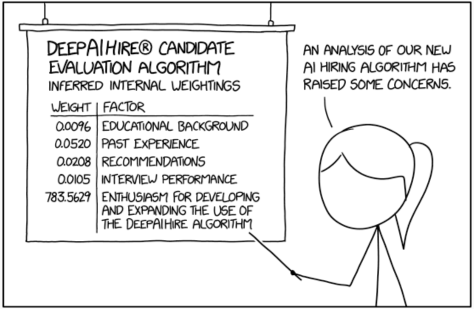
# Interpretability of Supervised Learning Models

BMI 773 Clinical Research Informatics


Yuriy Sverchkov

April 27, 2015

University of Wisconsin–Madison

# Why do we want to interpret models?

# Why do we want to interpret models?

- Trust: having an interpretation along with a prediction can bring a practitioner to agree with a model.

# Why do we want to interpret models?

- Trust: having an interpretation along with a prediction can bring a practitioner to agree with a model.
- Causality: understanding the associations driving model decisions can help uncover underlying mechanisms.

# Why do we want to interpret models?

- Trust: having an interpretation along with a prediction can bring a practitioner to agree with a model.
- Causality: understanding the associations driving model decisions can help uncover underlying mechanisms.
- Transferability: understanding how a model makes decisions informs about how it will perform on a different data distribution

# Why do we want to interpret models?

- Trust: having an interpretation along with a prediction can bring a practitioner to agree with a model.
- Causality: understanding the associations driving model decisions can help uncover underlying mechanisms.
- Transferability: understanding how a model makes decisions informs about how it will perform on a different data distribution
- Informativeness: pointing out evidence to support a decision (decision support systems)

# What makes a model interpretable?

# What makes a model interpretable?

- Simulatability: Can a person can look at the description of the model and figure out what the model's prediction about a given case would be?

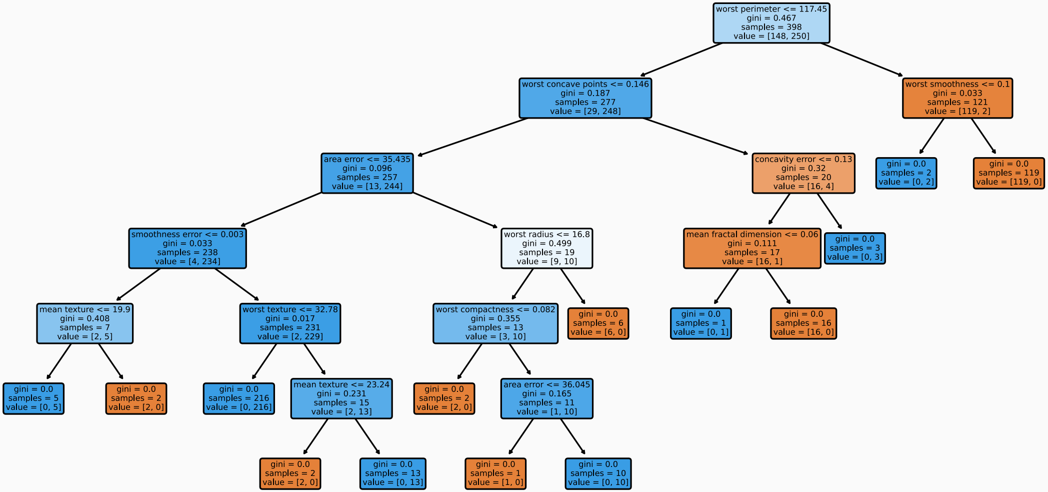
# What makes a model interpretable?

- Simulatability: Can a person can look at the description of the model and figure out what the model's prediction about a given case would be?
- Decomposability: Is the model's decision made up of semantically meaningful components?

# What constitutes an interpretation?

# Models that are interpretable by design

# Interpreting decision trees



# Interpreting linear regression coefficients

**Model:**

$$y = \beta_0 + \sum_{i=1}^{d} x_i \beta_i$$

**Interpretation:**
An increase in the value of feature $i$ by 1 unit corresponds to the increase in the outcome by $\beta_i$ units.

# Interpreting logistic regression coefficients

Model:

$$\overbrace{\log\left(\frac{P(y=1)}{P(y=0)}\right)}^{\text{log odds}} = \beta_0 + \sum_{i=1}^{d} x_i\beta_i$$

Interpretation:
An (additive) increase in the value of feature $i$ by 1 unit corresponds to the increase in the odds of the outcome by a (multiplicative) factor of $\beta_i$.

