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

Ribeiro, Singh, Guestrin (2016)

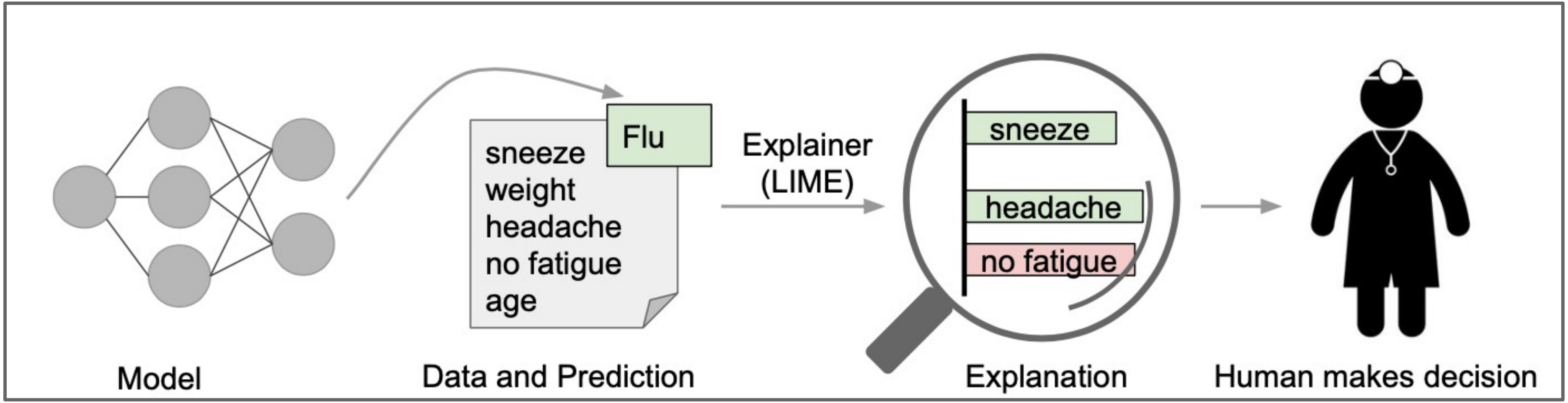
Presented by Robin Na, Paul Liu, Zelin (James) Li

Define Trust

- Trusting a prediction: whether a user trusts sufficiently to take some action based on it

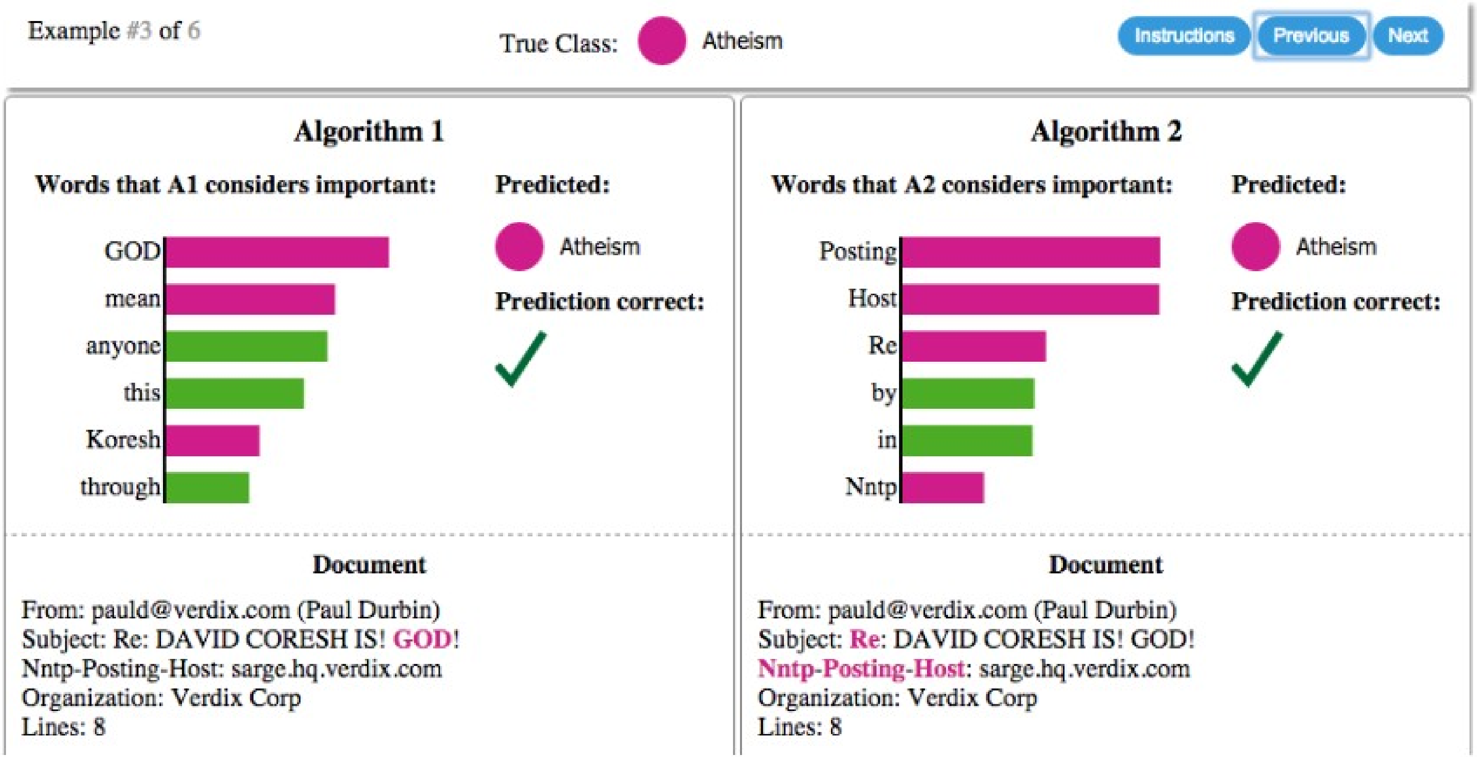
- Trusting a model: whether a user trusts a model to behave in reasonable ways if deployed.

