Fairness

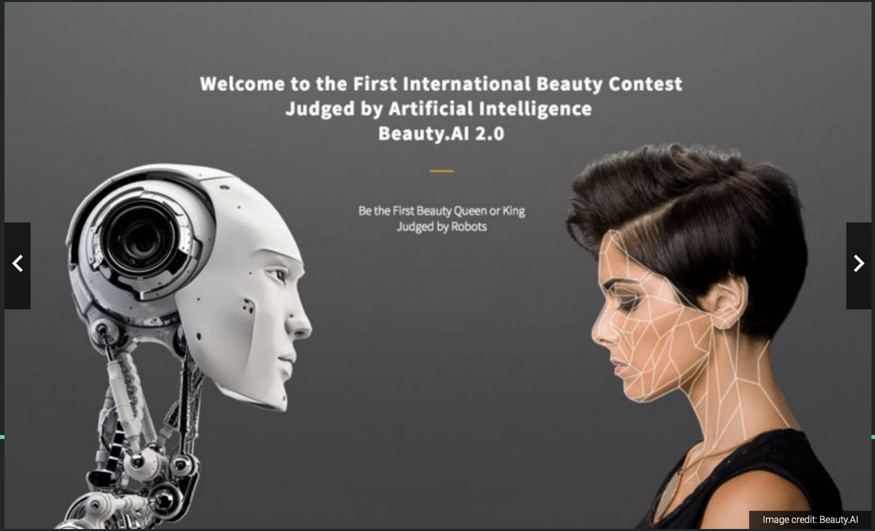
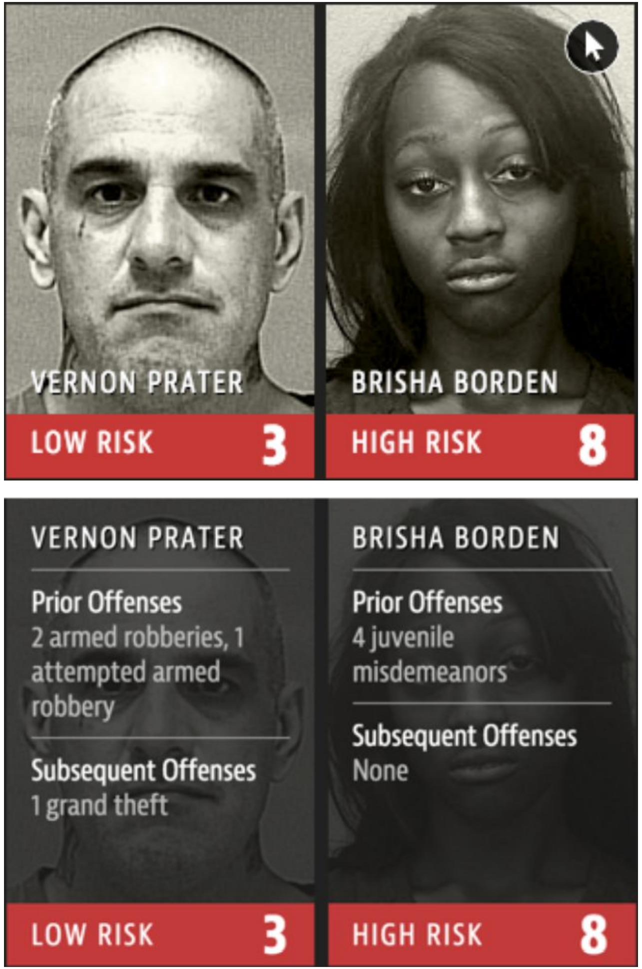


Image credit: Beauty.AI

Motivation

- When using machine learning in human centric decisions, we need to ensure that our algorithm is making fair choices and avoids discrimination.
- Discrimination can arise naturally from data, and ML algorithms can amplify existing bias.
- Just removing sensitive features is not the answer, as proxy features can encode sensitive information (e.g., zip code as a proxy for race)

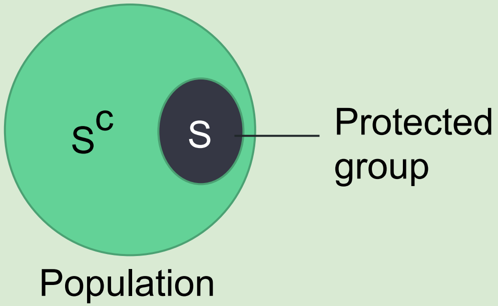


Fairness in Machine Learning: what is it?

- Multiple definitions. No unique answer. However, definitions can be grouped:

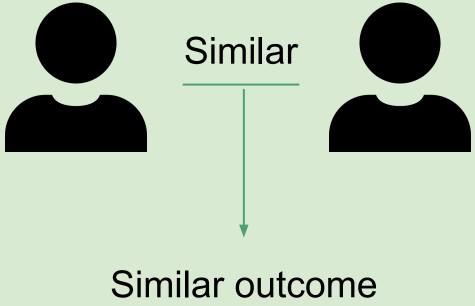
Group Fairness

Avoid discrimination with respect to protected group (or protected class) and the rest of the population.



Individual Fairness

Takes into account each individual. Similar individuals should be treated similarly.



Where do we measure fairness?

Data:

- The data itself might be biased because of how it was collected.
- The dataset might correspond to a biased subset of a larger set.

Classifier:

- The classifier might have learned from a specific unfair subset of data
- The classifier might be looking at patterns in the data that cause discrimination

A bias in the data leads to a biased classifier, but a classifier might discriminate from fair data depending on the training procedure.

Different metrics for different situations

Do we care about the balance of positive outcomes?

Statistical Parity

Do we want parity in error rates?

