FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

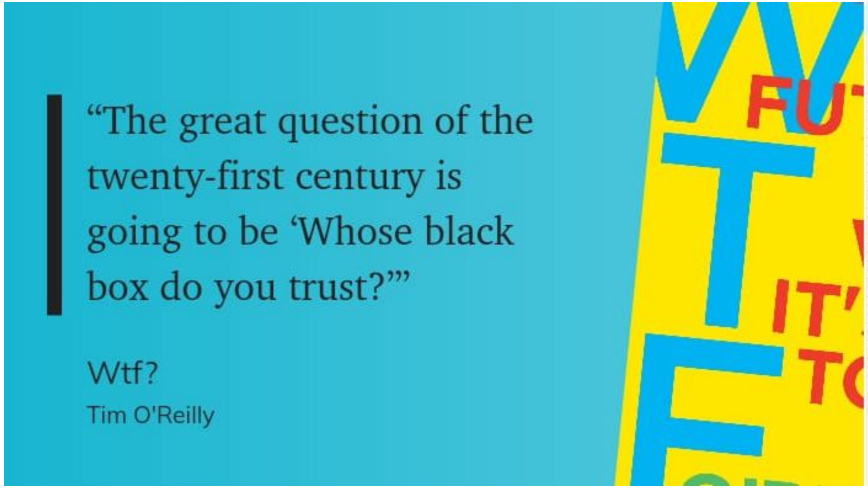
DATA
SCIENCE
LABORATORY

Interpretable and Explainable Machine Learning

Unraveling the Complexities

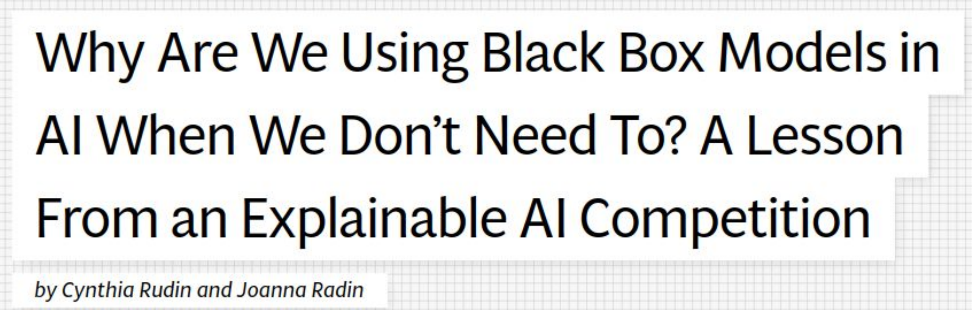
AML Spring 2023

Motivation



> “The great question of the twenty-first century is going to be ‘Whose black box do you trust?’”
>
> Wtf?
> Tim O'Reilly

WT
FU
WTF
IT'
TO



Why Are We Using Black Box Models in AI When We Don’t Need To? A Lesson From an Explainable AI Competition

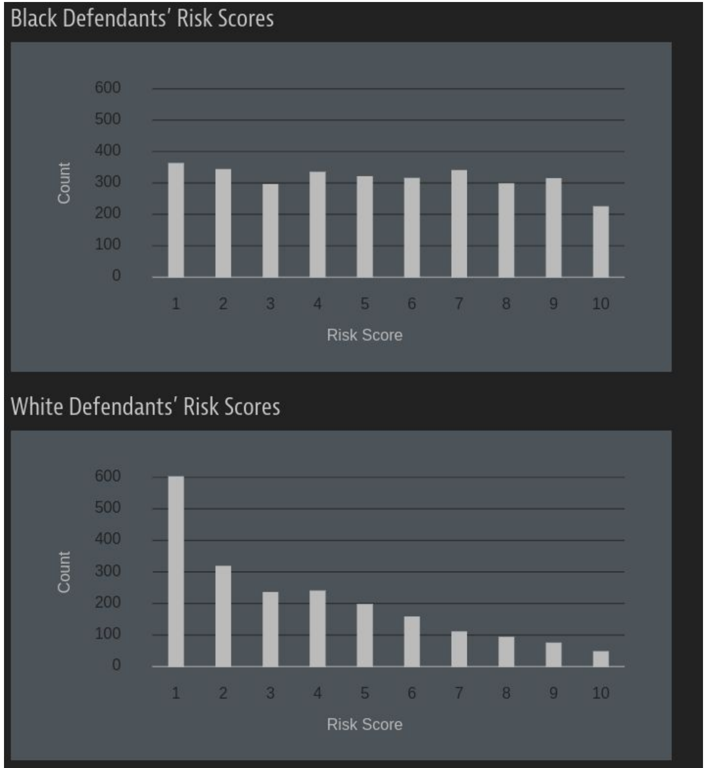
by Cynthia Rudin and Joanna Radin

COMPASS, ProPublica

- COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) is a risk assessment tool used in US the criminal justice system
  - Predicts the likelihood of recidivism for individual defendants
  - Influences decisions on bail, sentencing, and parole
  - 130+ factors
  - Might include socio-economic factors
  - expensive
- Propublica
  - Founded in 2007 by Paul Steiger, the former managing editor of The Wall Street Journal
  - investigative journalism in the public interest
  - Has won several Pulitzer Prizes and numerous other journalism awards

COMPASS vs ProPublica

- In 2016, a ProPublica investigation found that the COMPAS algorithm was biased against African-American defendants
- Black defendants were more likely to be falsely labeled as high risk, while white defendants were more likely to be falsely labeled as low risk

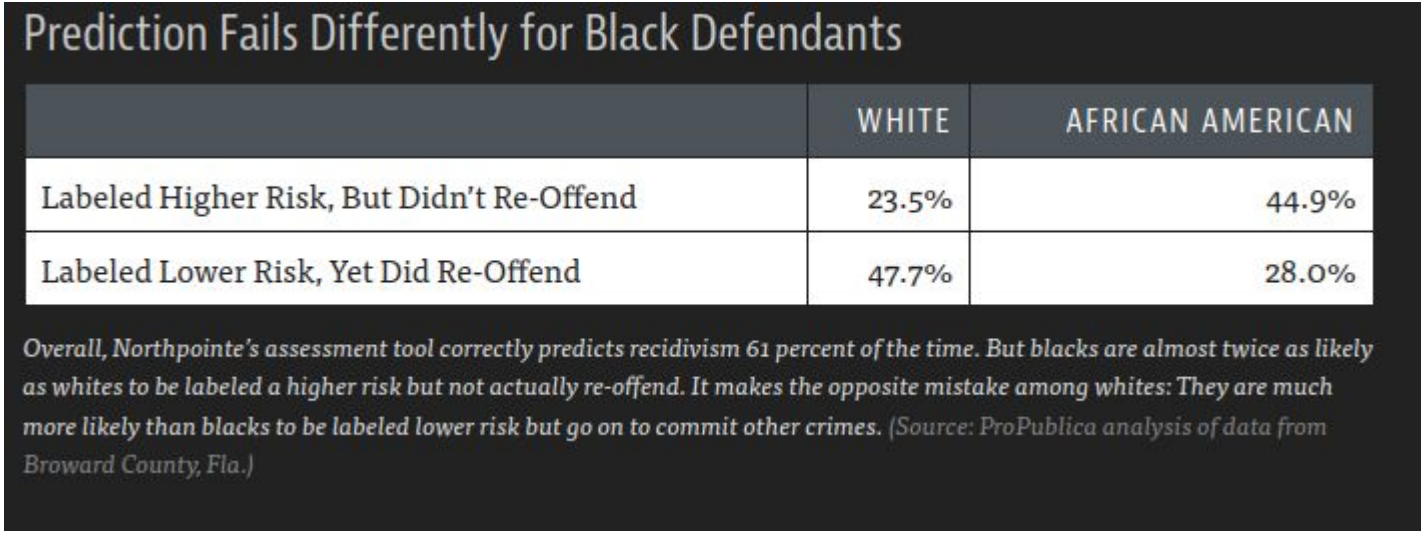


Black Defendants' Risk Scores

White Defendants' Risk Scores

These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. (Source: ProPublica analysis of data from Broward County, Fla.)

COMPASS vs ProPublica



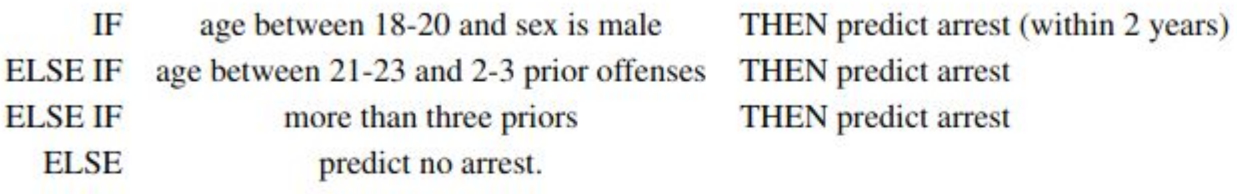
Prediction Fails Differently for Black Defendants

	WHITE	AFRICAN AMERICAN
Labeled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labeled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Overall, Northpointe's assessment tool correctly predicts recidivism 61 percent of the time. But blacks are almost twice as likely as whites to be labeled a higher risk but not actually re-offend. It makes the opposite mistake among whites: They are much more likely than blacks to be labeled lower risk but go on to commit other crimes. (Source: ProPublica analysis of data from Broward County, Fla.)

COMPAS vs CORRELS

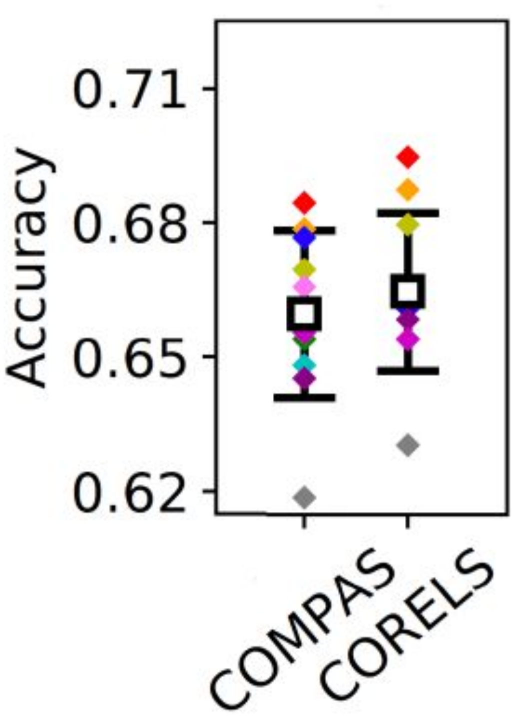
- CORELS (Certifiably Optimal RulE ListS) Angelino et al., KDD 2017 & JMLR 2018
- Model (Rule List) for prediction of recidivism within 2 years
- Free, transparent



IF	age between 18-20 and sex is male	THEN predict arrest (within 2 years)
ELSE IF	age between 21-23 and 2-3 prior offenses	THEN predict arrest
ELSE IF	more than three priors	THEN predict arrest
ELSE	predict no arrest.	