Define Explanation for a Prediction



Textual or visual artifacts to provide qualitative understanding of the relationship between instance’s components and the model’s prediction.

Explanations as a Means to Select Model



Example #3 of 6

True Class: ● Atheism

Algorithm 1

Words that A1 considers important:

Predicted:

● Atheism

Prediction correct:

✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Algorithm 2

Words that A2 considers important:

Predicted:

● Atheism

Prediction correct:

✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Why Do We Need Trust and Interpretability?

- Sometimes models can go wrong (in a way that’s obvious to humans)!
- **Data leakage:** patient ID being correlated with the target class
- **Dataset shift:** training data is different than the test data
- **Exploiting features:** users may favor recommender systems that don’t exploit on “clickbaits”

Desired Characteristics for Explainers

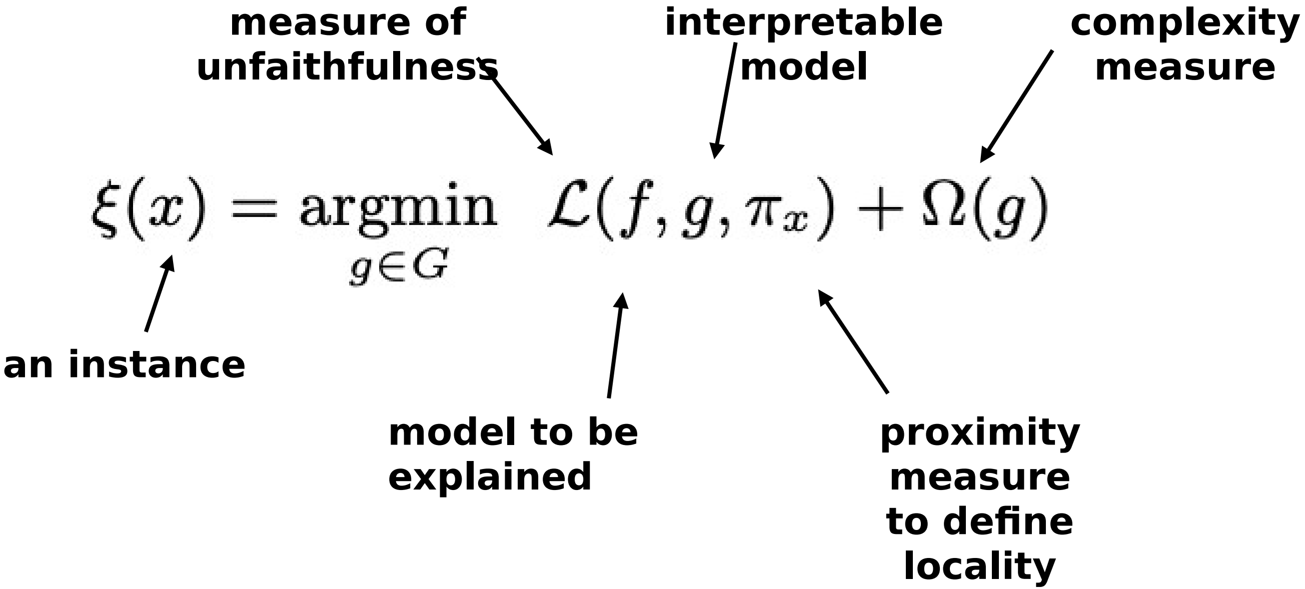
- **Interpretable**
- **Local fidelity:** at least locally faithful
- **Model-agnostic:** the ability to explain any model
- **Global perspective:** ability to explain the model not just single prediction

LIME as Interpretable Framework

Local Interpretable Model-agnostic Explanation

LIME as Interpretable Framework

Step 1: Define LIME Framework (Ensure both local fidelity and interpretability)



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

measure of unfaithfulness

interpretable model

complexity measure

an instance

model to be explained

proximity measure to define locality

LIME as Interpretable Framework

Step 2: Interpretable Data Representation

- Text: binary vector indicating the presence or absence of a word
- Image: binary vector indicating the presence or absence of a contiguous patch of similar pixels

$$x \in \mathbb{R}^d \rightarrow x' \in \{0,1\}^{d'}$$

Original Representation

Interpretable Representation

LIME as Interpretable Framework

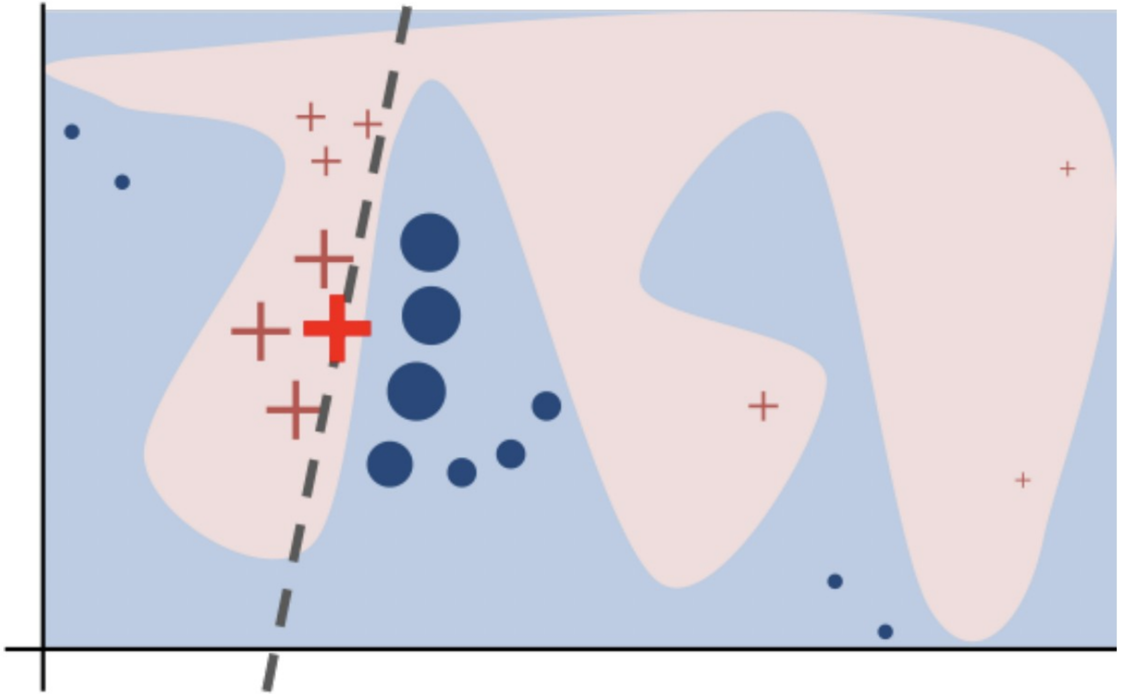
Step 3: Approximate Locality-Aware L $min.\ \mathcal{L}(f, g, \pi_x)$

