{3,p})$
$\vdots$
$x_n = (x_{n,1}, x_{n,2}, x_{6,3}, .., x_{n,p})$

Will the model error increase if we shuffle feature #3? If Yes, #3 will be an important feature

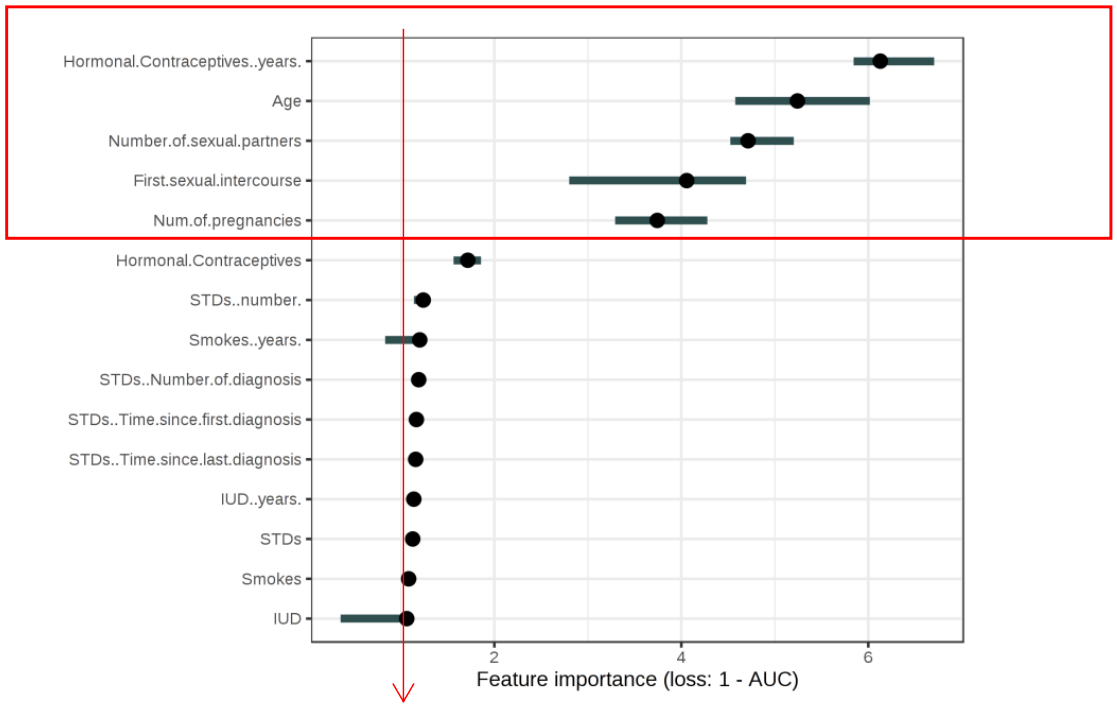
Model-agnostic Methods

- **Permutation feature importance**
  - Given:
    - Trained model **f**, feature matrix **X**, target vector **y**
    - Error measure **L(y, f)** (e.g., mean squared error)
  - Steps
    - Estimate the original model error: $e^{orig} = L(\ y, f(X)\ )$
    - For each feature j = 1, …, p do:
      - $X^{perm\ j} \leftarrow$ permute feature j in feature matrix **X**
      - Predict based on $X^{perm\ j} = f(\ X^{perm\ j}\ )$
      - Estimate error $e^{perm\ j} = L(\ \mathbf{y}, f(X^{perm\ j})\ )$
      - Get feature importance score $F_j = e^{perm\ j} / e^{orig}$ (alternatively, $e^{perm\ j} - e^{orig}$)
    - Sort feature importance score F



Model-agnostic Methods

- **Permutation feature importance**
  - **Cancer classification**
    - Fitting a random forest model to predict cancer, measuring the error increase by 1-AUC (1 minus the area under the ROC curve)



Important Features

Features associated with a model error increase by a factor of 1 (= no change) were not important for predicting cervical cancer

#Available importance_types = ['weight', 'gain', 'cover']

XGBClassifier.get_booster().get_score(importance_type= f)

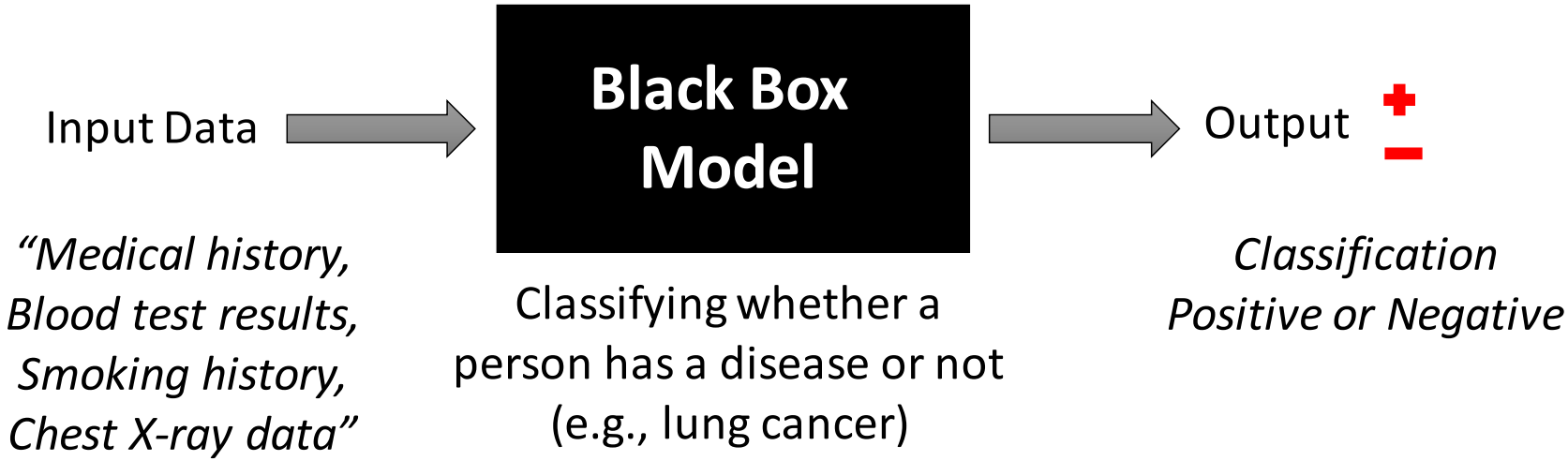
Detours: Gain-based Feature Importance

- **Gain-based feature importance** (e.g., information gain) is popular among tree-based algorithms
- **(Information) gain** implies the relative contribution of the corresponding feature to the model calculated by taking each feature's contribution for each tree in the model
  - A higher value of this metric when compared to another feature implies it is more important for generating a prediction
- **Coverage** metric means the relative number of observations related to this feature.
  - For example, if you have **100 observations**, 4 features and 3 trees, and suppose feature1 is used to decide the leaf node for 10, 5, and 2 observations in tree1, tree2 and tree3 respectively; then the metric will count cover for this feature as 10+5+2 = 17 observations.
  - This will be calculated for all the 4 features and the cover will be 17 expressed as a percentage for all features' cover metrics
- **Frequency (R)/Weight** is the percentage representing the relative number of times a particular feature occurs in the trees of the model.
  - In the above example, if feature1 occurred in 2 splits, 1 split and 3 splits in each of tree1, tree2 and tree3; then the weight for feature1 will be 2+1+3 = 6
  - The frequency for feature1 is calculated as its percentage weight over weights of all features

https://towardsdatascience.com/be-careful-when-interpreting-your-features-importance-in-xgboost-6e16132588e7

Model-agnostic Methods

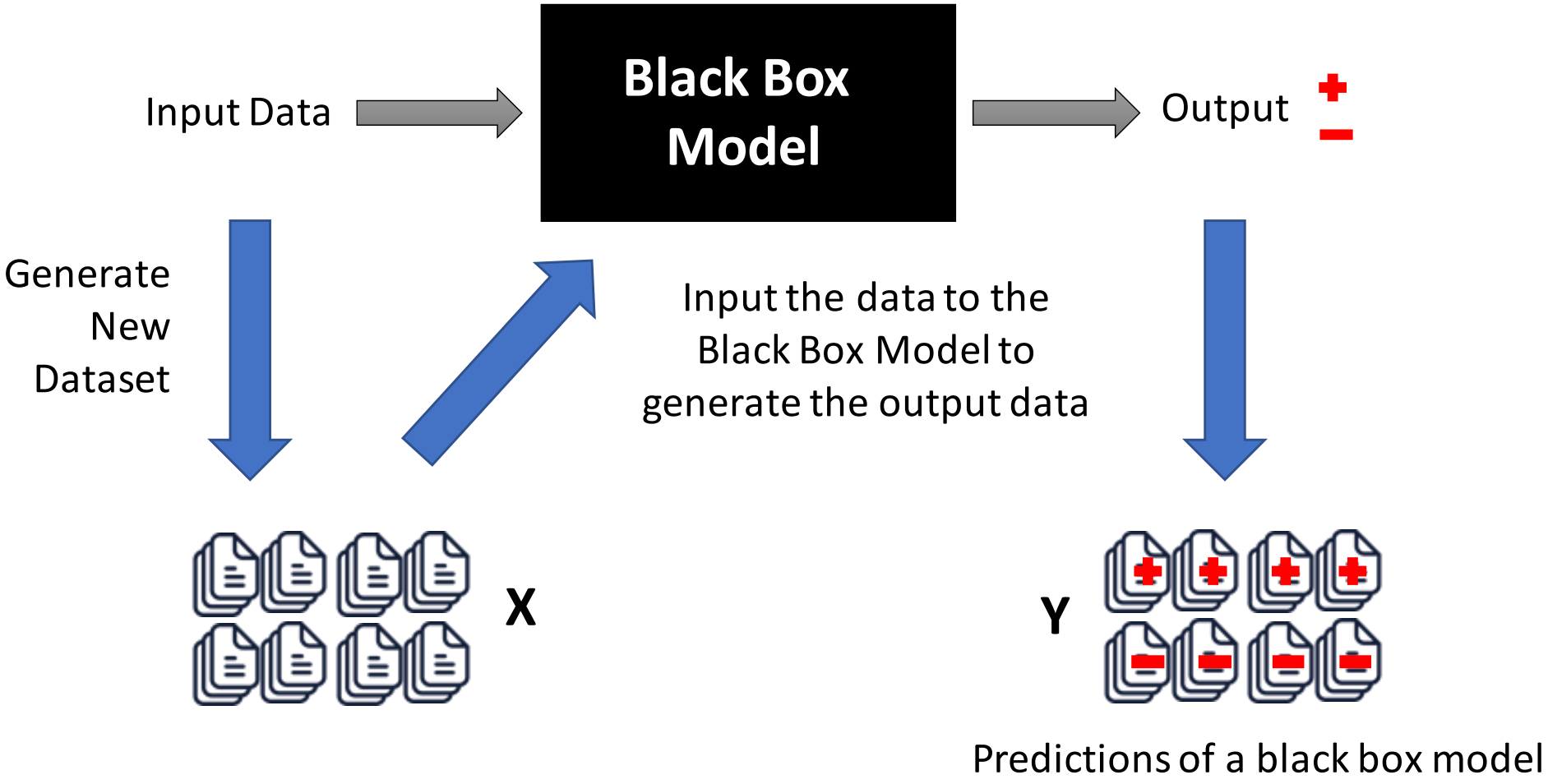
② • **Surrogate models (Global)**



How can we approximate the prediction of this "black box model"?