COMPAS vs CORRELS

- Simple CORRELS rule list is more accurate than COMPASS for prediction of recidivism in 2 years
- There’s no benefit from complicated models for re-arrest prediction in
- criminal justice.
- Perhaps we are using complicated models when we don't need them?



Outline

- Motivation
- Interpretable Machine Learning
- Explainable Machine Learning (XAI)
- Interactions and nonlinearities
- Reliability
- Contradiction

Interpretable Models

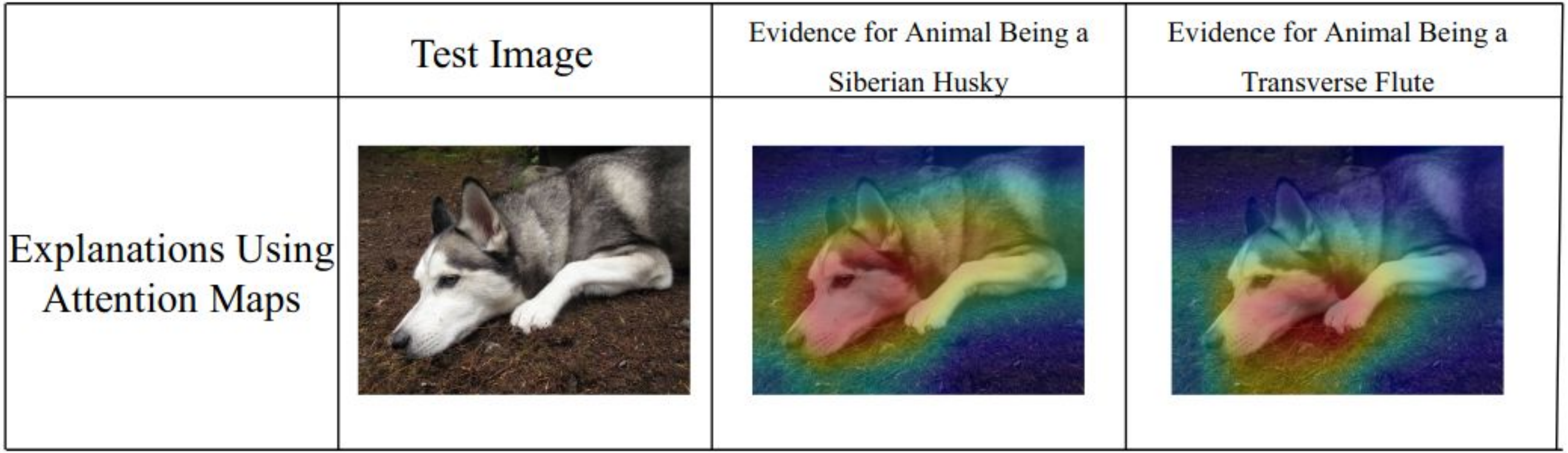
- In a full data science process, one interprets the results and tunes the processing of the data, the loss function, the evaluation metric, or anything else that is relevant. How can one do this without understanding how the model works?
- Avoid catastrophic consequences
- Black-box models often predicts the right answer for the wrong reason
- In cases where the underlying distribution of data changes (domain shift), problems arise if users cannot troubleshoot the model in real-time

Interpretable vs Explainable Models

- Interpretable Models:
  - Models that are inherently easy to understand and grasp by humans.
  - Simpler models like linear regression, decision trees
- Explainable Models:
  - Tools to explain decision of black box models
  - LIME, SHAP, Feature Importances

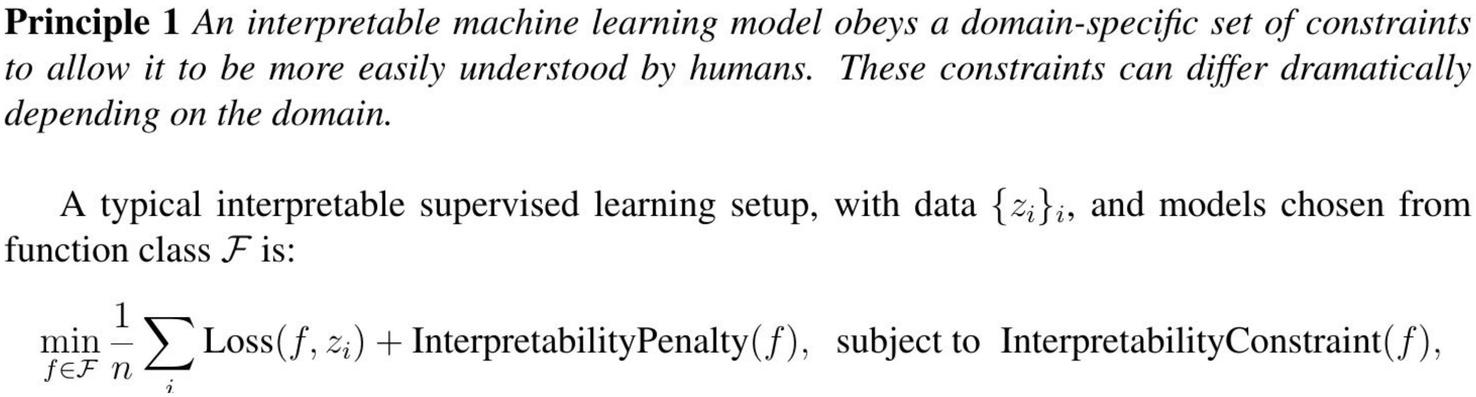
Interpretation vs Explanation

- Could make the situation worse by providing misleading or false characterizations or adding unnecessary authority to the model



	Test Image	Evidence for Animal Being a Siberian Husky	Evidence for Animal Being a Transverse Flute
Explanations Using Attention Maps			

General Principles



Principle 1 *An interpretable machine learning model obeys a domain-specific set of constraints to allow it to be more easily understood by humans. These constraints can differ dramatically depending on the domain.*

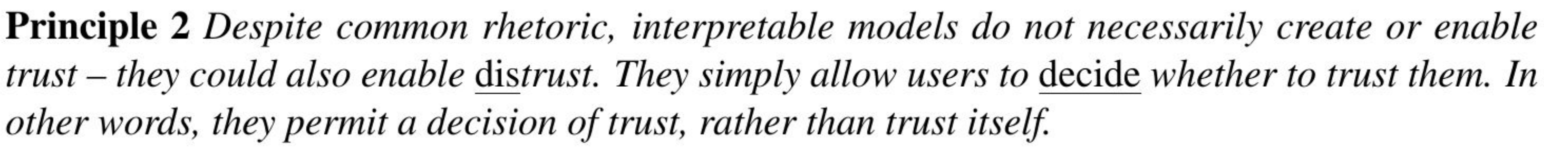
A typical interpretable supervised learning setup, with data $\{z_i\}_i$, and models chosen from function class $\mathcal{F}$ is:

$$\min_{f\in\mathcal{F}} \frac{1}{n}\sum_i \text{Loss}(f, z_i) + \text{InterpretabilityPenalty}(f), \quad \text{subject to} \quad \text{InterpretabilityConstraint}(f),$$

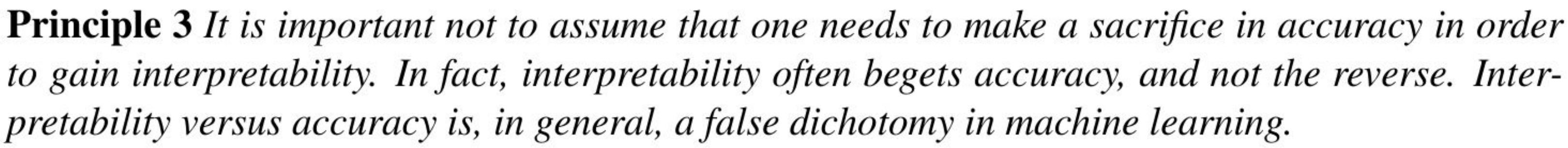
Interpretability constraints

- Sparsity of the model
- Monotonicity with respect to the variable
- Decomposability into sub-models
- Ability to perform case based-reasoning
- Disentanglement of certain types of information within the model reasoning process
- Generative constraints (laws of physics)
- Preferences among choice of variables

General Principles



Principle 2 *Despite common rhetoric, interpretable models do not necessarily create or enable trust – they could also enable* dis*trust. They simply allow users to* decide *whether to trust them. In other words, they permit a decision of trust, rather than trust itself.*



Principle 3 *It is important not to assume that one needs to make a sacrifice in accuracy in order to gain interpretability. In fact, interpretability often begets accuracy, and not the reverse. Interpretability versus accuracy is, in general, a false dichotomy in machine learning.*

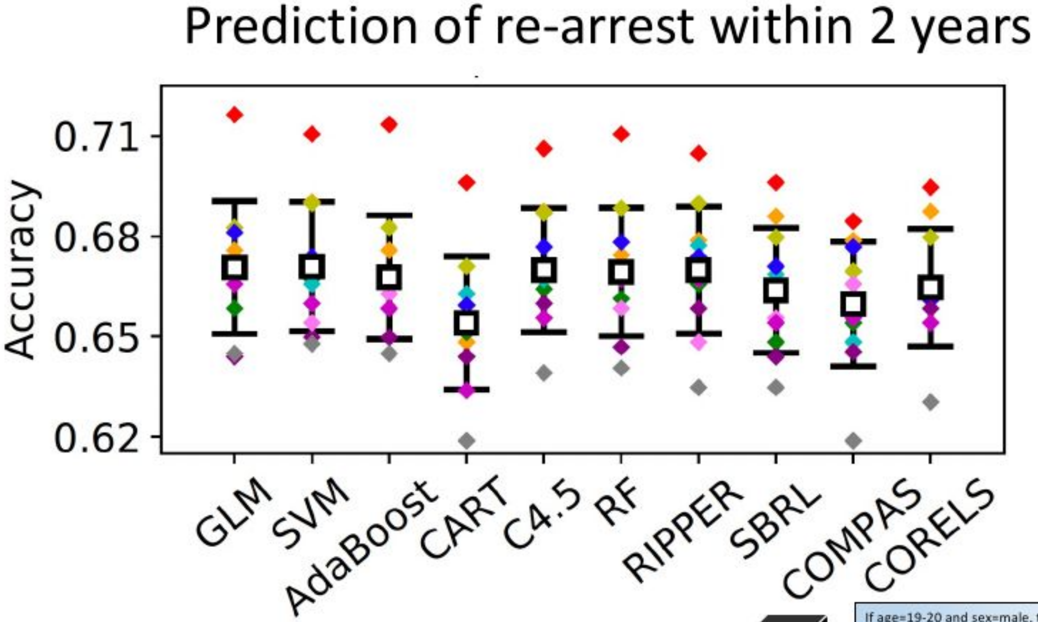
Roshomon set of good models

- Set of almost equally accurate models

$$R(\mathcal{F}, f^*, \epsilon) = \{f \in \mathcal{F} \text{ such that } Loss(f) \leq Loss(f^*) + \epsilon\},$$