Perturbed Sample (z’) Sampling Procedure:

- ❑ Sample around x’ by drawing nonzero elements of x’ uniformly at random
- ❑ The number of draws is also uniformly sampled
- ❑ z’ basically has a faction of nonzero elements of x’
- ❑ Samples are weighted by π_x

$$x \xrightarrow[\substack{\text{interpretable}\\ \text{representation}}]{} x' \xrightarrow[\substack{\text{sampling}\\ \text{local area}}]{} z' \xrightarrow[\substack{\text{inverse}\\ \text{(interpretable}\\ \text{representation)}}]{} z \xrightarrow[\text{obtain label}]{} f(z)$$

Sampling Procedure Intuition



Example: Sparse Linear Explanation

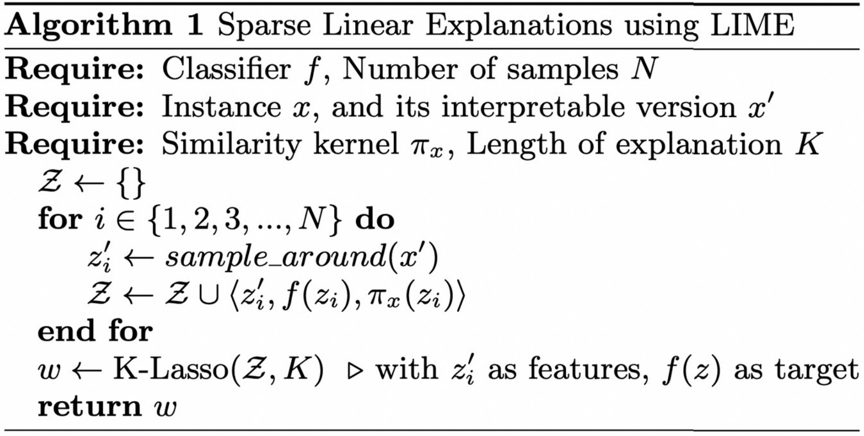
$$\xi(x) = \operatorname{argmin}_{g \in \mathcal{G}} \; \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$$g(z') = w_g \cdot z'$$

$$\pi_x(z) = exp\left(-\frac{D(x,z)^2}{\sigma^2}\right)$$

$$\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z)\big(f(z) - g(z')\big)^2$$

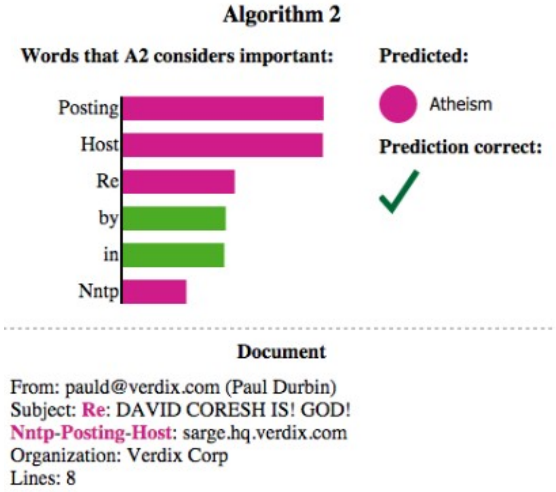
$$\Omega(g) = \infty \mathbb{I}[\| w_g \|_0 > K]$$



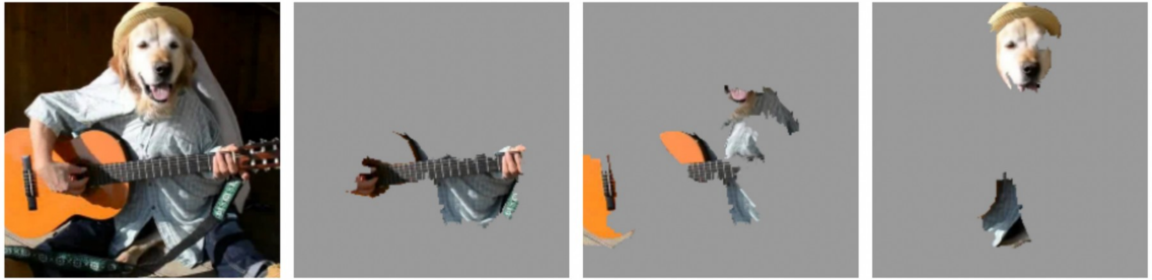
Algorithm 1 Sparse Linear Explanations using LIME

```
Require: Classifier $f$, Number of samples $N$
Require: Instance $x$, and its interpretable version $x'$
Require: Similarity kernel $\pi_x$, Length of explanation $K$
  $\mathcal{Z} \leftarrow \{\}$
  for $i \in \{1, 2, 3, ..., N\}$ do
      $z'_i \leftarrow sample\_around(x')$
      $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  end for
  $w \leftarrow$ K-Lasso($\mathcal{Z}, K$)  ▷ with $z'_i$ as features, $f(z)$ as target
  return $w$
```

