Faithful and Customizable Explanations of Black Box Models
Himabindu Lakkaraju$^1$, Ece Kamar$^2$, Rich Caruana$^2$, Jure Leskovec$^3$
$^1$Harvard University  $^2$Microsoft Research  $^3$Stanford University
AIES 2019

Presenter: Derrick Blakely
https://qdata.github.io/deep2Read
Outline

1 Background

2 Related Work

3 The MUSE Framework

4 Results

5 Conclusion
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
ML for High Stakes Decisions

- Medical diagnoses
- Recidivism prediction
- Air quality/pollution models
- Inspiring Big Data stock photos
- Finance
- Etc etc etc
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation
Issues

- Proprietary black box models
- Uninterpretable models (often deep learning)
- Often want explanations of model decisions
- Models should be trustworthy
- Want easy understanding and validation
A Popular Algorithm Is No Better at Predicting Crimes Than Random People

The COMPAS tool is widely used to assess a defendant’s risk of committing more crimes, but a new study puts its usefulness into perspective.
Problems in Explainable AI

- Fidelity
- Unambiguity
- Interpretability

Cat
Problems in Explainable AI

- Fidelity
- Unambiguity
- Interpretability

Cat

Presenter: Derrick Blakely  https://qdata.Faithful and Customizable Explanations of Bl:
Problems in Explainable AI

- Fidelity
- Unambiguity
- Interpretability

Cat
Related Work

- Logic-based approximations of black box
  - Decision trees
  - Decision lists
  - Decision sets
Related Work

- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets
Related Work

- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets
Related Work

- Logic-based approximations of black box
- Decision trees
- Decision lists
- Decision sets

Can you transfer information from a pre-trained neural network to this simple model?
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case: “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case: “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case: “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case: “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case:
  “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Related Work Shortcomings

- Explanations are often ambiguous
- Explanations too complicated
- Not customizable
- Don’t target an important use-case: “How does the model’s decision vary based on patient age?”
- Feature subspaces matter
Please Stop Explaining Black Box Models for High-Stakes Decisions

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

Abstract

Black box machine learning models are currently being used for high stakes decision-making throughout society, causing problems throughout healthcare, criminal justice, and in other domains. People have hoped that creating methods for explaining these black box models will alleviate some of these problems, but trying to explain black box models, rather than creating models that are interpretable in the first place, is likely to perpetuate bad practices and can potentially cause catastrophic harm to society. There is a way forward – it is to design models that are inherently interpretable.
Outline

1. Background
2. Related Work
3. The MUSE Framework
4. Results
5. Conclusion
Model Understanding through Subspace Explanations

“Differentiable explanation”: how does model vary across feature subspaces?

Uses 2-level decision sets
Model Understanding through Subspace Explanations

“Differentiable explanation”: how does model vary across feature subspaces?

Uses 2-level decision sets
Model Understanding through Subspace Explanations

“Differentiable explanation”: how does model vary across feature subspaces?

Uses 2-level decision sets
Explanation of the Black Box Model
(No user input)
Decision logic rules

Explanation of the Black Box Model w.r.t. Exercise & Smoking
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), \ldots, (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descriptions
- $\mathcal{DL}$: same as ND but for inner logic
MUSE Framework

- Dataset $D = \{x_1, x_2, ..., x_N\}$, black box $B$, class labels $C$
- Goal: create 2-level decision set $R = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $R$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating desctiprs
- $\mathcal{DL}$: same as ND but for inner logic
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descriptions
- $\mathcal{DL}$: same as ND but for inner logic
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), \ldots, (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descriptions
- $\mathcal{DL}$: same as ND but for inner logic
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descriptions
- $\mathcal{DL}$: same as ND but for inner logic

Presenter: Derrick Blakely  https://qdata.github.io/deep2Read
University of Virginia

Faithful and Customizable Explanations of Black Box Models
Himabindu Lakkaraju 1, Ece Kamar 2, Rich Caruana 2, Jure Leskovec 3
1 Harvard University 2 Microsoft Research 3 Stanford University
AIES 2019
16 / 33
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descripr
- $\mathcal{DL}$: same as ND but for inner logic
MUSE Framework