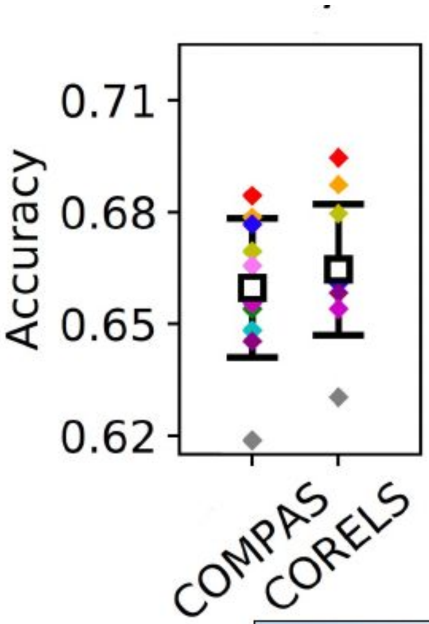
- Rashomon effect occurs there are multiple descriptions of the same event with possible no ground truth
- Seen in credit score estimation, medical imagining, health record analysis, recidivism prediction
- It has been argued that when Rashomon set is large, it must contain a simple model within

Roshomon set



If age=19-20 and sex=male, then predict arrest
else if age=21-22 and priors=2-3 then predict arrest
else if priors >3 then predict arrest
else predict no arrest

Roshomon set

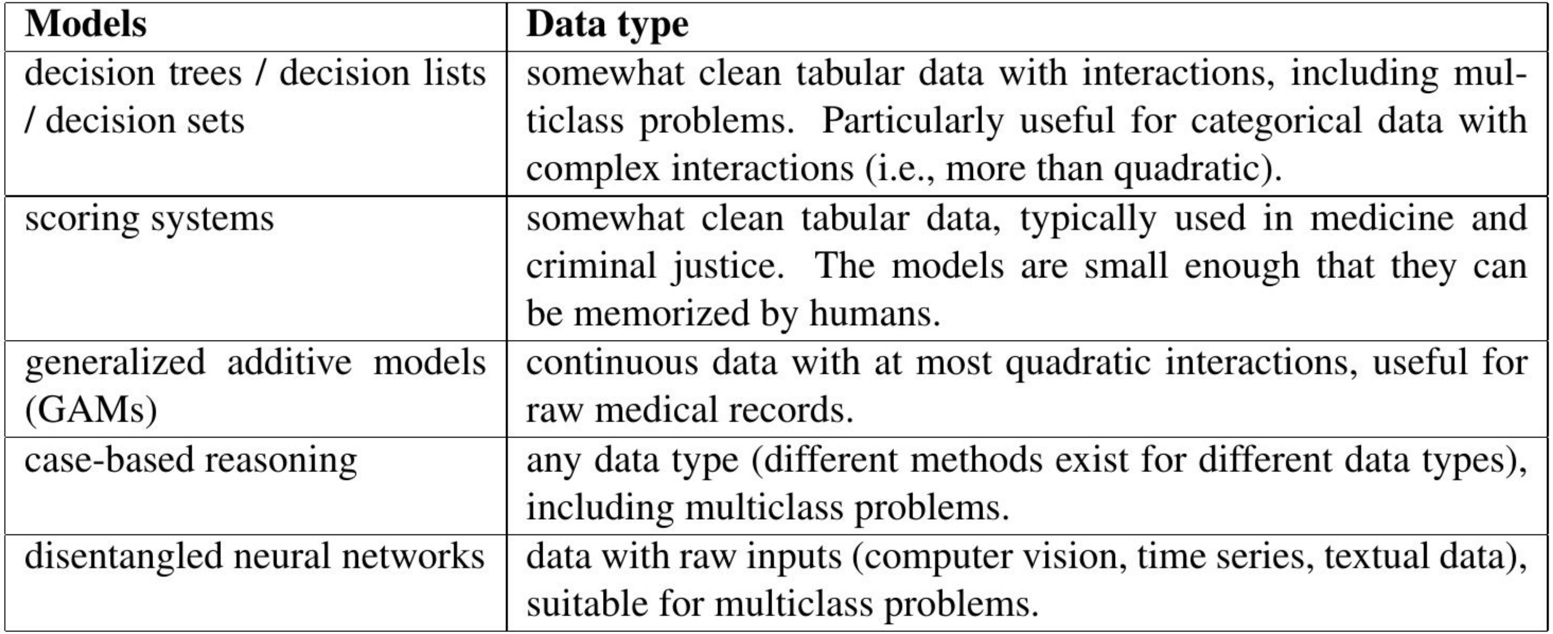


If age=19-20 and sex=male, then predict arrest
else if age=21-22 and priors=2-3 then predict arrest
else if priors >3 then predict arrest
else predict no arrest

Difficulties in creation of the model

- Solving the optimization problem may hard (i.e. finding the right decision tree)
- When one does create an interpretable model, on invariably realizes that the data are problematic and require troubleshooting, which slows down development
- It might not be initially clear which definition of interpretability use

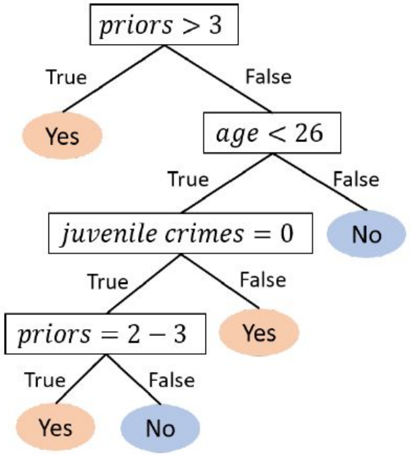
Algorithms for data types



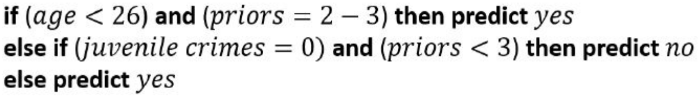
Models	Data type
decision trees / decision lists / decision sets	somewhat clean tabular data with interactions, including multiclass problems. Particularly useful for categorical data with complex interactions (i.e., more than quadratic).
scoring systems	somewhat clean tabular data, typically used in medicine and criminal justice. The models are small enough that they can be memorized by humans.
generalized additive models (GAMs)	continuous data with at most quadratic interactions, useful for raw medical records.
case-based reasoning	any data type (different methods exist for different data types), including multiclass problems.
disentangled neural networks	data with raw inputs (computer vision, time series, textual data), suitable for multiclass problems.

Logical Models

- Decision tree
- Decision list
- Decision set

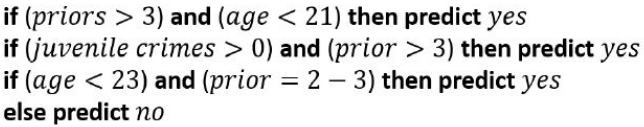


(a)



if $(age < 26)$ **and** $(priors = 2 - 3)$ **then predict** yes
else if $(juvenile\ crimes = 0)$ **and** $(priors < 3)$ **then predict** no
else predict yes

(b)

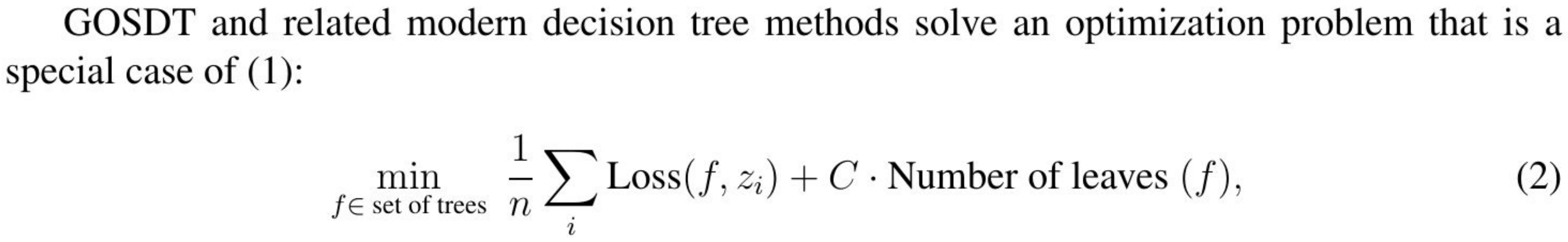


if $(priors > 3)$ **and** $(age < 21)$ **then predict** yes
if $(juvenile\ crimes > 0)$ **and** $(prior > 3)$ **then predict** yes
if $(age < 23)$ **and** $(prior = 2 - 3)$ **then predict** yes
else predict no

(c)

Decision Tree

- Current SOTA optimal decision tree methods can handle medium-sized datasets (thousands of samples, tens of binary variables) within 10 minutes when appropriate sparsity constraints are used
- Scale exponentially with dimension of data
- Handle **categorical** variables and complicated interactions better than e.g. linear models
- When fully optimized, single trees can be as accurate as ensembles of trees or NN

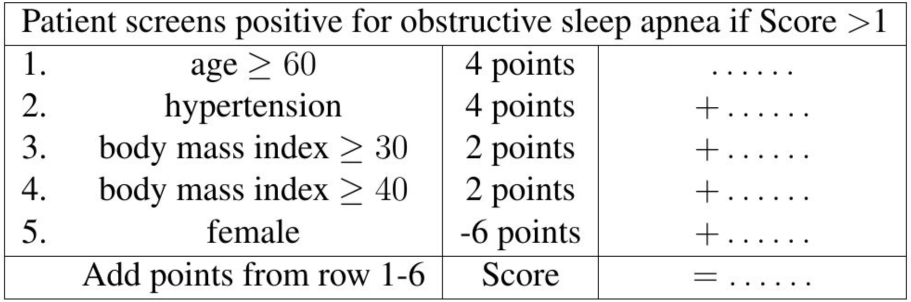


GOSDT and related modern decision tree methods solve an optimization problem that is a special case of (1):

$$\min_{f \in \text{ set of trees}} \frac{1}{n} \sum_i \text{Loss}(f, z_i) + C \cdot \text{Number of leaves } (f), \tag{2}$$