Some Results



From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8



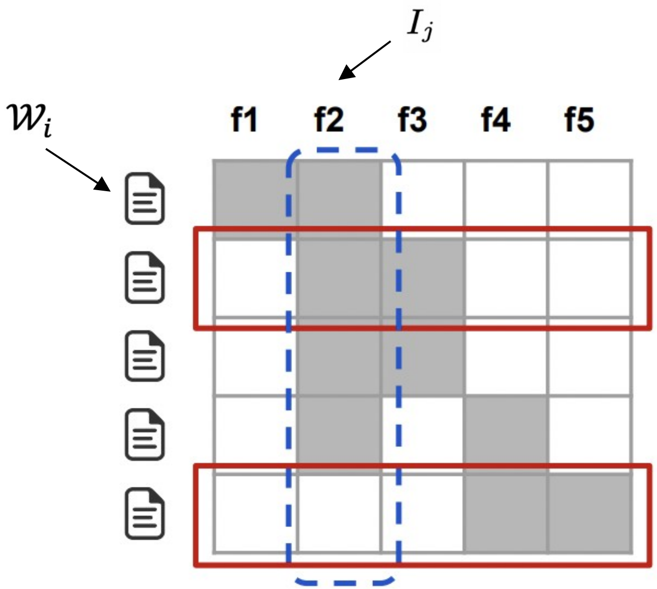
(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

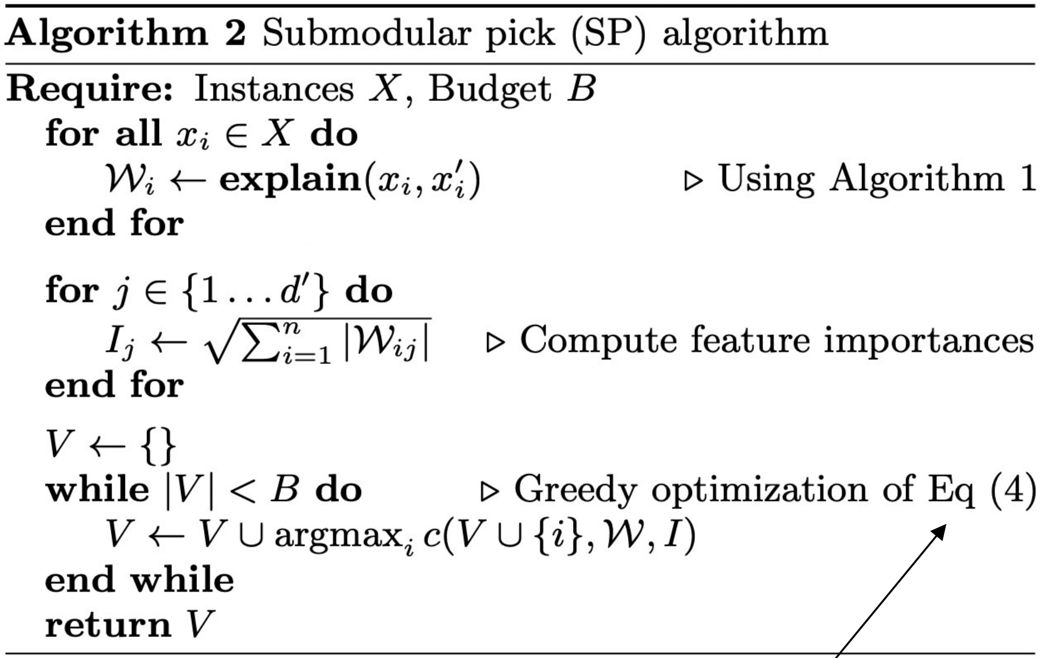
Gain Global Understanding of the Model

Proposal: Explain a set of individual instances

- ➔ The number of instances should be small (denoted by budget B)
- ➔ The pick step should account for the explanations that accompany each prediction
- ➔ Should pick a diverse, representative set



explanation matrix $\mathcal{W}_{n \times d'}$



Algorithm 2 Submodular pick (SP) algorithm

```
Require: Instances $X$, Budget $B$
  for all $x_i \in X$ do
    $\mathcal{W}_i \leftarrow$ explain$(x_i, x'_i)$          ▷ Using Algorithm 1
  end for
  for $j \in \{1 \dots d'\}$ do
    $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$   ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ do          ▷ Greedy optimization of Eq (4)
    $V \leftarrow V \cup \text{argmax}_i \, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return $V$
```

$$c(V, \mathcal{W}, I) = \sum_{j=1}^{d'} \mathbb{1}_{[\exists i \in V : \mathcal{W}_{ij} > 0]} I_j$$

(a submodular problem)

$$\boxed{Pick(\mathcal{W}, I) = \underset{V, |V| \leq B}{\text{argmax}} \, c(V, \mathcal{W}, I)}$$

recall submodularity has the property of diminishing return

Simulated User Experiment

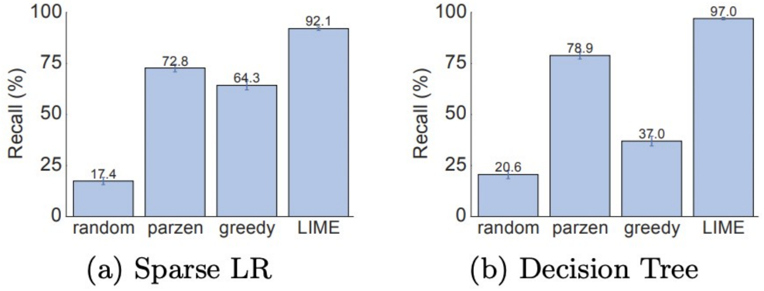
Experimental Setup

- Decision Tree, Logistic Regression, Nearest Neighbors, SVM, RandomForest
- Compare with **parzen, greedy** procedure, **random** procedure
  - Greedy Procedure: greedily remove features that contribute the most until prediction changes
  - Random Procedure: randomly select K features
- If involve pick procedure
  - Submodular Pick (SP)
  - Random Pick (RP)

Simulated User Experiment

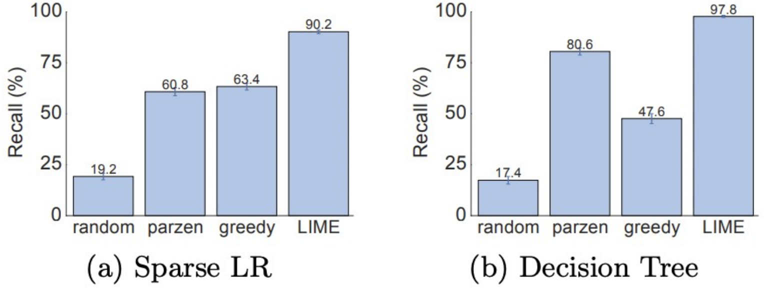
Are explanations faithful to the method?