FPR/FNR balance

Do we have a metric for similarity between individuals?

Fairness through awareness

Do we care about pairwise metrics?

Weakly Meritocratic Fairness

Statistical Parity

Given classifier decision `d` and protected group status `G`, estimating this metric is easy :

1. Calculate the proportion of `d=1` where `G=1` (`sum(d & g) / sum(g)`)
2. Calculate the proportion of `d=1` where `G=0` (`sum(d & ~g) / sum(~g)`)
3. Subtract the two values.

We'll code these steps in the function "evaluate_statistical_parity"

Conditional Parity

Given classifier decision `d`, protected group status `G`, and conditional value `c`, we can estimate it like so:

1. Calculate the proportion of `d=1` where `G=1` and the conditional is met (c=1) (`sum(d & g & c) / sum(g & c)`)
2. Calculate the proportion of `d=1` where `G=0` and the conditional is met (`sum(d & ~g & c) / sum(~g & c)`)
3. Subtract the two values.

We'll code these steps in the function "evaluate_conditional_parity"

False positive rate, true positive rate

This measure takes into account the error rates of the classifiers, and proposes to balance them out. The FPR balance makes sure that the false positive ratios be equal, which would help in scenarios where disproportionate mistakes on positive decisions would be discriminatory, such as deciding t stop and frisk an individual.

How to estimate them:

1. Calculate false positive (negative) rate for protected group
2. Subtract false positive (negative) rate for unprotected group.

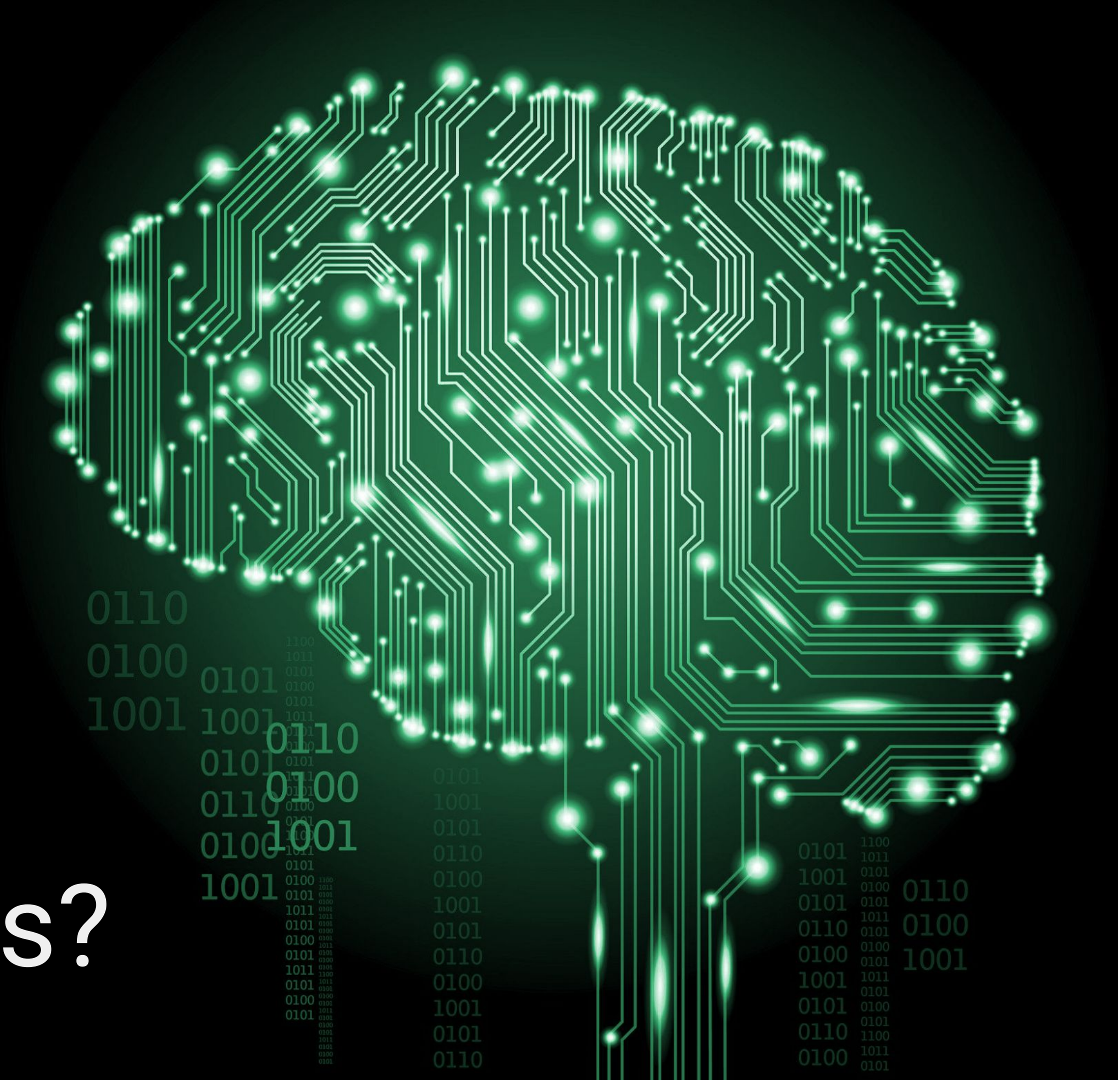
What we’ll do in this workshop

- Implement basic fairness measures such as statistical parity and conditional parity
- Demonstrate their use on an unfair dataset
- Build a model that tries to be fair