Scoring Systems

- Linear classification models models that require users to add, subtract and multiply only a few small numbers
- Do not handle interactions
- Good for counterfactual reasoning

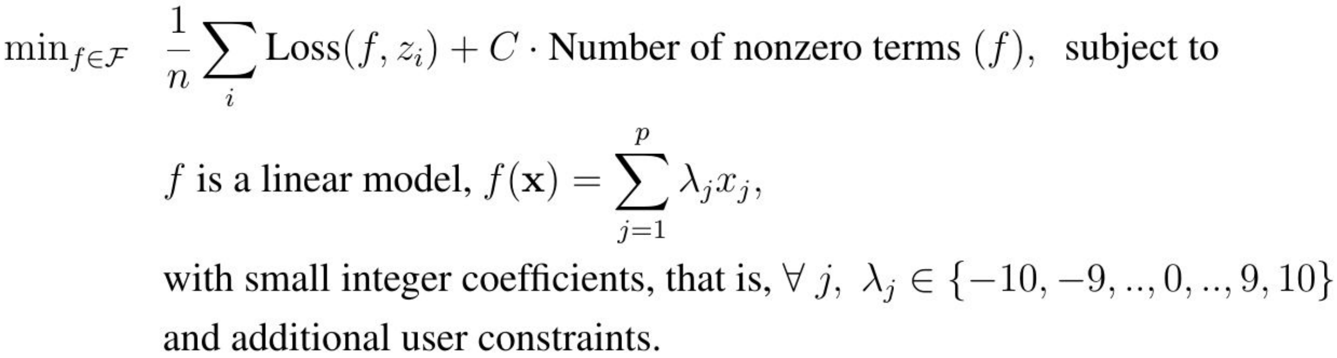


Patient screens positive for obstructive sleep apnea if Score >1			
1.	age ≥ 60	4 points	. . . . . .
2.	hypertension	4 points	+ . . . . . .
3.	body mass index ≥ 30	2 points	+ . . . . . .
4.	body mass index ≥ 40	2 points	+ . . . . . .
5.	female	-6 points	+ . . . . . .
	Add points from row 1-6	Score	= . . . . . .

Table 2: A scoring system for sleep apnea screening (Ustun et al., 2016). Patients that screen positive may need to come to the clinic to be tested.

Scoring Systems

- Optimization problem



$$\min_{f\in\mathcal{F}} \quad \frac{1}{n}\sum_i \text{Loss}(f, z_i) + C \cdot \text{Number of nonzero terms } (f), \quad \text{subject to}$$

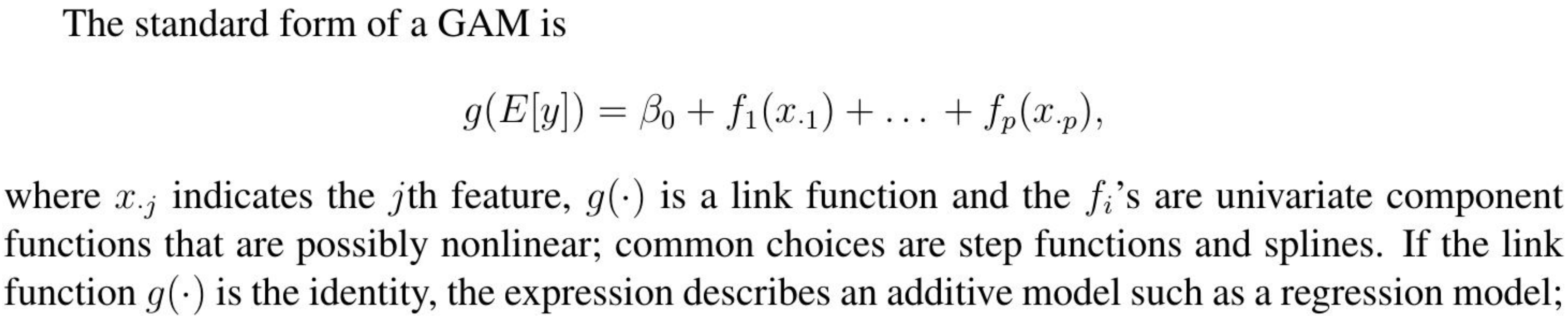
$$f \text{ is a linear model, } f(\mathbf{x}) = \sum_{j=1}^{p} \lambda_j x_j,$$

with small integer coefficients, that is, $\forall\ j,\ \ \lambda_j \in \{-10, -9, .., 0, .., 9, 10\}$

and additional user constraints.

- Practical implementation: round real coefficients -> loss of information
- Frameworks to allow Computer-aided exploration, human in the loop
- Risk scores
    - Scoring systems that have a conversion table to probabilities (1 point -> 15%, 2->33%)

Generalized Additive Models (GAMs)



The standard form of a GAM is

$$g(E[y]) = \beta_0 + f_1(x_{\cdot 1}) + \ldots + f_p(x_{\cdot p}),$$

where $x_{\cdot j}$ indicates the jth feature, $g(\cdot)$ is a link function and the f_i's are univariate component functions that are possibly nonlinear; common choices are step functions and splines. If the link function $g(\cdot)$ is the identity, the expression describes an additive model such as a regression model;

- Link function g
  - Identity -> regression
  - Logistic -> classification
- Component functions f
  - Step functions
  - Splines

Generalized Additive Models (GAMs)

- we can impose the prior belief that predictive relationships are inherently smooth in nature, even though the dataset at hand may suggest a more noisy relationship
- If the researcher could control the sparsity, smoothness, and monotonicity of the component functions, she might be able to design a model that not only predicts well but also reveals interesting relationships between observed variables and outcomes
- Could be used to troubleshoot complex datasets (raw medical data), find counterintuitive patterns
- GA2Ms

$$g(E[y]) = \beta_0 + \sum_{j} f_j(x_j) + \sum_{i \neq j} f_{ij}(x_i, x_j)$$

Case-Based Reasoning

- Solving a new problem using known solutions to similar past problems.
- Emulation of how humans reason
- Two types
  - Nearest neighbour-based techniques
  - Prototype-based techniques

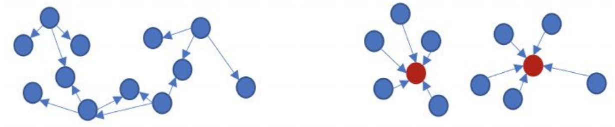


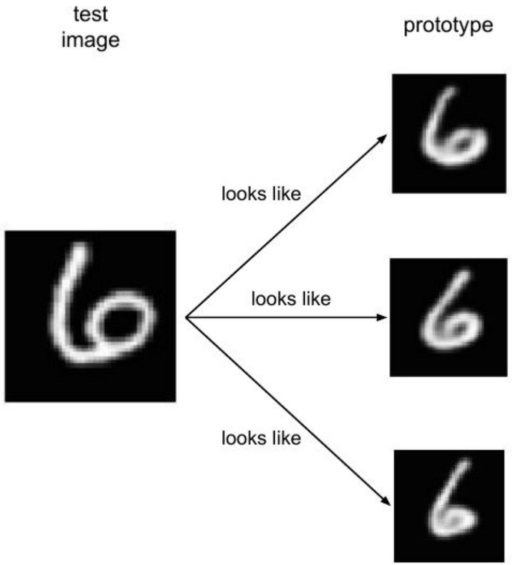
Figure 6: Case-based reasoning types. *Left:* Nearest neighbors (just some arrows are shown for 3-nearest neighbors). *Right:* Prototype-based reasoning, shown with two prototypes.

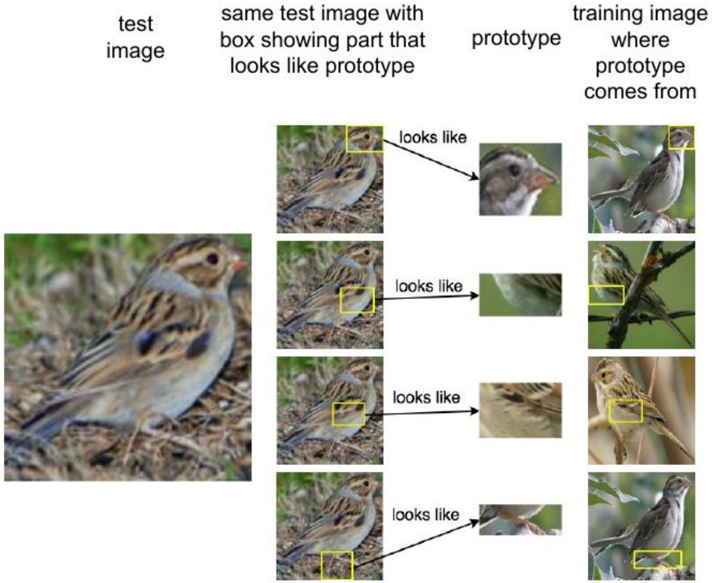
Prototype-Based Techniques

- Learn, from the training data, a set of prototypical cases for comparison
- Given a previously unseen test instance, they make a decision by finding prototypical cases that most closely resemble the particular test instance
- Part based prototypes compare parts of observations to parts of other observations
- Current methods do not take into account prior knowledge or expert opinions
- Sometimes the prototypes may not

Prototype-Based Techniques

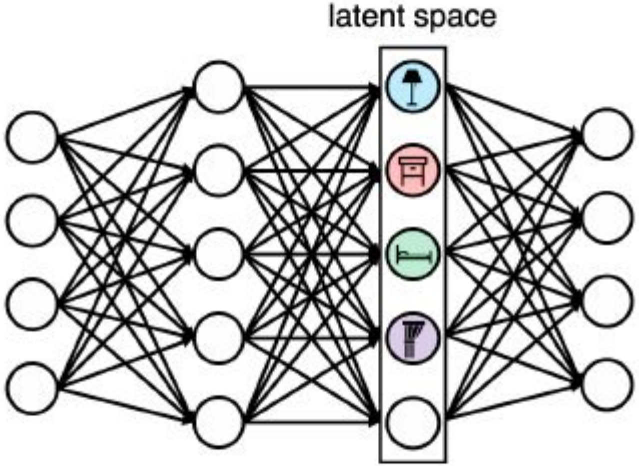
Whole vs part-based prototypes

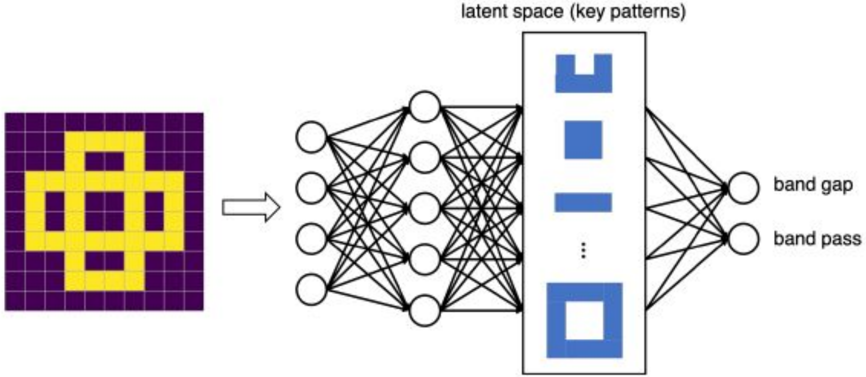



test image
same test image with box showing part that looks like prototype
prototype
training image where prototype comes from
looks like
looks like
looks like
looks like