Model-agnostic Methods

- **Surrogate models (Global)**

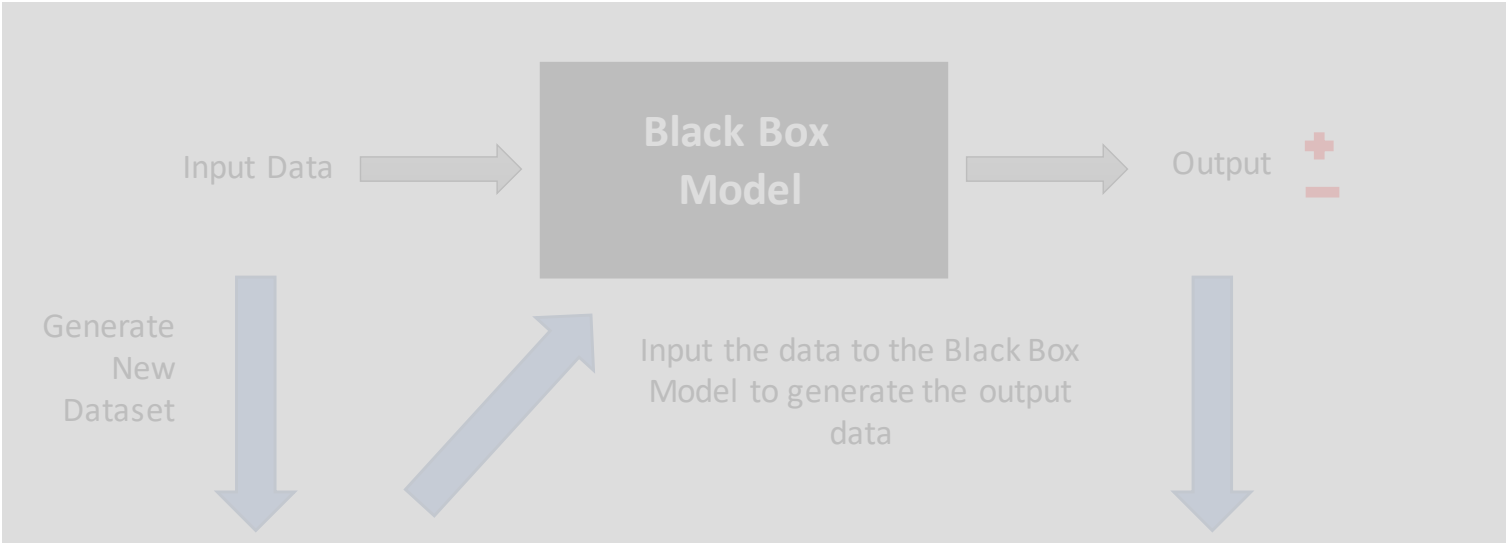


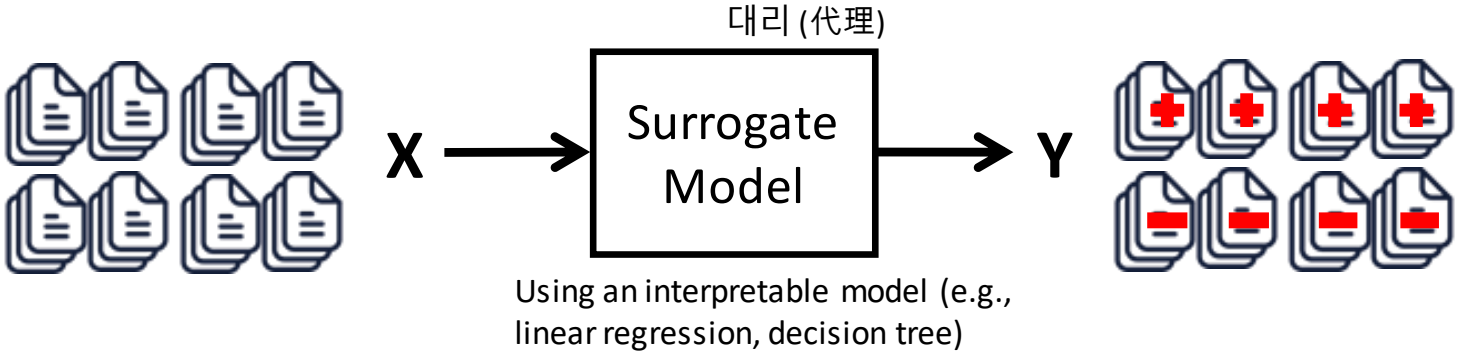
Model-agnostic Methods

- **Surrogate models (Global)**
  - Using an interpretable model (=surrogate model) that is trained to approximate the predictions of a black box model
    - Interpretable model (e.g., linear regression, decision tree)
  - Procedure
    - Data preparation
      - Select a dataset **X**
        - E.g., the same dataset used for training the black box model or a new dataset from the same distribution; or even a subset of the data could be used
      - For the selected dataset **X**, get the predictions **Y** from the black box model
    - Train a surrogate model
      - Select an interpretable model type (e.g., linear model, decision tree, ...)
      - Train the interpretable model on the dataset X and its predictions
    - Measure & interpret
      - Measure how well the surrogate model replicates the predictions of the black box model
      - Interpret the surrogate model

Model-agnostic Methods

- **Surrogate models (Global)**





Model-agnostic Methods

- **Surrogate models - Local Surrogate Model (called LIME)**
  - **Let's input a person's data (John) for diagnosis**

John

Input Data → **Black Box Model** → Output ±

"Medical history
Blood test results
Smoking history
Chest X-ray data"

Classifying whether a person has a disease or not (e.g., lung cancer)

Classification
Prob. of Positive?

25%

How can we explain this black box model?
(e.g., which feature is related to diagnosis?)

Model-agnostic Methods

- **Surrogate models - Local Surrogate Model (called LIME)**
  - **Let's input a person's data (John) for diagnosis**

John

Input Data → **Black Box Model** → Output (+ / −)

"Medical history
Blood test results
Smoking history
Chest X-ray data"

Classifying whether a person has a disease or not (e.g., lung cancer)

Classification
Prob. of Positive?

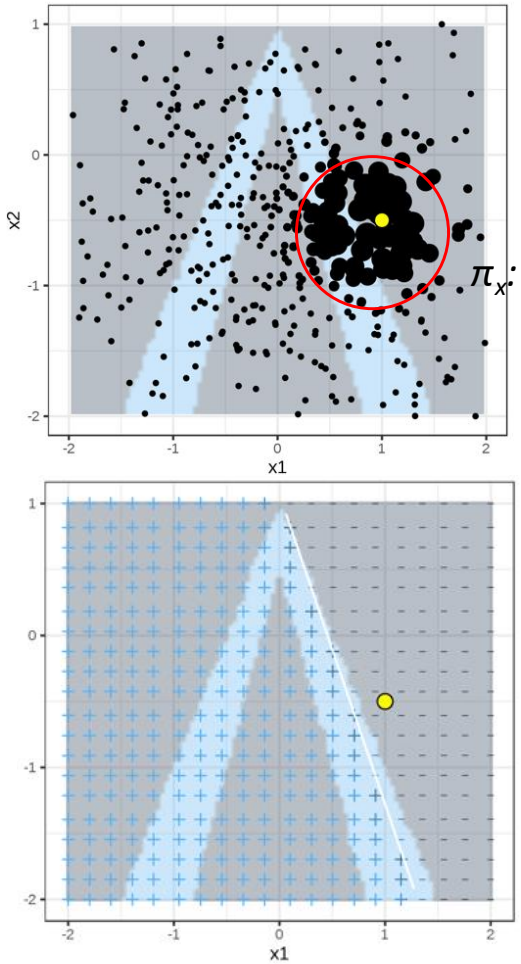
25%

How can we explain this black box model?

Idea: Let's build a (Local) Surrogate Model – Pick a set of data points that are similar to this person's data to train an interpretable model

Model-agnostic Methods

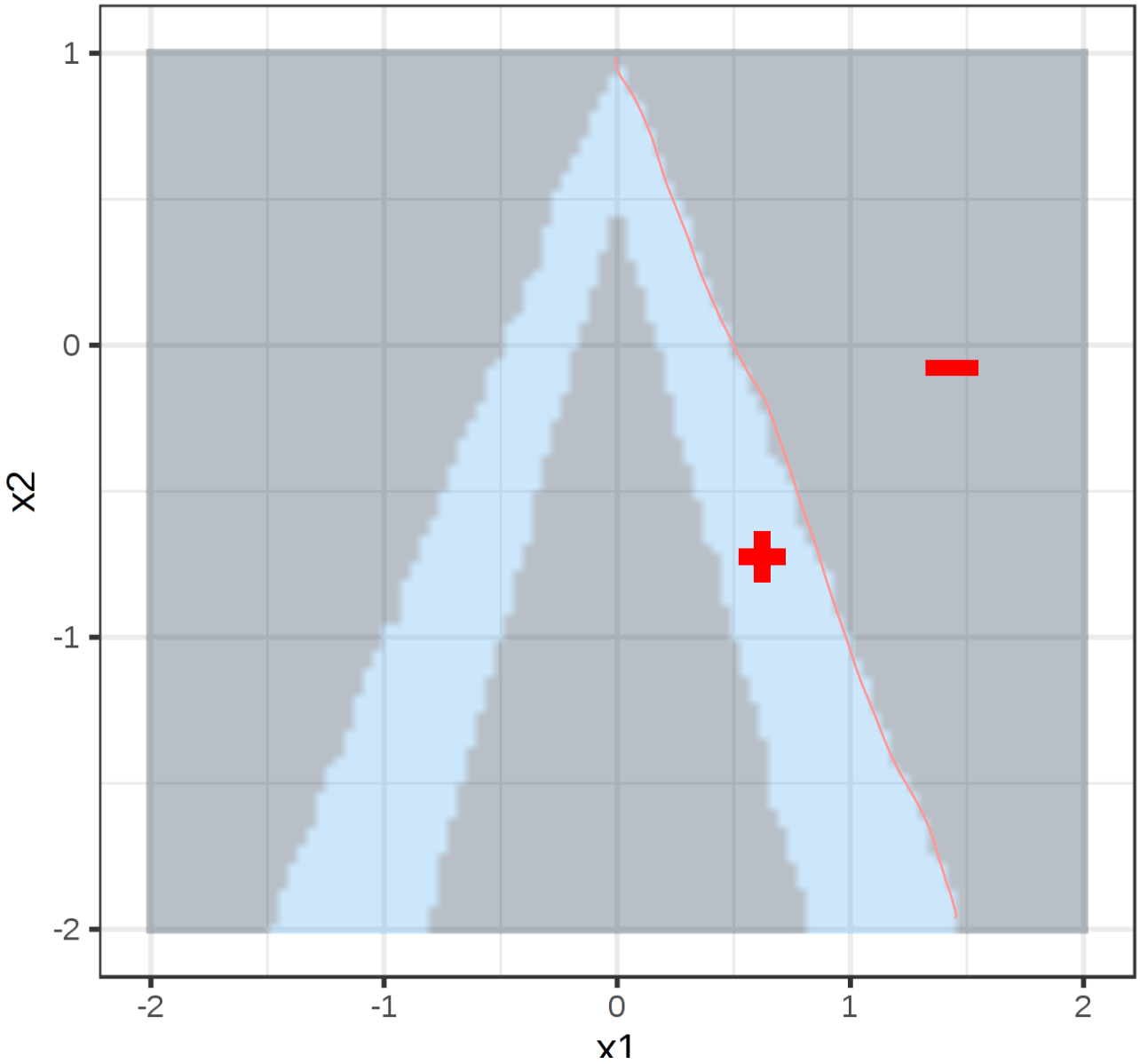
- **Surrogate models - Local Surrogate Model (LIME)**
  - LIME: Local interpretable model-agnostic explanations
  - Aims to explain individual predictions by training local surrogate models
  - Imagine you only have the black box model where you can input data points and get the predictions of the model
  - First generates a new dataset consisting of permuted samples and the corresponding predictions of the black box model
  - On this new dataset we train an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest
  - The learned model should be a good approximation of the machine learning model predictions locally, but it does not have to be a good global approximation. This kind of accuracy is also called local fidelity.