- Dataset $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, black box $\mathcal{B}$, class labels $\mathcal{C}$
- Goal: create 2-level decision set $\mathcal{R} = \{(q_1, s_1, c_1), ..., (q_M, s_M, c_M)\}$
  - $q_i$: subspace description; a conjunction
  - $s_i$: inner logic; a conjunction
  - $c_i$: label assigned by $\mathcal{R}$
- $\mathcal{ND}$: candidate set of conjunctions; for the generating descriptions
- $\mathcal{DL}$: same as ND but for inner logic
Label Assignment

- $x$ satisfies some $q_i \land s_i \in \mathcal{R} \rightarrow \text{label} = c_i$
- $x$ doesn’t satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- $x$ satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker
Label Assignment

- $x$ satisfies some $q_i \land s_i \in \mathcal{R} \rightarrow \text{label} = c_i$
- $x$ doesn’t satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- $x$ satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker
Label Assignment

- $x$ satisfies some $q_i \land s_i \in \mathcal{R} \rightarrow \text{label} = c_i$
- $x$ doesn’t satisfy any rules in $\mathcal{R} \rightarrow$ assigned with default function
- $x$ satisfies multiple rules $\rightarrow$ one rule is picked via a tie-breaker
Quantifying Fidelity

- Simply how often $R$ agrees with $B$ over $D$
  \[
  \text{disagreement}(R) = \sum \left| \{ x | x \in D, x \text{ satisfies } q_i \land s_i, B \neq c_i \} \right|
  \]
Quantifying Fidelity

- Simply how often $R$ agrees with $B$ over $D$
- $disagreement(R) = \sum \left| \{x | x \in D, x \text{ satisfies } q_i \land s_i, B \neq c_i \} \right|$
Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- goal 2: maximize coverage of the rules
- \( \text{ruleoverlap}(\mathcal{R}) = \text{number of times a conjunction was repeated in } \mathcal{R} \)
- \( \text{cover}(\mathcal{R}) = \text{number of } x \text{ covered by some rule in } \mathcal{R} \)
Goal 1: prevent rules from overlapping too much

Goal 2: maximize coverage of the rules

\[
\text{ruleoverlap}(\mathcal{R}) = \text{number of times a conjunction was repeated in } \mathcal{R}
\]

\[
\text{cover}(\mathcal{R}) = \text{number of } x \text{ covered by some rule in } \mathcal{R}
\]
Goal 1: prevent rules from overlapping too much
goal 2: maximize coverage of the rules

\( \text{ruleoverlap}(\mathcal{R}) = \text{number of times a conjunction was repeated in } \mathcal{R} \)

\( \text{cover}(\mathcal{R}) = \text{number of x covered by some rule in } \mathcal{R} \)
Quantifying Umambiguity

- Goal 1: prevent rules from overlapping too much
- Goal 2: maximize coverage of the rules
- $ruleoverlap(\mathcal{R}) =$ number of times a conjunction was repeated in $\mathcal{R}$
- $cover(\mathcal{R}) =$ number of $x$ covered by some rule in $\mathcal{R}$
Quantifying Interpretability

- \( \text{size}(\mathcal{R}) = \text{number of rule triples} \)
- \( \text{maxwidth}(\mathcal{R}) = \text{length of longest rule in number of predicates} \)
- \( \text{numpreds}(\mathcal{R}) = \text{number of predicates in } \mathcal{R} \text{ (non-unique)} \)
- \( \text{numdsets}(\mathcal{R}) = \text{number of descriptors} \)
- \( \text{featuroverlap}(\mathcal{R}) = \text{overlap of features in descriptors and inner logic} \)
Quantifying Interpretability

- $size(\mathcal{R}) = \text{number of rule triples}$
- $maxwidth(\mathcal{R}) = \text{length of longest rule in number of predicates}$
- $numpreds(\mathcal{R}) = \text{number of predicates in } \mathcal{R} \text{ (non-unique)}$
- $numdsets(\mathcal{R}) = \text{number of descriptors}$
- $featuroverlap(\mathcal{R}) = \text{overlap of features in descriptors and inner logic}$
Quantifying Interpretability

- \( \text{size}(\mathcal{R}) = \) number of rule triples
- \( \text{maxwidth}(\mathcal{R}) = \) length of longest rule in number of predicates
- \( \text{numpreds}(\mathcal{R}) = \) number of predicates in \( \mathcal{R} \) (non-unique)
- \( \text{numdsets}(\mathcal{R}) = \) number of descriptors
- \( \text{featuroverlap}(\mathcal{R}) = \) overlap of features in descriptors and inner logic
Quantifying Interpretability