Disentanglement of neural networks

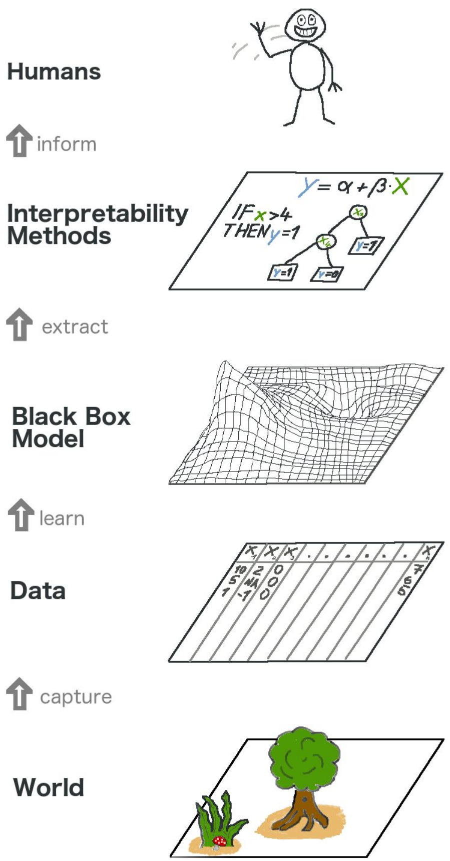
- Refers to the way information travels through the network: all information about a specific concept traverse through one part of the network
- Contains information about bed and room -> classify image as bedroom
- Supervised vs unsupervised ()





Explainable AI (XAI) Techniques

- Global XAI Techniques:
  - Methods that aim to explain the overall behavior of a model across all data points.
  - Provide insights into the general decision-making process of the model.
  - Methods
    - Feature Importance
    - Partial Dependence Plots (PDP)
- Local XAI Techniques:
  - Methods that focus on explaining specific individual predictions made by the model.
  - Offer insights into the model's decision-making process for a particular instance.
  - Methods
    - LIME (Local Interpretable Model-agnostic Explanations)
    - SHAP (SHapley Additive exPlanations)
    - Counterfactual Explanations:



Humans

inform

Interpretability Methods

extract

Black Box Model

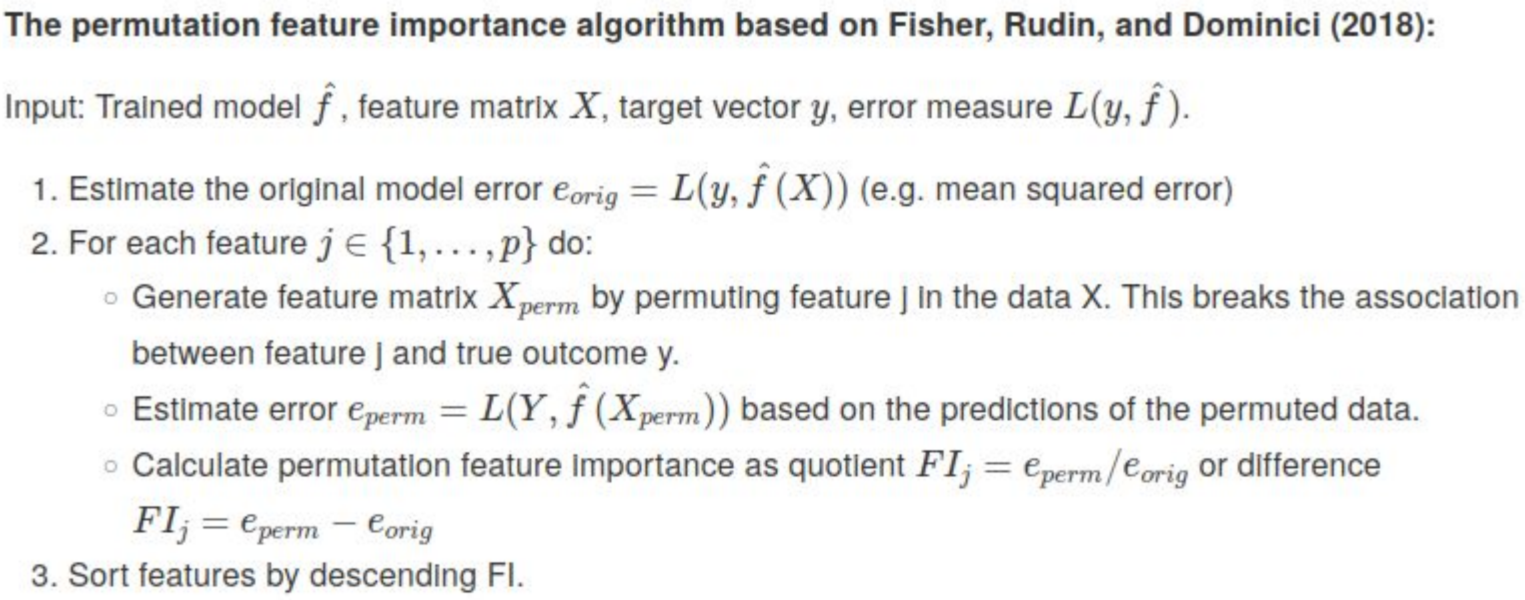
learn

Data

capture

World

Permutation Feature Importance



The permutation feature importance algorithm based on Fisher, Rudin, and Dominici (2018):

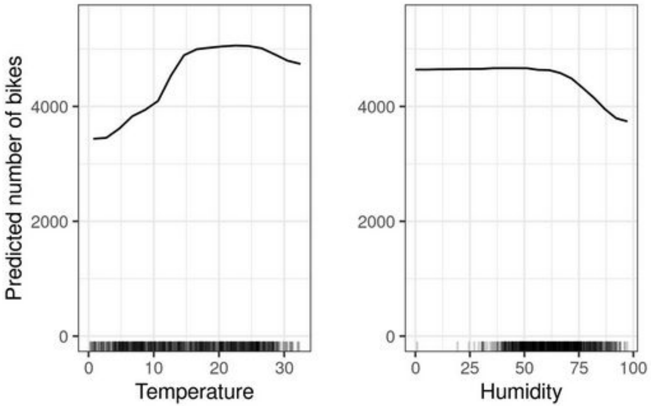
Input: Trained model $\hat{f}$, feature matrix X, target vector y, error measure $L(y, \hat{f})$.

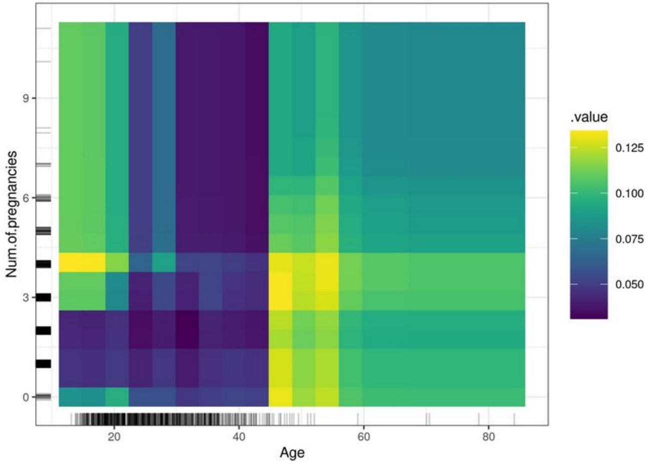
1. Estimate the original model error $e_{orig} = L(y, \hat{f}(X))$ (e.g. mean squared error)
2. For each feature $j \in \{1, \ldots, p\}$ do:
   - Generate feature matrix X_{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
   - Estimate error $e_{perm} = L(Y, \hat{f}(X_{perm}))$ based on the predictions of the permuted data.
   - Calculate permutation feature importance as quotient $FI_j = e_{perm}/e_{orig}$ or difference $FI_j = e_{perm} - e_{orig}$
3. Sort features by descending FI.

Partial Dependence Plot

- A visualization technique that shows the marginal effect of one or two features on the predicted outcome of a machine learning model.
- Reveals the relationship between the target and a feature: linear, monotonic, or complex.
- Plot of partial dependence function, for regression
- $f_S(x_S) = E_{X_c}[f(x_S, X_C)] = \int f(x_S, X_C) dP(X_C)$
- isolate the effect of the feature(s) of interest by averaging the model output over the distribution of other features.

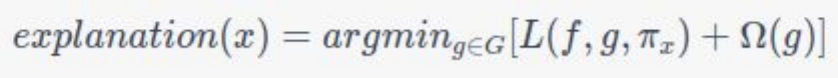
Partial Dependence Plot





Local interpretable model-agnostic explanations (LIME)

- Explain individual predictions of black box machine learning models using interpretable local surrogate models.