- Train sparse logistic regression and decision trees
  - Interpretable -> know gold set features
- Compute average fractions of gold features covered by the explanations (over all instance)



(a) Sparse LR (b) Decision Tree

Figure 6: Recall on truly important features for two interpretable classifiers on the books dataset.



(a) Sparse LR (b) Decision Tree

Figure 7: Recall on truly important features for two interpretable classifiers on the DVDs dataset.

Simulated User Experiment

Should I trust this prediction?

1. Randomly select 25% of features to be untrustworthy
2. Label predictions as untrustworthy if prediction changes when all untrustworthy features are removed from the instance
   a. For greedy and random, untrustworthy if untrustworthy features present

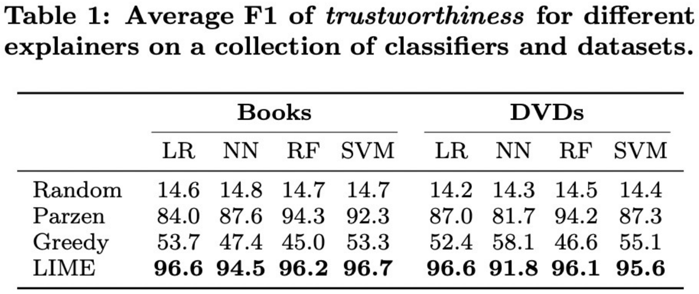


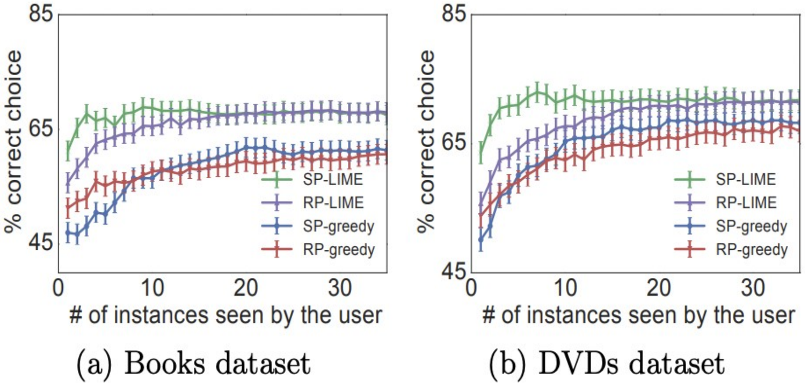
Table 1: Average F1 of *trustworthiness* for different explainers on a collection of classifiers and datasets.

	Books				DVDs			
	LR	NN	RF	SVM	LR	NN	RF	SVM
Random	14.6	14.8	14.7	14.7	14.2	14.3	14.5	14.4
Parzen	84.0	87.6	94.3	92.3	87.0	81.7	94.2	87.3
Greedy	53.7	47.4	45.0	53.3	52.4	58.1	46.6	55.1
LIME	**96.6**	**94.5**	**96.2**	**96.7**	**96.6**	**91.8**	**96.1**	**95.6**

Simulated User Experiment

Can I trust this model?

1. Add noisy features to data to create spurious correlations within dataset
2. Train two random forests
   a. Validation accuracy within 0.1% of each other
   b. Test accuracy differs by at least 5%
3. Marks the noisy features as untrustworthy and follow a similar procedure as before



(a) Books dataset

(b) DVDs dataset

Evaluating with Human Subjects

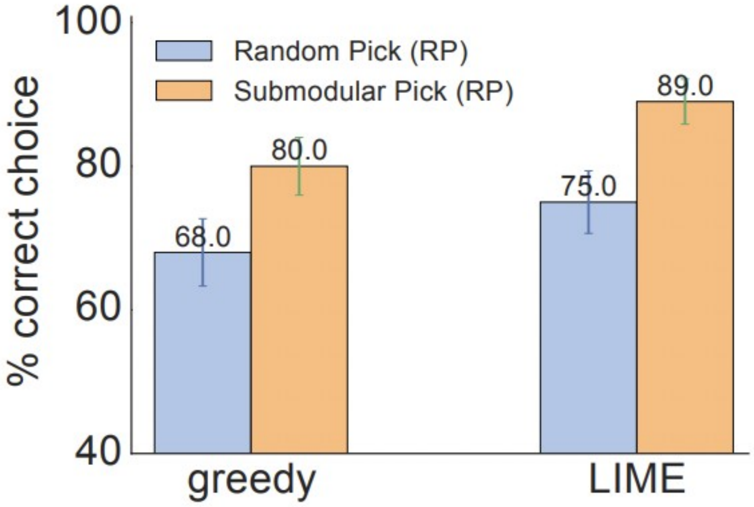
Experimental Setup

- Previous 20 newsgroups dataset (Problematic!)
  - Contain features that do not generalize
- Create a new religion dataset
  - 819 web pages in each of “Christianity” and “Atheism”
- Use SVM with RBF Kernel with hyperparameters chosen by cross-validation

Evaluating with Human Subjects

Can users select the best classifier?

