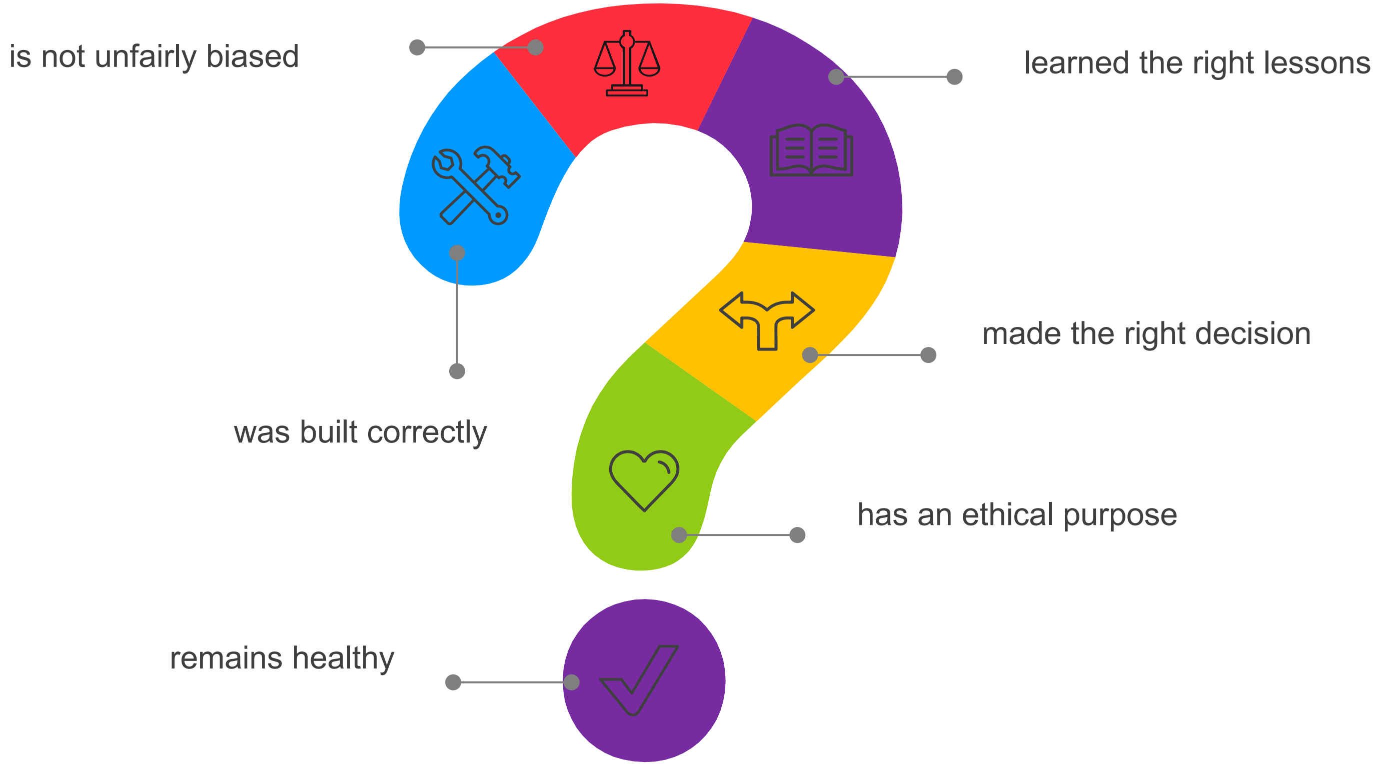

Rajiv Shah: @rajcs4
A Quest for Interpretability
https://bit.ly/inter_workshop


Predictive Model Around Aggression

Trust: The big picture



Today, We are focusing on Just a Small Part


Interpretable Predictive Model Around Aggression

Why Interpretability?

Why?

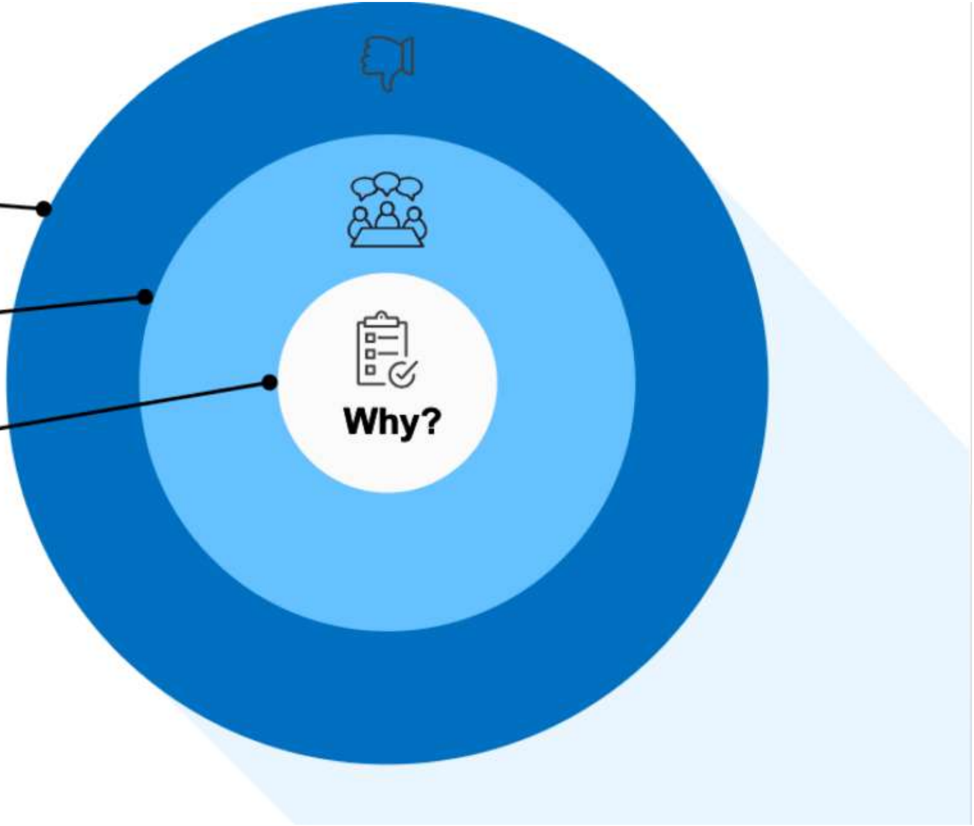
It's easy for things to go wrong

Why?

You need buy-in from human staff

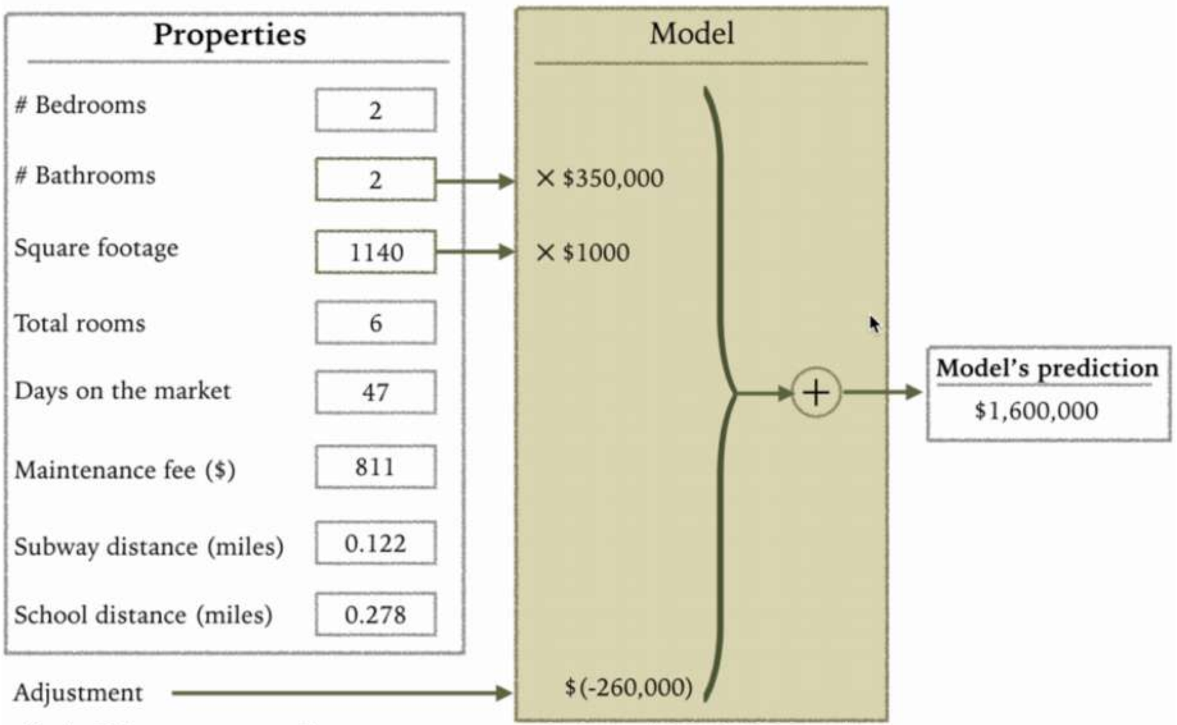
Why?

You need buy-in from regulators



An Understandable White Box Model

All the features and calculations are exposed

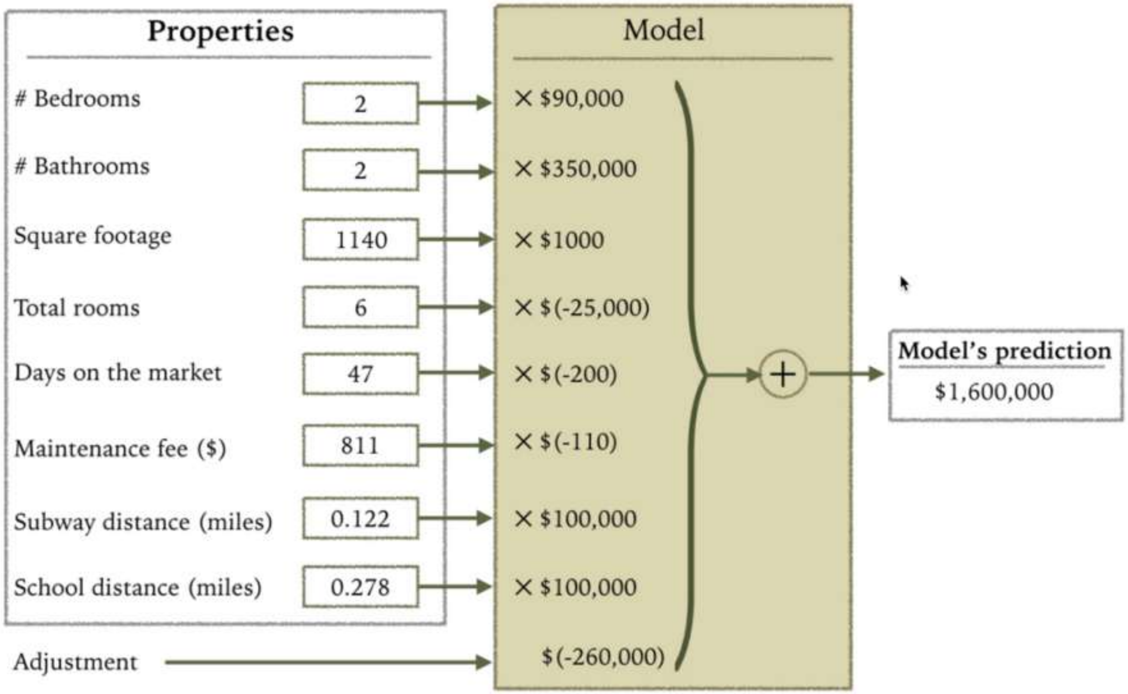


(b) Clear, two-feature condition (CLEAR-2)

Source: [Poursabzi-Sangdeh 2017]()

An Understandable White Box Model? #$@&%*!

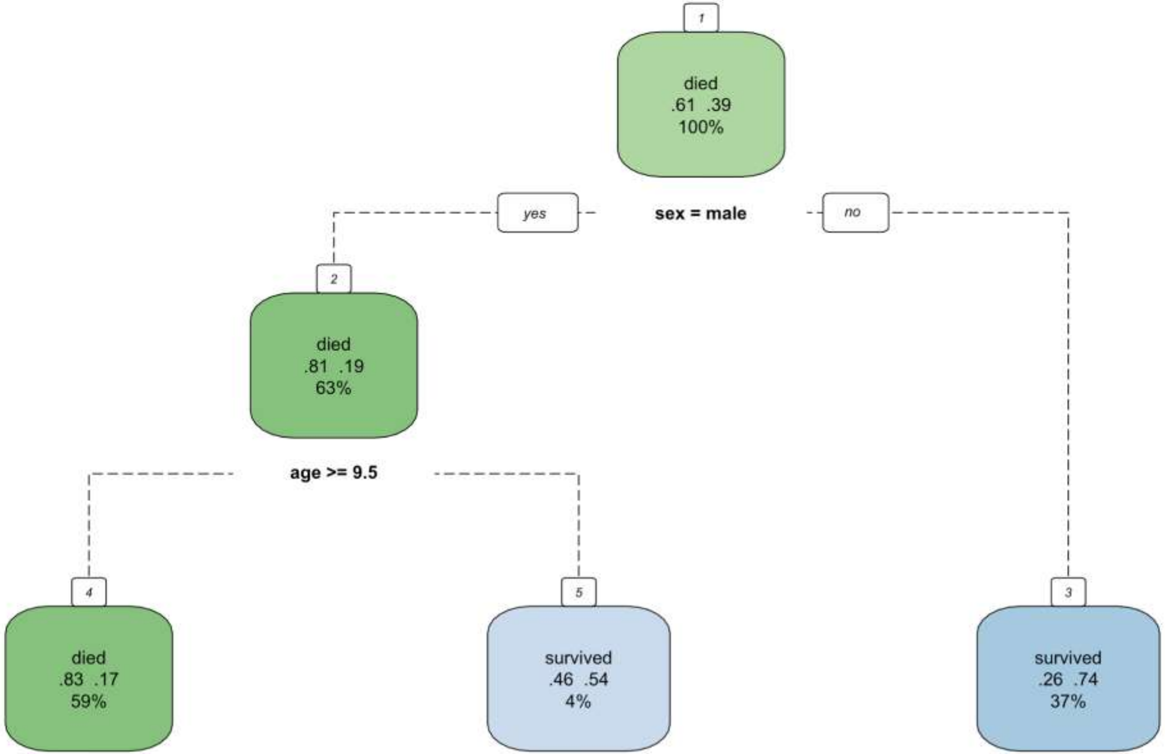
More features and correlated features make it difficult to understand



(d) Clear, eight-feature condition (CLEAR-8)

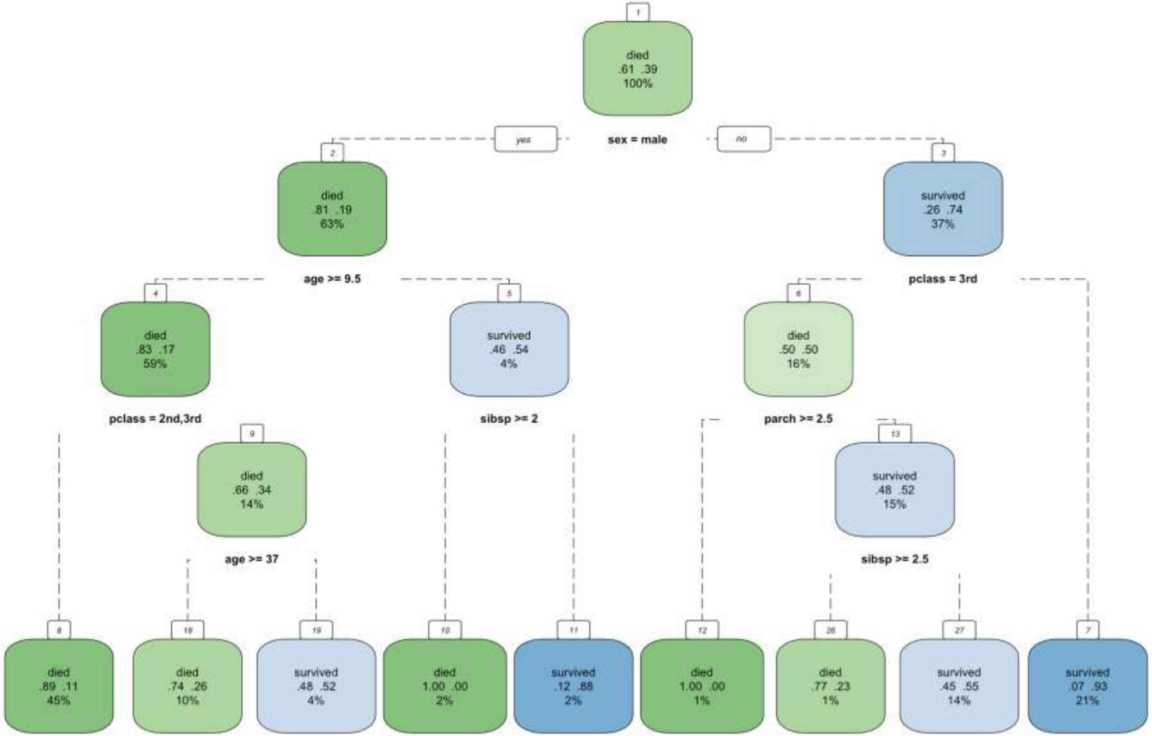
Source: Poursabzi-Sangdeh 2017

An Understandable White Box Model



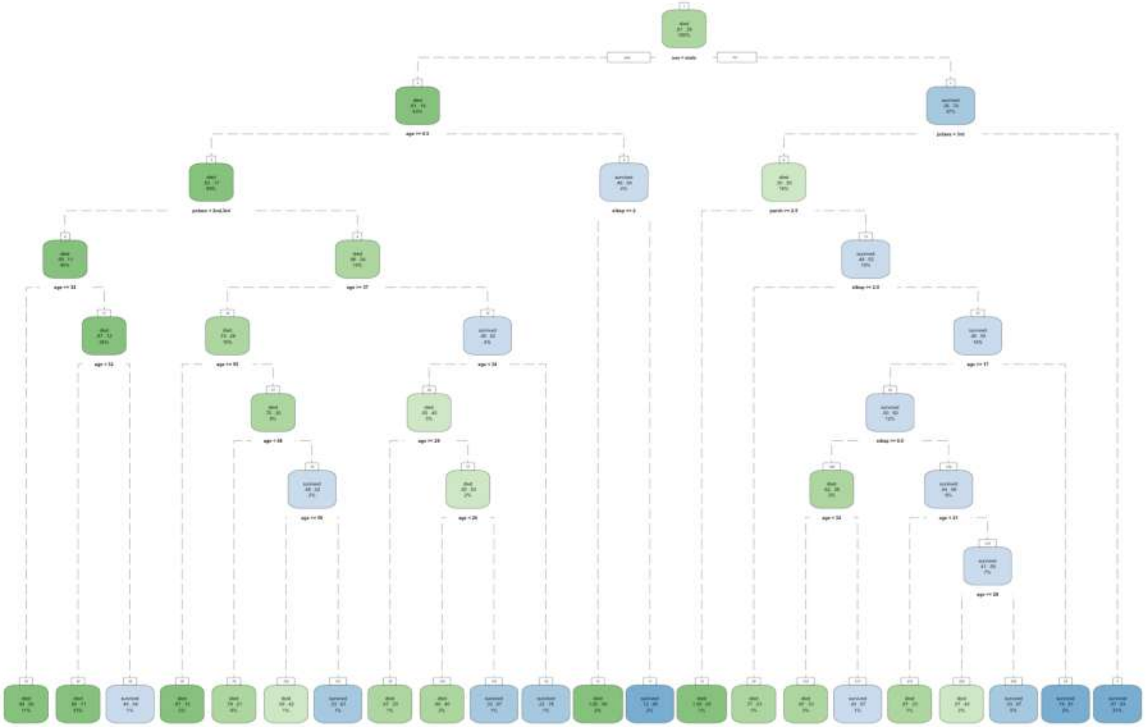
AUC = 0.74

An Understandable White Box Model?