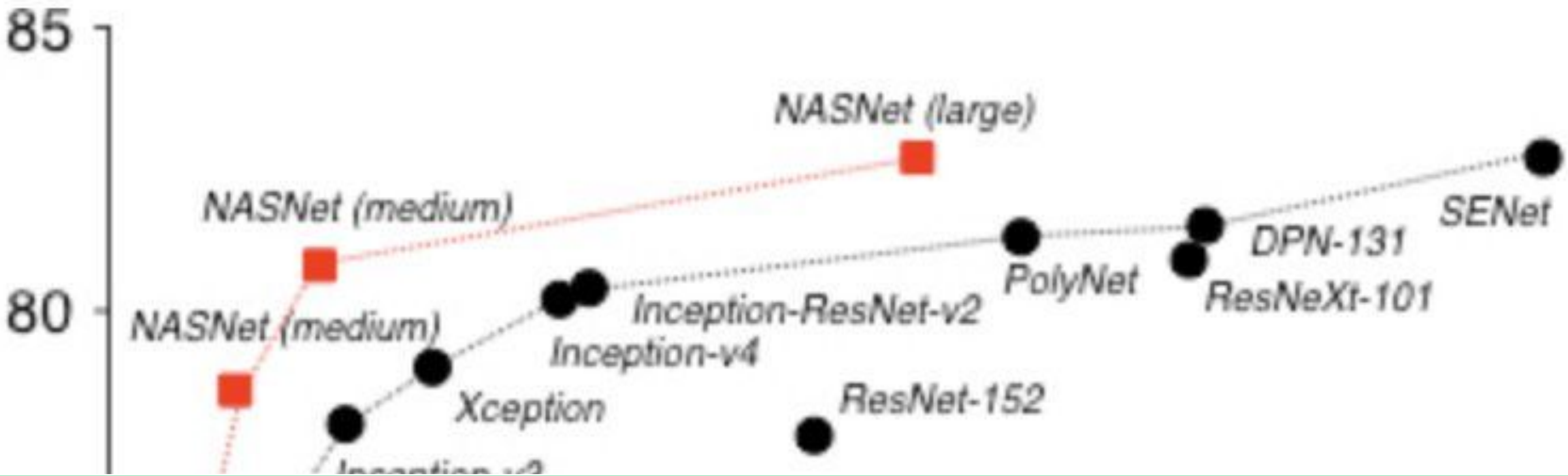
To the notebooks!

Interpretability

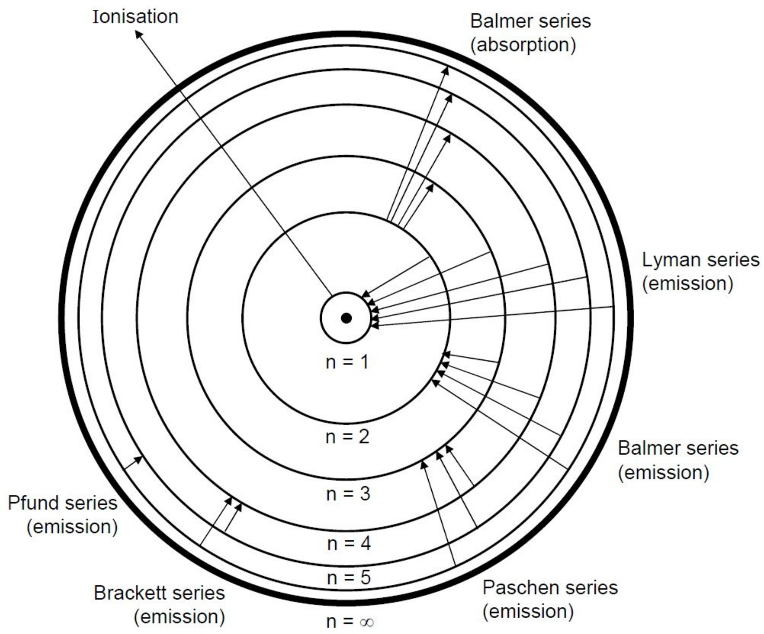
Why do we build models?



Is it to get an extra 0.5% on ImageNet?

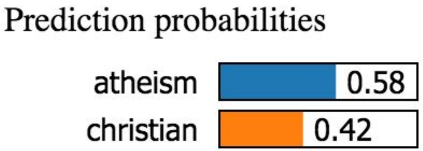


No!



We build models to explain
how the world works.

It also makes us better data scientists.



Text with highlighted words

From: johnchad@triton.unm.edu (jchadwic)
Subject: Another request for Darwin Fish
Organization: University of New Mexico, Albuquerque
Lines: 11
NNTP-Posting-Host: triton.unm.edu

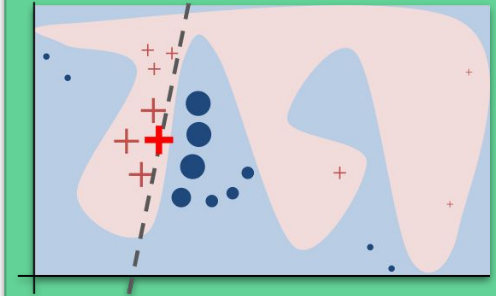
Hello Gang,

There have been some notes recently asking where to obtain the
DARWIN fish.
This is the same question I have and I have not seen an answer on
the
net. If anyone has a contact please post on the net or email me.

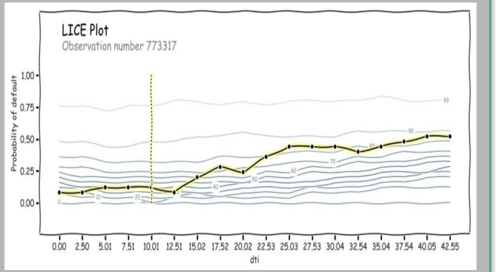
This model has 92.4% accuracy.