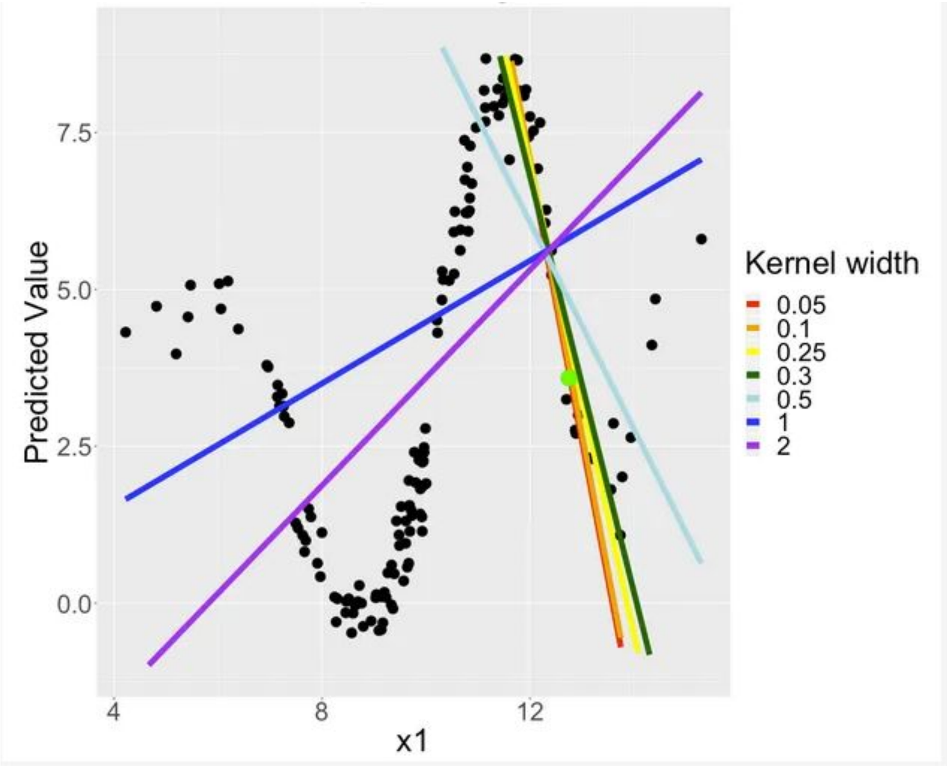
$$explanation(x) = argmin_{g \in G}[L(f, g, \pi_x) + \Omega(g)]$$

L…loss, *G*… family of possible explanations, *π* … proximity measure for neigh. definition

- LIME Process:
  - ○ Select an instance of interest.
  - ○ Perturb the dataset and obtain the black box predictions for the new points.
  - ○ Weight the new samples based on their proximity to the instance of interest.
  - ○ Train a weighted, interpretable model on the perturbed dataset.
  - ○ Explain the prediction by interpreting the local model.

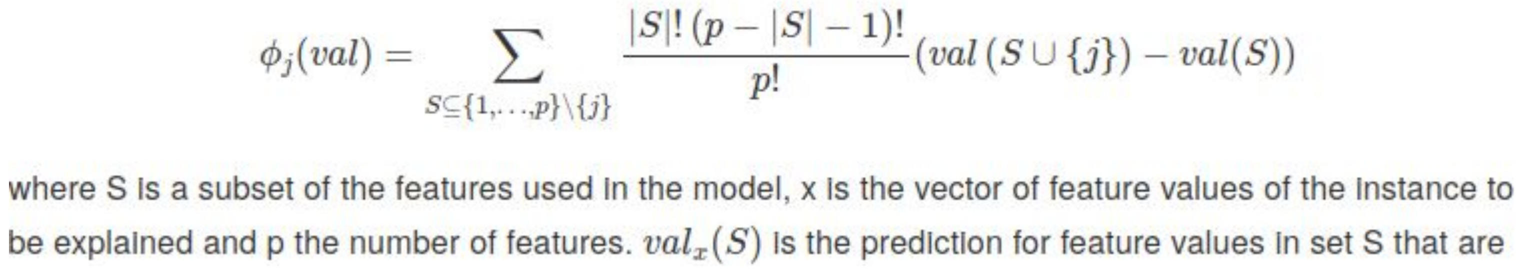
LIME

- Depends strongly of the proximity measure (kernel)



Shapley Values for Explaining Predictions

- Fairly distribute the contribution of each feature to a model's prediction using Shapley values from coalitional game theory.
- **Coalition**:combination of feature values working together to produce a specific prediction
- Algorithm (example on appartement price)
  - Determine all possible coalitions of feature values.
  - Compute the predicted apartment price with and without the feature value of interest for each coalition.
  - Calculate the marginal contribution as the difference between the predicted apartment prices.
  - Compute the (weighted) average of marginal contributions to obtain the Shapley value.

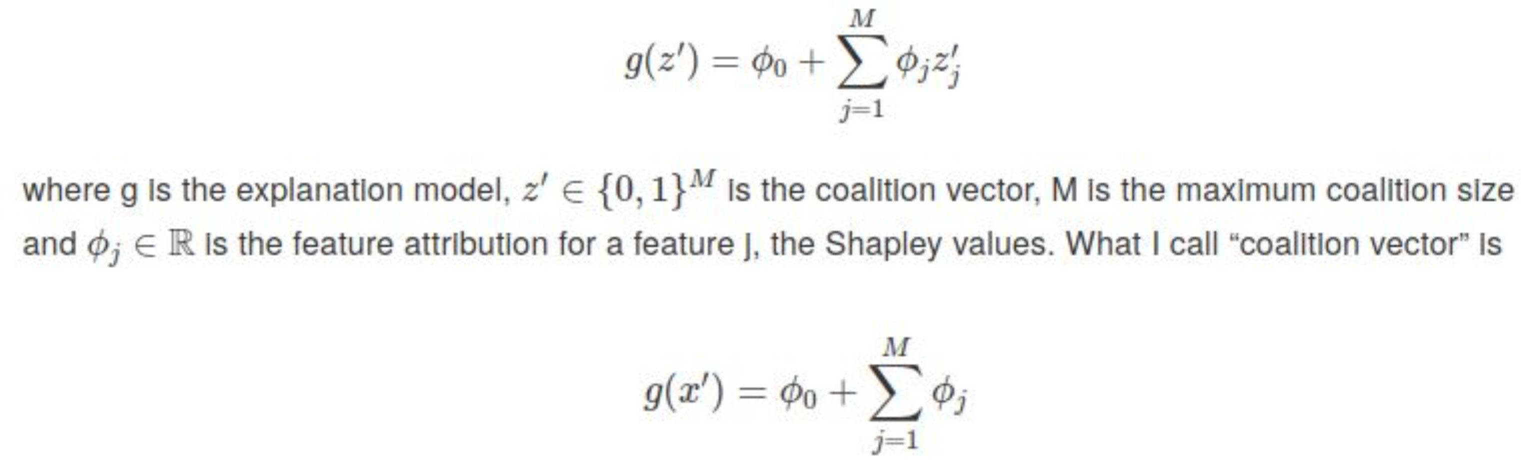


$$\phi_j(val)=\sum_{S\subseteq\{1,\ldots,p\}\setminus\{j\}}\frac{|S|!\,(p-|S|-1)!}{p!}(val\,(S\cup\{j\})-val(S))$$

where S is a subset of the features used in the model, x is the vector of feature values of the instance to be explained and p the number of features. $val_x(S)$ is the prediction for feature values in set S that are

SHAP (SHapley Additive exPlanations)

- Additive feature attribution method represents the Shapley value explanation as a linear model (of coalitions).
    - Connecting Shapley value and surrogates (LIME)



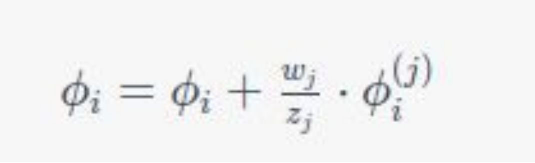
$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

where g is the explanation model, $z' \in \{0,1\}^M$ is the coalition vector, M is the maximum coalition size and $\phi_j \in \mathbb{R}$ is the feature attribution for a feature j, the Shapley values. What I call “coalition vector” is

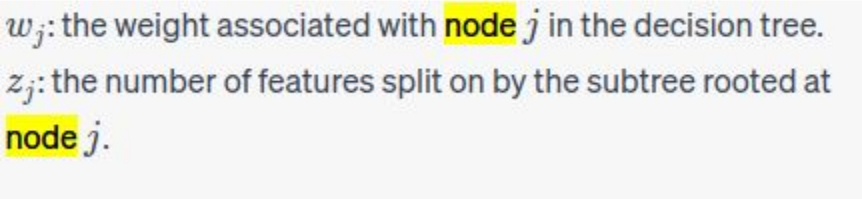
$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j$$

TreeSHAP

- Reduces complexity from O(TL2^M) to O(TLD^2)
- traversing the decision tree recursively. At each node *j*, the algorithm calculates the contribution of the split feature and updates the Shapley values accordingly. The update rule for the Shapley values is:



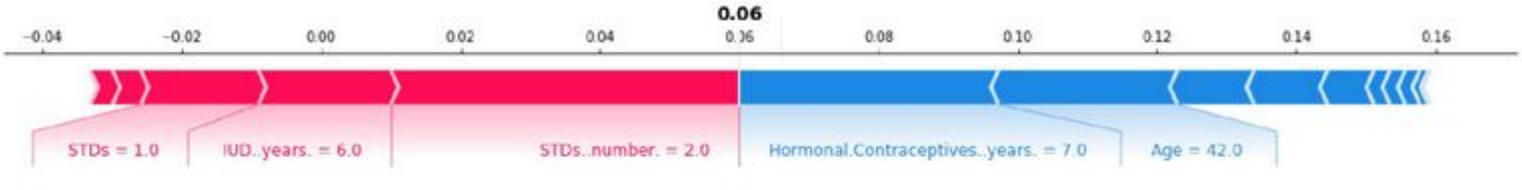
$$\phi_i = \phi_i + \frac{w_j}{z_j} \cdot \phi_i^{(j)}$$