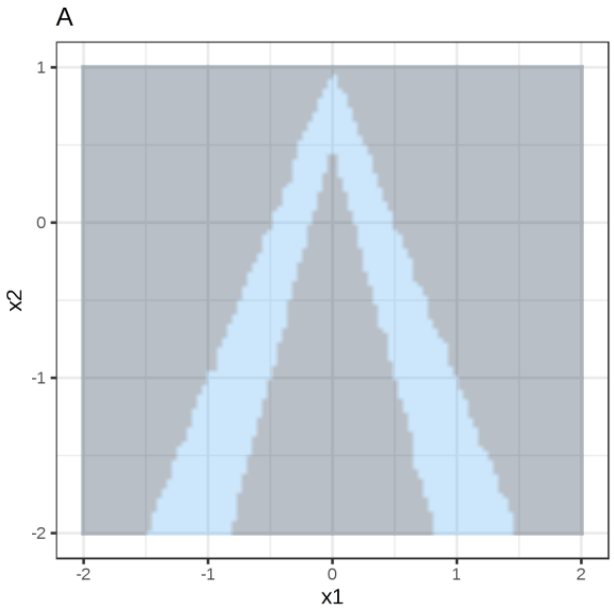
π_x: *proximity*

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin.
"Why should I trust you?: Explaining the predictions of any classifier." KDD 2016

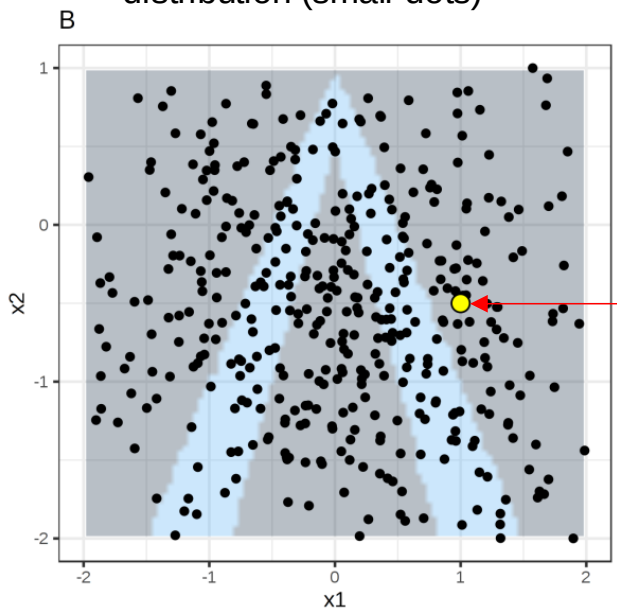
Random forest predictions given features x1 and x2



Random forest predictions given features x1 and x2



Instance of interest (big dot) and data sampled from a normal distribution (small dots)



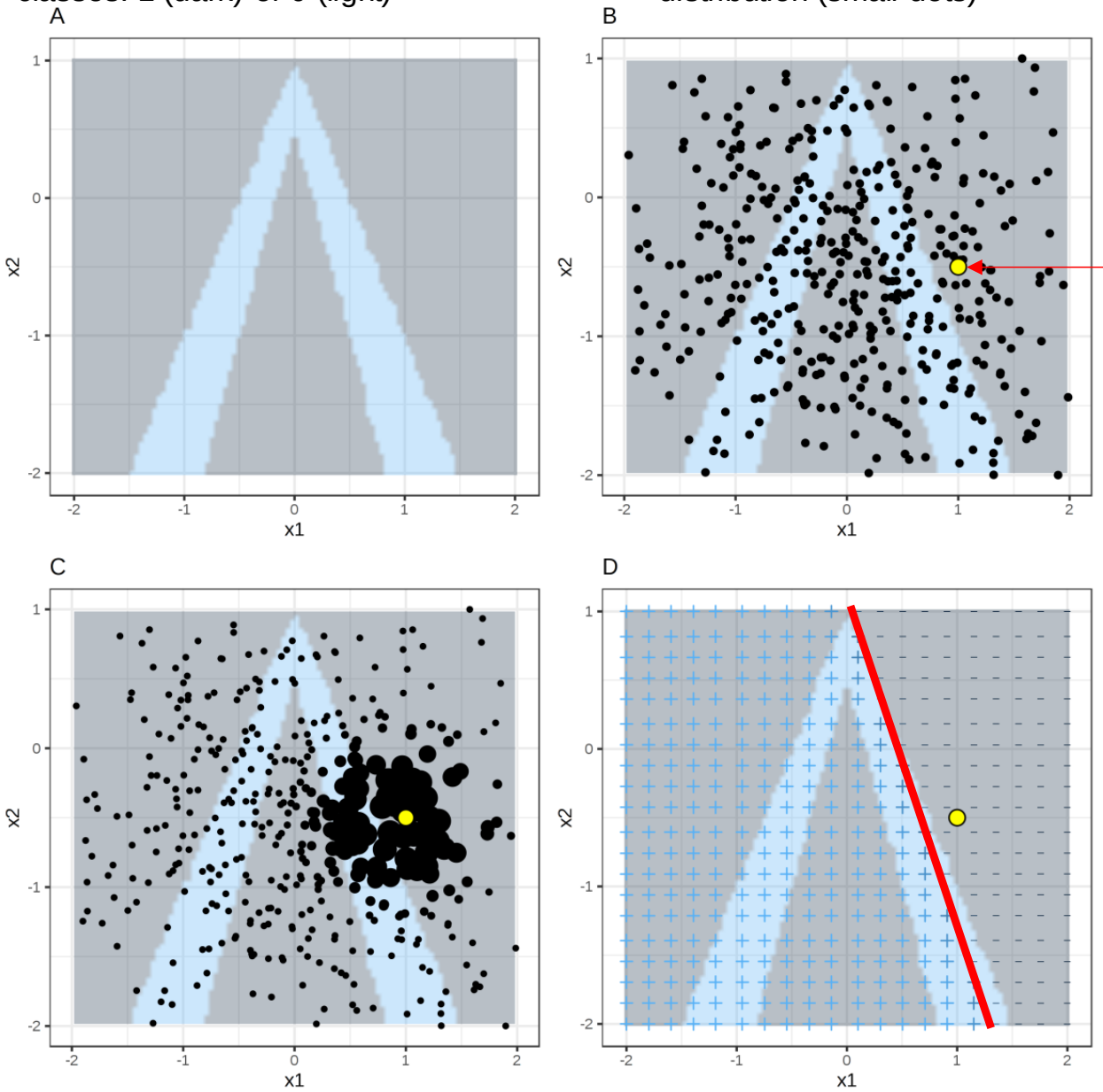
Instance of interest

John's data point

John

Random forest predictions given features x1 and x2. Predicted classes: 1 (dark) or 0 (light)

Instance of interest (big dot) and data sampled from a normal distribution (small dots)



Instance of interest

John's data point

John

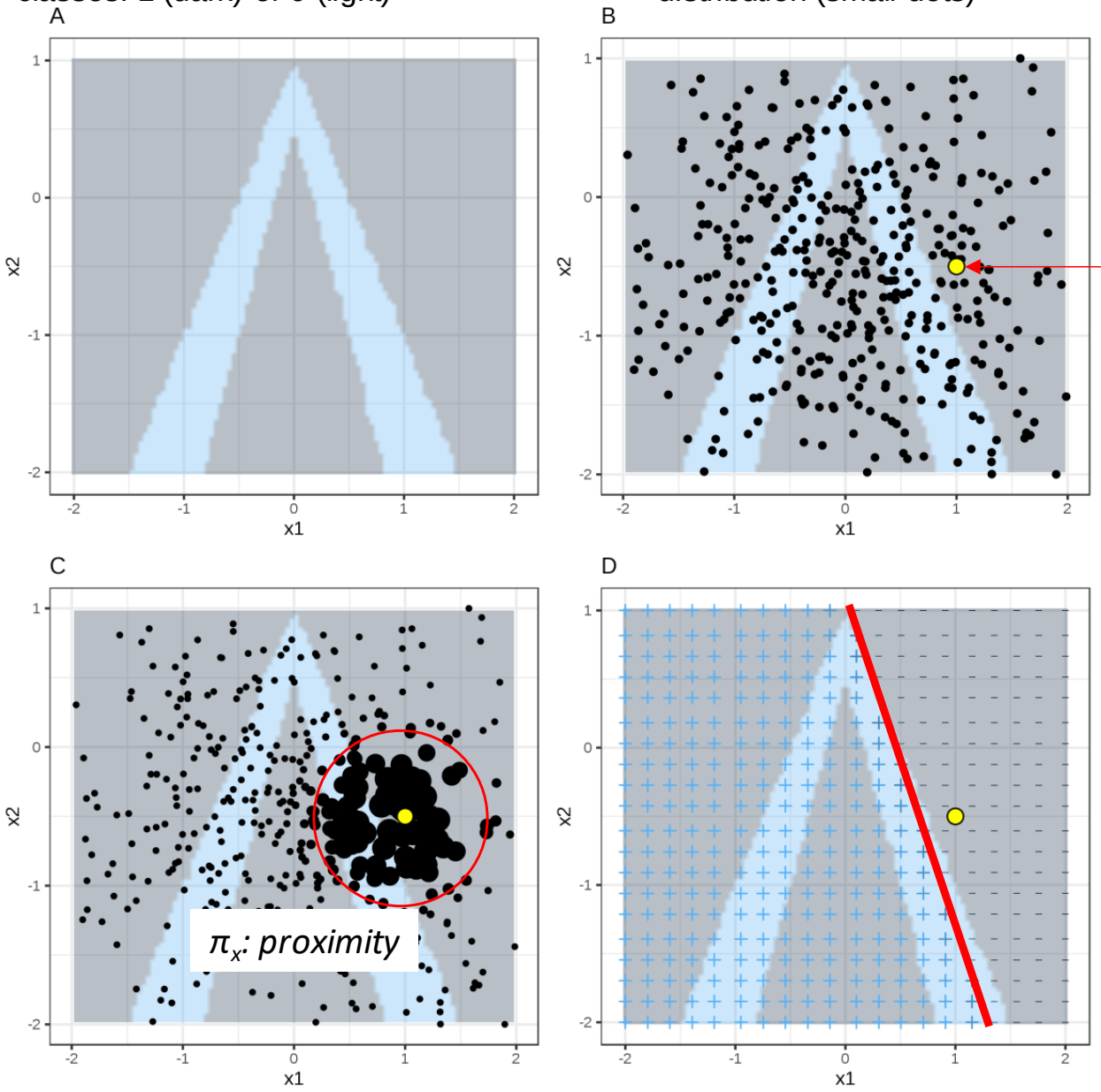
The Red line marks the decision boundary (P(class=1) = 0.5).

Assign higher weight to points near the instance of interest

Signs of the grid show the classifications of the locally learned model from the weighted samples

Random forest predictions given features x1 and x2. Predicted classes: 1 (dark) or 0 (light)

Instance of interest (big dot) and data sampled from a normal distribution (small dots)



Instance of interest

John's data point

John

The Red line marks the decision boundary (P(class=1) = 0.5).

Assign higher weight to points near the instance of interest

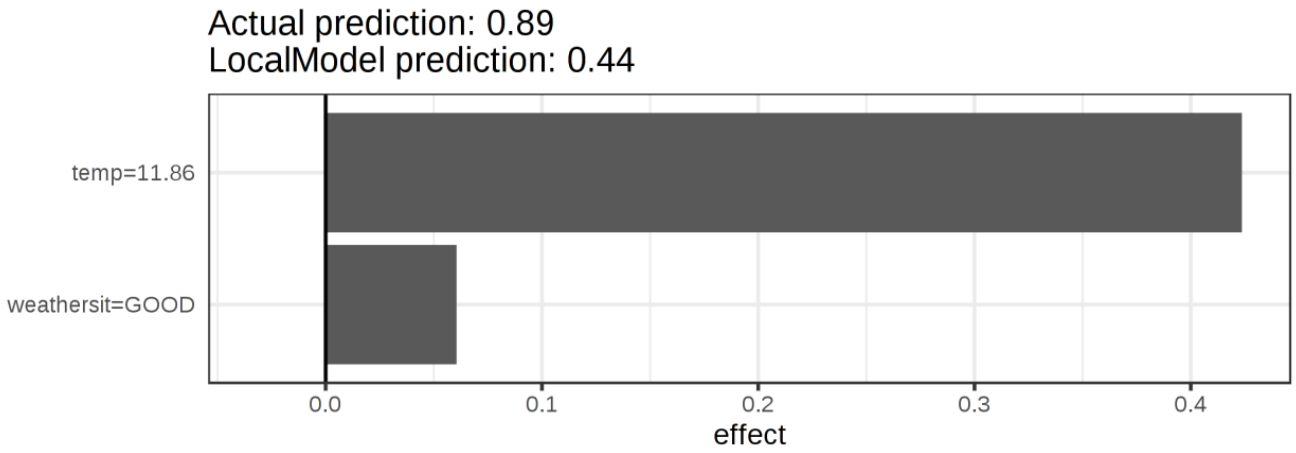
Signs of the grid show the classifications of the locally learned model from the weighted samples

LIME - Example

- **Bike rental classification: classifying whether a given day will be above the trend line?**
  - **Blackbox model:** random forest with 100 trees
  - **Local model:** sparse local linear model