# Generalized additive models

Model:

$$\overbrace{\underbrace{g(y) = \beta_0 + \sum_i f_i(x_i)}_{\text{Standard GAM}} + \sum_{i \neq j} f_{ij}(x_i, y_j)}^{\text{Caruana, KDD 2015}}$$

# Generalized additive models

Model:

$$\overbrace{\underbrace{g(y) = \beta_0 + \sum_i f_i(x_i)}_{\text{Standard GAM}} + \sum_{i\neq j} f_{ij}(x_i, y_j)}^{\text{Caruana, KDD 2015}}$$

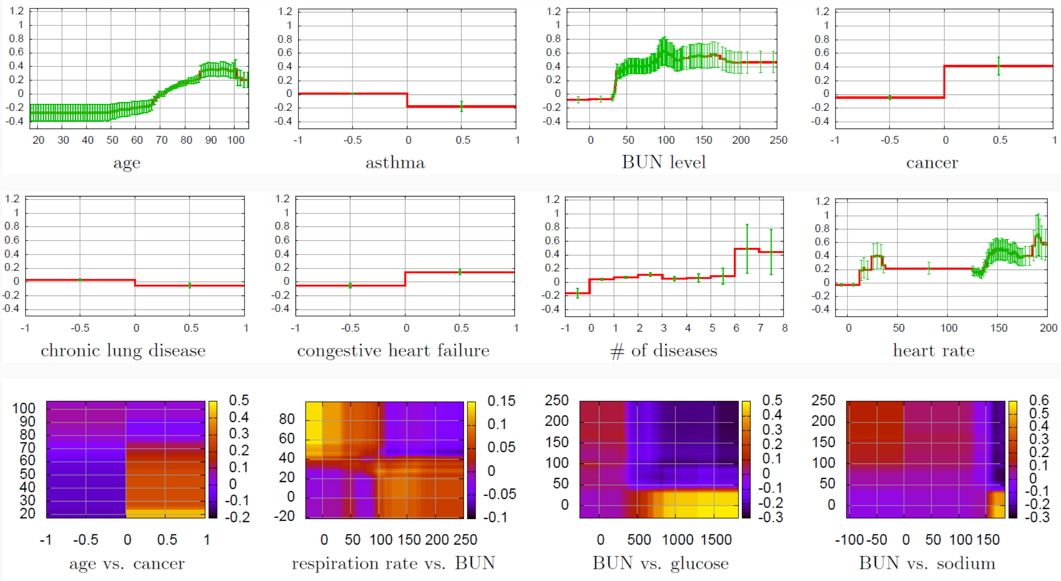
RESEARCH-ARTICLE

**Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-day Readmission**

**Authors:** Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, Noemie Elhadad

Authors Info & Affiliations

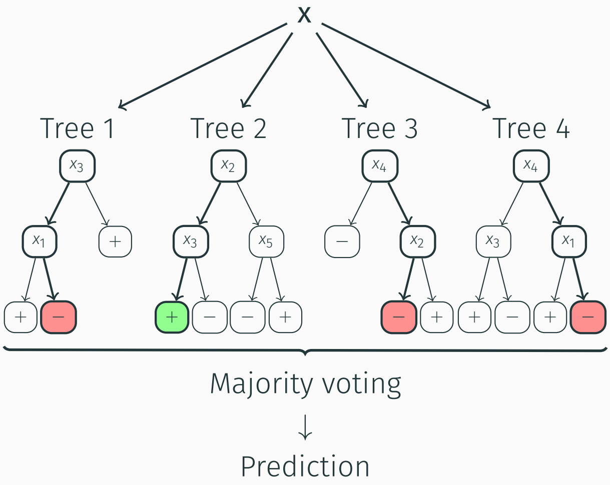
**Publication:** KDD '15: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining • August 2015 • Pages 1721–1730 • https://doi.org/10.1145/2783258.2788613


age
asthma
BUN level
cancer
chronic lung disease
congestive heart failure
# of diseases
heart rate
age vs. cancer
respiration rate vs. BUN
BUN vs. glucose
BUN vs. sodium

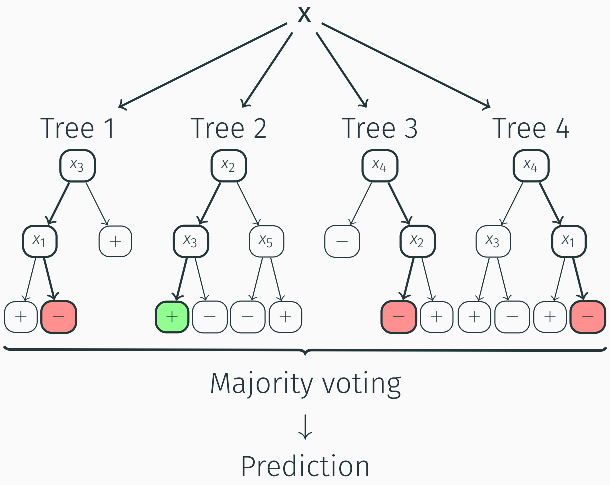
# Post-hoc model-aware interpretation

- Post-hoc — the interpretation is not built into the predictive model
- Model-aware — the interpretation exploits knowledge about the model's internals