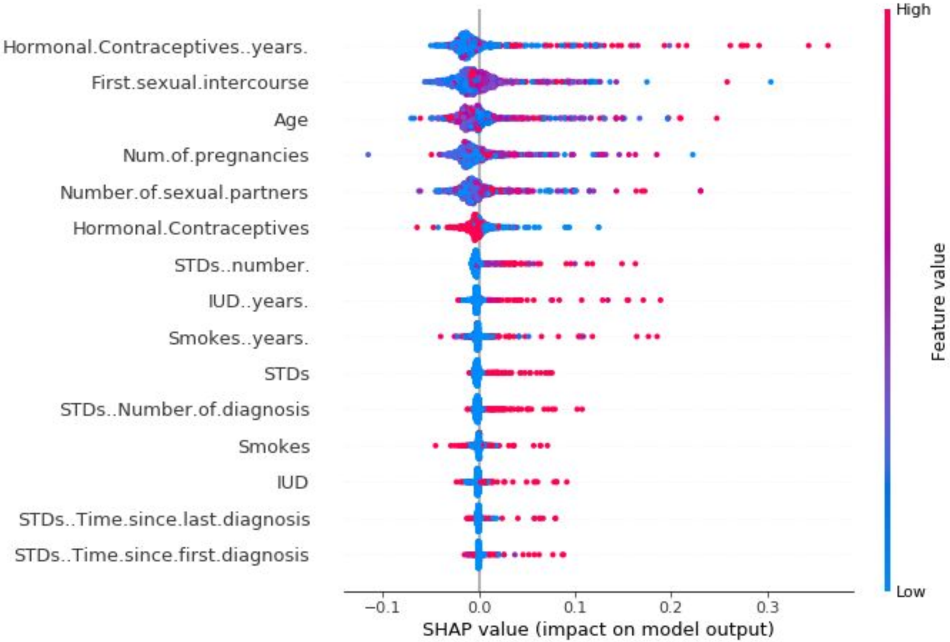


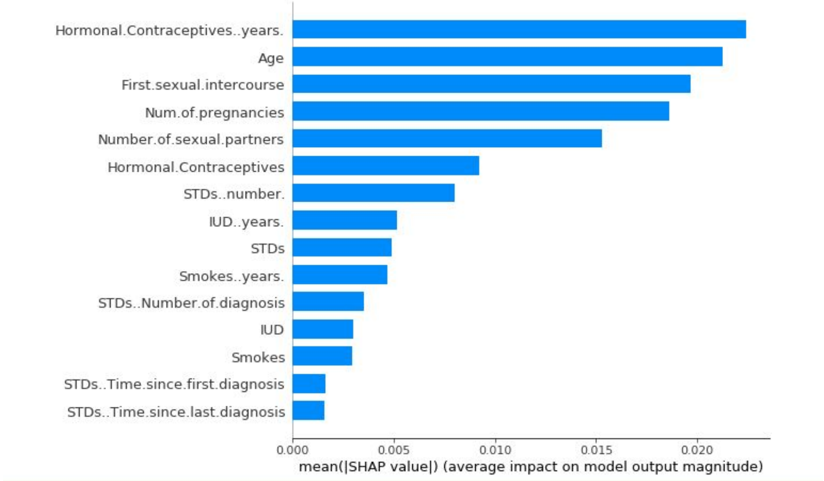
w_j: the weight associated with node j in the decision tree.

z_j: the number of features split on by the subtree rooted at node j.

SHAP Plots

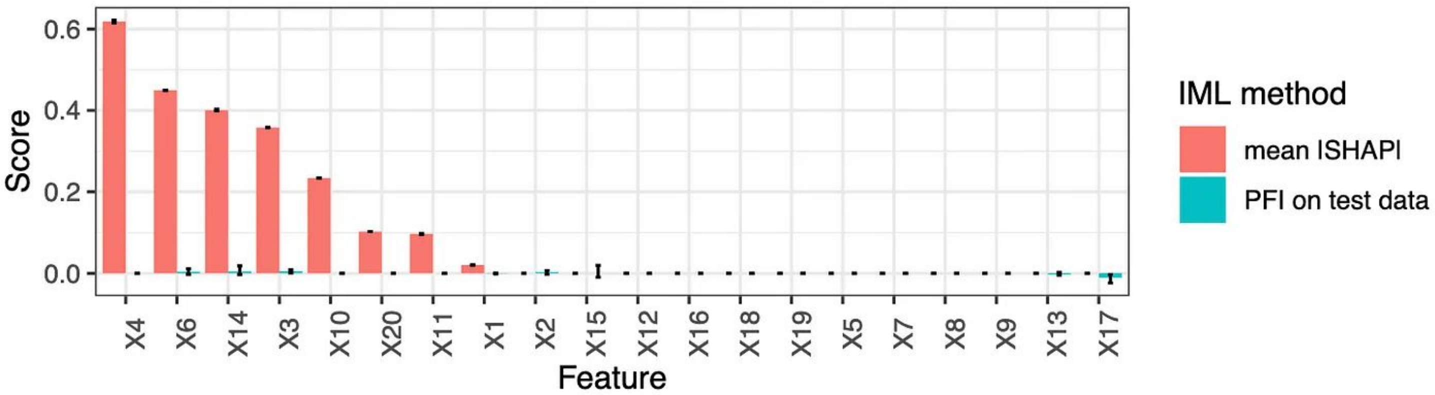

0.06
−0.04
−0.02
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
STDs = 1.0
IUD..years. = 6.0
STDs..number. = 2.0
Hormonal.Contraceptives..years. = 7.0
Age = 42.0


Hormonal.Contraceptives..years.
First.sexual.intercourse
Age
Num.of.pregnancies
Number.of.sexual.partners
Hormonal.Contraceptives
STDs..number.
IUD..years.
Smokes..years.
STDs
STDs..Number.of.diagnosis
Smokes
IUD
STDs..Time.since.last.diagnosis
STDs..Time.since.first.diagnosis
−0.1
0.0
0.1
0.2
0.3
SHAP value (impact on model output)
High
Feature value
Low

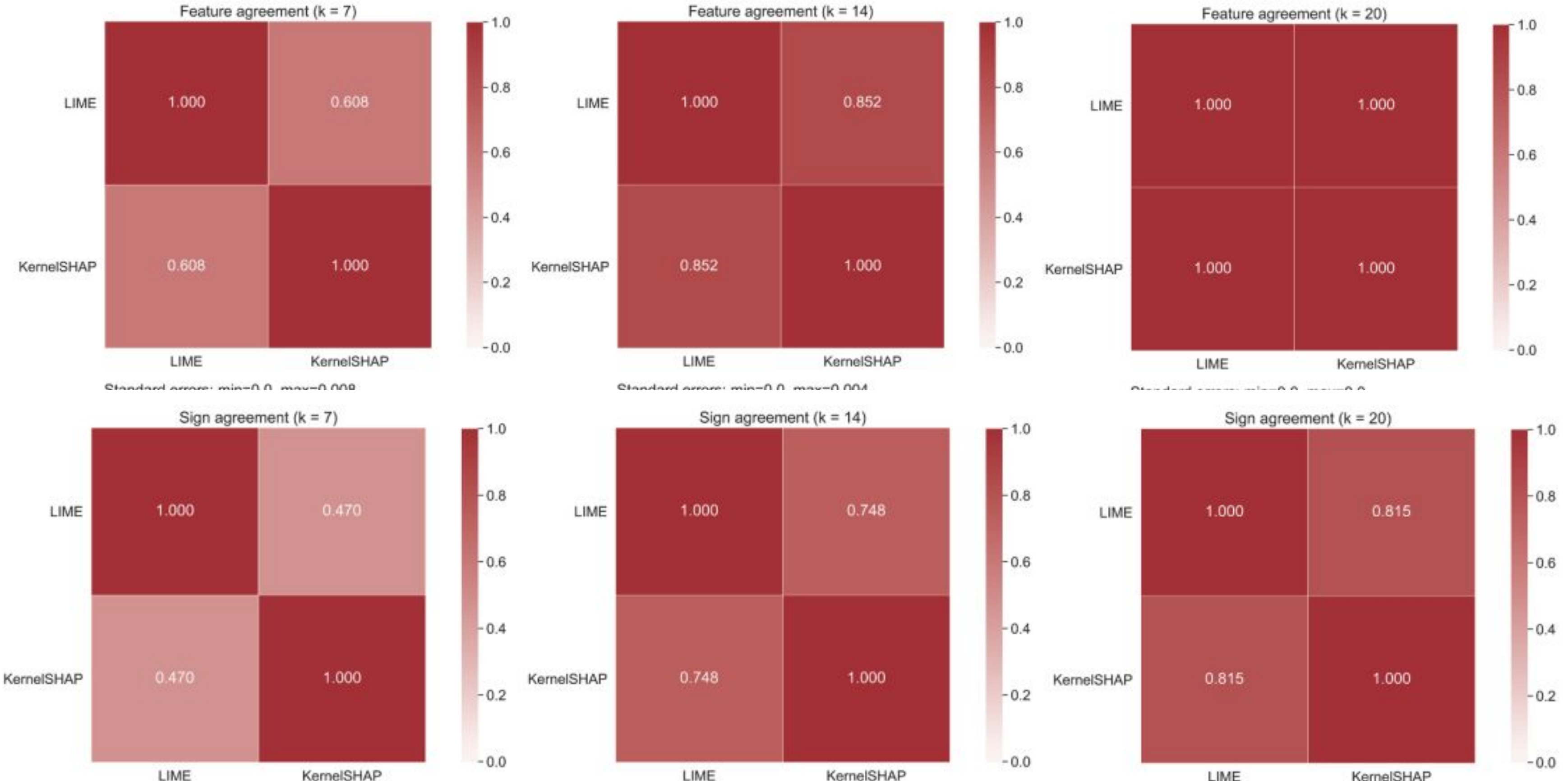

Hormonal.Contraceptives..years.
Age
First.sexual.intercourse
Num.of.pregnancies
Number.of.sexual.partners
Hormonal.Contraceptives
STDs..number.
IUD..years.
STDs
Smokes..years.
STDs..Number.of.diagnosis
IUD
Smokes
STDs..Time.since.first.diagnosis
STDs..Time.since.last.diagnosis
0.000
0.005
0.010
0.015
0.020
mean(|SHAP value|) (average impact on model output magnitude)

SHAP vs PFI on Simulated Data

- All features are random and has no relation to the target
- PFI can detect it, SHAP not



SHAP vs LIME


Feature agreement (k = 7)
LIME
KernelSHAP
1.000
0.608
0.608
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0
Standard errors: min=0.0, max=0.008
Feature agreement (k = 14)
LIME
KernelSHAP
1.000
0.852
0.852
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0
Standard errors: min=0.0, max=0.004
Feature agreement (k = 20)
LIME
KernelSHAP
1.000
1.000
1.000
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0
Standard errors: min=0.0, max=0.0
Sign agreement (k = 7)
LIME
KernelSHAP
1.000
0.470
0.470
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0
Sign agreement (k = 14)
LIME
KernelSHAP
1.000
0.748
0.748
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0
Sign agreement (k = 20)
LIME
KernelSHAP
1.000
0.815
0.815
1.000
LIME
KernelSHAP
1.0
0.8
0.6
0.4
0.2
0.0