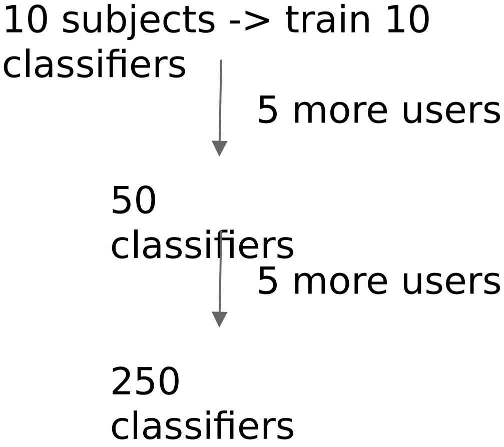
- Train two SVM
  - one on problematic dataset, and the other on “cleaned” dataset
- Recruit human to select which algorithm will perform best



Evaluating with Human Subjects

Can non-experts improve a classifier?

Human marks which words should be removed after seeing B = 10 instances and K = 10 words in explanation.



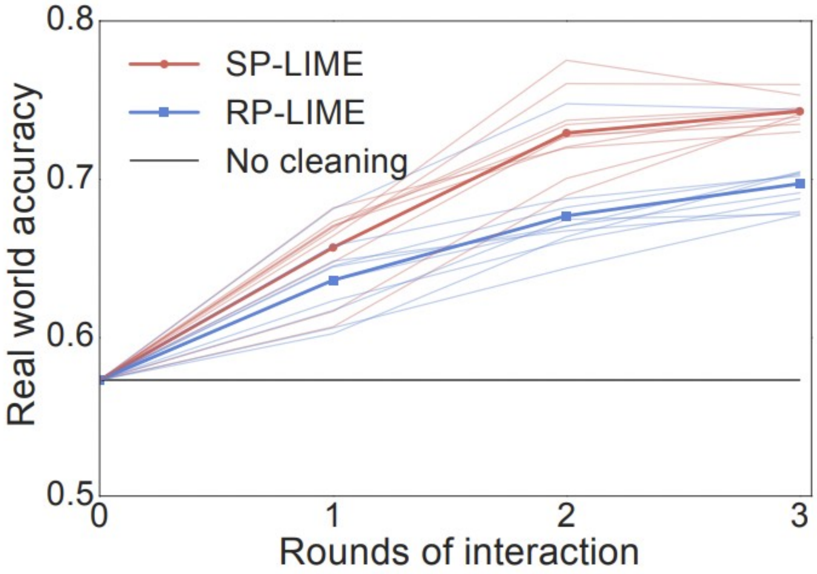
10 subjects -> train 10 classifiers

5 more users

50 classifiers

5 more users

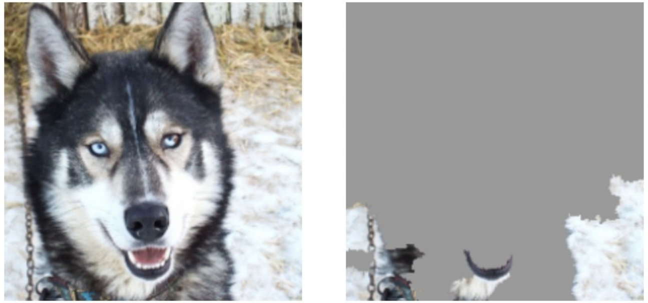
250 classifiers



Evaluating with Human Subjects

Do explanations lead to insights?

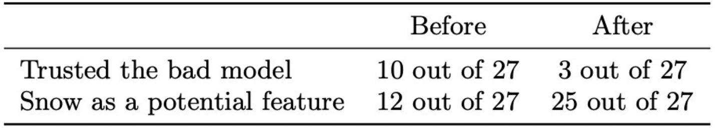
1. Find pictures such that classifier predicts “wolf” if there is snow, and “husky” otherwise.
2. Ask subjects (grad students) questions
   a. Do they trust model to work well in real world
   b. Why?
   c. How do they think the model distinguishes
3. Showed explanations and asked again



(a) Husky classified as wolf

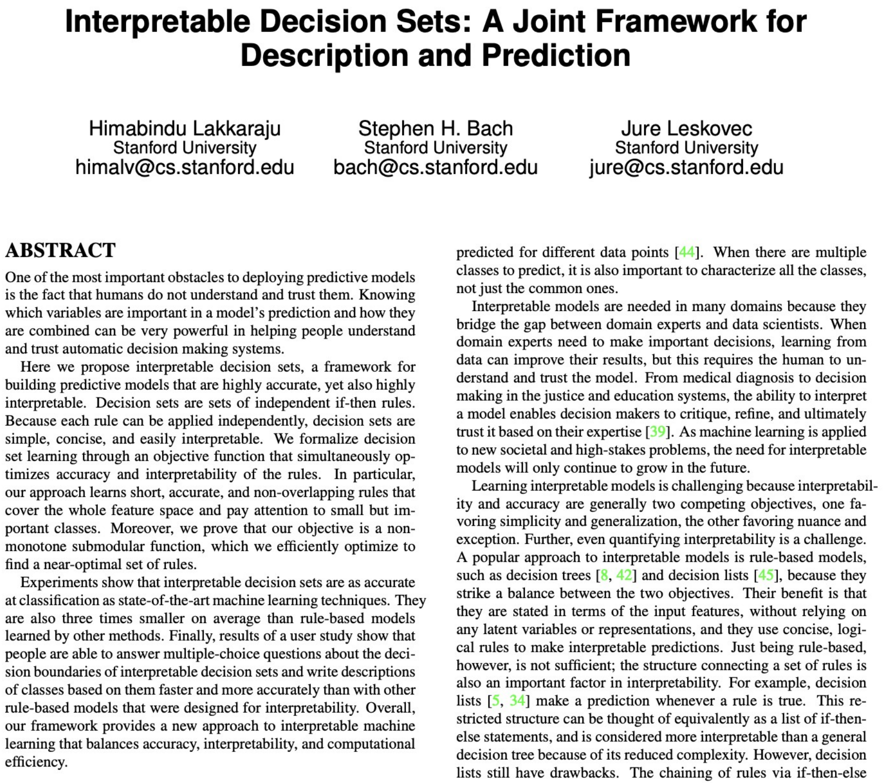
(b) Explanation

Figure 11: Raw data and explanation of a bad model’s prediction in the “Husky vs Wolf” task.