Interpretability tools we’ll discuss today:

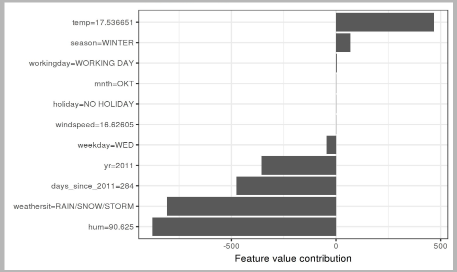
Local Perturbations



Decision Boundaries



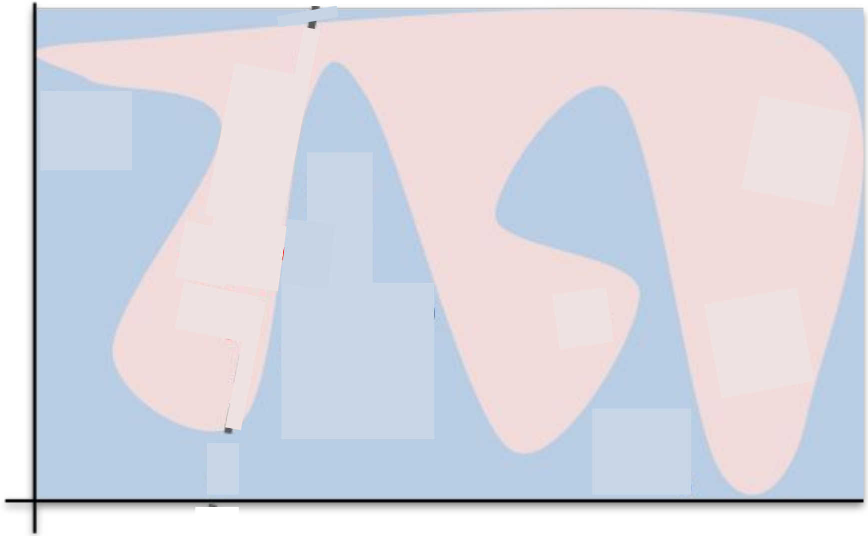
Shapley Values



Local Perturbations: LIME

Local interpretability:

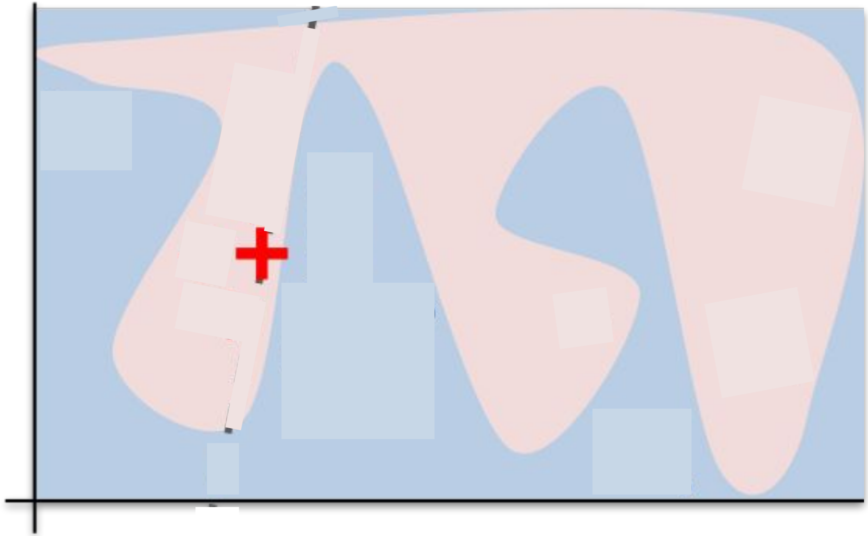
1: Take a black-box model with arbitrary decision boundary



Ribiero, Singh, Guestrin, 2016. '"Why Should I Trust You?" Explaining the Predictions of Any Classifier'
https://arxiv.org/pdf/1602.04938v1.pdf

Local Perturbations: LIME

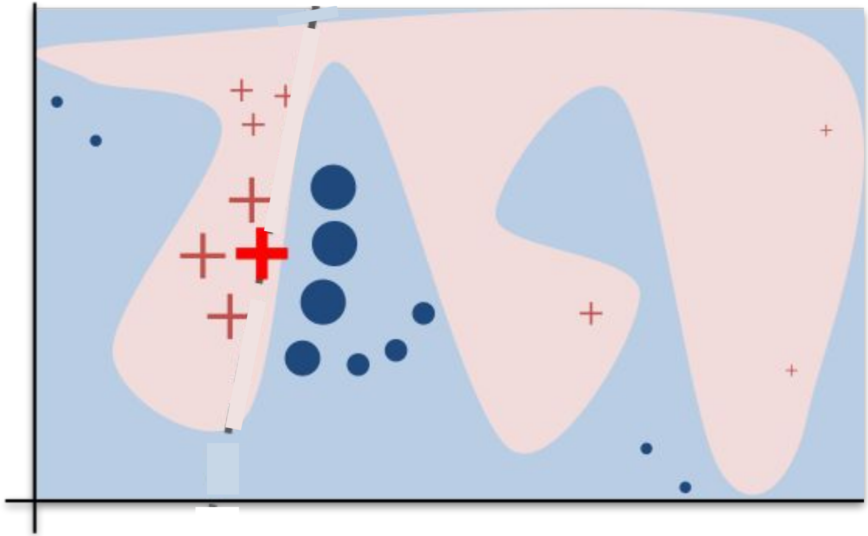
2: Select observation to explain



Ribiero, Singh, Guestrin, 2016. '"Why Should I Trust You?" Explaining the Predictions of Any Classifier'
https://arxiv.org/pdf/1602.04938v1.pdf