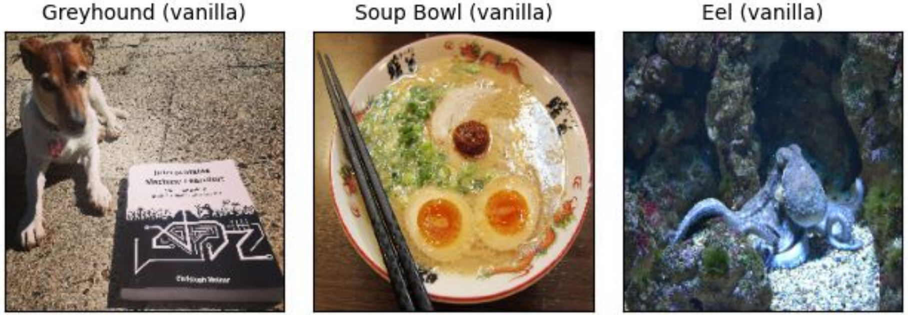
Saliency Maps

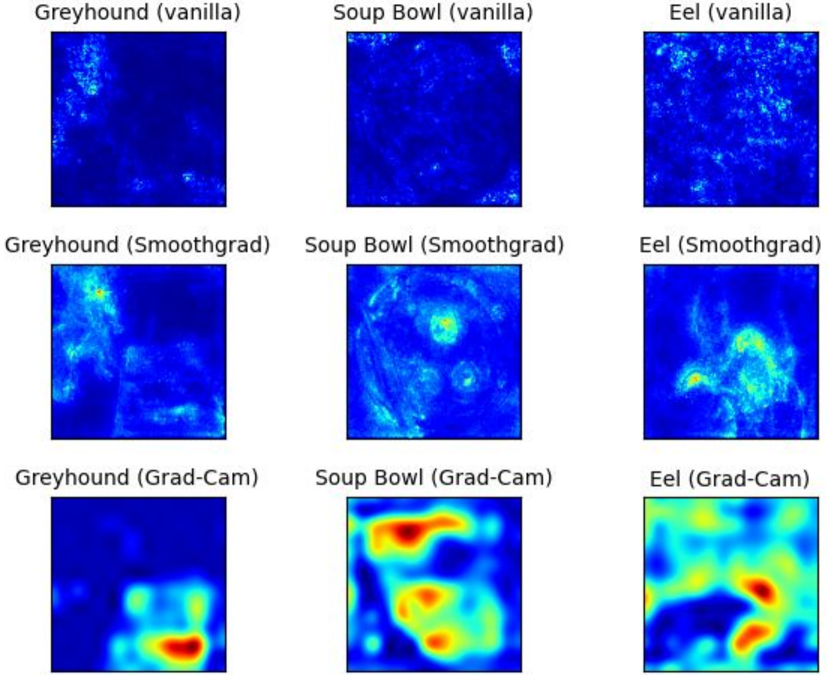
- Saliency maps are visual representations that highlight important regions or features in an input image that contribute to a model's prediction.
- Recipe
  - Perform a forward pass of the image of interest.
  - Compute the gradient of class score of interest with respect to the input pixels:

$$E_{grad}(I_0) = \frac{\delta S_c}{\delta I}|_{I=I_0}$$

  - Visualize the gradients. You can either show the absolute values or highlight negative and positive contributions separately.
-

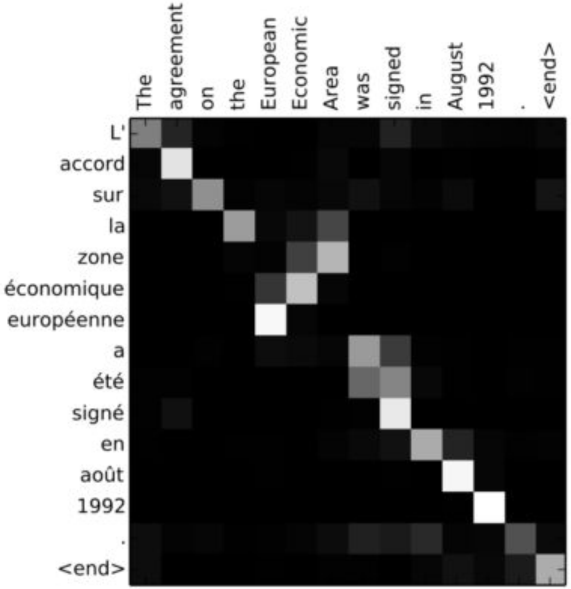
Saliency Maps





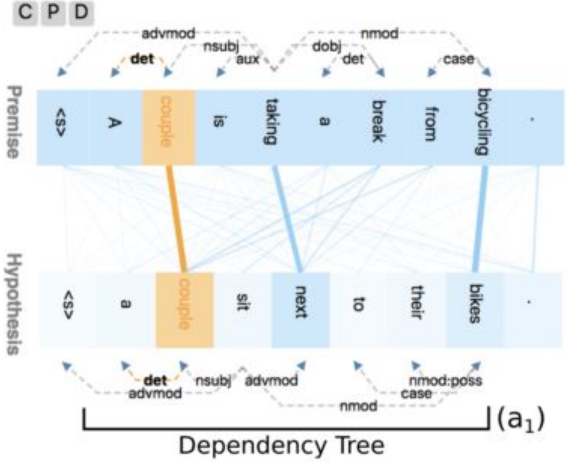
Attention-based models

- attention describes the ability of a model to pay attention to the important parts of a sentence (or image, or any other sequential input). It does this by assigning weights to input features based on their importance and their position in the sequence.



Attention-matrix heatmap

Bahdanau, et al. 2015. Neural machine translation by jointly learning to align and translate. In Proc. ICLR.

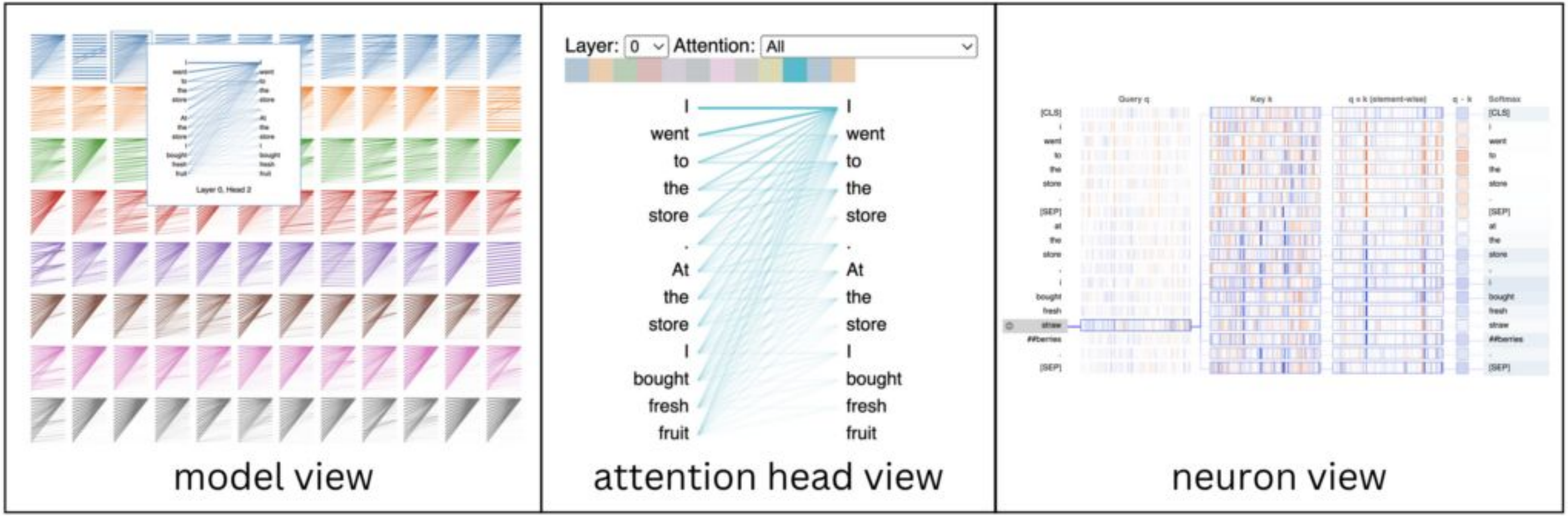


Bi-partite graph representation

Shusen Liu, et al. 2018. Visual interrogation of attention-based models for natural language inference and machine comprehension. In EMNLP: System Demonstrations.

Attention-based models

- BertViz



model view

attention head view

neuron view

Visual Transformers

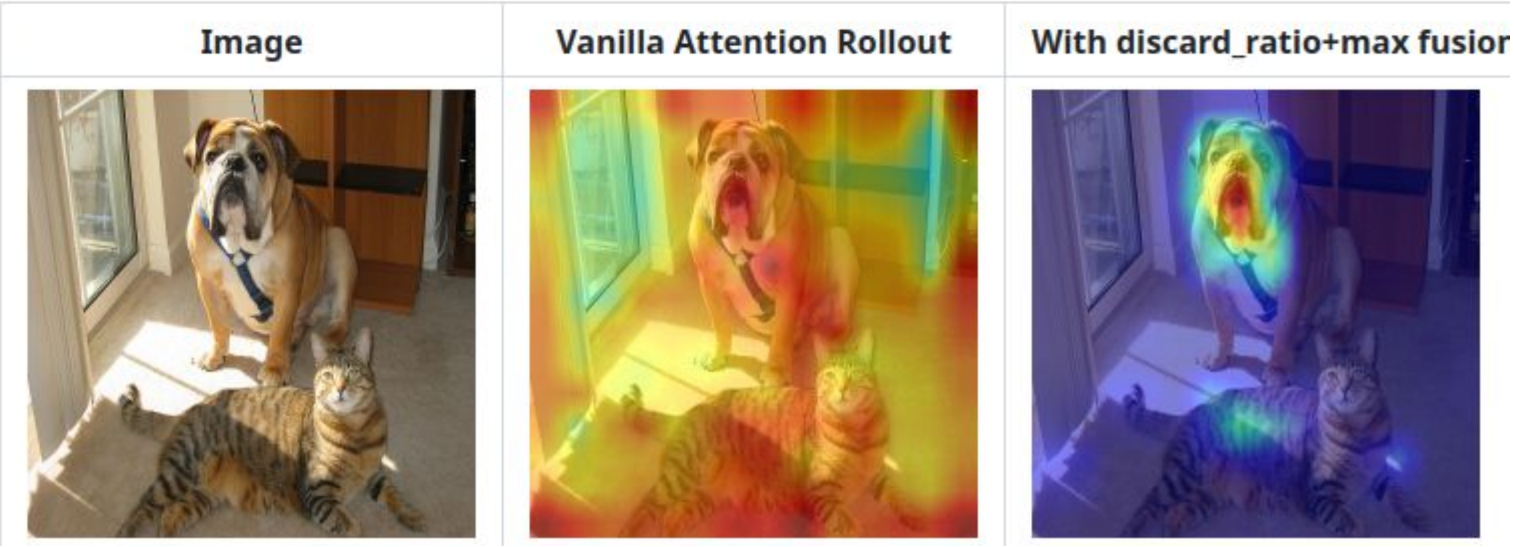


Image	Vanilla Attention Rollout	With discard_ratio+max fusior