	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Related Work



Interpretable Decision Sets: A Joint Framework for Description and Prediction

Himabindu Lakkaraju
Stanford University
himalv@cs.stanford.edu

Stephen H. Bach
Stanford University
bach@cs.stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

ABSTRACT

One of the most important obstacles to deploying predictive models is the fact that humans do not understand and trust them. Knowing which variables are important in a model's prediction and how they are combined can be very powerful in helping people understand and trust automatic decision making systems.

Here we propose interpretable decision sets, a framework for building predictive models that are highly accurate, yet also highly interpretable. Decision sets are sets of independent if-then rules. Because each rule can be applied independently, decision sets are simple, concise, and easily interpretable. We formalize decision set learning through an objective function that simultaneously optimizes accuracy and interpretability of the rules. In particular, our approach learns short, accurate, and non-overlapping rules that cover the whole feature space and pay attention to small but important classes. Moreover, we prove that our objective is a non-monotone submodular function, which we efficiently optimize to find a near-optimal set of rules.

Experiments show that interpretable decision sets are as accurate at classification as state-of-the-art machine learning techniques. They are also three times smaller on average than rule-based models learned by other methods. Finally, results of a user study show that people are able to answer multiple-choice questions about the decision boundaries of interpretable decision sets and write descriptions of classes based on them faster and more accurately than with other rule-based models that were designed for interpretability. Overall, our framework provides a new approach to interpretable machine learning that balances accuracy, interpretability, and computational efficiency.

predicted for different data points [44]. When there are multiple classes to predict, it is also important to characterize all the classes, not just the common ones.

Interpretable models are needed in many domains because they bridge the gap between domain experts and data scientists. When domain experts need to make important decisions, learning from data can improve their results, but this requires the human to understand and trust the model. From medical diagnosis to decision making in the justice and education systems, the ability to interpret a model enables decision makers to critique, refine, and ultimately trust it based on their expertise [39]. As machine learning is applied to new societal and high-stakes problems, the need for interpretable models will only continue to grow in the future.

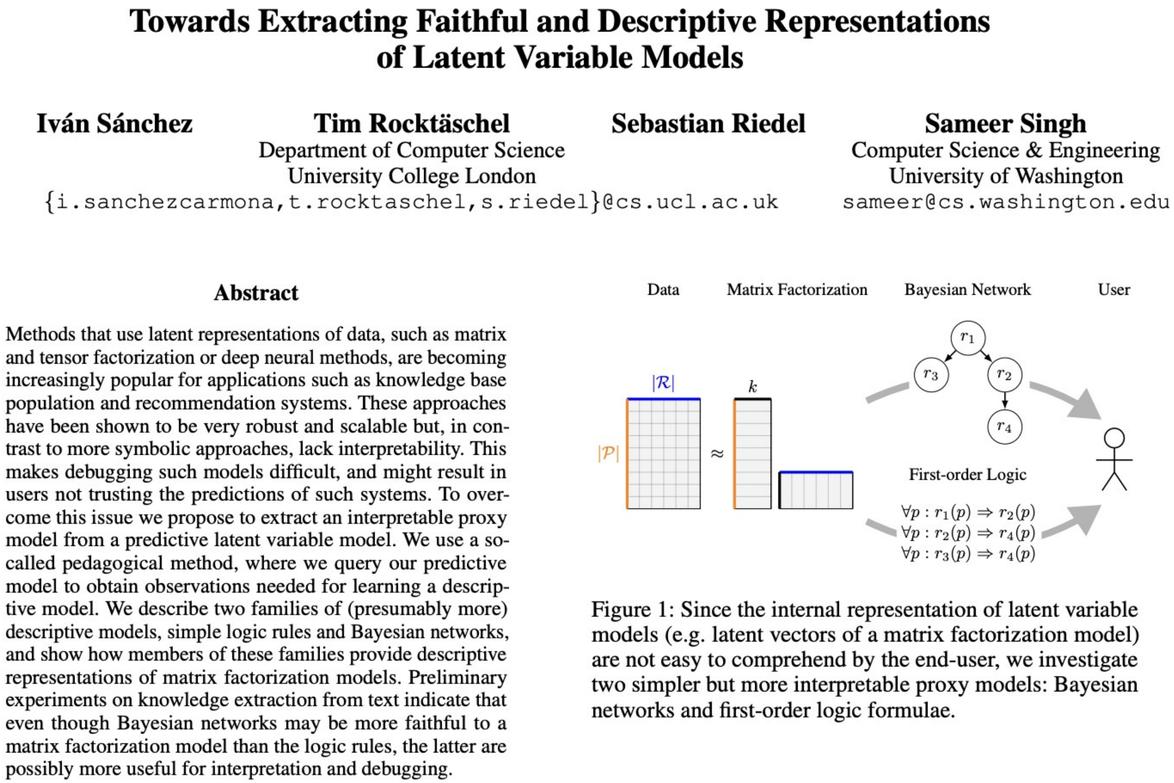
Learning interpretable models is challenging because interpretability and accuracy are generally two competing objectives, one favoring simplicity and generalization, the other favoring nuance and exception. Further, even quantifying interpretability is a challenge. A popular approach to interpretable models is rule-based models, such as decision trees [8, 42] and decision lists [45], because they strike a balance between the two objectives. Their benefit is that they are stated in terms of the input features, without relying on any latent variables or representations, and they use concise, logical rules to make interpretable predictions. Just being rule-based, however, is not sufficient; the structure connecting a set of rules is also an important factor in interpretability. For example, decision lists [5, 34] make a prediction whenever a rule is true. This restricted structure can be thought of equivalently as a list of if-then-else statements, and is considered more interpretable than a general decision tree because of its reduced complexity. However, decision lists still have drawbacks. The chaining of rules via if-then-else

Inherently interpretable models

- Trees, lists, sets, (generalized) linear models and additive models

LIME: one of the first significant papers for post-hoc, model-agnostic analysis

Related Work



Towards Extracting Faithful and Descriptive Representations of Latent Variable Models