Local Perturbations: LIME

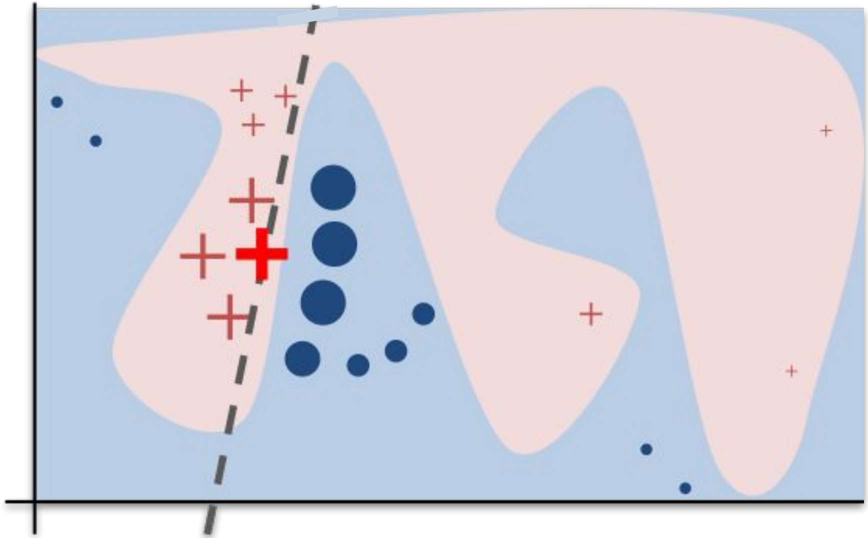
3: Sample around the chosen selected point



Ribiero, Singh, Guestrin, 2016. '"Why Should I Trust You?" Explaining the Predictions of Any Classifier'
https://arxiv.org/pdf/1602.04938v1.pdf

Local Perturbations: LIME

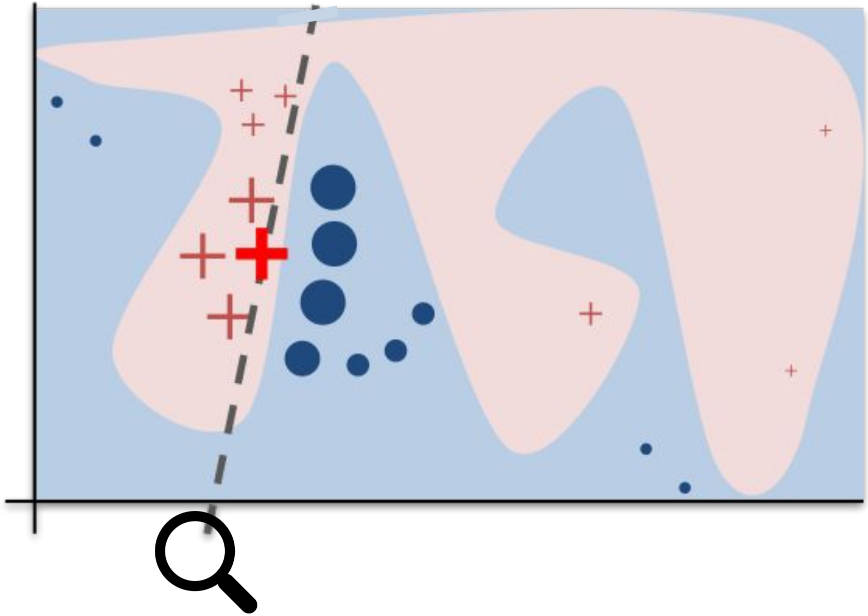
4: Build a local linear regression with the samples



Ribiero, Singh, Guestrin, 2016. '“Why Should I Trust You?” Explaining the Predictions of Any Classifier'
https://arxiv.org/pdf/1602.04938v1.pdf

Local Perturbations: LIME

5: Feature weights correspond to local explanation!



Ribiero, Singh, Guestrin, 2016. '"Why Should I Trust You?" Explaining the Predictions of Any Classifier'
https://arxiv.org/pdf/1602.04938v1.pdf

Local Perturbations: Input Gradients

$$\frac{\partial \hat{f}}{\partial x_k} \approx \frac{\hat{f}(x_1,\ldots,x_k+h,\ldots,x_p)-\hat{f}(x_1,\ldots,x_k,\ldots,x_p)}{h}$$

$$g_k(x) \equiv \frac{\partial \hat{f}}{\partial x_k}(x)$$

Non-linear local interpretability

Use backpropagation to calculate gradient wrt each feature at the selected observation

Hechtlinger, 2016. 'Interpretation of Prediction Models Using the Input Gradient'
https://arxiv.org/pdf/1611.07634.pdf

Local Perturbations: Input Gradients

$$\frac{\partial \hat{f}}{\partial x_k} \approx \frac{\hat{f}(x_1,\ldots,x_k+h,\ldots,x_p)-\hat{f}(x_1,\ldots,x_k,\ldots,x_p)}{h}$$

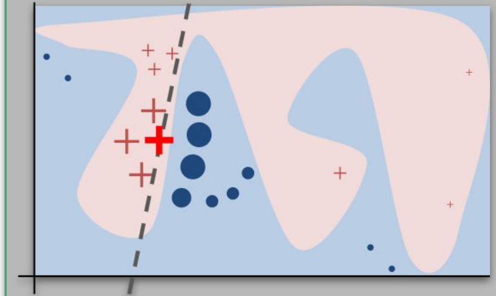
$$g_k(x) \equiv \frac{\partial \hat{f}}{\partial x_k}(x)$$