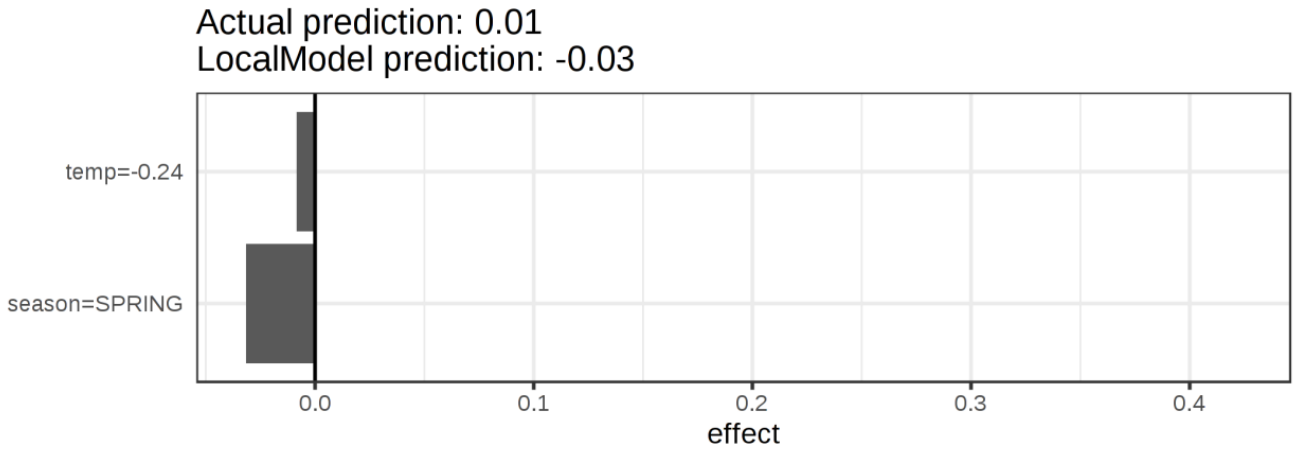
Day #1

Warmer temp
Good weather



Day #2

Cold temperature
Season



Model-agnostic Methods

- Consider a linear model

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$$

x is the instance for which the contribution is calculated
Each x_j is a feature value, with j = 1, ... , p
The β_j is the weight corresponding to feature j

- Contribution φ_j of j-th feature on prediction f^(x) can be defined as feature effect of x_j – avg. effect

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

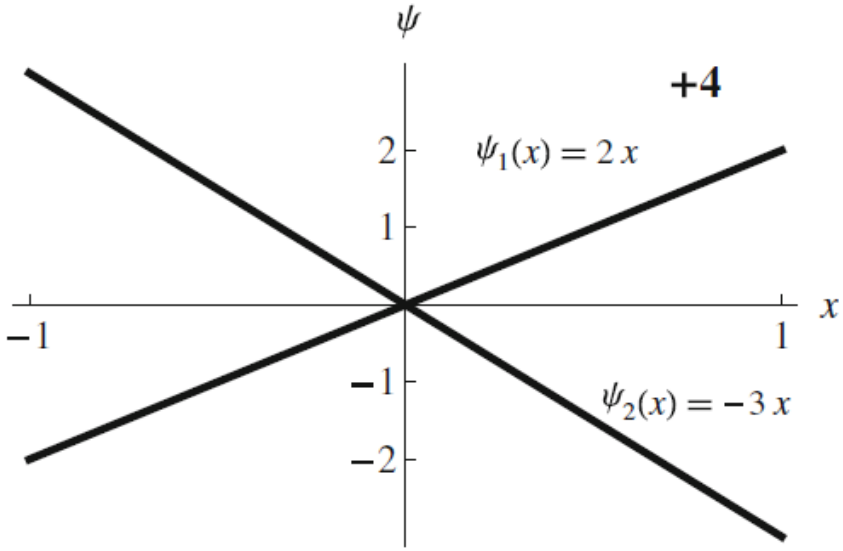
Feature j-th contribution (or situational importance)

Feature effect of x_j

Avg. effect of feature j

Model-agnostic Methods

- Consider a model: $f(x_1, x_2) = 2x_1 - 3x_2 + 4$
  - Both input features uniformly distributed on [-1,1]
- How much do the two input features contribute for the prediction of f(1/2, 1/3)?
  - Contribution of the 1st feature x_1: $2*1/2 - E[X_1] = 1 - 0 = 1$
  - Contribution of the 2nd feature x_2: $-3*1/3 - E[X_2] = -1 - 0 = -1$



Model-agnostic Methods

- Sum of all the feature contributions

$$\sum_{j=1}^{p} \phi_j(\hat{f}) = \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j))$$

$$= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j))$$

$$= \hat{f}(x) - E(\hat{f}(X))$$

predicted value for data point *x*

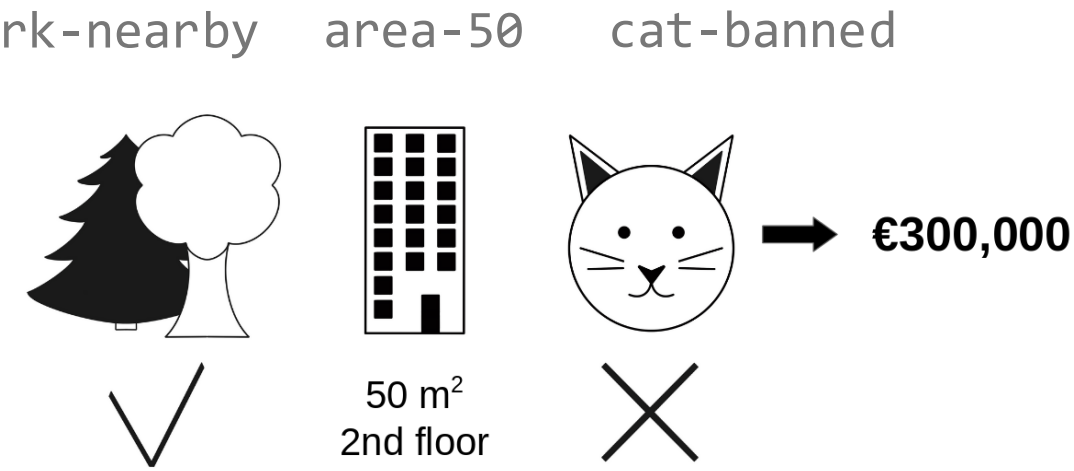
avg predicted value

Critical Issue: Linearity assumption!
(interaction among features not considered)

Model-agnostic Methods

- Consider a model that predicts apartment prices

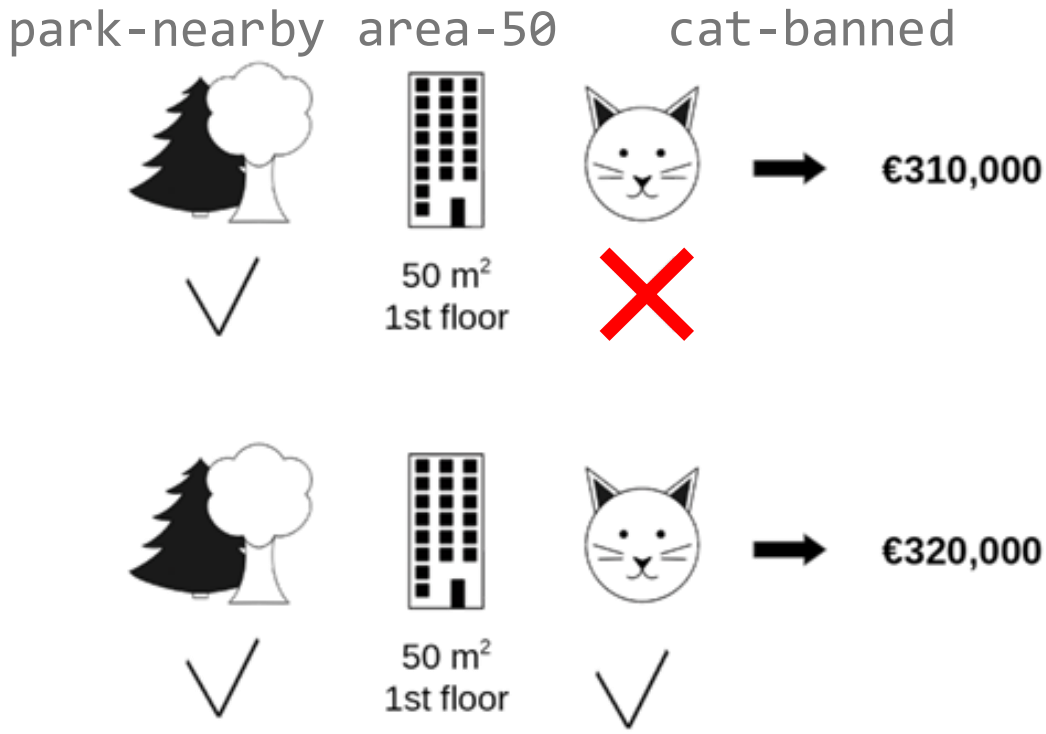
The apartment has a size of 50 m^2, is located on the 2nd floor, has a park nearby and cats are banned

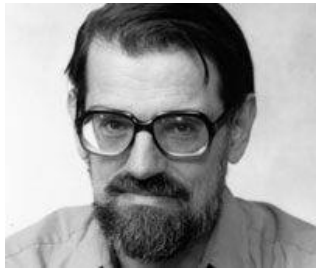


Can we explain how each of these feature values contributed to the prediction?

Model-agnostic Methods

- Estimating the contribution of cat-banned





Lloyd Shapley

Contribution of cat-banned: €310,000 - €320,000 = -€10.000
(this relative contribution is also known as *Shapley value*)

Model-agnostic Methods

- Consider all possible combinations (=coalitions)

Without cat-banned	With cat-banned
No feature values	No feature values + **Cat-banned**
park-nearby	park-nearby + **Cat-banned**
size-50	size-50 + **Cat-banned**
floor-2nd	floor-2nd + **Cat-banned**
park-nearby+size-50	park-nearby+size-50 + **Cat-banned**
park-nearby+floor-2nd	park-nearby+floor-2nd + **Cat-banned**
size-50+floor-2nd	size-50+floor-2nd + **Cat-banned**
park-nearby+size-50+floor-2nd	park-nearby+size-50+floor-2nd + **Cat-banned**

Contribution of cat-banned
= ∑ { Price(w/o cat-banned) – Price(w/ cat-banned) }

Model-agnostic Methods

- But it's too expensive to consider all combinations
- Approximation via random sampling (*m* instances)

select, at random, permutation $\mathcal{O} \in \pi(n)$
select, at random, $w \in \mathcal{X}$
construct two instances:

take values from x | take values from w

$\vec{b}_1 \leftarrow$ [preceding i-th in $\mathcal{O}$] [i] [succeeding i-th in $\mathcal{O}$]

take values from x | take values from w

$\vec{b}_2 \leftarrow$ [preceding i-th in $\mathcal{O}$] [i] [succeeding i-tega v $\mathcal{O}$]

$$\varphi_i(x) \leftarrow \varphi_i(x) + f(\vec{b}_1) - f(\vec{b}_2)$$

Explaining prediction models and individual predictions with feature contributions, Erik Štrumbelj and Igor Kononenko 2014

Model-agnostic Methods

3

- SHAP (SHapley Additive exPlanations)
  - **f** is a black box machine learning model to be explained
  - **g** is the explanation model

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

    - $z' \in \{0,1\}^M$ is the coalition vector (=simplified/reduced features)
    - **M** is the maximum coalition size
    - $\varphi_j \in R$ is the feature attribution for a feature j, the Shapley values

  - Train the linear model **g** by optimizing the following loss function L:

$$L(f, g, \pi_x) = \sum_{z' \in Z} [f(h_x(z')) - g(z')]^2 \pi_x(z') \quad \textit{where} \quad \pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|}|z'|(M-|z'|)}$$

Z is the training data

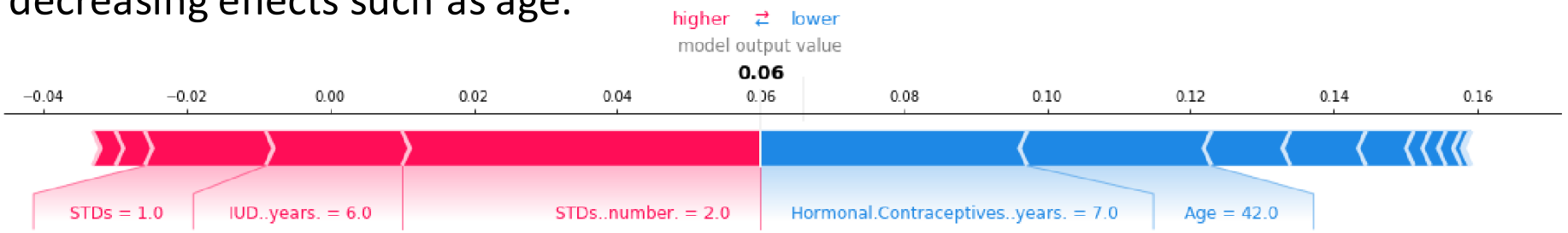
$h_x: \{0,1\}^M \rightarrow \boldsymbol{R}^p$

SHAP kernel

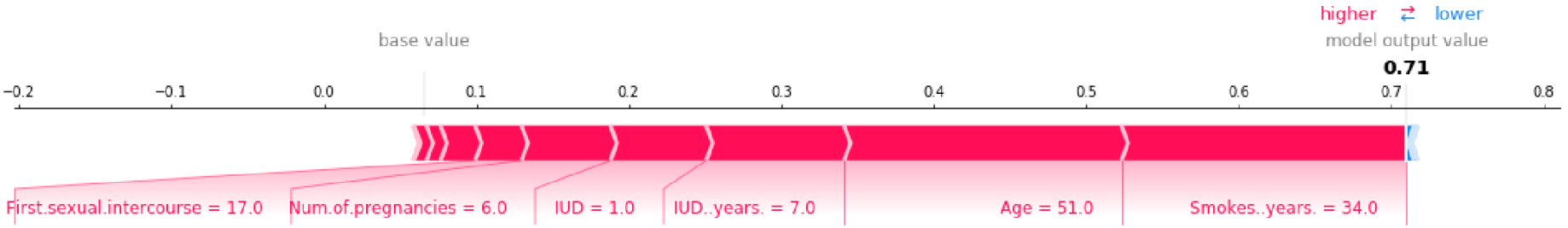
Explaining prediction models and individual predictions with feature contributions, Erik Štrumbelj and Igor Kononenko 2014

Explain the predicted cancer probabilities of two individuals

The first woman has a low predicted risk of 0.06. The baseline – the average predicted probability – is 0.066. Risk increasing effects such as STDs are offset by decreasing effects such as age.