Iván Sánchez **Tim Rocktäschel** **Sebastian Riedel**
Department of Computer Science
University College London
{i.sanchezcarmona,t.rocktaschel,s.riedel}@cs.ucl.ac.uk

Sameer Singh
Computer Science & Engineering
University of Washington
sameer@cs.washington.edu

Abstract

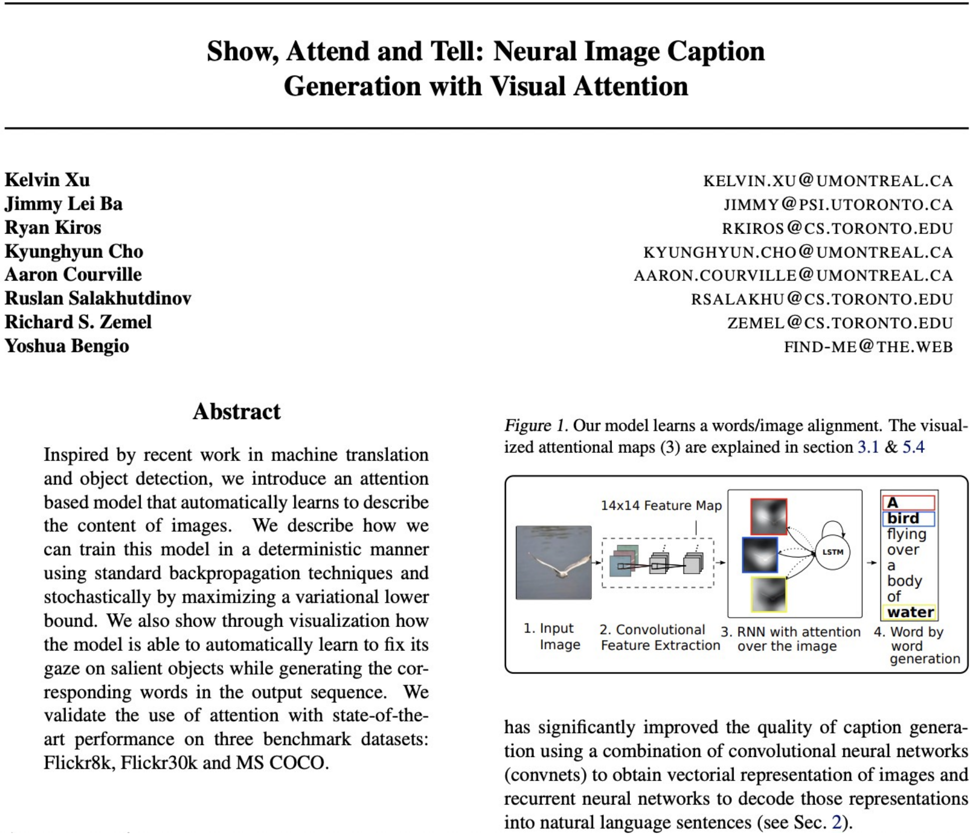
Methods that use latent representations of data, such as matrix and tensor factorization or deep neural methods, are becoming increasingly popular for applications such as knowledge base population and recommendation systems. These approaches have been shown to be very robust and scalable but, in contrast to more symbolic approaches, lack interpretability. This makes debugging such models difficult, and might result in users not trusting the predictions of such systems. To overcome this issue we propose to extract an interpretable proxy model from a predictive latent variable model. We use a so-called pedagogical method, where we query our predictive model to obtain observations needed for learning a descriptive model. We describe two families of (presumably more) descriptive models, simple logic rules and Bayesian networks, and show how members of these families provide descriptive representations of matrix factorization models. Preliminary experiments on knowledge extraction from text indicate that even though Bayesian networks may be more faithful to a matrix factorization model than the logic rules, the latter are possibly more useful for interpretation and debugging.

Figure 1: Since the internal representation of latent variable models (e.g. latent vectors of a matrix factorization model) are not easy to comprehend by the end-user, we investigate two simpler but more interpretable proxy models: Bayesian networks and first-order logic formulae.

Extracting descriptive features

Approximating the model globally through gradient methods is a challenge → Local fidelity

Related Work



Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Kelvin Xu KELVIN.XU@UMONTREAL.CA
Jimmy Lei Ba JIMMY@PSI.UTORONTO.CA
Ryan Kiros RKIROS@CS.TORONTO.EDU
Kyunghyun Cho KYUNGHYUN.CHO@UMONTREAL.CA
Aaron Courville AARON.COURVILLE@UMONTREAL.CA
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Richard S. Zemel ZEMEL@CS.TORONTO.EDU
Yoshua Bengio FIND-ME@THE.WEB

Abstract

Inspired by recent work in machine translation and object detection, we introduce an attention based model that automatically learns to describe the content of images. We describe how we can train this model in a deterministic manner using standard backpropagation techniques and stochastically by maximizing a variational lower bound. We also show through visualization how the model is able to automatically learn to fix its gaze on salient objects while generating the corresponding words in the output sequence. We validate the use of attention with state-of-the-art performance on three benchmark datasets: Flickr8k, Flickr30k and MS COCO.

Figure 1. Our model learns a words/image alignment. The visualized attentional maps (3) are explained in section 3.1 & 5.4

14x14 Feature Map
LSTM
A
bird
flying
over
a
body
of
water
1. Input Image
2. Convolutional Feature Extraction
3. RNN with attention over the image
4. Word by word generation

has significantly improved the quality of caption generation using a combination of convolutional neural networks (convnets) to obtain vectorial representation of images and recurrent neural networks to decode those representations into natural language sentences (see Sec. 2).

Interpretable description of images

LIME aims for a more model-agnostic approach

Limitations and Future Work

- Expected to work on feature spaces that are not too complex
    - Can similar ideas be applied to sequential decision making (e.g., RL)?

- User experiment: results could depend on how the study notified the definition of “trustworthiness” to participants

- A principled way for the algorithm to automatically improve based on its explanation (without humans having to modify the data or algorithm manually)