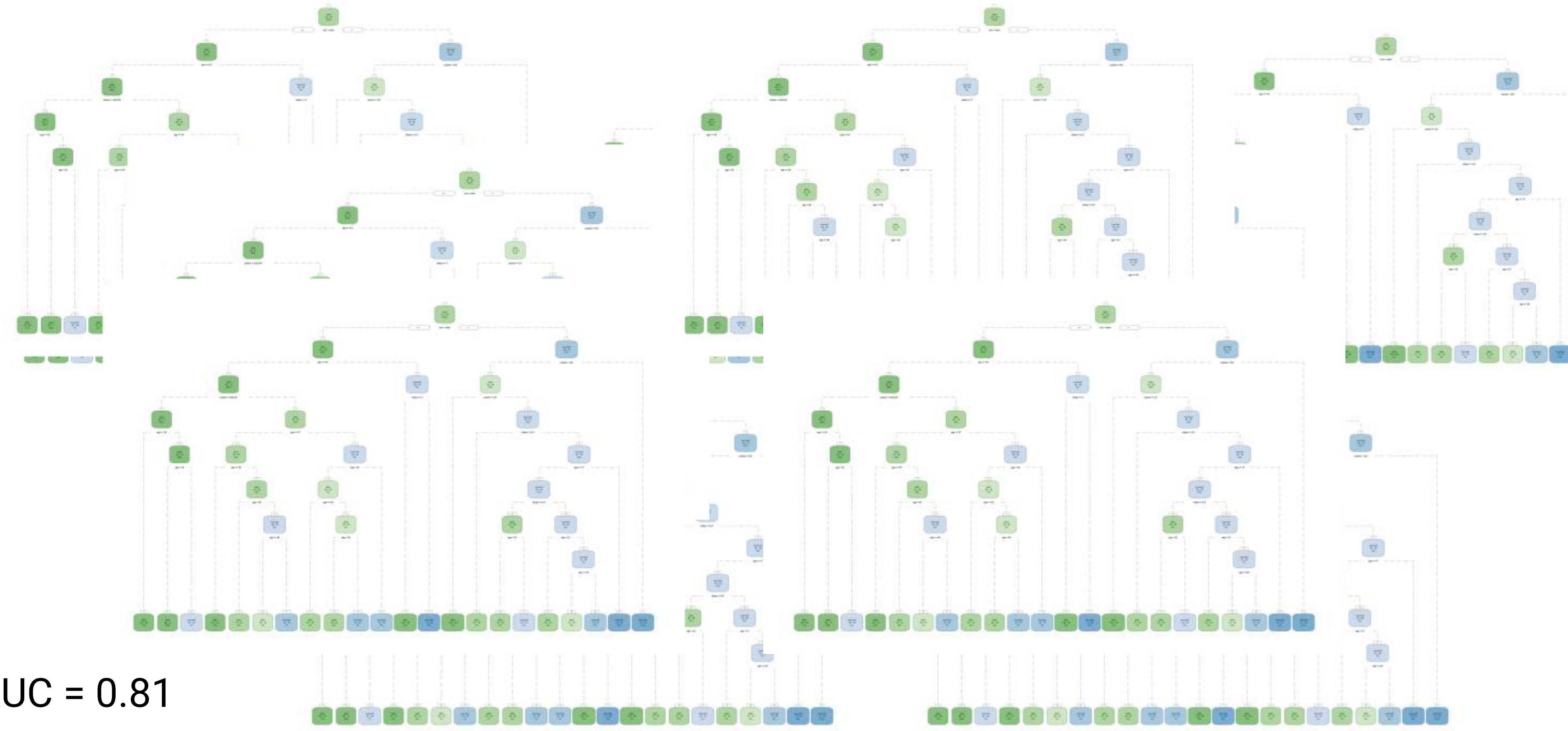
AUC = 0.78

An Understandable White Box Model? #$@&%*!



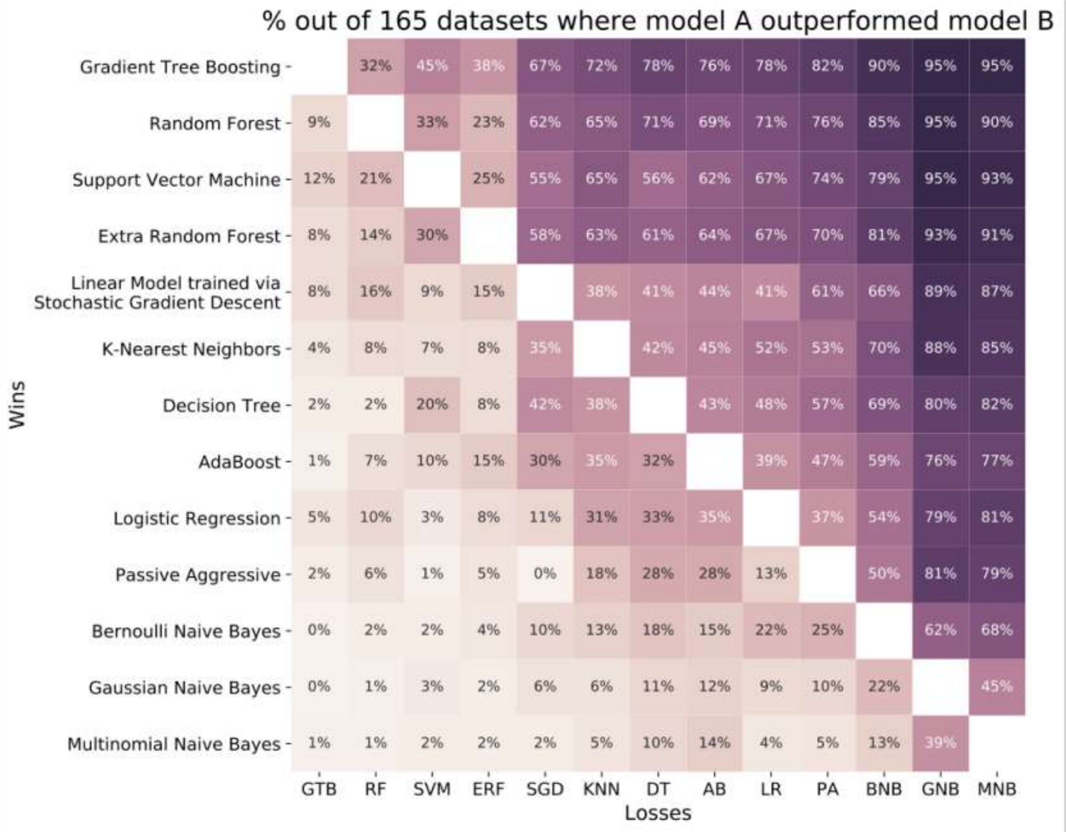
AUC = 0.79

Better Performance but too much to Comprehend



AUC = 0.81

There are so many algorithms to try



Source: [Olson 2018](#)
[Penn ML Benchmarks](#)

Algorithms matter

If the model is inaccurate,
we are toast

Simple models != Accurate

Only very very simple models are human understandable

Further study:

- Interpretability in models with multicollinearity: Brieman
- Limits of human understanding: Poursabzi
- Simple models are unfair: Kleinberg
- In defense of the black box: Holm

There are tools that can explain **any** black box model

Model Agnostic Explanation Tools

Most impactful features - Feature importance

Directionality of the feature - Partial dependence

Explain a prediction - Explanation techniques (LIME, XEMP, SHAP . . .)

Feature Importance

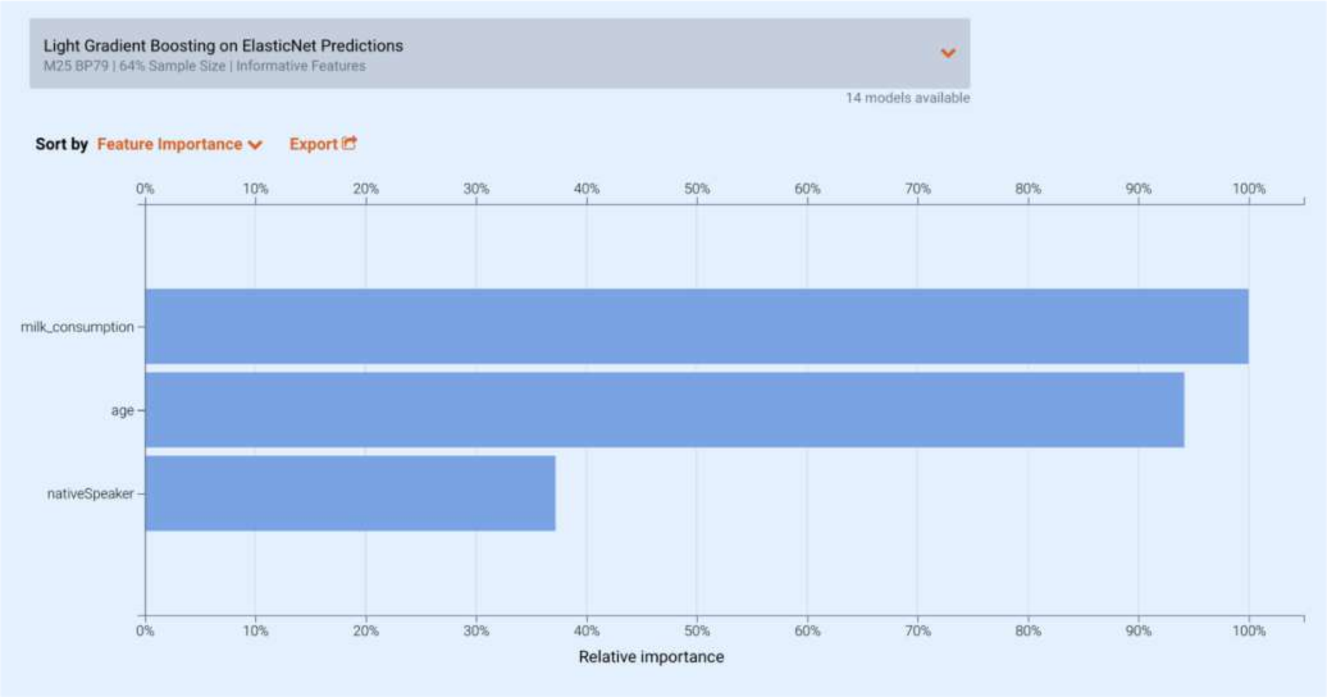
Age
Weight
Gender
Color
Breath Fire
# of Kills
Winged
# of Heads
Spiked tail
Demeanor
Children

WHAT AFFECTS READING?

AGE (IT DOES)

MILK CONSUMPTION (IT DOESN'T)

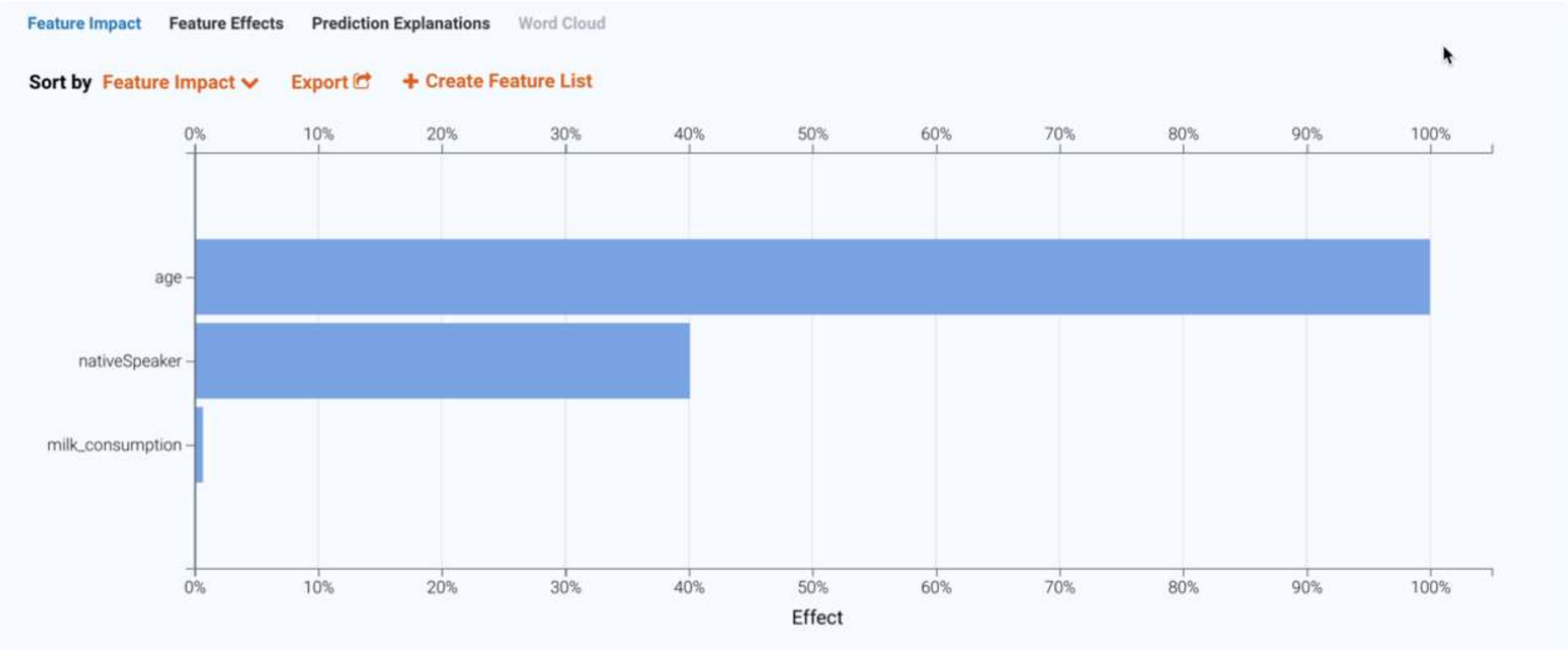
Split Based Variable Importance



Split Falls for Milk Consumption

Source: [Strobl 2009]

Permutation Based Variable Importance



Permutation Recognizes that Age Affects Reading

Source: [Strobl 2009]

Feature Impact Ranking:

1. # of Kills
2. # of Heads
3. Children
4. Age
5. Weight
6. Demeanor
7. Gender
8. Breath Fire
9. Color
10. Spiked tail
11. Winged

Feature impact has consequences, so you better get it right

If your feature impact is wrong, you are toast.

Feature Importance

Model AB
A & B
R^2=0.9

Model A
A
R^2=0.7

Model B
B
R^2=0.8

Build 3 different models based on different sets of features

FEATURE B IS MORE IMPORTANT TO THE MODEL

Ablation Methodology

Compare performance with and without the features

‘Leave it Out’ Feature Importance

Model ABC
A & B &C
R^2=0.9

Model AB
AB
R^2=0.7

Model BC
BC
R^2=0.8

Model AC
AC
R^2=0.75

Build 4 different models based on ‘Leave it Out’ importance

FEATURE C IS MORE IMPORTANT TO THE MODEL

Permutation based Feature Importance

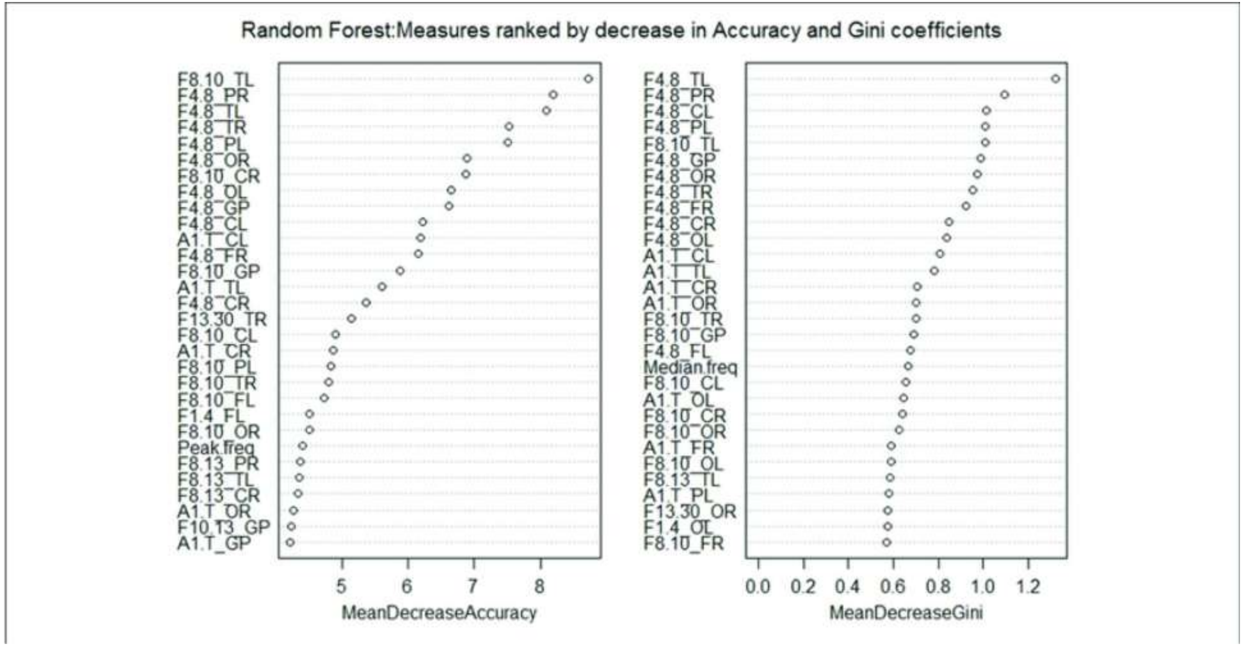
Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...	...	...	...
156	142	...	8
153	130	...	24