The second woman has a high predicted risk of 0.71. Age of 51 and 34 years of smoking increase her predicted cancer risk.



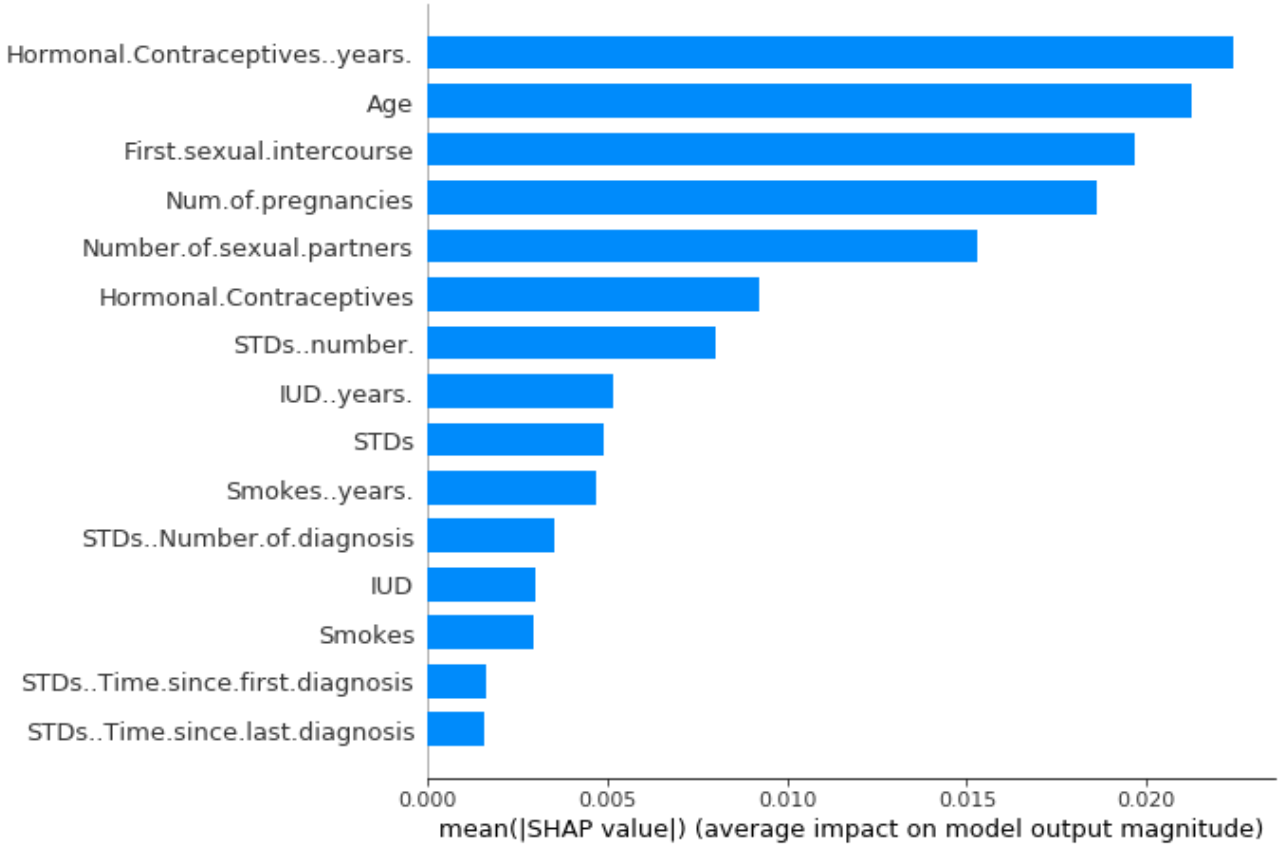
Each feature value is a force that either increases or decreases the prediction. The prediction starts from the baseline. The baseline for Shapley values is the average of all predictions. In the plot, each Shapley value is an arrow that pushes to increase (positive value) or decrease (negative value) the prediction. These forces balance each other out at the actual prediction of the data instance.

Explain the predicted cancer probabilities of two individuals

The idea behind SHAP feature importance is simple: Features with large absolute Shapley values are important. Since we want the global importance, we average the absolute Shapley values per feature across the data:

Sum of SHAP values of all samples (= n)

$$I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$$

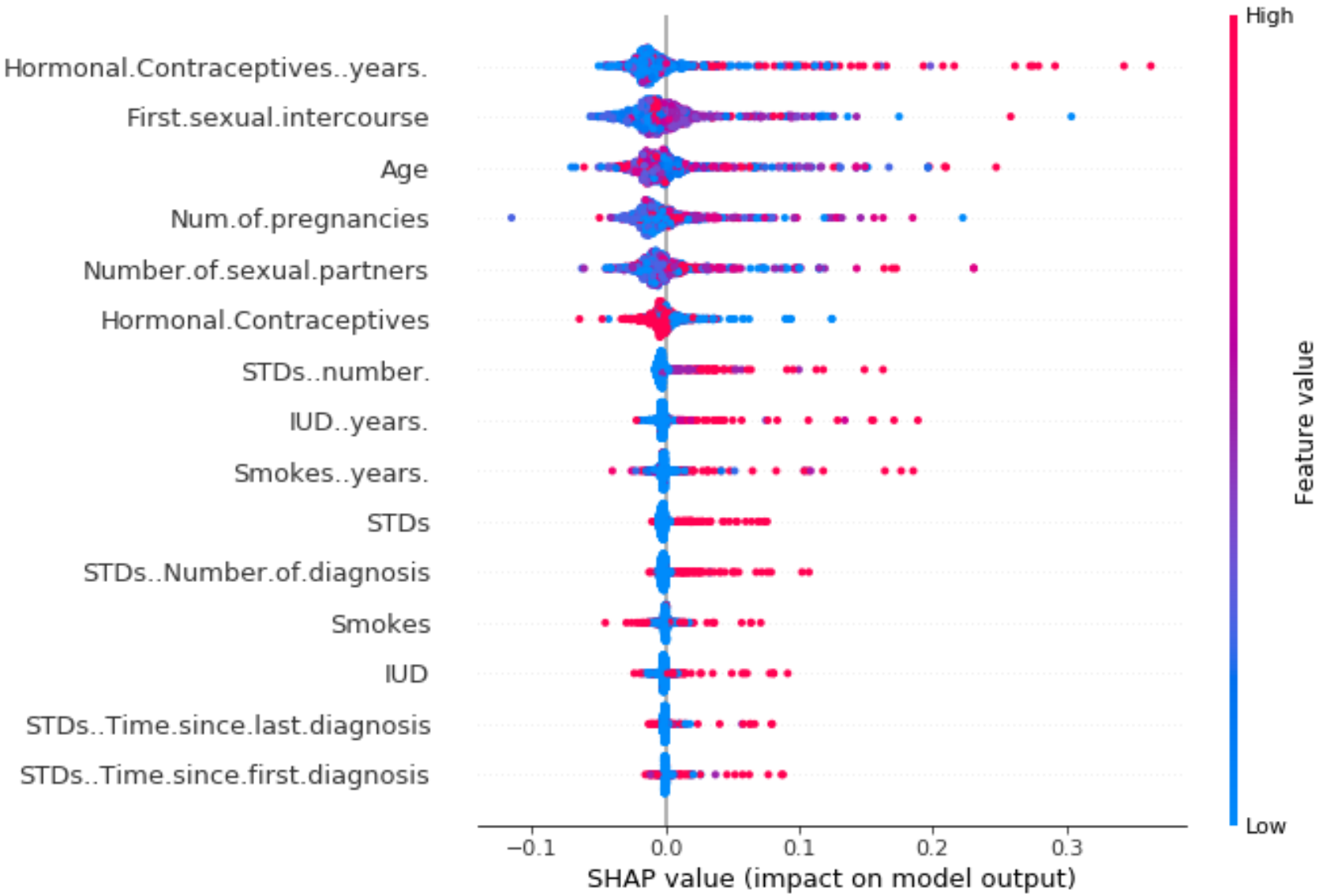


SHAP feature importance measured as the mean absolute Shapley values. The number of years with hormonal contraceptives was the most important feature, changing the predicted absolute cancer probability on average by 2.4 percentage points (0.024 on x-axis).

SHAP Summary Plot

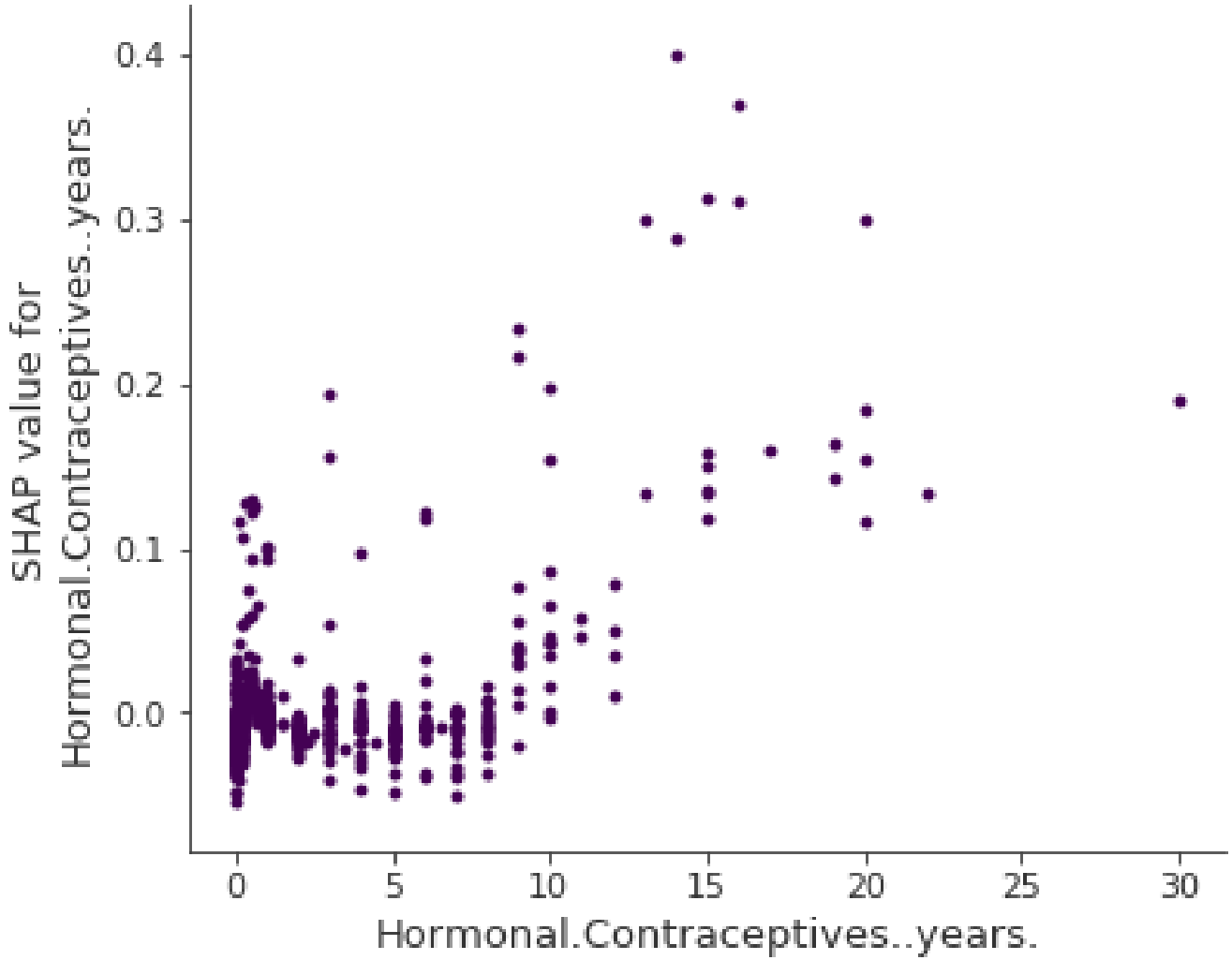
Each feature ranked by its mean (scatter plot with jittering)

Color – Actual Feature Value



SHAP summary plot. Low number of years on hormonal contraceptives reduce the predicted cancer risk, a large number of years increases the risk. Your regular reminder: All effects describe the behavior of the model and are not necessarily causal in the real world.