Pros: Handles highly non-linear decision surfaces better than LIME

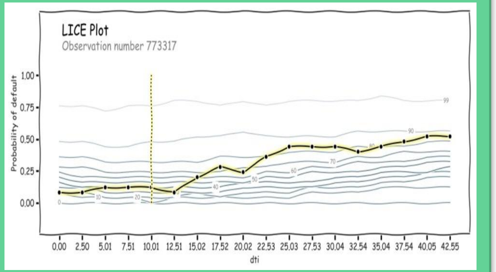
Cons: Must specify an h.
Poorly-specified h values compromise accuracy

Hechtlinger, 2016. 'Interpretation of Prediction Models Using the Input Gradient'
https://arxiv.org/pdf/1611.07634.pdf

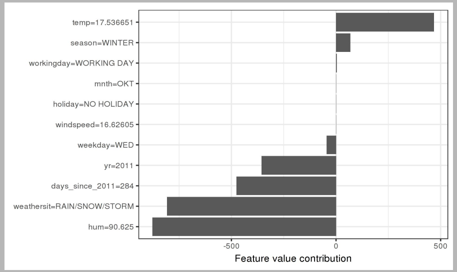
Local Perturbations



Decision Boundaries



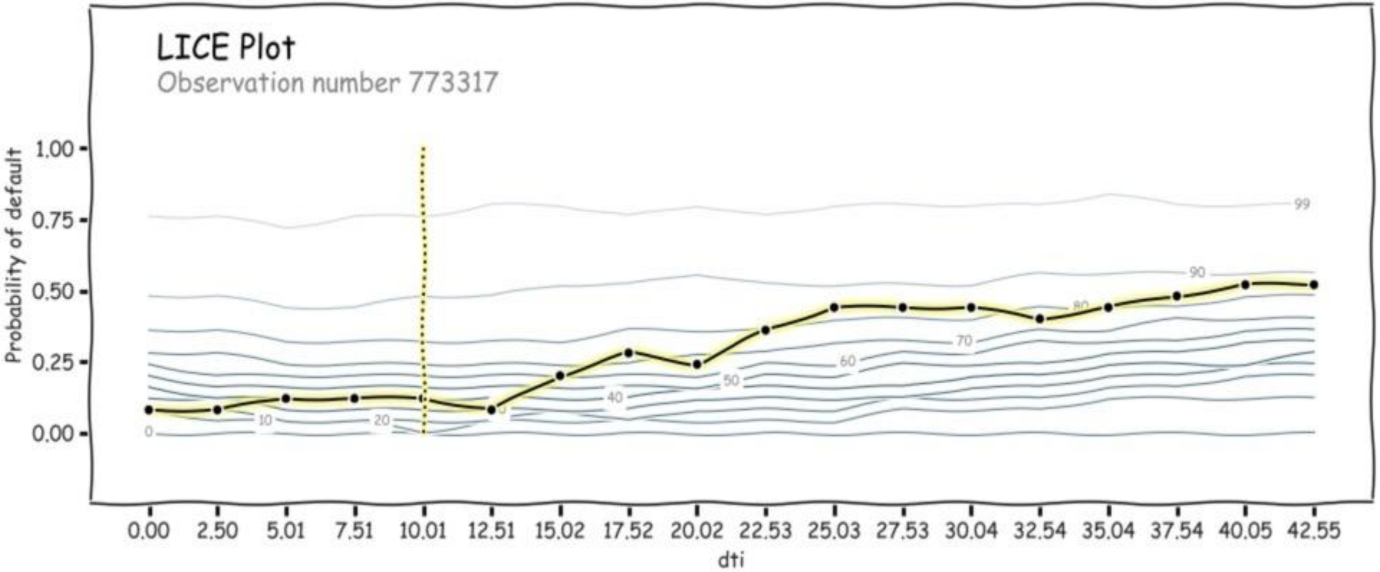
Shapley Values



Partial Dependencies: Local ICE

model predictions, by decile

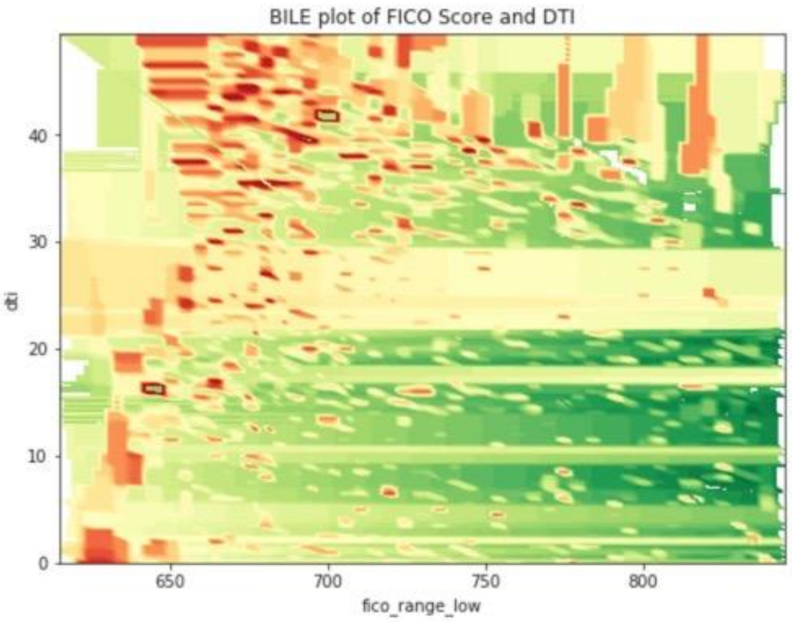
(observation in yellow)



“what-if” analysis of the same observation with adjusted DTI

Goldstein et al, 2014. ‘Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation’
https://arxiv.org/pdf/1309.6392.pdf