Shuffle the feature (permute) which
removes the signal within the same model

Source: Breiman 2001

R

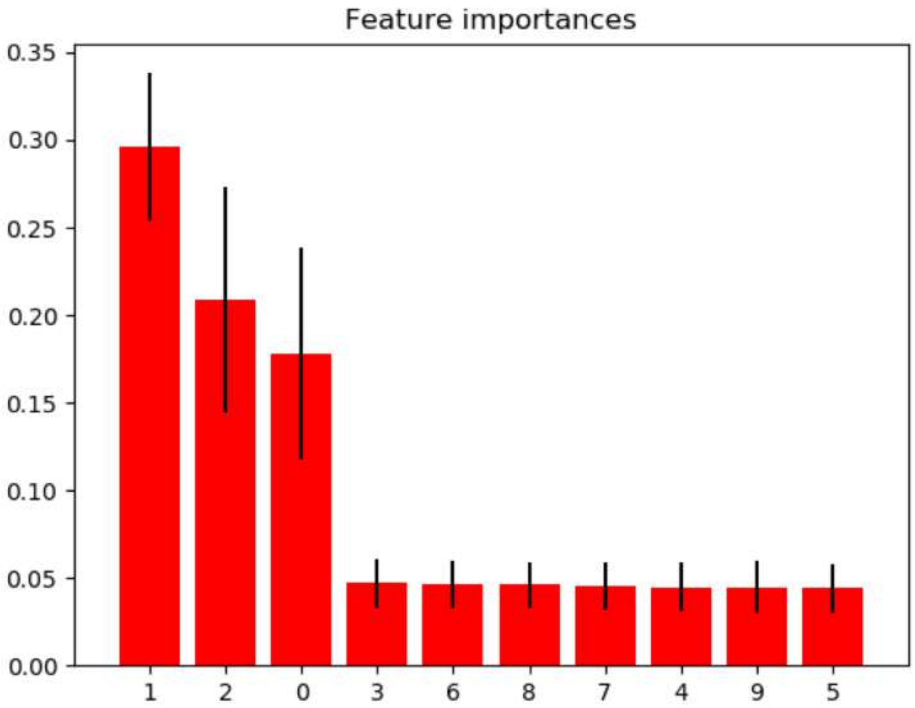
R randomforest shows both permutation and gini based importance



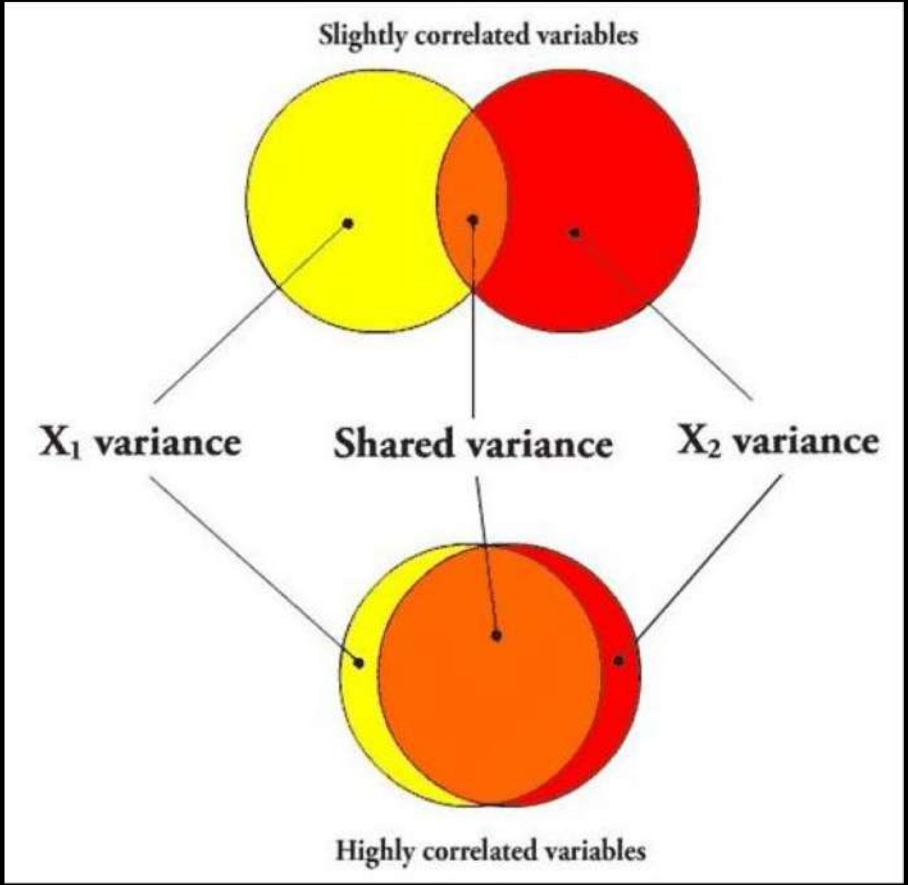
YEA, R SUPPORTS PERMUTATION!

Python

Python sklearn only uses gini for feature importance . . . go find ELI5

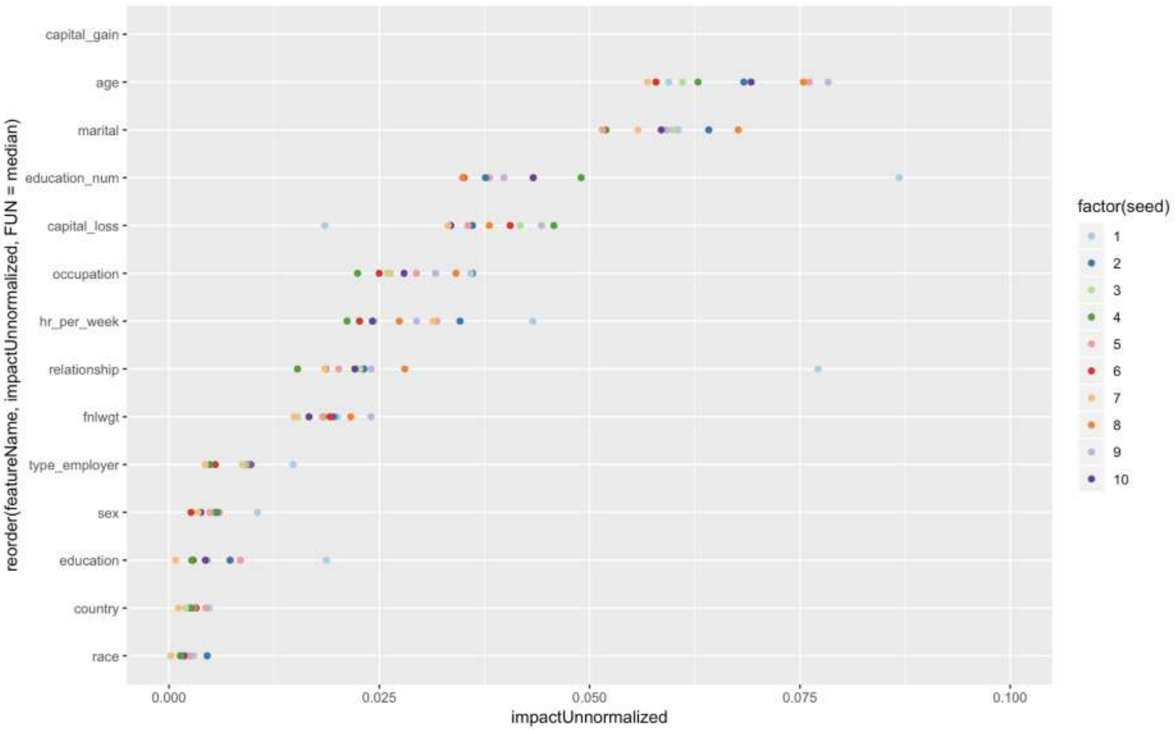


Boo!, Python Does not support Permutation!



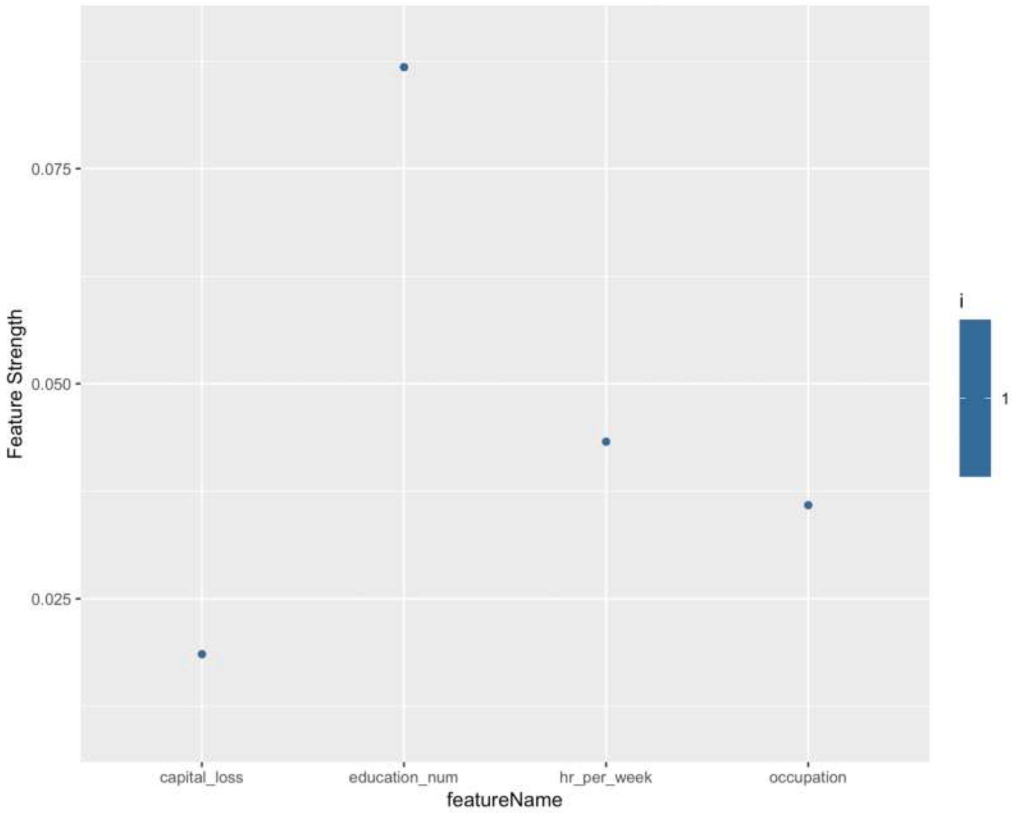
Multicollinearity

Run It Again



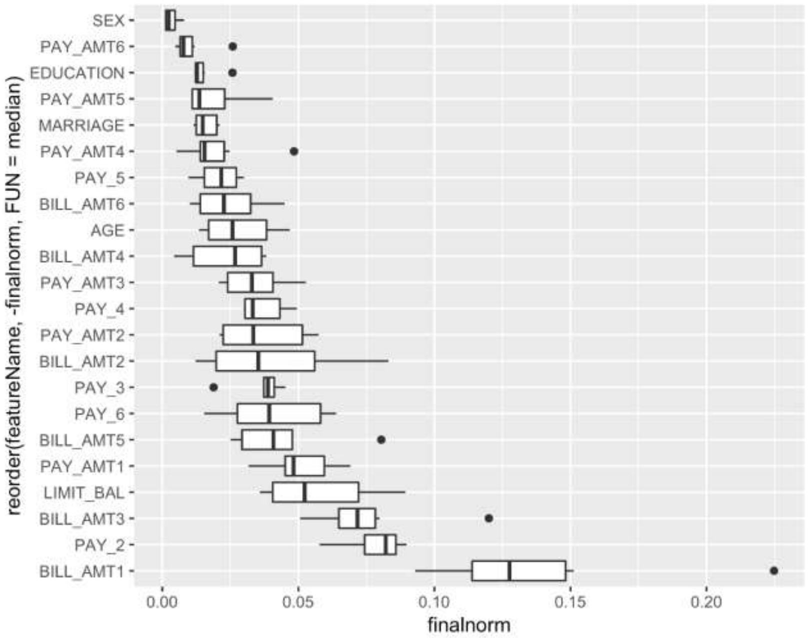
10 different models, 10 different feature importances

Multicollinearity affects Interpreting models



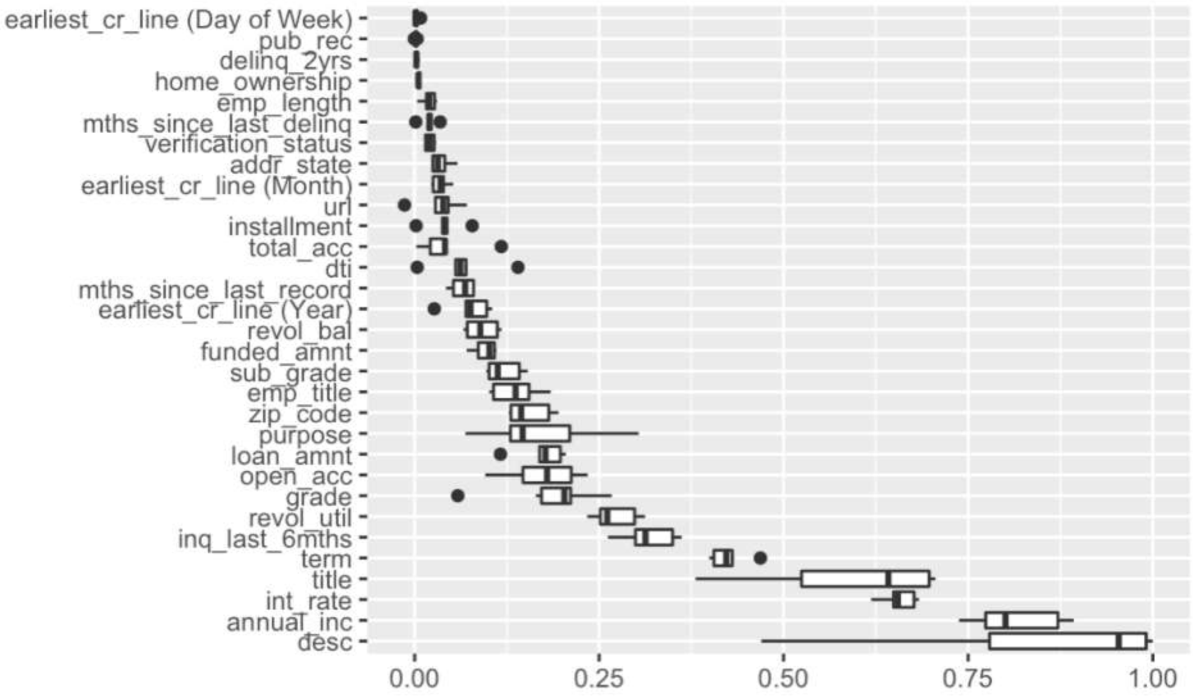
Features trade off against each other in different model runs

Pro Tip: Aggregate Feature Importance to Provide a Richer Understanding



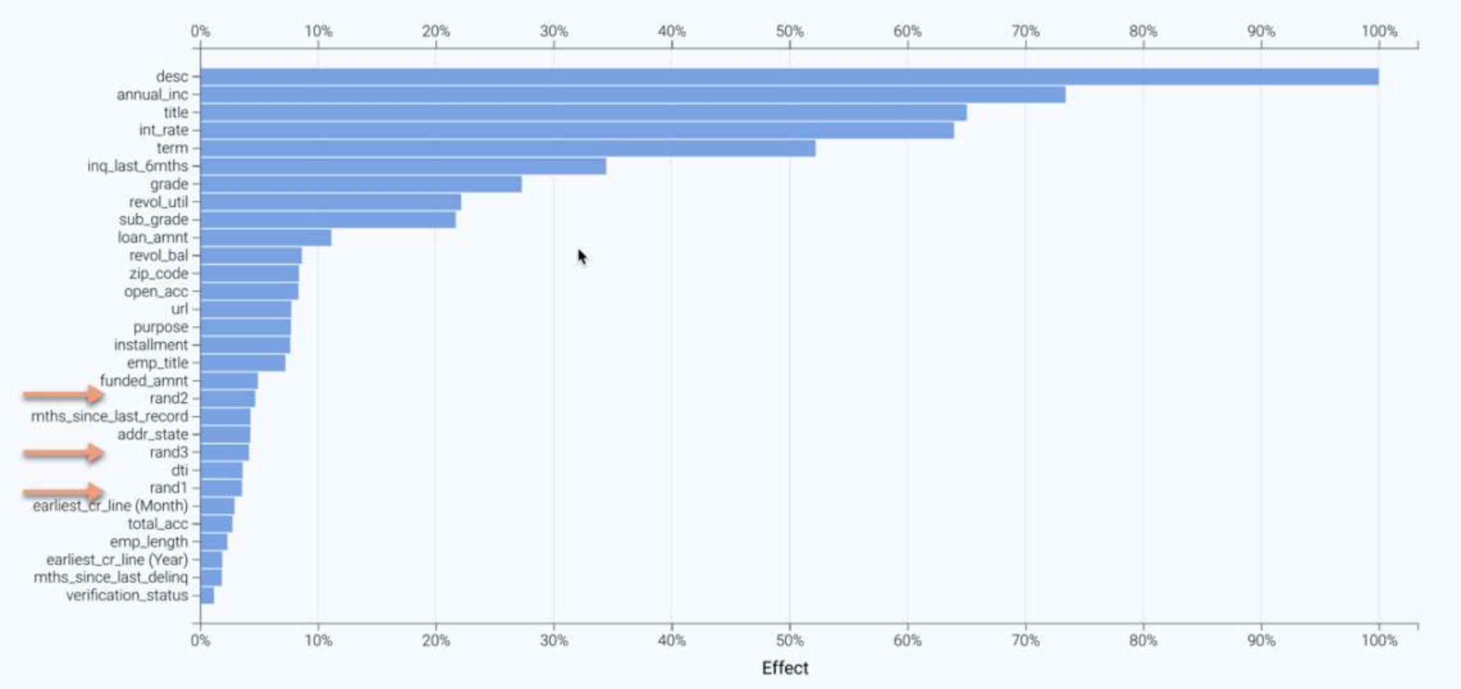
This plots show how the ranking of feature importance varies across multiple model runs of the same model

Pro Tip: Aggregate Feature Importance to Provide a Richer Understanding



This plots show how the ranking of feature importance varies across multiple model runs of different models

Pro Tips: Add Random Features



Helps you understand the line between signal and noise

Permutation based importance is a good balance of computation and performance for any model

Further study:

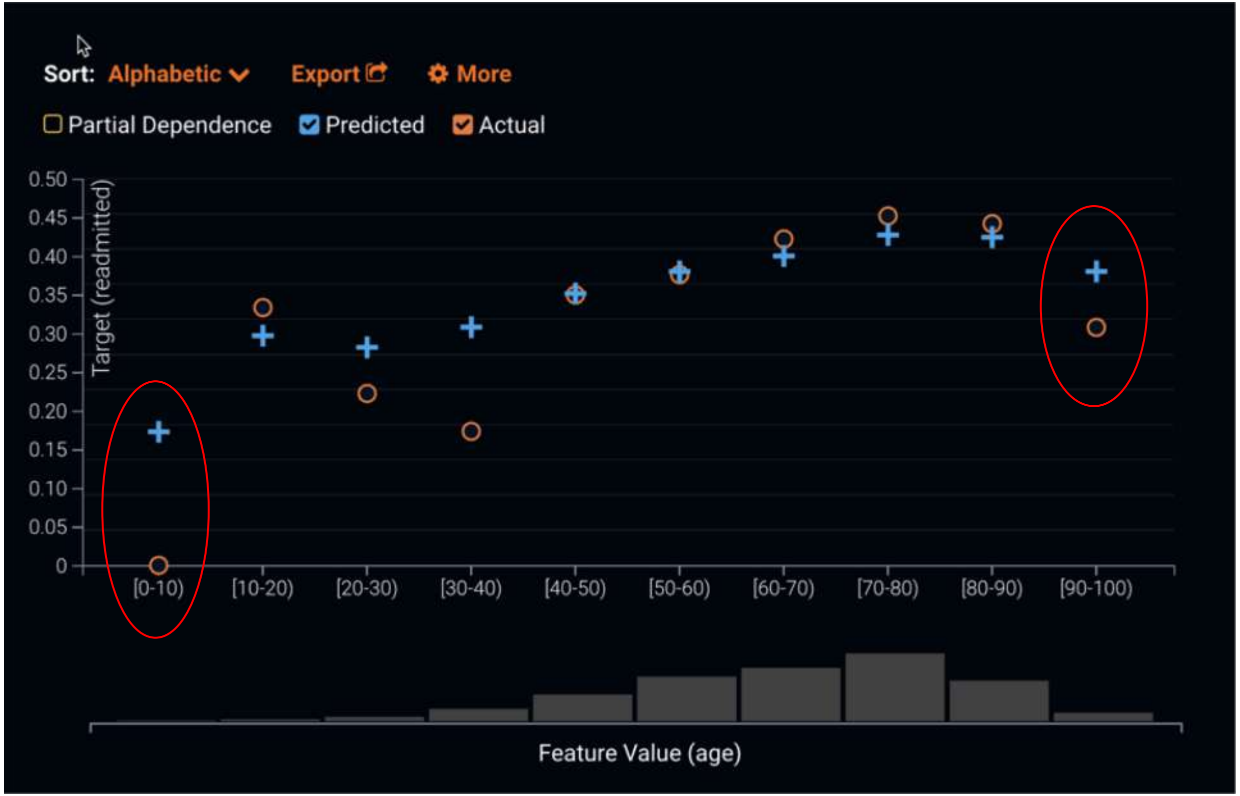
Studies on permutation based importance: Strobl 2008 and Lundberg 2018 and explained.ai and datadive . . . more advanced approaches - Party, Shap, and Boruta

Partial Dependence

Age 

Weight 

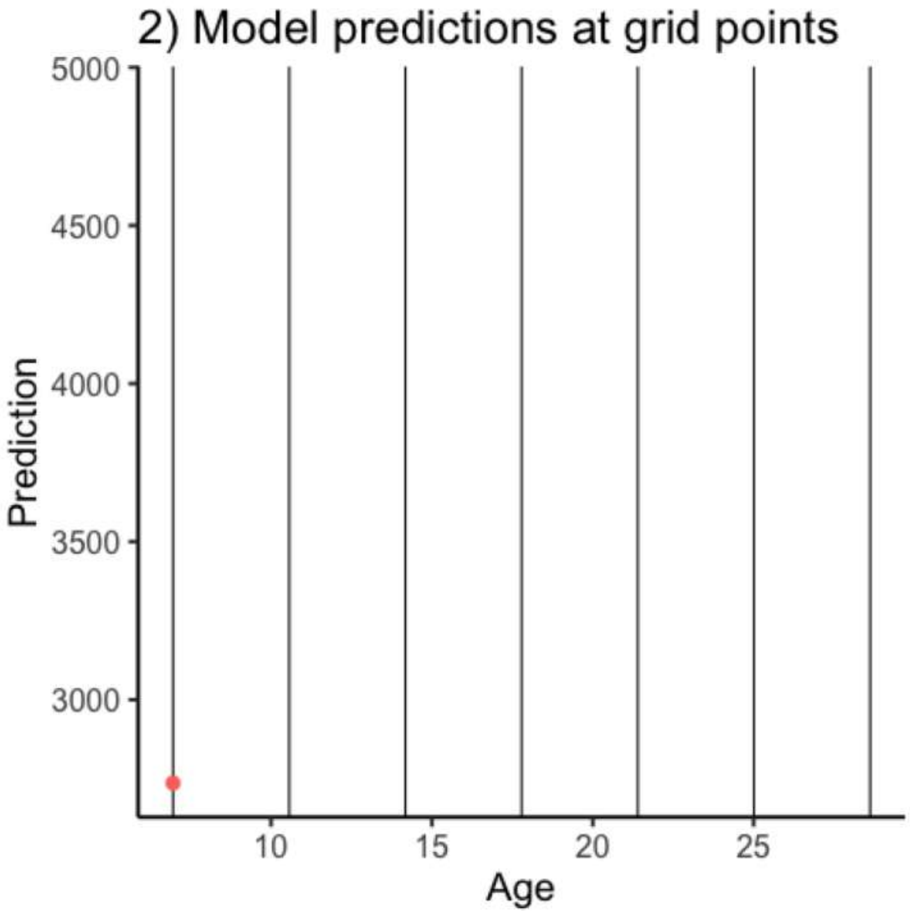
Effect of Age on our Target



What is the average weight for each of these bins?

THIS PLOT DOES NOT ISOLATE THE EFFECT OF AGE

Calculating Partial Dependence



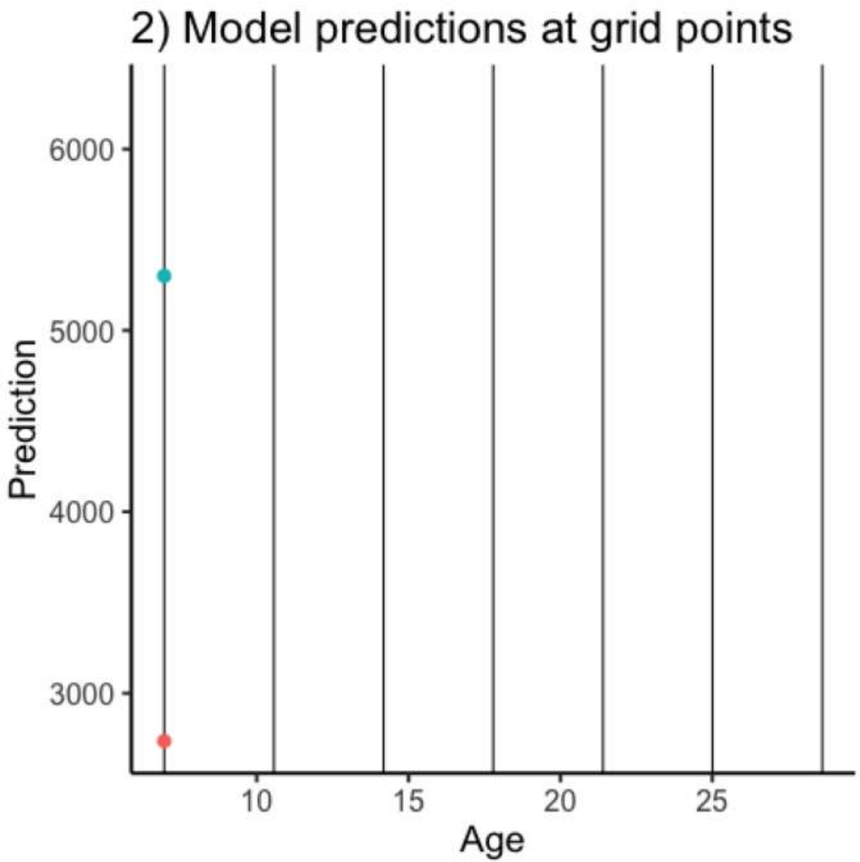
Start with an observation and get predictions for different values

Source: [Christoph Molnar]

Calculating Partial Dependence

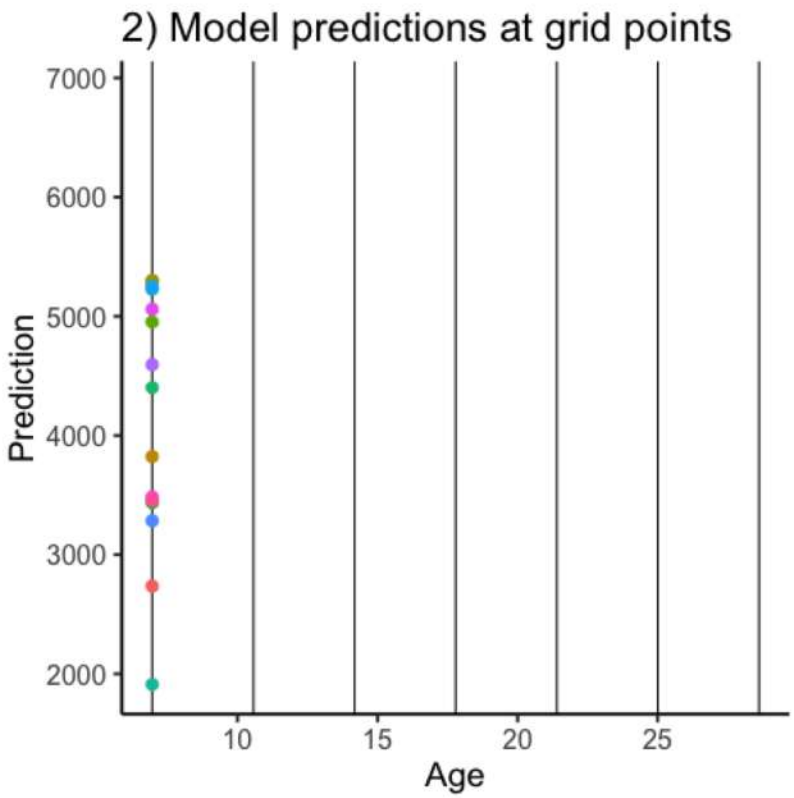
Start with another observation and get predictions for different values

Source: [Christoph Molnar]



How Partial Dependence is Calculated

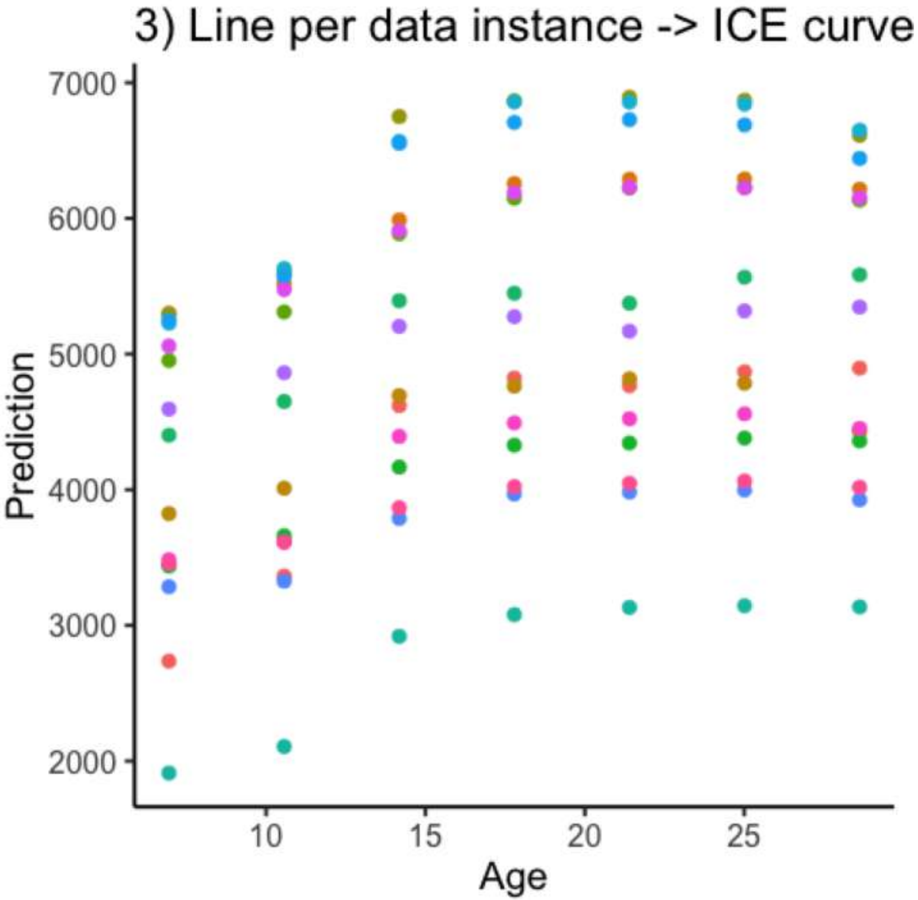
Start with a set of observations from our dataset



Source: Christoph Molnar

How Partial Dependence is Calculated

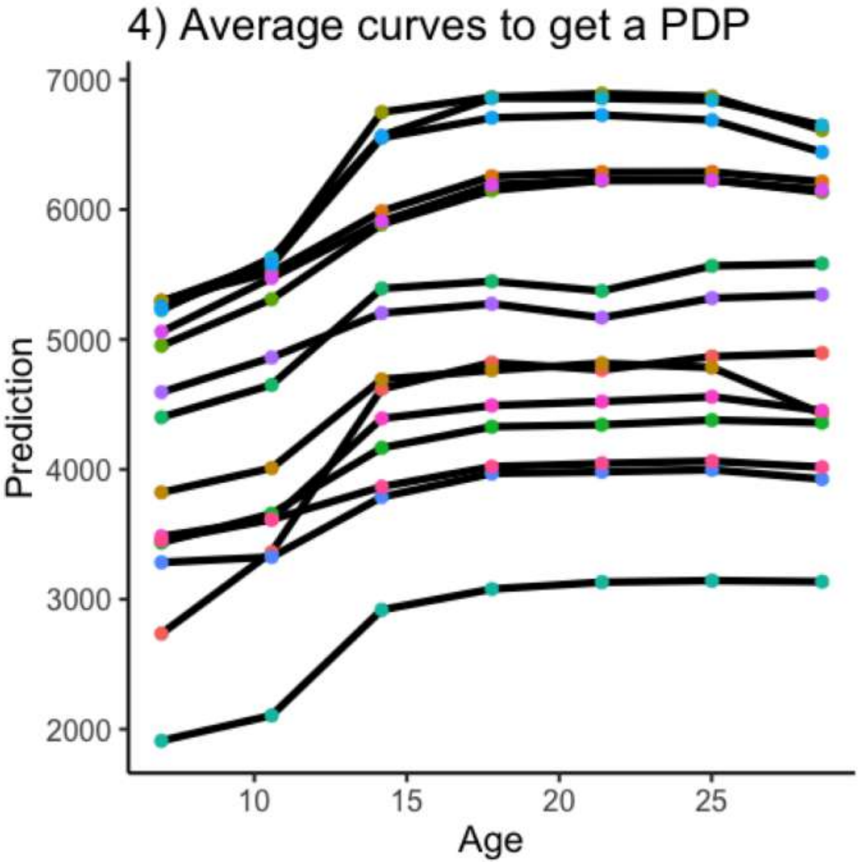
Get a line per instance



Source: [Christoph Molnar]