SHAP Summary Plot



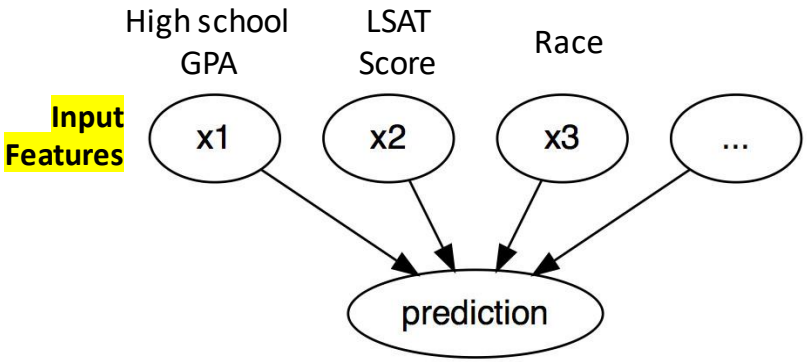
HAP dependence plot for years on hormonal contraceptives. Compared to 0 years, a few years lower the predicted probability and a high number of years increases the predicted cancer probability.

Example-Based Explanations

- Counterfactual Explanations
  - Finding the smallest change (what if) to the feature values that modifies the prediction to a predefined output
    - Cf) counter(opposing)-factual = not a fact → it's because we're doing "what if" analysis (e.g., what if kangaroos had no tails), by arbitrarily modifying input data (and using that data for prediction) (e.g., how that results in walk stability)
- Adversarial Examples
  - Adversarial examples are counterfactual examples with the aim to deceive the model, not to interpret it (goal is to check model reliability)
  - Small, intentional feature perturbations that cause a machine learning model to make a false prediction
- Prototypes & Criticisms
  - A prototype is a data instance that is representative of all the data.
  - A criticism is a data instance that is not well represented by the set of prototypes
- Influential Instances
  - Removing an instance from the training data will considerably change the parameters or predictions of the model

Example-Based Explanations

- Counterfactual Explanations
  - Finding the smallest change (what if) to the feature values that changes the prediction to a predefined output
    - Cf) counter(opposing)-factual = not a fact → it's because we're doing "what if" analysis (e.g., what if kangaroos had no tails), by arbitrarily modifying input data (and using that data for prediction) (e.g., how that results in walk stability)
  - Predict a student's (normalized) avg. grade of the first year at law school, based on high school GPA, law school entrance exam (LSAT) score, and Race

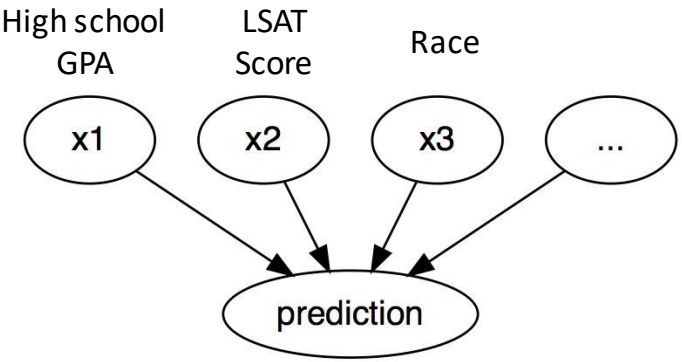


	Score	GPA	LSAT	Race
Above-avg prediction	0.17	3.1	39.0	0
	0.54	3.7	48.0	0
Below-avg prediction	-0.77	3.3	28.0	1
	-0.83	2.4	28.5	1
	-0.57	2.7	18.3	0
	Predicted normalized score		Original	

Find counterfactual explanations for each student: How would the input features need to be changed, to get a predicted score of 0 (=avg)?

Wachter et. al (2017)

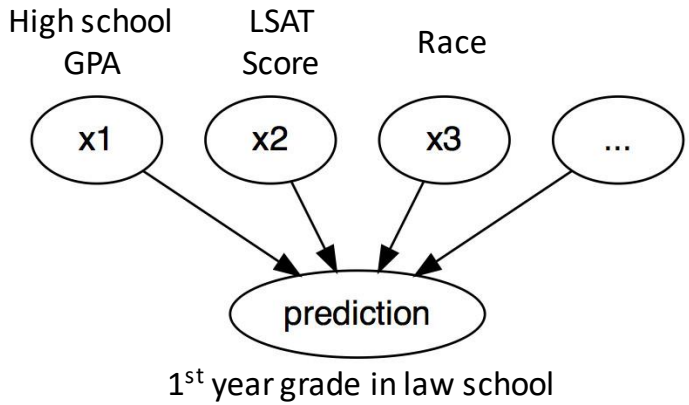
Example-Based Explanations



1st year score (grade) in law school

- Counterfactual Explanations
  - Finding the smallest change (what if) to the feature values that changes the prediction to a predefined output
    - Cf) counter(opposing)-factual = not a fact → it's because we're doing "what if" analysis (e.g., what if kangaroos had no tails), by arbitrarily modifying input data (and using that data for prediction) (e.g., how that results in walk stability)
  - Predict a student's (normalized) avg. grade of the first year at law school, based on high school GPA, law school entrance exam (LSAT) score, and Race
  - Find counterfactual explanations for each student: How would the input features need to be changed, to get a predicted score of 0 (=avg)?

Example-Based Explanations



- Counterfactual Explanations
  - Finding the smallest change (what if) to the feature values that changes the prediction to a predefined output
    - Cf) counter(opposing)-factual = not a fact → it's because we're doing "what if" analysis, pretending something would happen using the model with modified data
  - Predict a student's (normalized) avg. grade of the first year at law school, based on high school GPA, law school entrance exam (LSAT) score, and Race
  - Find counterfactual explanations for each student: How would the input features need to be changed, to get a predicted score of 0 (avg score)?
    - Counterfactuals should be **small changes**, but lead to the **desired outcomes (=0)**

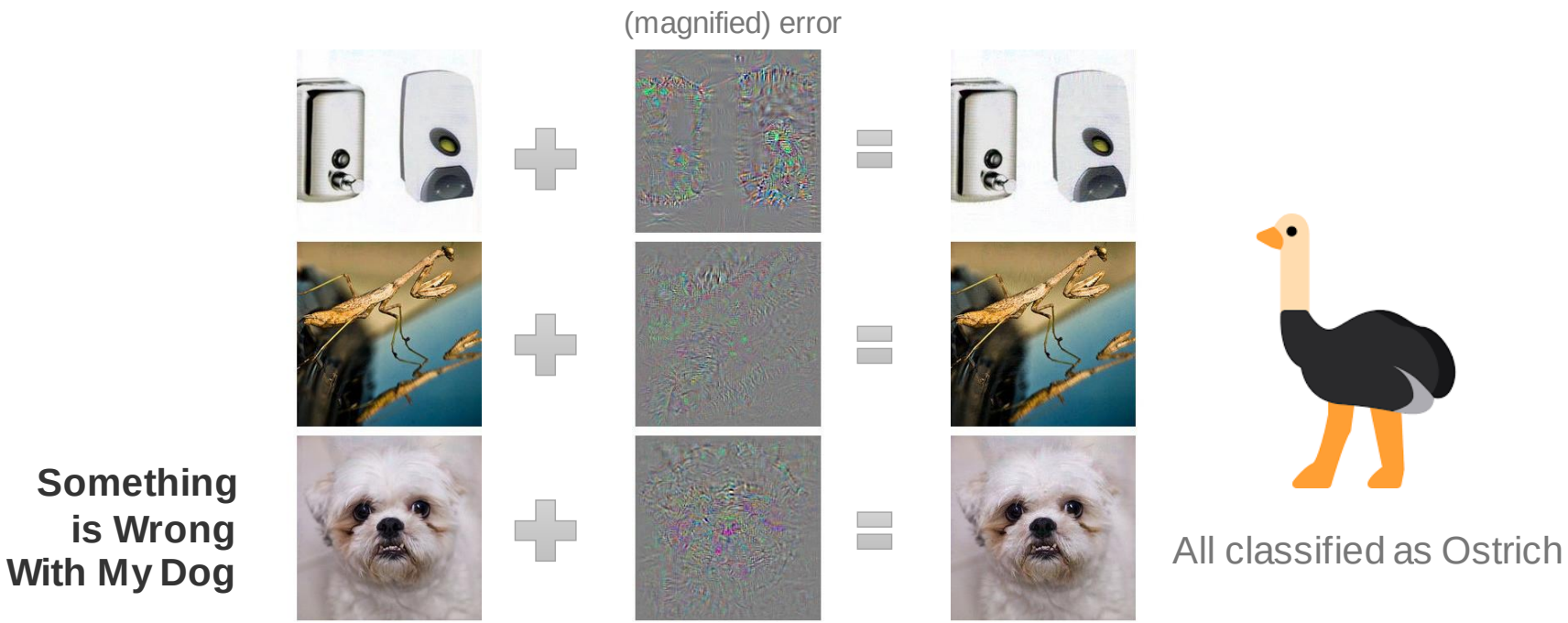
	Score	GPA	LSAT	Race	GPA x'	LSAT x'	Race x'
Above-avg prediction	0.17	3.1	39.0	0	3.1	34.0	0
	0.54	3.7	48.0	0	3.7	32.4	0
Below-avg prediction	-0.77	3.3	28.0	1	3.3	33.5	0
	-0.83	2.4	28.5	1	2.4	35.8	0
	-0.57	2.7	18.3	0	2.7	34.9	0
	Predicted score	Original			Counterfactuals (as small changes as possible)		

Example-Based Explanations

- Counterfactual Explanations
  - Finding the smallest change (what if) to the feature values that changes the prediction to a predefined output
    - Cf) counter(opposing)-factual = not a fact → it's because we're doing "what if" analysis, pretending something happens using the model
- Adversarial Examples
  - Adversarial examples are counterfactual examples with the aim to deceive the model, not interpret it
  - Small, intentional feature perturbations that cause a machine learning model to make a false prediction
- Prototypes
  - A prototype is a data instance that is representative of all the data. A criticism is a data instance that is not well represented by the set of prototypes
- Influential Instances
  - Removing an instance from the training data will considerably change the parameters or predictions of the model? (outlier vs. influential instance)