- \( \text{size}(\mathcal{R}) = \) number of rule triples
- \( \text{maxwidth}(\mathcal{R}) = \) length of longest rule in number of predicates
- \( \text{numpreds}(\mathcal{R}) = \) number of predicates in \( \mathcal{R} \) (non-unique)
- \( \text{numdsets}(\mathcal{R}) = \) number of descriptors
- \( \text{featuroverlap}(\mathcal{R}) = \) overlap of features in descriptors and inner logic
Quantifying Interpretability

- \( \text{size}(\mathcal{R}) = \) number of rule triples
- \( \text{maxwidth}(\mathcal{R}) = \) length of longest rule in number of predicates
- \( \text{numpreds}(\mathcal{R}) = \) number of predicates in \( \mathcal{R} \) (non-unique)
- \( \text{numdsets}(\mathcal{R}) = \) number of descriptors
- \( \text{featuroverlap}(\mathcal{R}) = \) overlap of features in descriptors and inner logic
### Quantified Metrics

**Fidelity**

\[
disagreement(R) = \sum_{i=1}^{M} \left| \{ x \mid x \in D, x \text{ satisfies } q_i \land s_i, \mathcal{B}(x) \neq c_i \} \right|
\]

**Unambiguity**

\[
ruleoverlap(R) = \sum_{i=1}^{M} \sum_{j=1, i \neq j}^{M} overlap(q_i \land s_i, q_j \land s_j)
\]

\[
cover(R) = \left| \{ x \mid x \in D, x \text{ satisfies } q_i \land s_i \text{ where } i \in \{1 \cdots M\} \} \right|
\]

\[
size(R): \text{ number of rules (triples of the form } (q, s, c) \text{) in } R
\]

\[
maxwidth(R) = \max_{e \in \bigcup_{i=1}^{M} (q_i \cup s_i)} width(e)
\]

**Interpretability**

\[
umpreds(R) = \sum_{i=1}^{M} width(s_i) + width(q_i)
\]

\[
umdssets(R) = |dset(R)| \text{ where } dset(R) = \bigcup_{i=1}^{M} q_i
\]

\[
featureoverlap(R) = \sum_{q \in dset(R)} \sum_{i=1}^{M} featureoverlap(q, s_i)
\]
Setting up the Objective Function

- Goal: maximize each $f_i$ reward function
- $W_{max} = \max$ width of any rule in either $ND$ or $DL$

$$f_1(\mathcal{R}) = P_{max} - numpreds(\mathcal{R}), \text{ where } P_{max} = 2 * W_{max} * |ND| * |DL|$$

$$f_2(\mathcal{R}) = O_{max} - featureoverlap(\mathcal{R}), \text{ where } O_{max} = W_{max} * |ND| * |DL|$$

$$f_3(\mathcal{R}) = O'_{max} - ruleoverlap(\mathcal{R}), \text{ where } O'_{max} = N \times (|ND| * |DL|)^2$$

$$f_4(\mathcal{R}) = cover(\mathcal{R})$$

$$f_5(\mathcal{R}) = F_{max} - disagreement(\mathcal{R}), \text{ where } F_{max} = N \times |ND| * |DL|$$
Objective Function

Find $\mathcal{R} \subseteq \mathcal{ND} \times \mathcal{DL} \times \mathcal{C}$ to maximize:

$$\sum_{i=1}^{M} \lambda_i f_i(\mathcal{R})$$

subject to:

$$\text{size}(\mathcal{R}) \leq \epsilon_1$$
$$\text{maxwidth}(\mathcal{R}) \leq \epsilon_2$$
$$\text{numdsets}(\mathcal{R}) \leq \epsilon_3$$

$\lambda_i$: non-negative weight set by user or found via CV.
$\epsilon_i$: set by user.
Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use “approximate local search” algo (Lee at al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set
Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use “approximate local search” algo (Lee et al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set
Objective Function Optimization

- Optimization is NP-Hard; instance of Budgeted Maximum Coverage Problem
- Use “approximate local search” algo (Lee at al. 2009) for 1/5-approximation
- Gist: select a rule that maximizes the objective; repeatedly perform delete or exchange operations to optimize the solution set
Algorithm 1 Optimization Procedure (Lee et al. 2009)