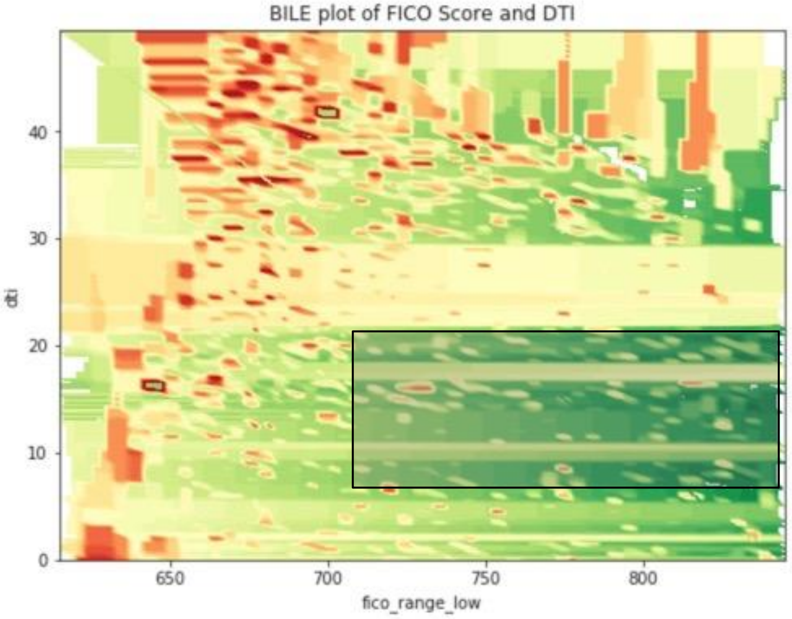
Partial Dependencies: BILE Decision Boundary



- Compresses high-dimension feature set into 2d visualizations:
    1. For each [x, y], find nearest real observation using kd-tree
    2. Assign nearest neighbor's predicted response
    3. Heatmap!

Goldstein et al, 2014. 'Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation'
https://arxiv.org/pdf/1309.6392.pdf

Partial Dependencies: BILE Decision Boundary



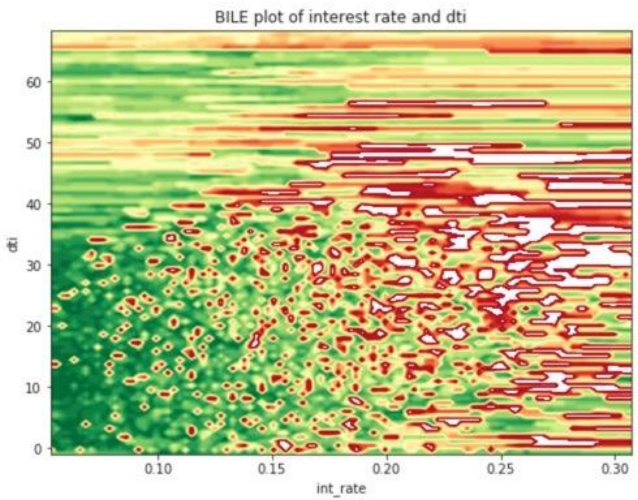
Note the region where loans are generally approved:

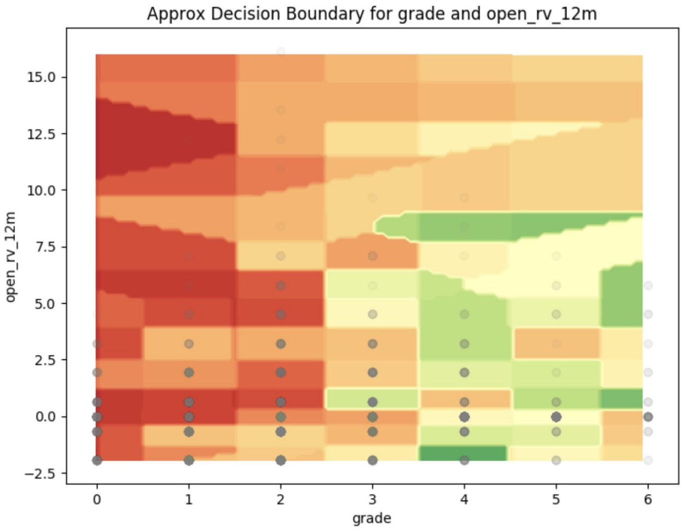
FICO > 700

5 < Debt:Income < 20

Goldstein et al, 2014. ‘Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation’
https://arxiv.org/pdf/1309.6392.pdf

Partial Dependencies: BILE Decision Boundary

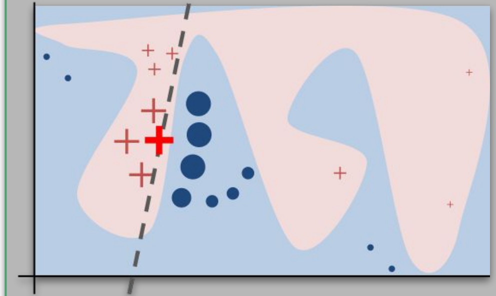




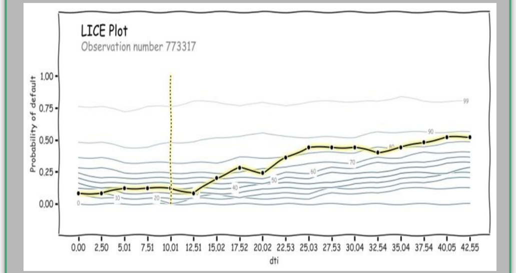
#TODO: Include this slide to show what overfitting looks like??