Example-Based Explanations

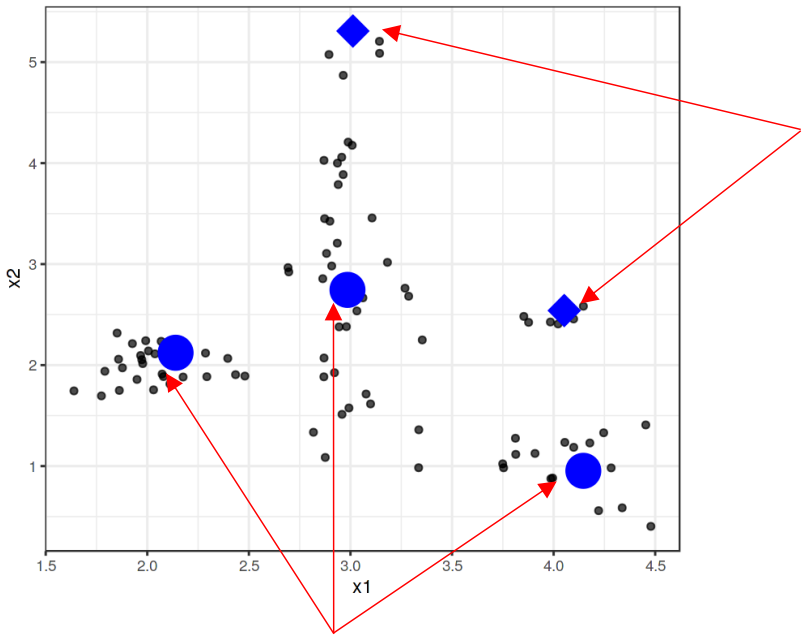
- Adversarial Examples
  - Adversarial examples are counterfactual examples with the aim to deceive the model, not to interpret it
  - Small, intentional feature perturbations that cause a machine learning model to make a false prediction



Adversarial examples for AlexNet by Szegedy et. al (2013)

Example-Based Explanations

- Prototypes
  - Prototypes & criticisms represent the dataset, and they can be used to create an interpretable model



A criticism is a data instance that is *not well represented* by the set of prototypes

A prototype is a data instance that is *representative* of all the data

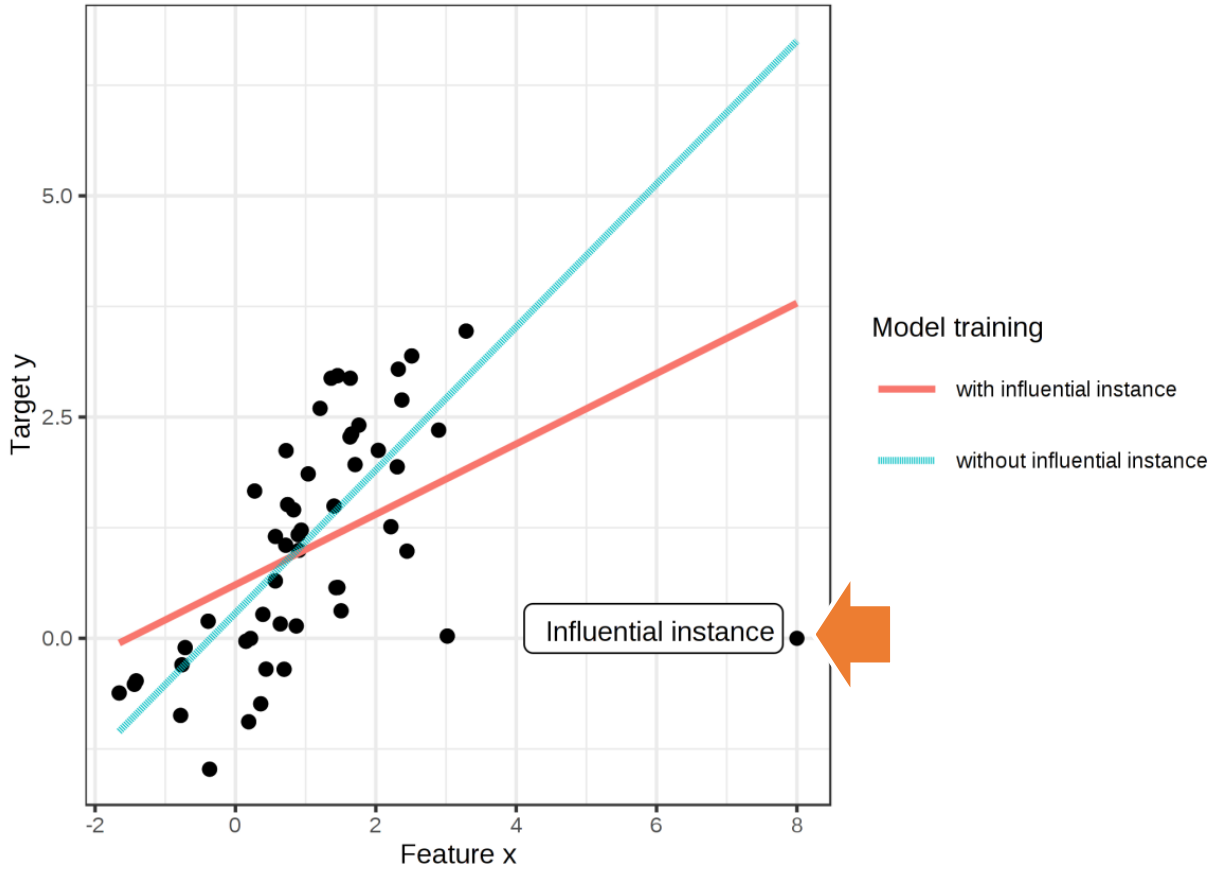
Prototypes for a handwritten digits dataset

Example-Based Explanations

- Influential Instances
  - Removing an instance from the training data will considerably change the parameters or predictions of the model? (testing model reliability/trust)

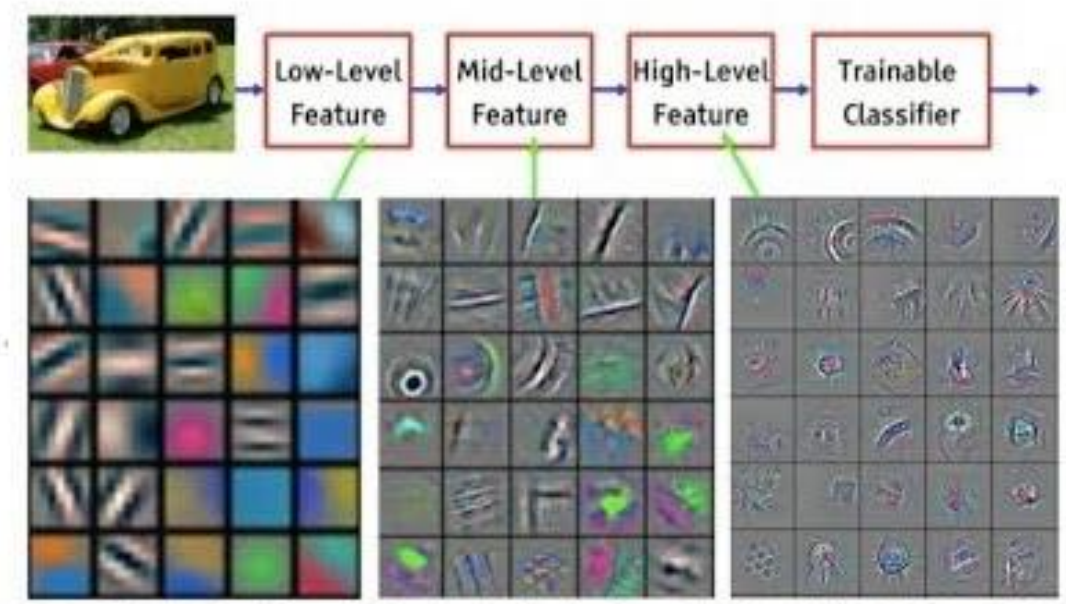
Outliers can be interesting data points (e.g. criticisms). When an outlier influences the model it is also an influential instance.

How to find influential instances? (e.g., delete & retrain)



Neural Network Interpretation

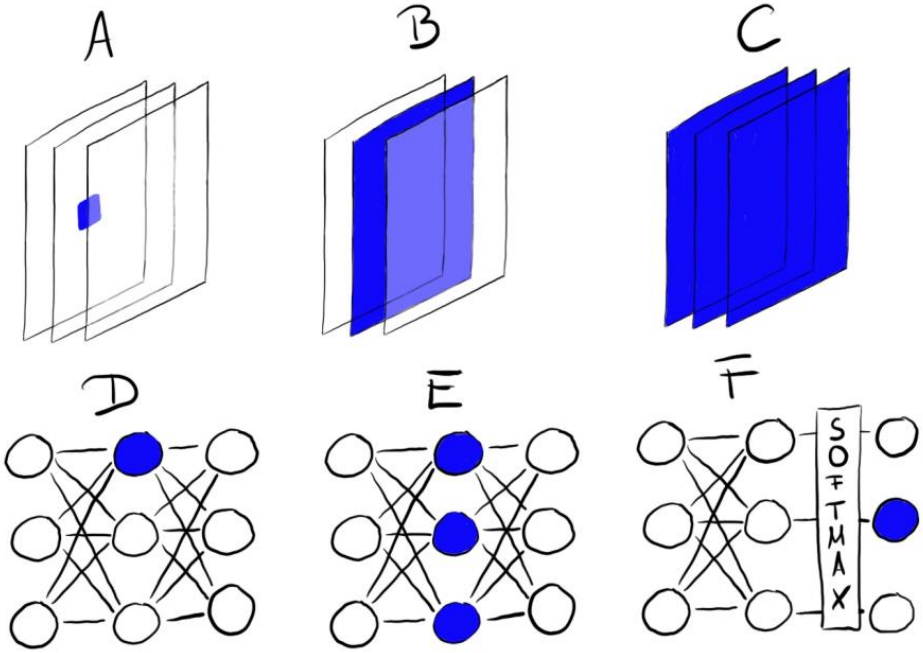
- Using model-agnostic methods such as LIME and SHAP or example-based explanations
- Alternatives: examining internal structures of neural networks!
  - Feature visualization: what features has the neural network learned?



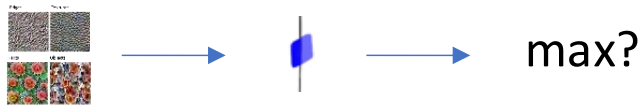
Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Neural Network Interpretation

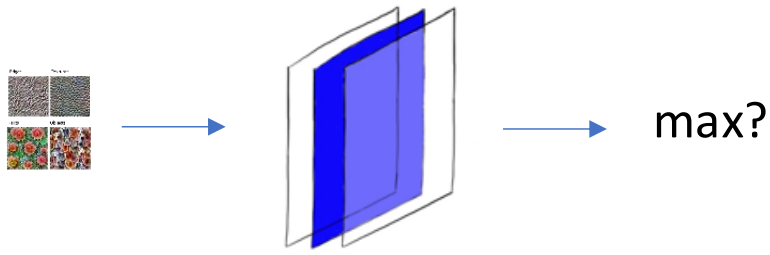
- Alternatives: examining internal structures of neural networks!
  - Feature visualization: what features has the neural network learned?
    - Find an image that maximizes the activation of a given unit (e.g., A-E below)



Feature visualization can be done for different units. A) Convolution neuron, B) Convolution channel, C) Convolution layer, D) Neuron, E) Hidden layer, F) Class probability neuron (or corresponding pre-softmax neuron)



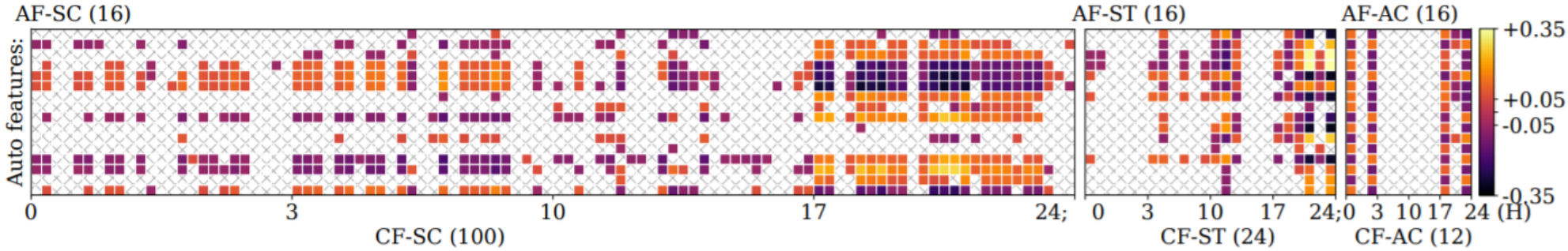
$$img^* = \arg\max_{img} h_{n,x,y,z}(img)$$



$$img^* = \arg\max_{img} \sum_{x,y} h_{n,x,y,z}(img)$$

Neural Network Interpretation

- Sensor data interpretation?
  - Correlation matrix between the corresponding modalities of auto-learned features and hand-crafted features
    - auto-learned features (AF-) and hand-crafted features (CF-) pairs