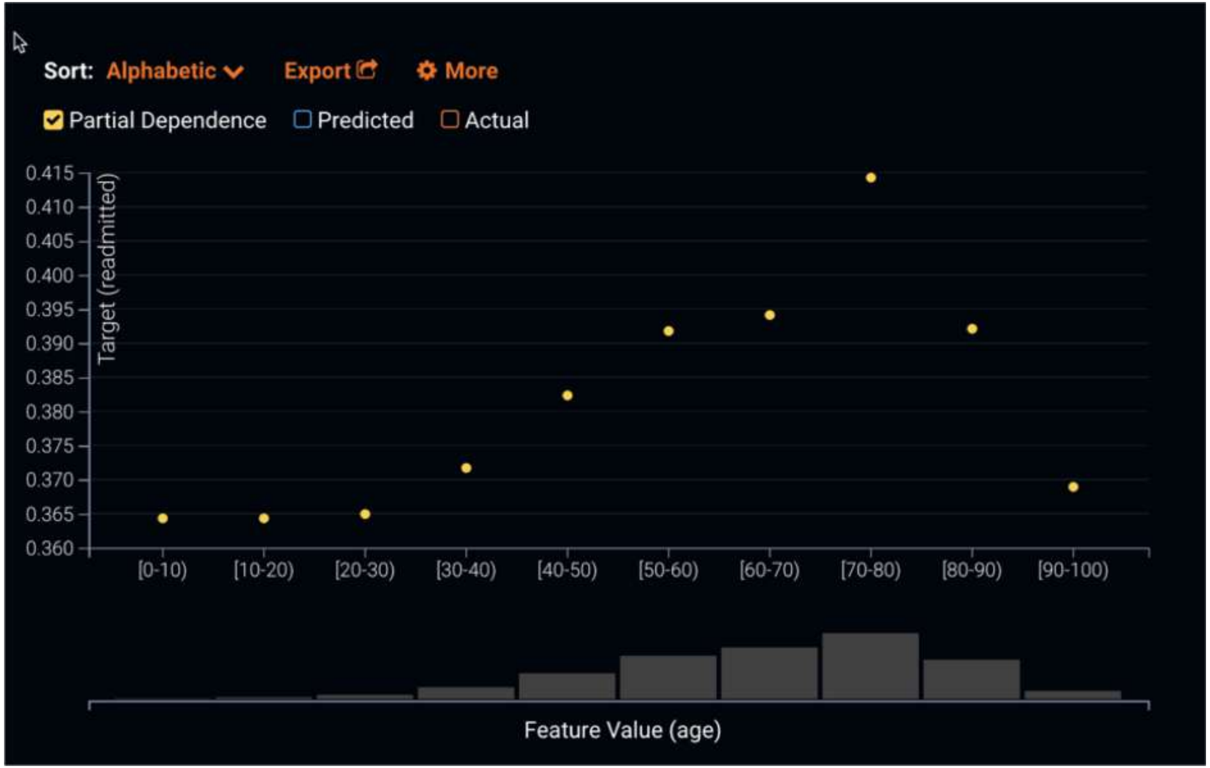
How Partial Dependence is Calculated

Average the curves to get the partial dependence curve

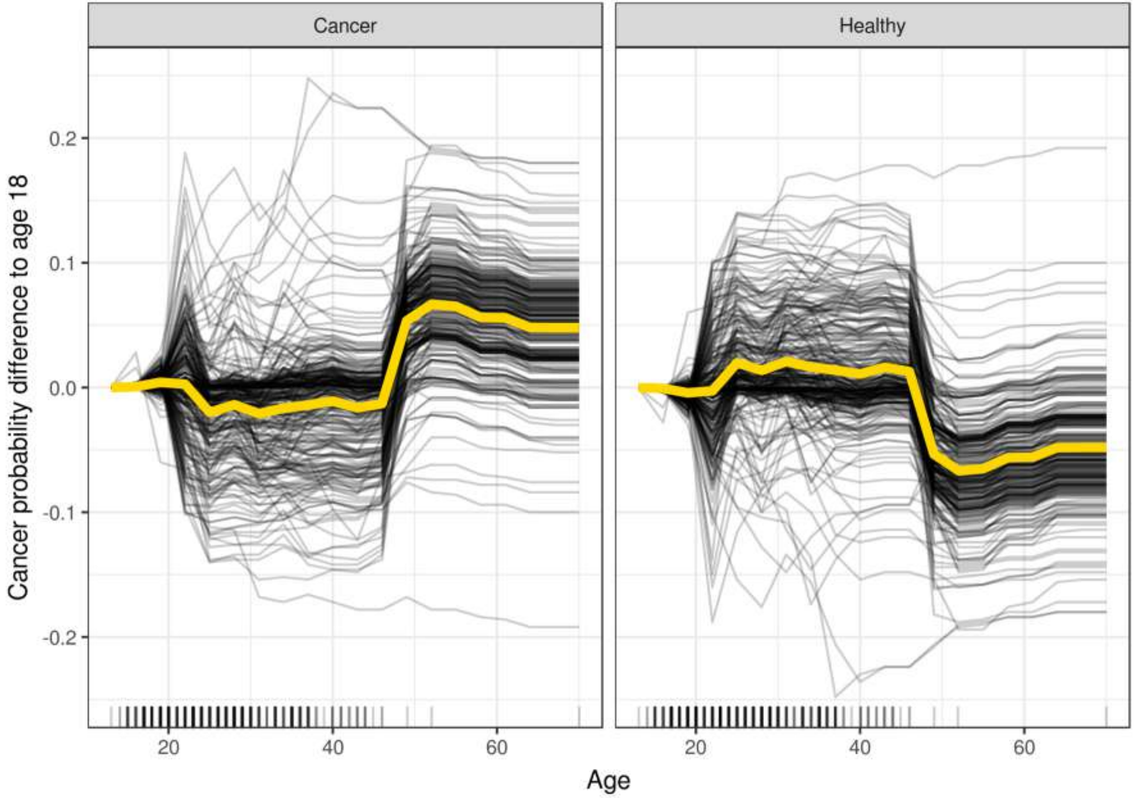


Source: [Christoph Molnar]()

Partial Dependence to Isolate the Effect of Age



ICE Plots



Individual Conditional Expectation plots draw one line per instance

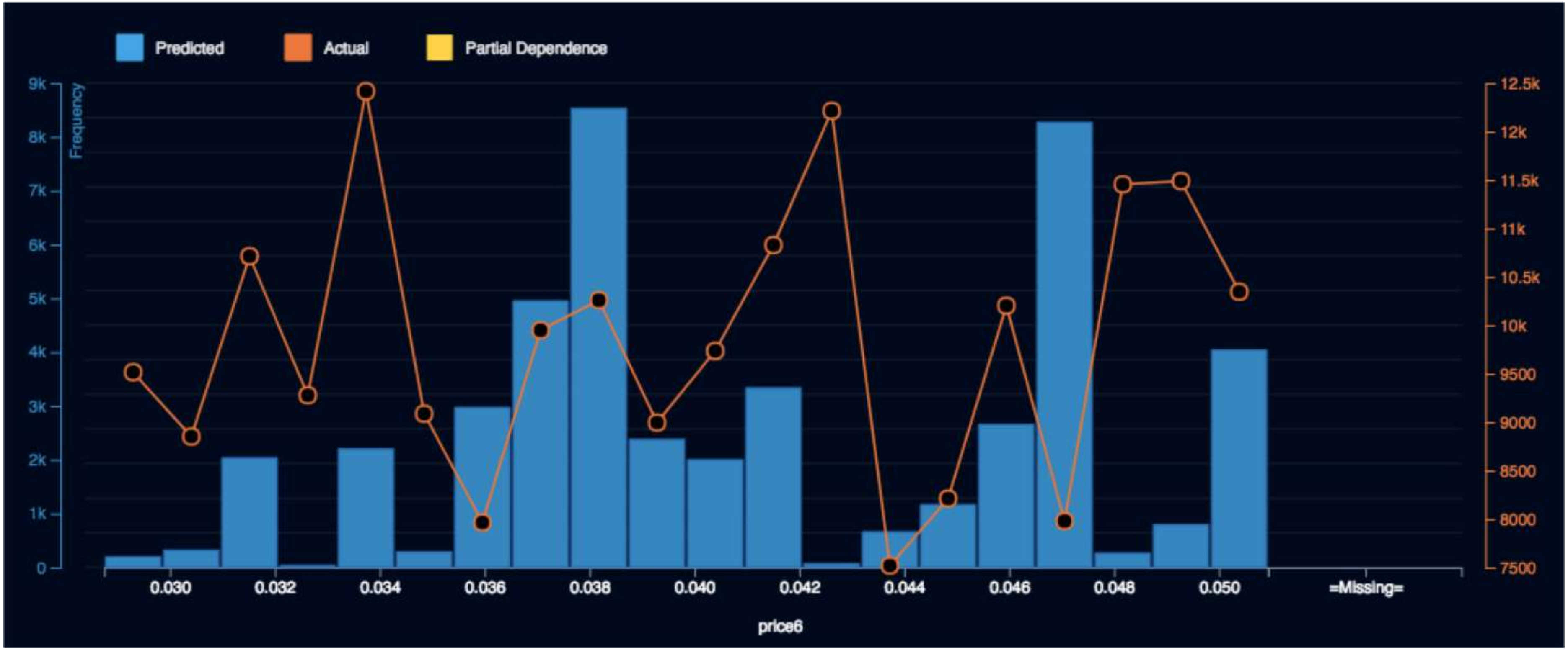
Partial Dependence to show Price Elasticity

Effect of price on sales of orange juice

Features include:

- store location
- date
- coupons
- advertising
- prices for 10 other brands

Change in Price Affects Sales?



Ahh, Price does affect Sales!

Partial dependence is a best practice for understanding the features in your model

Further study:

Friedman, 2001 on PDP

Goldstein, 2013 on ICE Plots

Predictions

Prediction: 9.1

Explanations:
1. # of Past Kills (+0.8)
2. Color (+0.3)
3. Gender (-0.2)

Predictions & Explanations

Charge Nurse

Nurse Susan

Floor

1st

3

Urgent Risk Patients

2

High Risk Patients

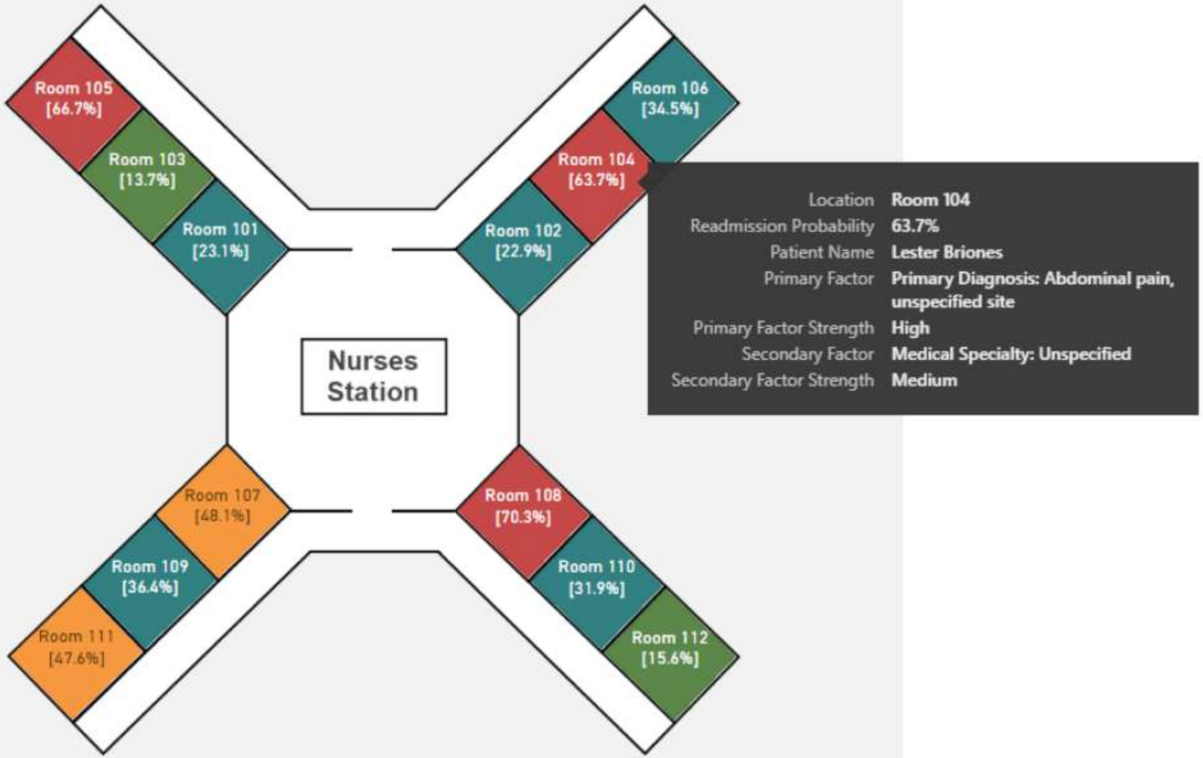
5

Moderate Risk Patients

2

Low Risk Patients

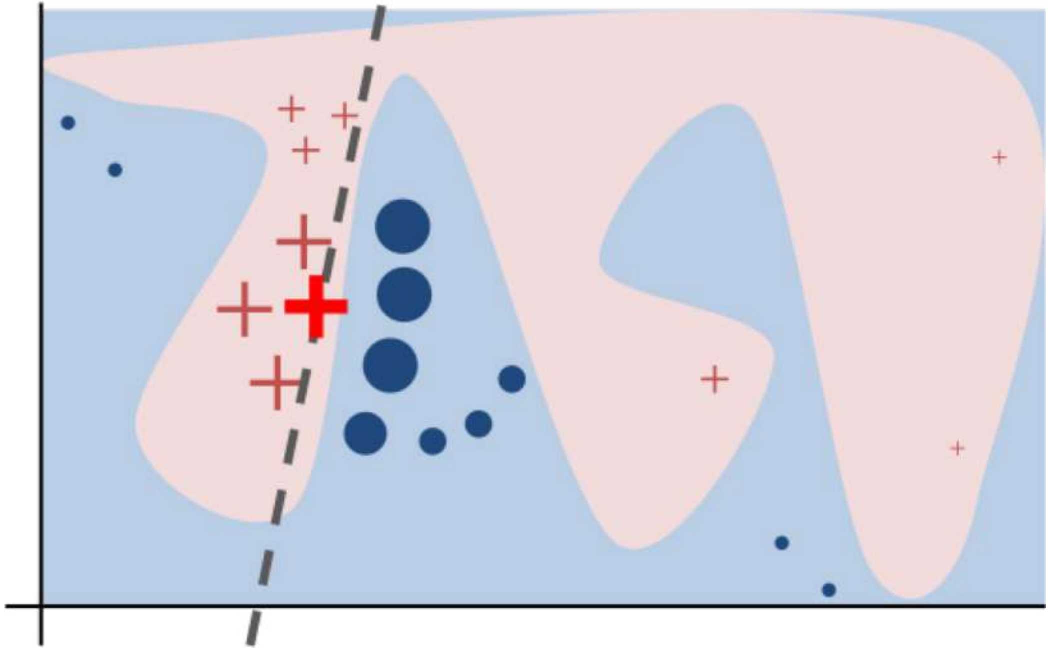
Floor Map with Readmission Probability by Room



Explanation Methods:
Local Interpretable Model-Agnostic Explanations (LIME)

For any prediction:

LIME gives you an ordered list of the most important features for that prediction



SPEND SOME TIME WITH LIME

Prediction: 9.1

Explanation (1)
1. # of Past Kills (+0.8)
2. Color (+0.4)
3. Gender (-0.2)

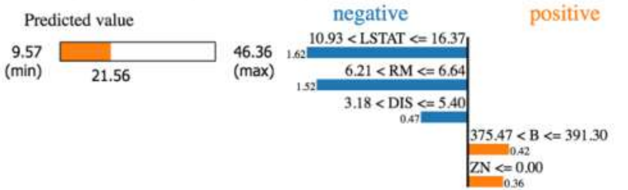
Explanation (2)
1. Gender (+0.5)
2. Breath Fire (+0.3)
3. Weight (+0.1)

Explanations should be Identical

Identical Explanations

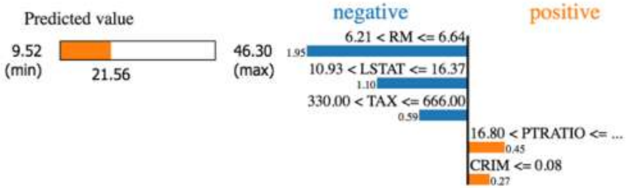
```
In [56]: i = 13
         exp = explainer.explain_instance(test[i], rf.predict, num_features=5)
         print (exp)
         exp.show_in_notebook(show_table=True)
         i = 13
         exp = explainer.explain_instance(test[i], rf.predict, num_features=5)
         print (exp)
         exp.show_in_notebook(show_table=True)
```

```
Intercept 24.51223189433816
Prediction_local [21.66593303]
Right: 21.558500000000006
<lime.explanation.Explanation object at 0x1a2acd6860>
```



Feature	Value
LSTAT	12.34
RM	6.23
DIS	5.01
B	386.40
ZN	0.00

```
Intercept 24.84927186407495
Prediction_local [21.91763542]
Right: 21.558500000000006
<lime.explanation.Explanation object at 0x1a1c040c50>
```



Feature	Value
RM	6.23
LSTAT	12.34
TAX	398.00
PTRATIO	18.70
CRIM	0.06

Same Data, Same Model . . . Two Different Explanations!!

Prediction: 9.1

Explanations:
1. # of Past Kills (+0.8)
2. Color (+0.4)
3. Gender (-0.2)

Prediction: 2.4

Explanations:
1. # of Past Kills (+0.8)
2. Color (+0.4)
3. Gender (-0.2)

Prediction: 8.3

Explanations:
1. # of Past Kill (+0.8)
2. Color (+0.4)
3. Gender (-0.2)

EXPLANATIONS SHOULD HAVE FIDELITY TO THE DATA

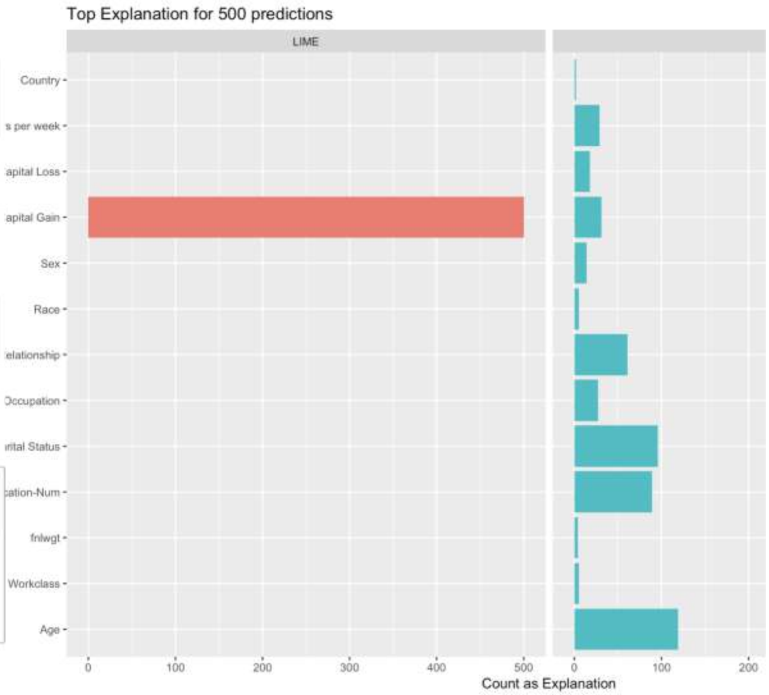
Local Fidelity

```
In [22]: np.random.seed(1)
         i = 1653
         exp = explainer.explain_instance(test[i], predict_fn, num_features=2)
         exp.show_in_notebook(show_all=False)
```

Note that capital gain has very high weight. This makes sense. Now let's see an example where the person has a capital gain below the mean:

```
In [23]: i = 59
         exp = explainer.explain_instance(test[i], predict_fn, num_features=2)
         exp.show_in_notebook(show_all=False)
```


```
In [24]: i = 26
         exp = explainer.explain_instance(test[i], predict_fn, num_features=2)
         exp.show_in_notebook(show_all=False)
```



LIME EXPLANATIONS AREN'T RESPONSIVE TO THE DATA

Anyone relying on LIME is toast

What can we learn from this?

We want consistency and accuracy . . what else . . .

Further study:

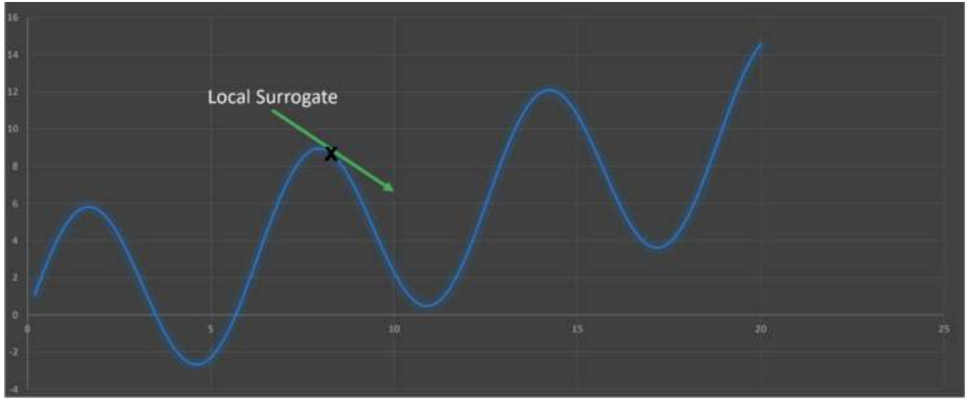
Explanations affect fairness: Dodge

Shap: Lundberg

Live and Breakdown: Biecek

Your Model or a Surrogate Model?