1: Input: Objective $f$, domain $\mathcal{N} \mathcal{D} \times \mathcal{G} \times \mathcal{C}$, parameter $\delta$, number of constraints $k$$
2: V_1 = \mathcal{N} \mathcal{D} \times \mathcal{G} \times \mathcal{C}$$
3: for $i \in \{1, 2 \cdots k + 1\}$ do
4: \hspace{1em} $X = V_i$; $n = |X|$; $S_i = \emptyset$
5: \hspace{1em} Let $v$ be the element with the maximum value for $f$ and set $S_i = v$
6: \hspace{1em} while there exists a delete/update operation which increases the value of $S_i$ by a factor of at least $\left(1 + \frac{\delta}{n^4}\right)$ do
7: \hspace{2em} \textbf{Delete Operation}: If $e \in S_i$ such that $f(S_i \setminus \{e\}) \geq (1 + \frac{\delta}{n^4})f(S_i)$, then $S_i = S_i \setminus e$
8: \hspace{1em} \textbf{Exchange Operation} If $d \in X \setminus S_i$ and $e_j \in S_i$ (for $1 \leq j \leq k$) such that
9: \hspace{2em} $(S_i \setminus e_j) \cup \{d\}$ (for $1 \leq j \leq k$) satisfies all the $k$ constraints and
10: \hspace{2em} $f(S_i \setminus \{e_1, e_2 \cdots e_k\} \cup \{d\}) \geq (1 + \frac{\delta}{n^4})f(S_i)$, then $S_i = S_i \setminus \{e_1, e_2, \cdots e_k\} \cup \{d\}$
11: \hspace{1em} end while
12: \hspace{1em} $V_{i+1} = V_i \setminus S_i$
13: \hspace{1em} end for
14: \hspace{1em} return the solution corresponding to $\max\{f(S_1), f(S_2), \cdots f(S_{k+1})\}$
Outline

1 Background

2 Related Work

3 The MUSE Framework

4 Results

5 Conclusion
Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients
Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients
Datasets

1. Bail outcomes (released on bail or not) for 86K defendants
2. High school performance (graduated on time or not) for 21K students
3. Depression diagnoses (depressed or not) 33K patients
Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)
Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)
Baselines

- Locally Interpretable Model agnostic Explanations (LIME) (Ribeiro, Singh, and Guestrin 2016)
- Interpretable Decision Sets (IDS) (Lakkaraju, Bach, and Leskovec 2016)
- Bayesian Decision Lists (BDL) (Letham et al. 2015)
Results

(a) Number of Rules

(b) Avg. Number of Predicates
### Results

<table>
<thead>
<tr>
<th>Approach</th>
<th>Human Accuracy</th>
<th>Avg. Time (in secs.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSE (No customization)</td>
<td>94.5%</td>
<td>160.1</td>
</tr>
<tr>
<td>IDS</td>
<td>89.2%</td>
<td>231.1</td>
</tr>
<tr>
<td>BDL</td>
<td>83.7%</td>
<td>368.5</td>
</tr>
<tr>
<td>MUSE (Customization)</td>
<td>98.3%</td>
<td>78.3</td>
</tr>
</tbody>
</table>

(c) Results of User Study
Outline

1. Background
2. Related Work
3. The MUSE Framework
4. Results
5. Conclusion
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L$, and $D_L$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
  - Asks end users to do a lot of work
    - create $D, N, L$, and $D_{\epsilon}$ sets
    - Set objective function weights
    - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L,$ and $DL$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
**Shortcomings**

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{N}, \mathcal{L}$, and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L$, and $DL$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L$, and $DL$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L$, and $DL$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
  - Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $D, N, L,$ and $D_L$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
  - Might be better as an analysis tool for ML developers
Shortcomings

- Can’t work on image classifiers; needs to be combined with feature extraction from middle layers of NN
- What if we value some features more than others?
- Asks end users to do a lot of work
  - create $\mathcal{D}, \mathcal{N}, \mathcal{L}$, and $\mathcal{DL}$ sets
  - Set objective function weights
  - Set $\epsilon$ constraint values
- Is 85% tolerable in high stakes situations?
- Possibly encourages bad practice
- Might be better as an analysis tool for ML developers
Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules

- Valuable for sanity checking or helping others use model

- More work is needed on interpretable black box algorithms
Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules
- Valuable for sanity checking or helping others use model
- More work is needed on interpretable black box algorithms
Lessons Learned

- Potentially good idea to build an interpretable approximation of your model using logic rules
- Valuable for sanity checking or helping others use model
- More work is needed on interpretable black box algorithms