Goldstein et al, 2014. ‘Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation’
https://arxiv.org/pdf/1309.6392.pdf

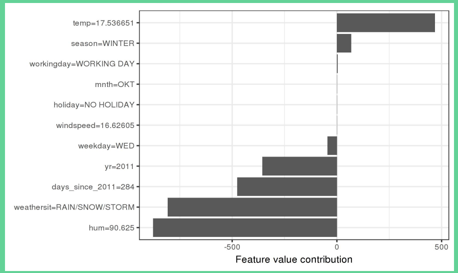
Local Perturbations



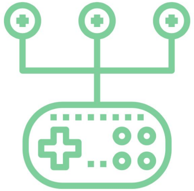
Decision Boundaries



Shapley Values



Shapley Values



If you played a game with multiple players, how would you assign rewards?

Everyone should be rewarded based on his or her contribution, right?

The difficult problem is properly assigning credit for contributions (e.g. what if two features contribute the same thing?)

How does this apply to interpretability?

We can treat each feature as a player in a game, and use Shapley values to determine their contribution to the model's value.

This must satisfy several axioms...

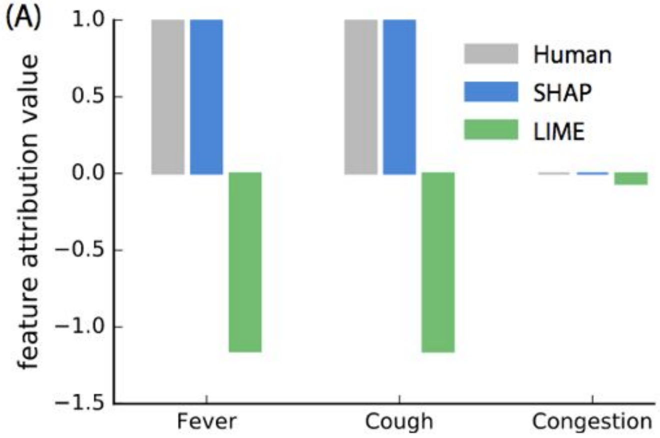
Shapley Values

Axioms

1. If a player never adds any marginal value, their payoff portion should be 0 (Dummy Player)
2. If two players always add the same marginal value to any subset to which they're added, their payoff portion should be the same (Substitutability)
3. If a game is composed of two subgames, you should be able to add the payoffs calculated on the subgames, and that should match the payoffs calculated for the full game (Additivity)

[Source: https://towardsdatascience.com/one-feature-attribution-method-to-supposedly-rule-them-all-shapley-values-f3e04534983d]

Shapley Values: Matching human intuition



In empirical tests, Shapley values more consistently match human intuition than LIME and related methods.

[Source: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf]

Shapley Values

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right].$$

Notation: |F| is the size of the full coalition. S represents any subset of the coalition that doesn't include player i, and |S| is the size of that subset. The bit at the end is just "how much bigger is the payoff when we add player i to this particular subset S"

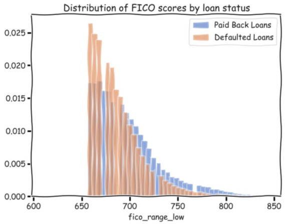
Pros: Matches human intuition and provably satisfies axioms

Cons: Computationally intensive, even with sampling approximation ($2^{|F|}$ subsets)

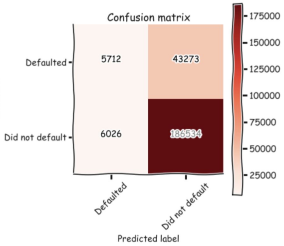
[Source: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf]

To the notebooks!

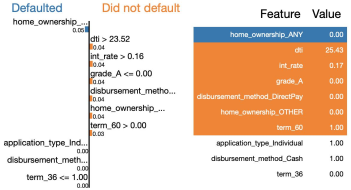
```
$ git clone https://github.com/pblankley/interp-workshop-2019.git
$ cd interp-workshop-2019
$ ./make_env.sh
```



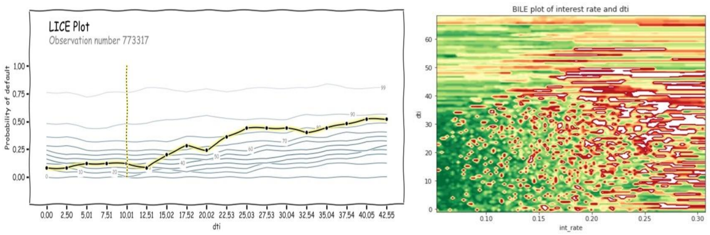
Step 0: EDA



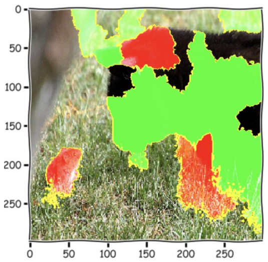
Step 1: Build model



Step 2: LIME



Step 3: Decision boundaries



Secret bonus step!
LIME for images/text

Clardic Fug 112 113 84
Snowbonk 201 199 165
Catbabel 97 93 68
Bunflow 190 174 155
Ronching Blue 121 114 125
Bank Butt 221 196 199
Caring Tan 171 166 170
Stargoon 233 191 141
Sink 176 138 110
Stummy Beige 216 200 185
Dorkwood 61 63 66
Flower 178 184 196
Sand Dan 201 172 143
Grade Bat 48 94 83
Light Of Blast 175 150 147
Grass Bat 176 99 108
Sindis Poop 204 205 194
Dope 219 209 179
Testing 156 101 106
Stoner Blue 152 165 159
Burble Simp 226 181 132
Stanky Bean 197 162 171
Turdly 190 164 116