Imagined model



Source: Mehrnoosh Sameki, Microsoft, ODSC 2019

Real Models

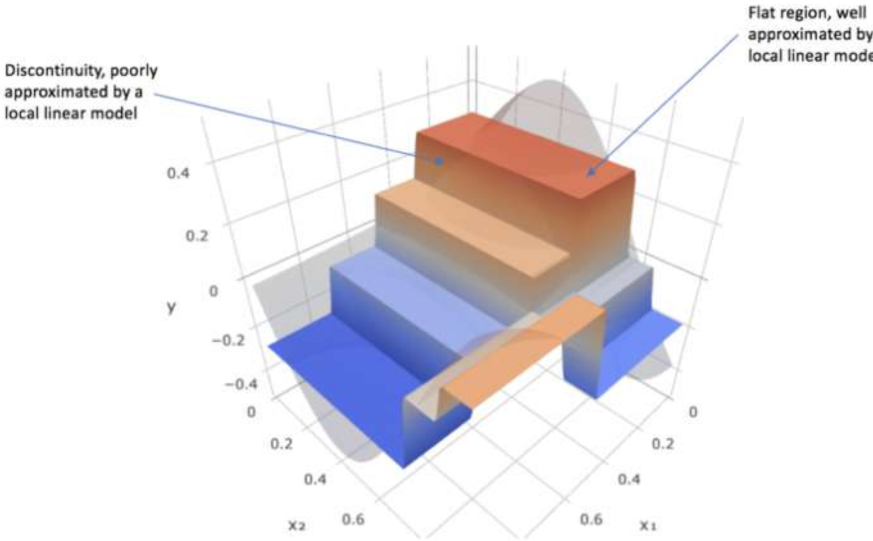
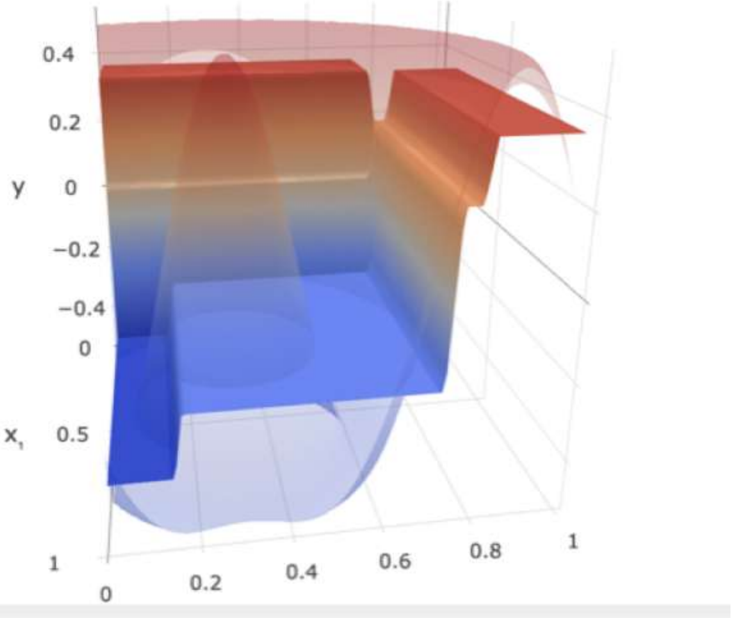


Image source: http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

SURROGATE MODELS ARE APPROXIMATIONS

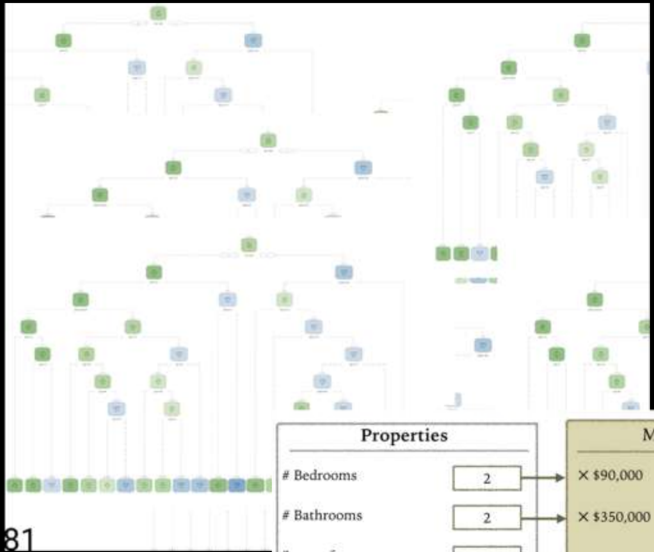
What is local?

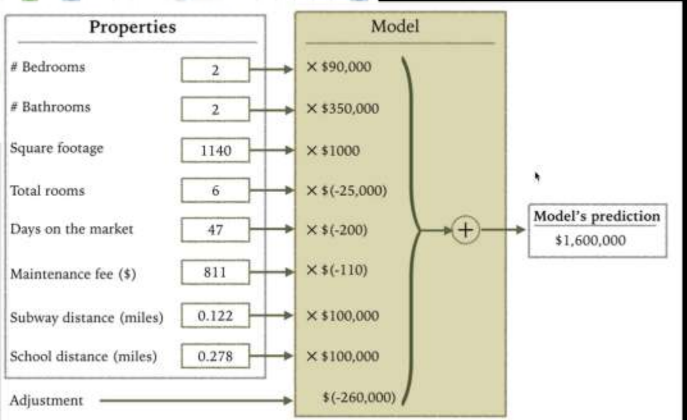


Source: Christoph Molnar

UGH!

YOU DON'T WANT HYPERPARMETERS TO DECIDE HOW TO SET





(d) Clear, eight-feature condition (CLEAR-8)

Explanation methods should be Model Agnostic

Time in seconds

	LIME	
Boston Housing (100 explanations)	43	0.3
Adult (1000 explanations)	423	3

Explanations should be Fast

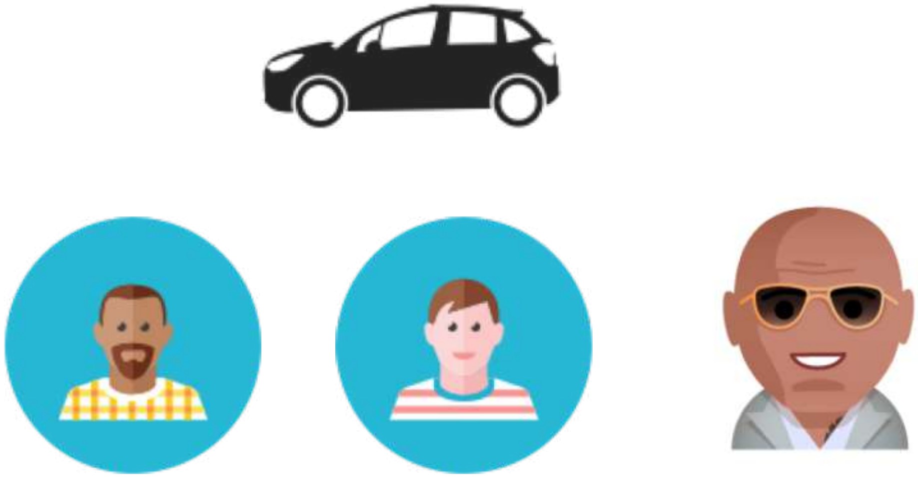
Shapley Values

Beautiful Idea from Game Theory . . . Won some Awards

Intuition of Shapley Values

My car is stuck and needs 85 units of force to move it

Here we can understand how much each person contributed to getting the car unstuck

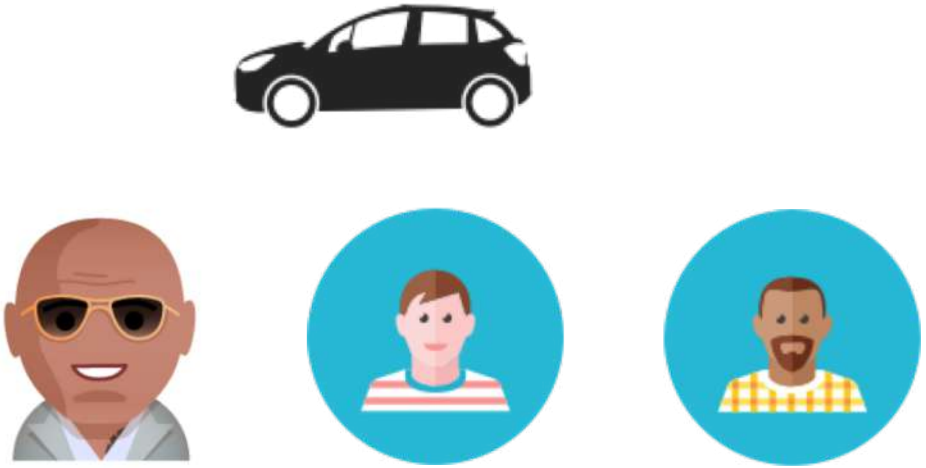


60 20 10

Intuition of Shapley Values

But I loaned my car to the Rock and he got it stuck, how much effort would it be to move it?

Note, the order matters when calculating Shapley value

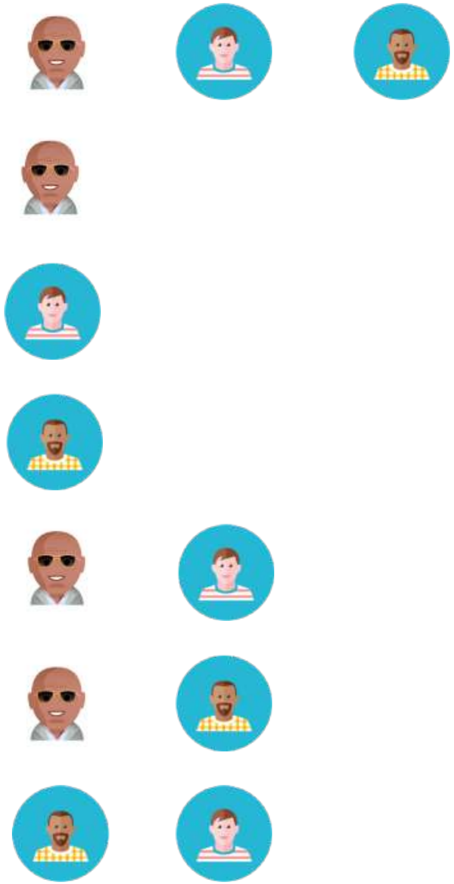


60 20 10

ON AVERAGE, WHAT WOULD BE EACH PERSON'S CONTRIBUTION

Subsets

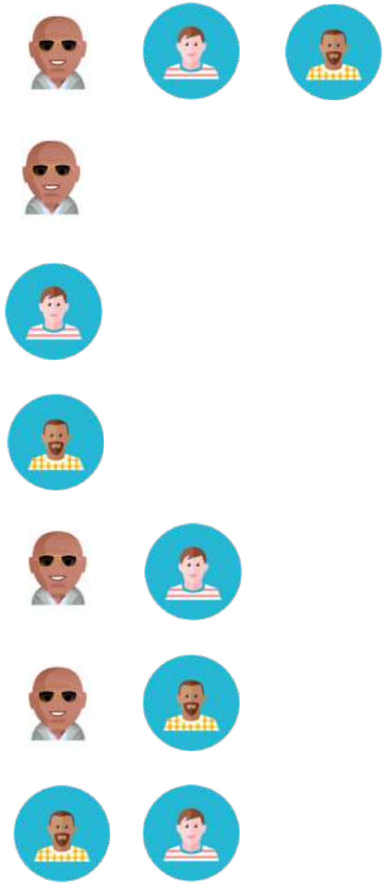
Total Force



The first step is calculating the output for all the subsets

Which combinations can move the car?

Calculating Shapley Values



90

80

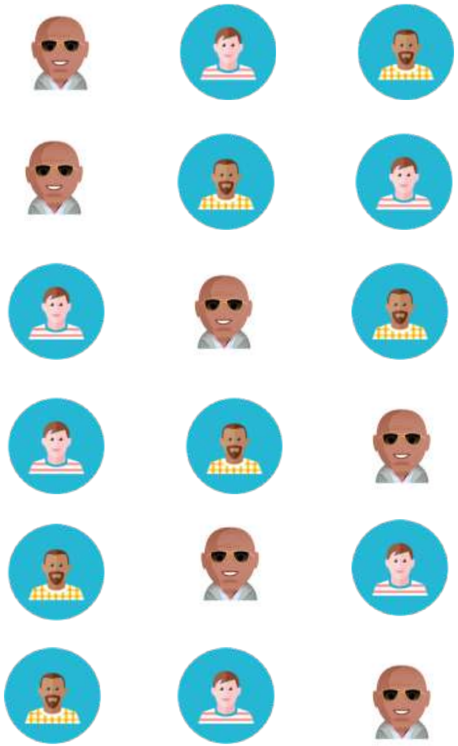
56

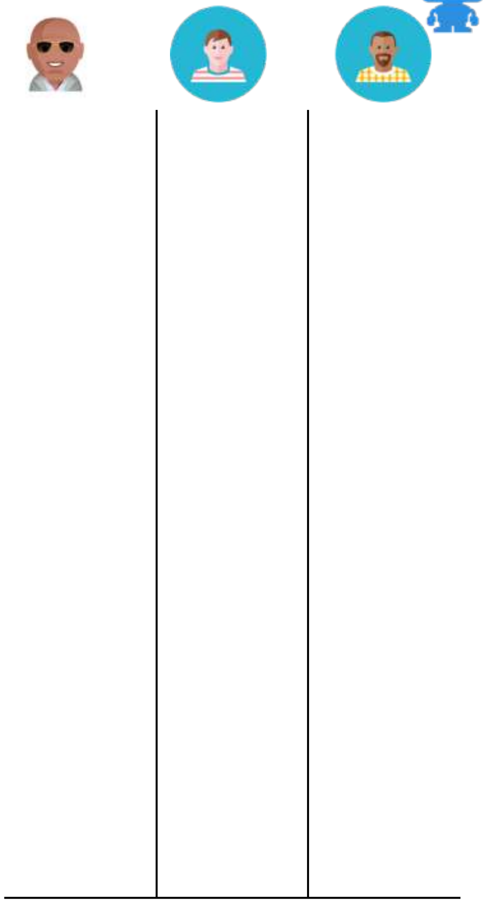
70

80

85

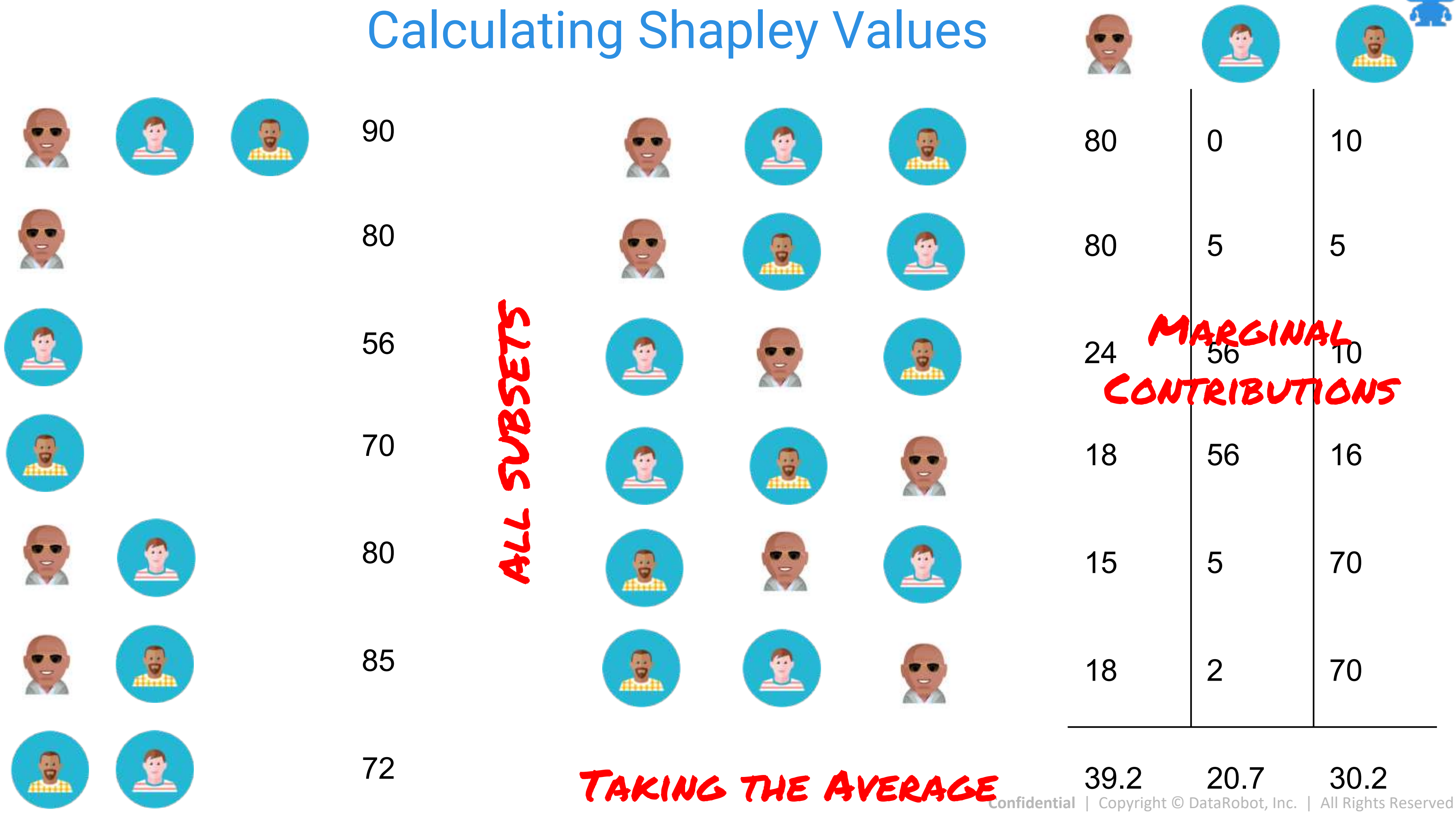
72





SHAPLEY VALUE -->

Calculating Shapley Values



Calculating Shapley Values

$$R_i = \sum_{S \subseteq P \setminus \{i\}} \frac{|S|!(|P| - |S| - 1)!}{|P|!} [f(S \cup \{i\}) - f(S)]$$

ALL SUBSETS

TAKING THE AVERAGE

MARGINAL CONTRIBUTIONS

"The **average marginal contribution** of a feature with respect to **all subsets** of other features"