# Feature importances in random forests



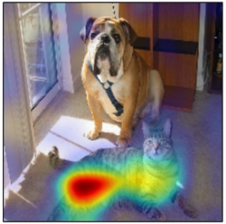
# Feature importances in random forests




- Model-aware approach to feature importances for random forests:
  - Count the number of times a feature is selected for a split, or
  - Average the impurity score (Gini, entropy, variance) gains for each feature $x_i$ over the all trees.
- Model-agnostic approach to feature importances: permutation (more on this later)

# Saliency maps


(a) Original Image


(c) Grad-CAM 'Cat'


(i) Grad-CAM 'Dog'

**Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization**

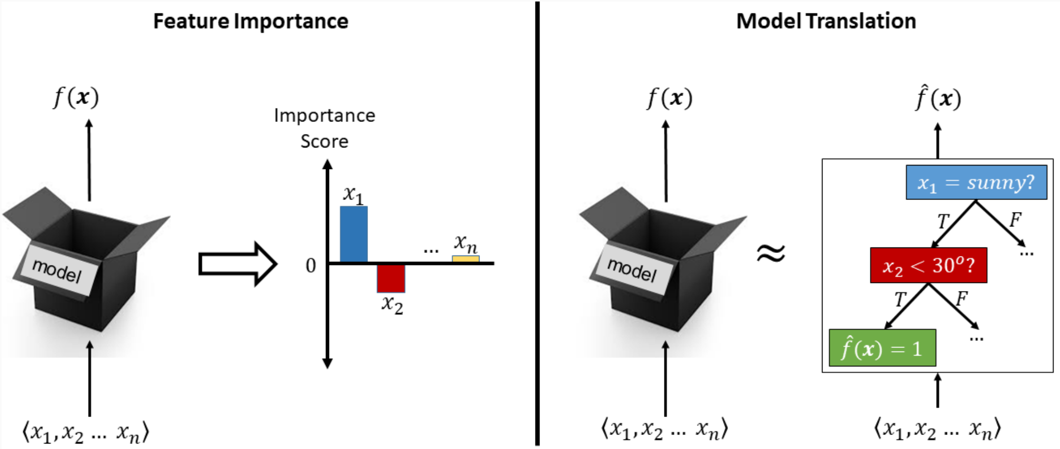
**Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra**

- Highlight regions of the input space that drive a particular prediction
- Commonly used with image data

# Post-hoc model-agnostic interpretation

- **Model-agnostic** — the interpretation assumes no knowledge about the model's internals

- **Model-agnostic** — the interpretation assumes no knowledge about the model's internals



# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$
2. and a training instance $(\mathrm{x}^{(j)}, y^{(j)})$,

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \rightarrow \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \rightarrow \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.
4. Construct a *perturbed* training instance $\mathbf{x}_{\sim i}^{(j)} = \langle x_1^{(j)}, \ldots, x_{i-1}^{(j)}, z_i, x_{i+1}^{(j)}, \ldots, x_d^{(j)} \rangle$ in which we replaced the value of the $i$-th feature

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.
4. Construct a *perturbed* training instance $\mathbf{x}^{(j)}_{\sim i} = \langle x^{(j)}_1, \ldots, x^{(j)}_{i-1}, z_i, x^{(j)}_{i+1}, \ldots, x^{(j)}_d \rangle$ in which we replaced the value of the $i$-th feature
    - The best perturbation is domain-dependent (zeroing out, random/permuted values, population mean).

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.
4. Construct a *perturbed* training instance $\mathbf{x}^{(j)}_{\sim i} = \langle x^{(j)}_1, \ldots, x^{(j)}_{i-1}, z_i, x^{(j)}_{i+1}, \ldots, x^{(j)}_d \rangle$ in which we replaced the value of the $i$-th feature
    - The best perturbation is domain-dependent (zeroing out, random/permuted values, population mean).
5. Compute the loss on the perturbed instance $L(f(\mathbf{x}^{(j)}_{\sim i}), y^{(j)})$

# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.
4. Construct a *perturbed* training instance $\mathbf{x}^{(j)}_{\sim i} = \langle x^{(j)}_1, \ldots, x^{(j)}_{i-1}, z_i, x^{(j)}_{i+1}, \ldots, x^{(j)}_d \rangle$ in which we replaced the value of the $i$-th feature
   - The best perturbation is domain-dependent (zeroing out, random/permuted values, population mean).
5. Compute the loss on the perturbed instance $L(f(\mathbf{x}^{(j)}_{\sim i}), y^{(j)})$
6. Consider average difference in loss across the training set

$$\frac{1}{n} \sum_{j=0}^{n} \left( L(f(\mathbf{x}^{(j)}_{\sim i}), y^{(j)}) - L(f(\mathbf{x}^{(j)}), y^{(j)}) \right)$$

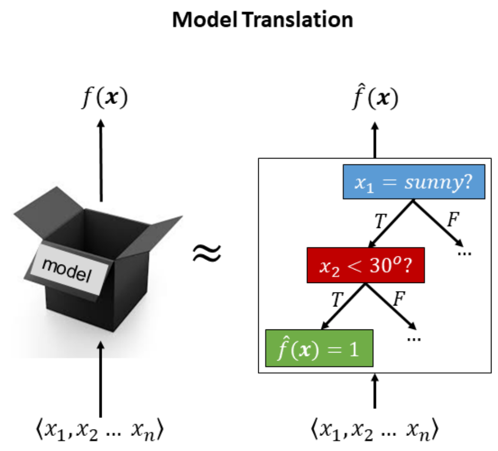
# Eliciting feature importances from black-box models

1. Given a black-box model $f : \mathcal{X} \to \mathcal{Y}$
2. and a training instance $(\mathbf{x}^{(j)}, y^{(j)})$,
3. compute a loss function $L(f(\mathbf{x}^{(j)}), y^{(j)})$.
4. Construct a *perturbed* training instance $\mathbf{x}^{(j)}_{\sim i} = \langle x^{(j)}_1, \ldots, x^{(j)}_{i-1}, z_i, x^{(j)}_{i+1}, \ldots, x^{(j)}_d \rangle$ in which we replaced the value of the $i$-th feature
    - The best perturbation is domain-dependent (zeroing out, random/permuted values, population mean).
5. Compute the loss on the perturbed instance $L(f(\mathbf{x}^{(j)}_{\sim i}), y^{(j)})$
6. Consider average difference in loss across the training set

$$\frac{1}{n} \sum_{j=0}^{n} \left( L(f(\mathbf{x}^{(j)}_{\sim i}), y^{(j)}) - L(f(\mathbf{x}^{(j)}), y^{(j)}) \right)$$

- This gives an importance score for feature $i$.

# Learning high-fidelity mimic models




- Learn an interpretable model that has high fidelity to the black box
- **Fidelity:** how well the interpreting model's outputs match the black box output (given the same inputs)
- Instead of learning only from a training set, learning is also based on the black box's outputs
- The black box is queried throughout the learning process

# Black box $\rightarrow$ decision tree

# Black box $\rightarrow$ decision tree

- Decision trees learned using the black-box model as an oracle
- Better results than learning a decision tree from the training data

- Craven and Shavlik 1995
- Breiman and Shang 1996
- Bastani et al. 2017
- Frosst and Hinton 2017

Appears in *Advances in Neural Information Processing Systems, Vol. 8.*
MIT Press, Cambridge, MA, 1996.

**Extracting Tree-Structured Representations of Trained Networks**

**Mark W. Craven and Jude W. Shavlik**
Computer Sciences Department
University of Wisconsin-Madison

Table 2: Test-set accuracy and fidelity.

| domain | accuracy | | | | fidelity |
|---|---|---|---|---|---|
| | networks | C4.5 | ID2-of-3 | TREPAN | TREPAN |
| heart | 84.5% | 71.0% | 74.6% | 81.8% | 94.1% |
| promoters | 90.6 | 84.4 | 83.5 | 87.6 | 85.7 |
| protein coding | 94.1 | 90.3 | 90.9 | 91.4 | 92.4 |
| voting | 92.2 | 89.2 | 87.8 | 90.8 | 95.9 |

# Black box $\rightarrow$ (locally) linear

# Black box → (locally) linear

**“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

$$\underset{g}{\operatorname{argmin}} \quad \underbrace{\mathcal{L}(f, g, \pi_x)}_{\text{fidelity loss around } x} + \underbrace{\Omega(g)}_{\text{complexity}}$$

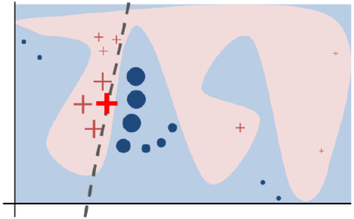


**Figure 3: Toy example to present intuition for LIME. The black-box model’s complex decision function $f$ (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bold red cross is the instance being explained. LIME samples instances, gets predictions using $f$, and weighs them by the proximity to the instance being explained (represented here by size). The dashed line is the learned explanation that is locally (but not globally) faithful.**