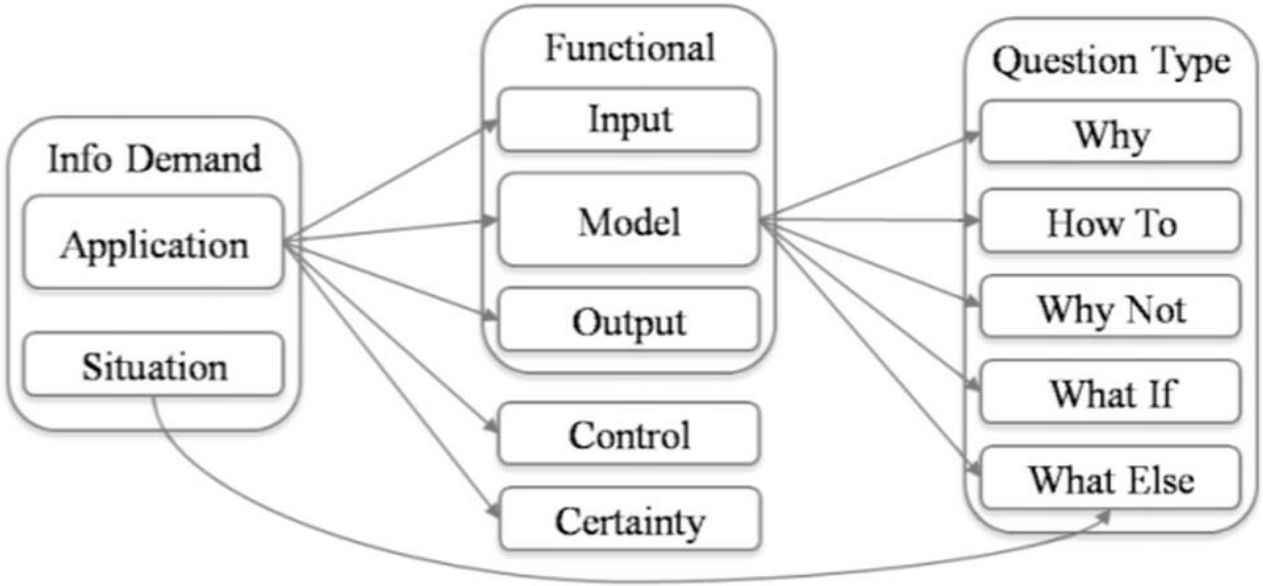
Auto Features	Crafted Features				
	top-1	top-2	top-3	top-4	top-5
AF-SC#6	17H+: unnormalized mean -0.30	17H+: normalized s.d. -0.30	17H+: count of peaks -0.29	17H+: 30-min median peaks -0.28	17H+: area under curve (AUC) -0.28
AF-ST#5	17H+: median while still -0.31	17H+: raw value median -0.30	10H-17H: raw value min 0.18	17H+: raw value s.d. -0.15	3H-10H: raw value min 0.11
AF-AC#5	17H+: stillness percent 0.16	17H+: step count -0.13	0H-3H: stillness percent 0.13	0H-3H: step count -0.12	17H+: mean movement step time 0.08

Most interpreted auto features for each sensor modality, as well as the top-5 highest correlated crafted features

Extraction and Interpretation of Deep Autoencoder-based Temporal Features from Wearables for Forecasting Personalized Mood, Health, and Stress, Boning Li, Akane Sano, IMWUT 2020

Users Demand for Intelligibility

- Hierarchical representation of intelligibility types users want to know

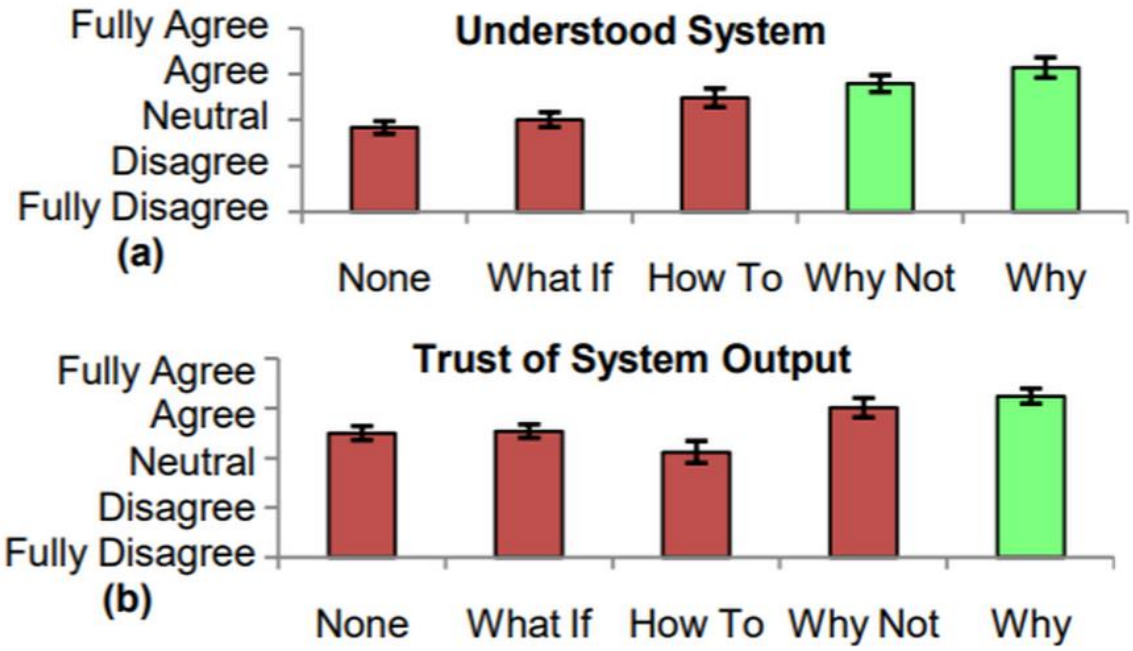


User demands were extracted based on these four applications: (1) Desktop interruptibility management application (an Instant Messenger plugin), (2) Remote person monitoring peripheral display (Digital Family Portrait), (3) **Context-aware reminder application (CybreMinder), and (4)** Mobile context-aware tour guide (CyberGuide)

Assessing Demand for Intelligibility in Context-Aware Applications, Ubicomp 2009

Intelligibility Affects Trust

- Different types of explanations would result in changes in user experience (i.e., understanding of the system, perceptions of trust)

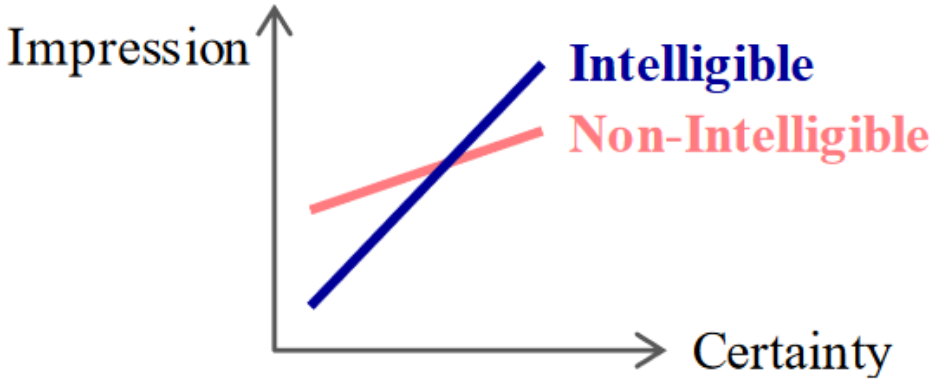


Green: Significant differences observed

Why and Why Not Explanations Improve the Intelligibility of Context-Aware Intelligent Systems, CHI 2009

Intelligibility Affects Trust

- **Inference uncertainty** will negatively influence user impression/trust
- But providing intelligibility will help improve impression (when an application is certain of its actions, but it will harm impressions when it is uncertain)



Conceptual diagram
(linearity assumed)

Investigating Intelligibility for Uncertain Context-Aware Applications, Ubicomp 2011

Intelligibility Affects Labeling Quality

- Improving labeling accuracy by providing contextual features of the sample to be labeled (Rosenthal and Dey 2010)
- Highest labeling accuracy occurred when the system provided sufficient contextual features and current predictions without uncertainty information
- This line of research demonstrates that the way in which information is presented (for example, with or without context) can greatly affect the quality of the response elicited from the user

Dimension	Description	Activity Recognition Example
Uncertainty	Notify labeler that it is uncertain of the label	"Cannot determine your activity."
Amount of Context	Provide varying amounts of contextual information (none, sufficient, extra)	**Sufficient**: "Your feet are leaving the ground." **Extra:** "Your feet are leaving the ground together and repeatedly."
High/Low-Level Context	Give either low (sensor) level context or high (activity) level context	**Low**: "Shaking motion detected." **High**: "Your feet are leaving the ground."
Question	Ask for a label	"What activity are you doing?"
Prediction	Share the expected label for the data	"Prediction: Jumping."
User Feedback	Ask labeler to describe the important features	"How can this action be detected in the future?"

Towards Maximizing the Accuracy of Human-Labeled Sensor Data, Rosenthal and Dey, IUI 2010

Intelligibility + Feedback

- Classification explanation & user feedback
  - Classification explanation (explaining why it classified so)
  - User's feedback (for possible improvement)
    - Instant-level feedback: whether each instance is correct or not?
    - Feature-level feedback: which features would be helpful?
- Effects on user perception
  - Frustration, trust, accuracy, understanding, acceptance
  - Feedback importance, expected improvement (after feedback)
- Poor quality model (75% accuracy)
  - Explanation increased frustration; trust/acceptance reduced due to wrong explanation, but feedback helped improve these factors
- High quality model (95% accuracy)
  - No effect on frustration (good explanation)

No Explainability without Accountability: An Empirical Study of Explanations and Feedback in Interactive ML, ACM CHI 2020

Possible to Generate Explanations

- Survey of various context-aware services
- Generate explanations for four decision model types (Decision Tree, Naive Bayes, HMM, KNN)

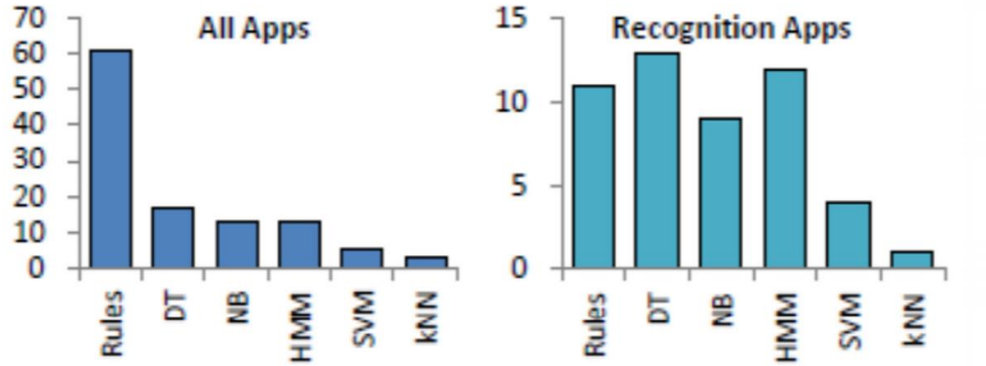


Figure 1: (Left) Counts of model types used in 109 of 114 reviewed context-aware applications. (Right) Counts for 50 recognition applications; classifiers are used most often for applications that do recognition. Key: decision tree (DT), naïve Bayes (NB), hidden Markov models (HMM), support vector machines (SVM), k-Nearest Neighbor (kNN).

Toolkit to Support Intelligibility in Context-Aware Applications, Ubicomp 2010

HCI & Ubicomp Challenges for Sensing Application Development

- Rigorous and usable intelligibility
  - How to explain, how users understand/make use of, how to interact?
  - Required to define efficacy, usability, user experiences
- Designing interactive explanations and interfaces for intelligibility
  - So far mostly focused on supporting data scientists in using and understanding machine-learning algorithms
  - How can we go beyond that and provide end users with such tools?

Summary

- Explainable, Accountable, Intelligible
- FAT-ML: Fair, Accountable, Transparent
- Citation Network Analysis
  - Intelligent and Ambient Systems
  - Fair, Accountable, Transparent (FAT) and Interpretable Machine Learning (iML)
  - Interaction Design & Software Learnability
- Why Explanation?
- Taxonomy of Interpretability
- Model-agnostic Methods: Permutation feature importance , Global surrogates vs. LIME (local surrogates), Sharpley value & SHAP
- Example-Based Explanations: Counterfactual Explanations, Adversarial Examples, Prototypes, Influential Instances
- Neural Network Interpretation
- Intelligibility Affects Trust