Source: Marco Ancona, ICML 2019

Shapley Values for Feature Attribution

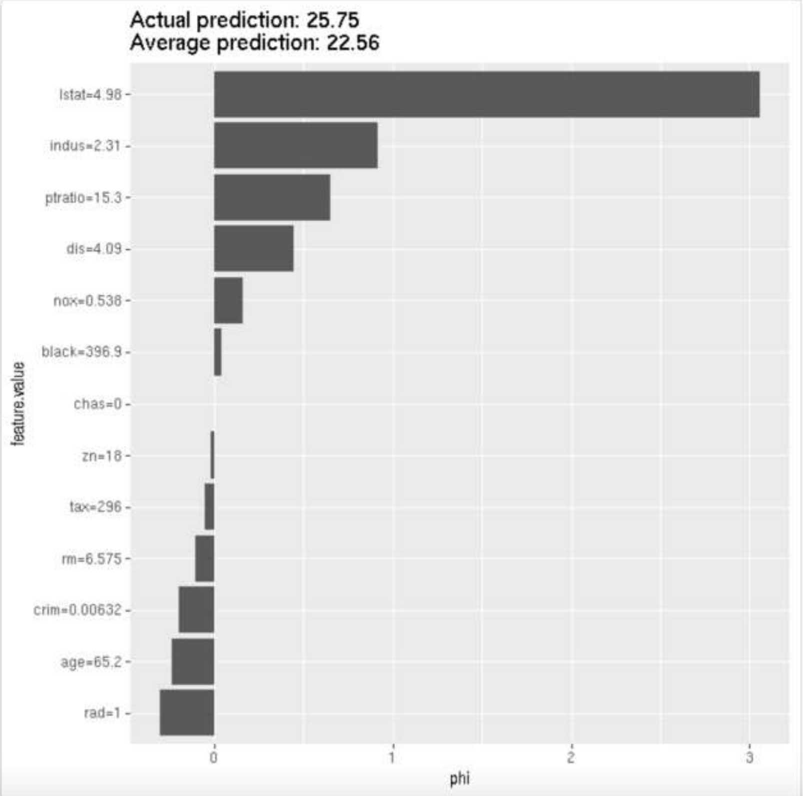
```
> X[1,]
      crim zn indus chas   nox    rm  age  dis rad tax ptratio black lstat
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.09   1 296    15.3 396.9  4.98
```

Apply the concept of Shapley Values to machine learning models.   We want to understand the contribution of each feature for a single prediction (X1)

We start with the average prediction (22.56) and then the feature contributions (their shapley values) sum up to the actual prediction (25.75)

Example is from R - IML package

```
shapley = Shapley$new(predictor, x.interest = X[1,])
shapley$plot()
```



So many methods for Shapley values: Partial List of Implementations

marcoancona / DASP

fastshap

pbiecek / breakDown

iml

- IML - R
- Shap - Python
- Shapper - R (Wrapper of Shap)
- FastShap - R
- Breakdown - R
- GkmExplain - Python
- DASP - Python
- Deep Explain - Python

marcoancona / DeepExplain

kundajelab / gkmexplain

Shap

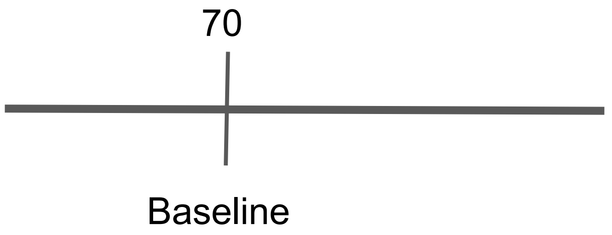
Calculating Shapley Values - Linear Model

Start with a simple linear model; Y = 0.5*weight + 2*age

What are the shapley values for the below example?

The baseline is:

Average weight is 40 and the average age is 25



.

Weight	Age	Y	Shap (Weight)	Shap (Age)
40	25	70		

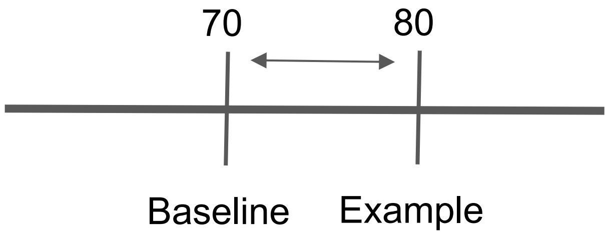
Calculating Shapley Values - Linear Model

Start with a simple linear model; Y = 0.5*weight + 2*age

What are the shapley values for the below example?

The baseline is:

Average weight is 40 and the average age is 25



.

Weight	Age	Y	Shap (Weight)	Shap (Age)
40	30	70		

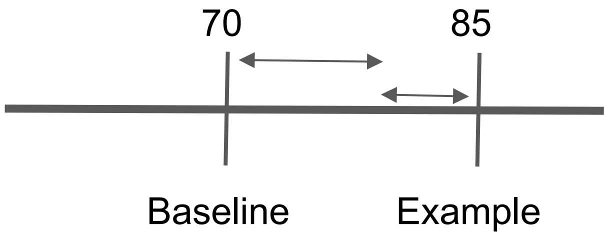
Calculating Shapley Values - Linear Model

Start with a simple linear model; Y = 0.5*weight + 2*age

What are the shapley values for the below example?

The baseline is:

Average weight is 40 and the average age is 25



.

Weight	Age	Y	Shap (Weight)	Shap (Age)
50	30	85		

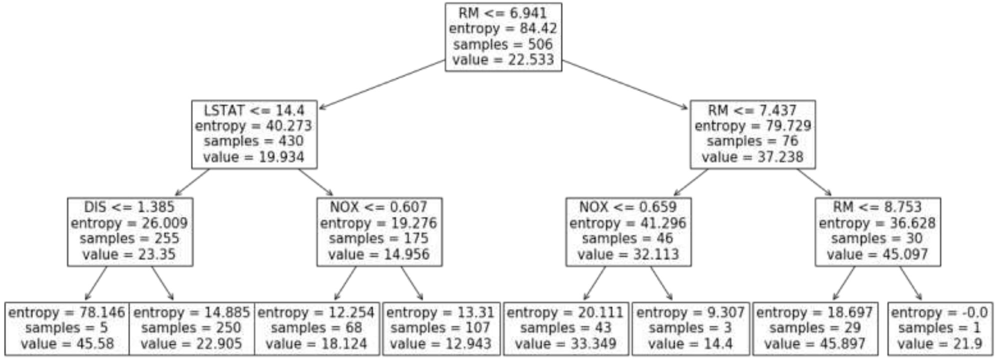
SIMPLE TO GET SHAPLEY VALUES FOR A LINEAR MODEL -

ASSUMPTION INDEPENDENT AND ADDITIVE

Shapley Values for Trees: Tree Shap

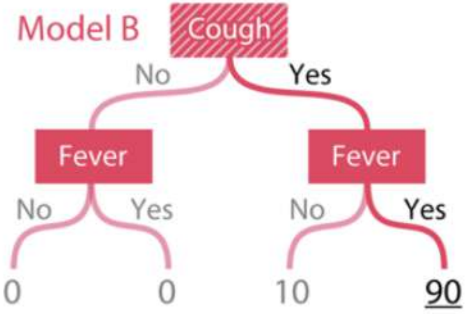
A fast and exact algorithm to compute SHAP values for trees and ensembles of trees. Computes in polynomial time.

Intuition is that by parsing a built tree it is possible to identify various sequences of features and their effect on predictions




```
E[y|LSTAT=4.98,NOX=0.538,RM=6.575] = 5/255 * 45.58 + 250/255 * 22.905 = 23.3496
```

Tree Shap Calculation



Feature combinations	{}	{F}	{C}	{F, C}
Model B: Fever = Yes, Cough = Yes	25	45	50	90

$$\phi_F = \frac{1}{2}[45 - 25] + \frac{1}{2}[90 - 50] = 30$$

$$\phi_C = \frac{1}{2}[50 - 25] + \frac{1}{2}[90 - 45] = 35$$

$$\phi = \phi_0 + \phi_F + \phi_C = 25 + 30 + 35 = 90$$

Source: Meichen Lu

Approximating Shapley Values

No longer have the additive and independence assumption outside of linear models

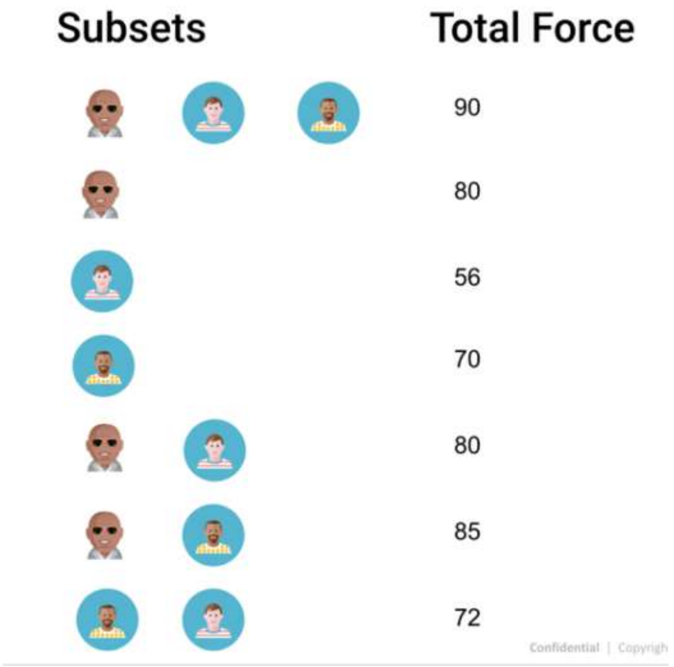
No longer have a tree structure

. . . this means there are 2^N combinations for exact shapley values!

This has lead to **many** methods for approximating Shapley values

Partial List

1. Strumbelj approximation / Sampling
2. Kernel Shap / Local linear model
3. Mimic Shap (Approximates using gbm/Tree Shap)
4. Gradient Shap (differentiable models/deep learning)



Approximating Shapley Values: Strumbelj

Strumbelj approximation:

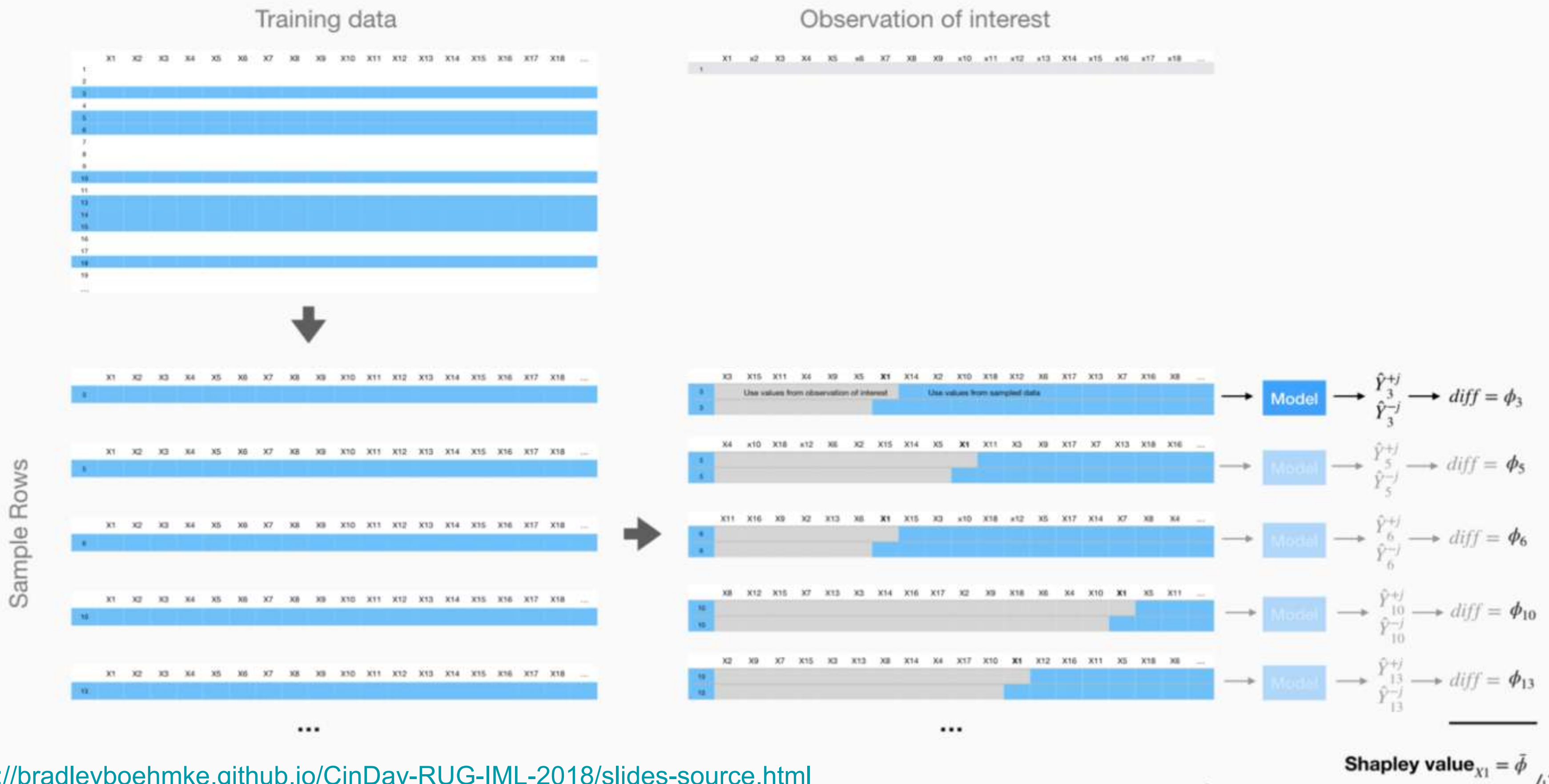
Instead of calculating every sequence, use the concept of permutation to generate a sample of sequences upon which the shapley values are estimated

Work in polynomial time and is model agnostic.

An approximation

1. Repeat for M times.
2. Pick a feature j from the instance x_i.
3. Generate synthetic instances x_L and x_U using x_{ij} pivot.
4. Estimate individual contribution $\phi_{ijm} = \hat{f}(x_L) - \hat{f}(x_U)$
5. Estimate average contribution of feature j at the prediction of the i-th subject as: $\phi_{ij}(x) = \frac{1}{M} \sum_{m=1}^{M} \phi_{ijm}$

Approximating Shapley Values: Strumbelj

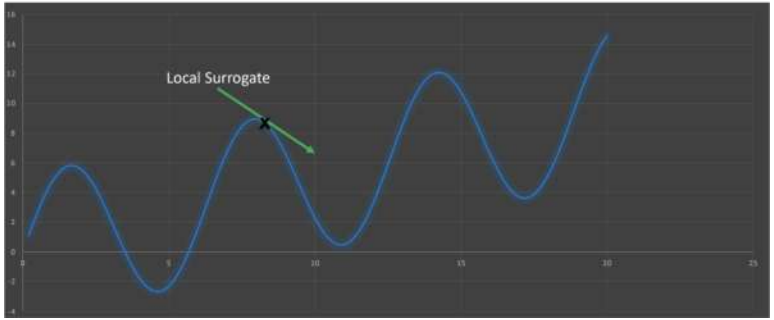


https://bradleyboehmke.github.io/CinDay-RUG-IML-2018/slides-source.html

Approximating Shapley Values: Shap Kernel

Kernel Shap uses a specially-weighted local linear regression to estimate SHAP values for any prediction. It is inspired by LIME. It is model agnostic.

The intuition is we can use least squares point estimate to get the mean values, i.e., the Shapley values



Shap Kernel: Generating Data

The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X).

1. **Background dataset for integrating out features.**
2. Permute values to identify the impact of features
3. Learn a linear model whose coefficients provide the importance of features
4. Returns a matrix of the background dataset and all the features . . . each row sums to the difference between the model output for that sample and the base (expected) value of the model output

This will be the base dataset and bring the base value. The larger the dataset, then the better the approximation. Often the centers of clusters chosen by Kmeans are used.

Shap Kernel: Generating Data

The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X).

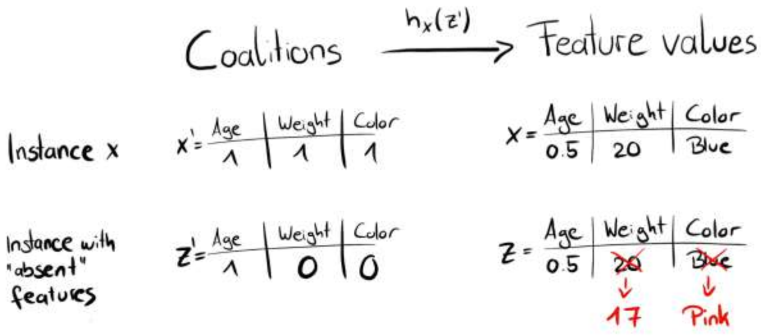
1. Background dataset for integrating out features.
2. **Permute values to identify the impact of features**
3. Learn a linear model whose coefficients provide the importance of features
4. Returns a matrix of the background dataset and all the features . . . each row sums to the difference between the model output for that sample and the base (expected) value of the model output

Compare the background rows to the prediction using permutation to treat value as “missing” . . . This method generates sequences necessary to calculate shapley values

Shap Kernel: Generating Data

The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X).

1. Background dataset for integrating out features.
2. **Permute values to identify the impact of features**
3. Learn a linear model whose coefficients provide the importance of features
4. Returns a matrix of the background dataset and all the features . . . each row sums to the difference between the model output for that sample and the base (expected) value of the model output



Create coalitions and then permute

Shap Kernel: Generating Data

The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X).

1. Background dataset for integrating out features.
2. Permute values to identify the impact of features
3. **Learn a linear model whose coefficients provide the importance of features**
4. Returns a matrix of the background dataset and all the features . . . each row sums to the difference between the model output for that sample and the expected value of the model output

Linear model is an approximation of the model for this prediction

The shapley values can be obtained from this linear model

Shap Kernel: Generating Data

The original model's decision function is represented by the blue/pink background, and is clearly nonlinear. The bright red cross is the instance being explained (let's call it X).

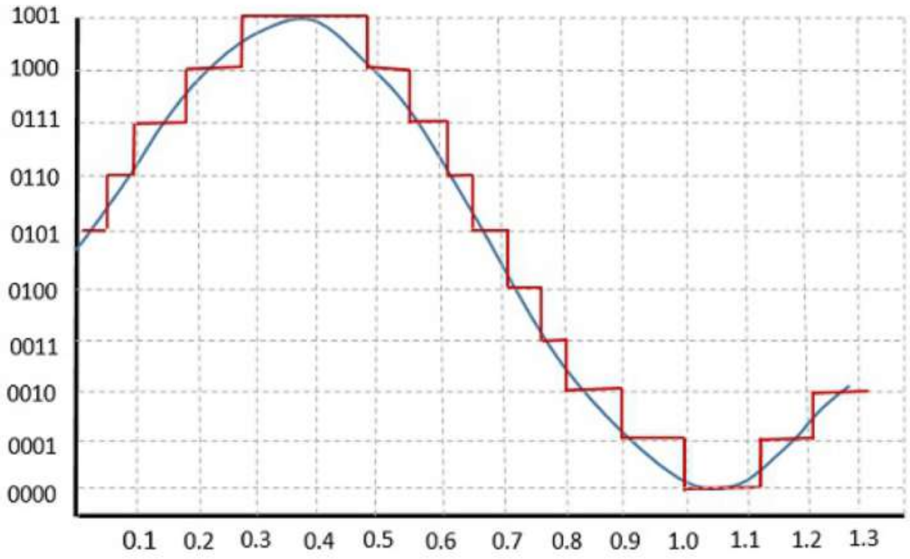
1. Background dataset for integrating out features.
2. Permute values to identify the impact of features
3. Learn a linear model whose coefficients provide the importance of features
4. Returns a matrix of the background dataset and all the features . . . each row sums to the difference between the model output for that sample and the expected value of the model output

Returns the shapley value for that prediction (X)

Mimic Shap

Created an approximation model using a gradient boosted decision tree model. These types of models are so flexible we can train them to mimic any black-box model and then using Tree SHAP we can explain them.

https://github.com/slundberg/shap/blob/master/shap/explainers/mimic.py



Gradient Shap

Integrated gradients values are a bit different from SHAP values, and require a single reference value to integrate from. As an adaptation to make them approximate SHAP values,

Expected gradients reformulates the integral as an expectation and combines that expectation with sampling reference values from the background dataset.

This leads to a single combined expectation of gradients that converges to attributions that sum to the difference between the expected model output and the current output.

Shap Gradient Explainer - https://github.com/slundberg/shap/blob/master/shap/explainers/gradient.py

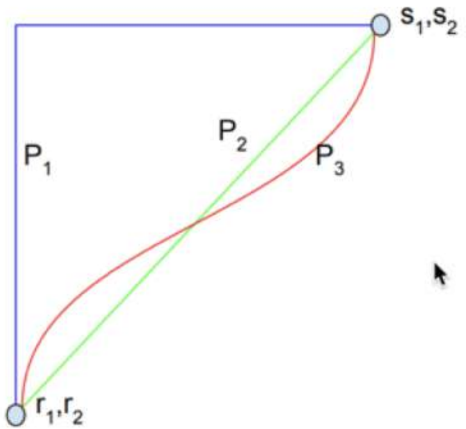


Figure 1. Three paths between an a baseline (r_1, r_2) and an input (s_1, s_2). Each path corresponds to a different attribution method. The path P_2 corresponds to the path used by integrated gradients.

GkmExplain: Fast and Accurate Interpretation of Nonlinear Gapped k-mer Support Vector Machines

Support Vector Machines with gapped k-mer kernels (gkm-SVMs) have been used to learn predictive models of regulatory DNA sequence. However, interpreting predictive sequence patterns learned by gkm-SVMs can be challenging.

Here, we propose gkmexplain: a novel approach inspired by the method of Integrated Gradients for interpreting gkm-SVM models.

Source:

https://www.biorxiv.org/content/biorxiv/early/2018/11/06/457606.full.pdf

https://github.com/kundajelab/gkmexplain

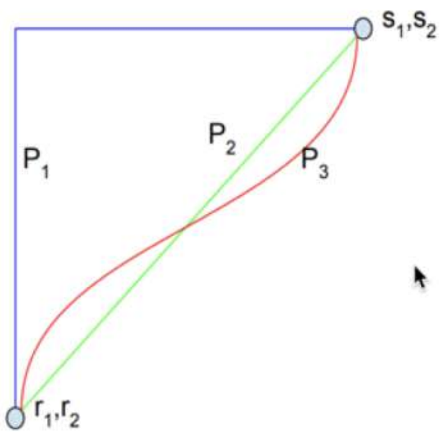
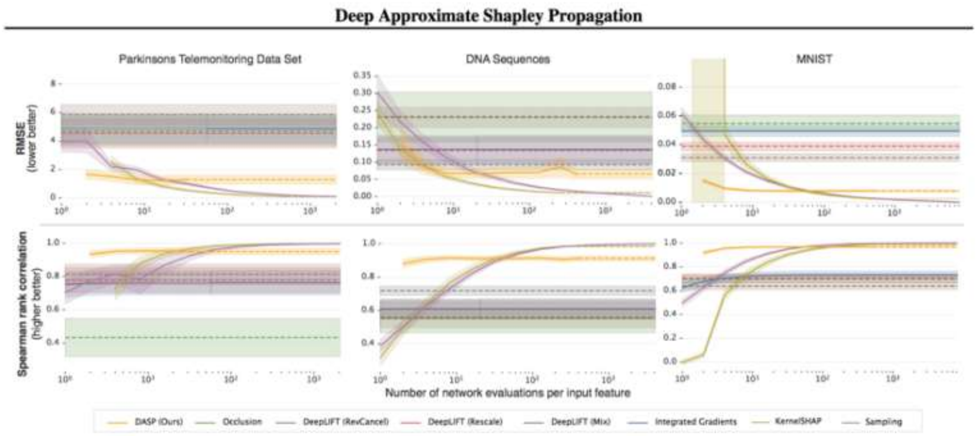


Figure 1. Three paths between an a baseline (r_1, r_2) and an input (s_1, s_2). Each path corresponds to a different attribution method. The path P_2 corresponds to the path used by integrated gradients.

Explaining Deep Neural Networks with a Polynomial Time Algorithm for Shapley Values Approximation

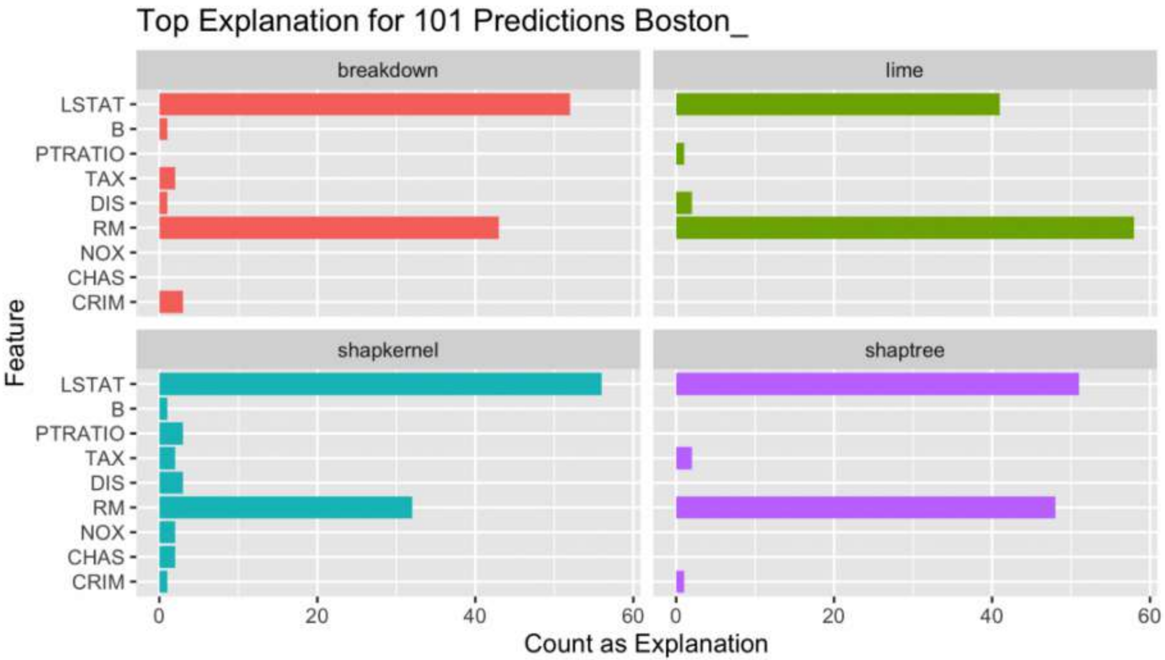
Deep Approximate Shapley Propagation (DASP), a novel perturbation-based method that approximates Shapley values using uncertainty propagation in DNNs.

Source: http://proceedings.mlr.press/v97/ancona19a/ancona19a.pdf



Comparing Explainers

We can see that explainers can differ, in this case lime is differing from Breakdown, Shap Kernel, and Shap Tree

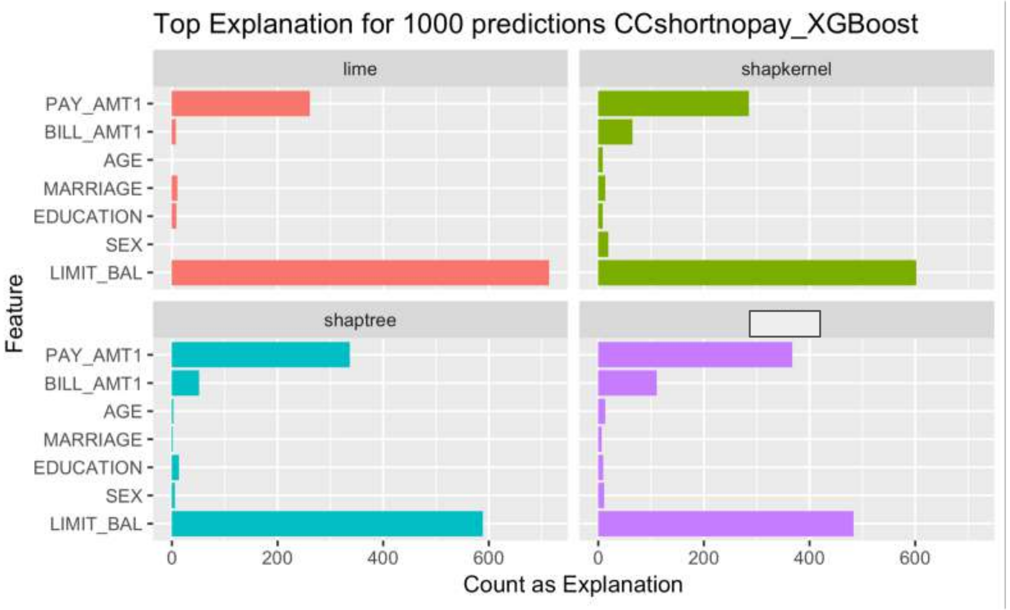


Credit Card: No Pay0

This is a variant of the credit card dataset - I have removed all the strongest feature, Pay0, and the correlated features (e.g, just Bill_Amt1 and drop Bill_Amt2, Bill Amt3). This model still performs well.

The visualization counts what top explanation for each explainer.

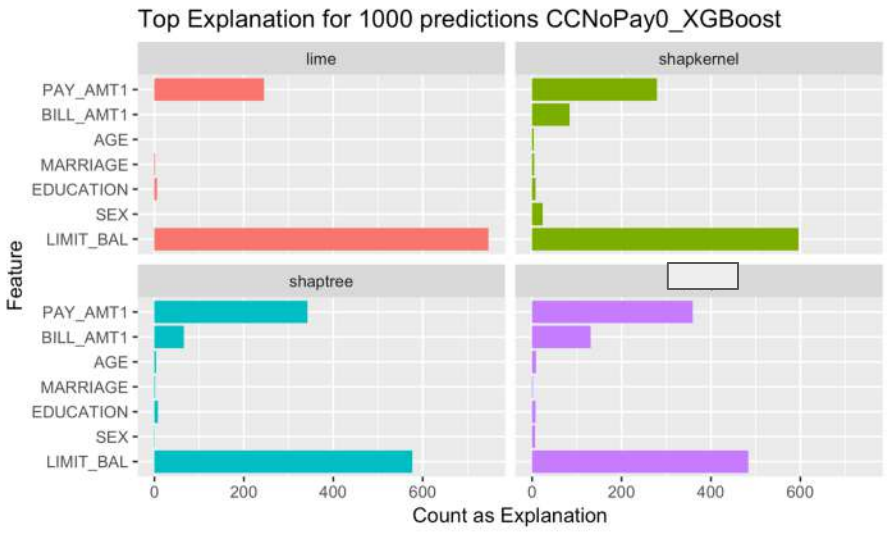
Interestingly, all 4 explainers are pretty similar . . . LIME has a tendency towards more on Limit_Bal . . . this will change.

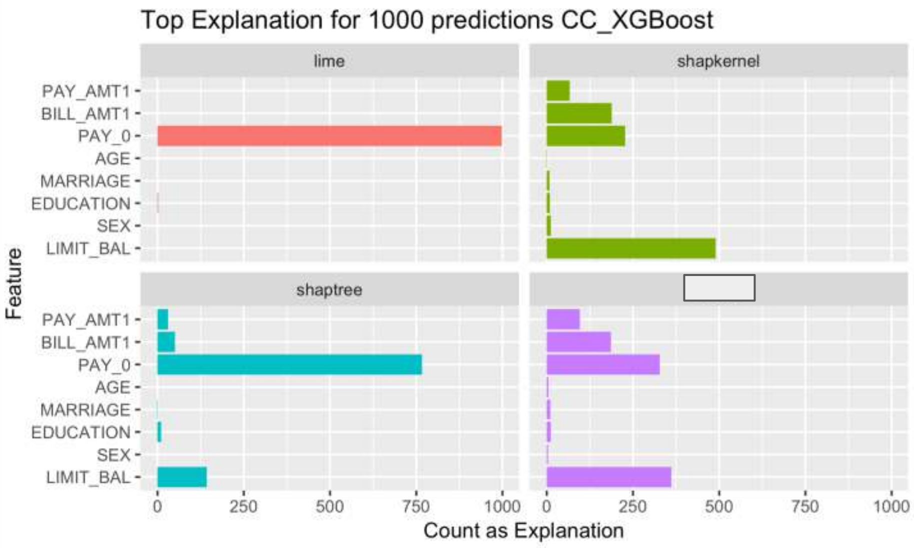


Credit Card: Add Pay0

These results show what happens when we add Pay0, which is a very strong feature to the explanations.

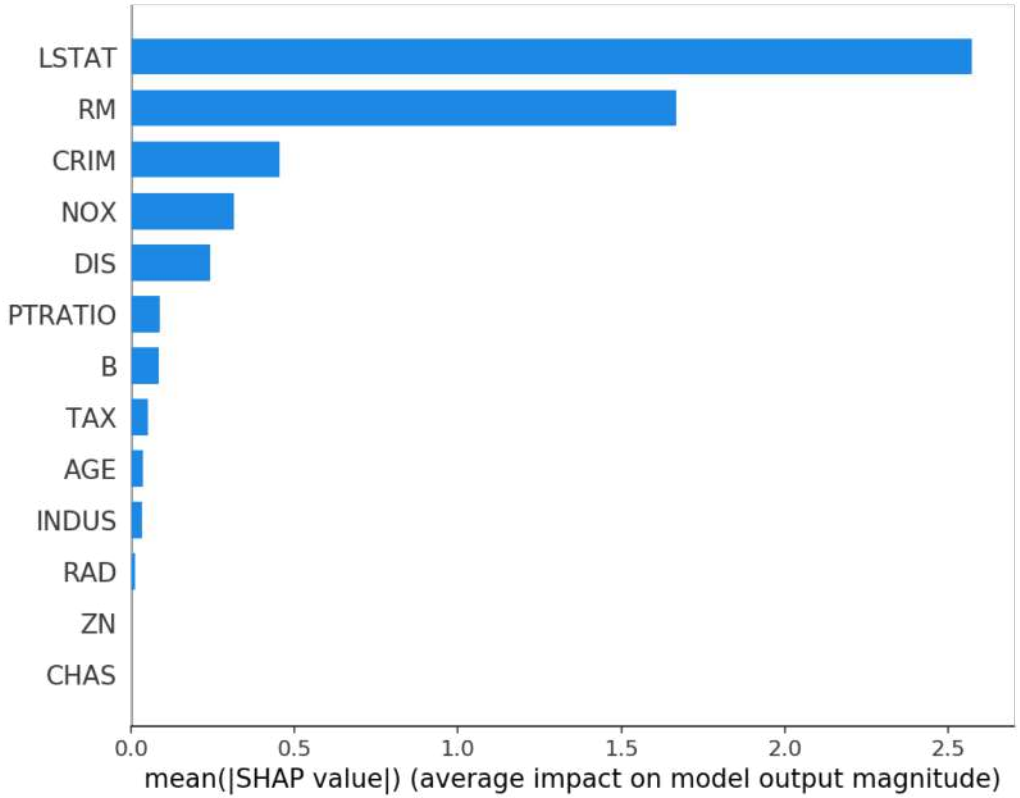
LIME now only explains one feature, Shap Tree also takes a big jump towards Pay0, while XEMP and Shap Kernel use Pay0, but it's not dominant. . . . So what is going on?





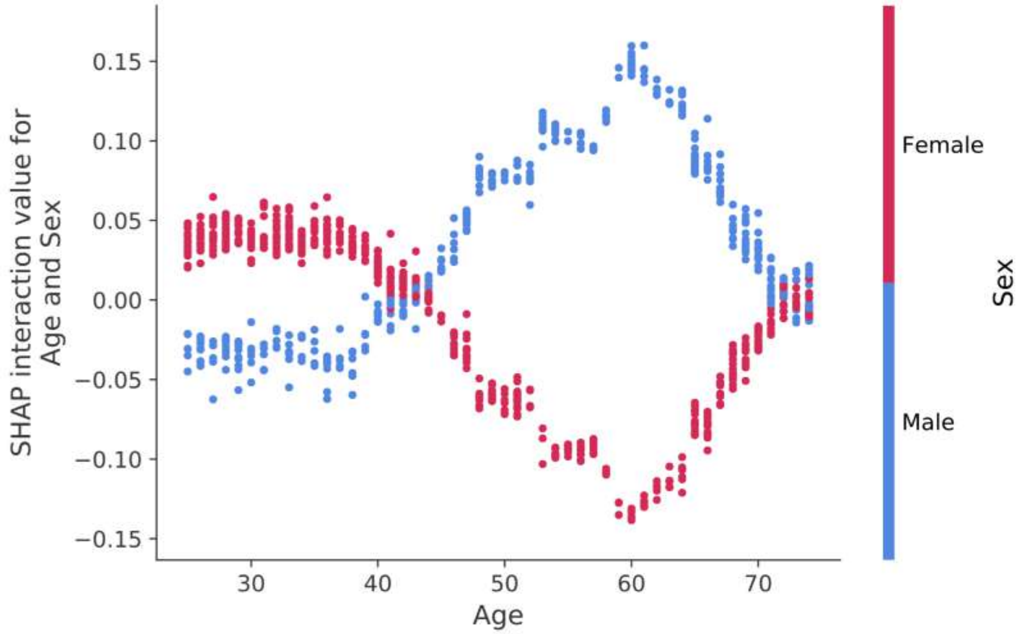
Aggregating Shapley Values

1. **Feature Importance**
2. Feature interactions
3. Feature Selection
4. Supervised Clustering



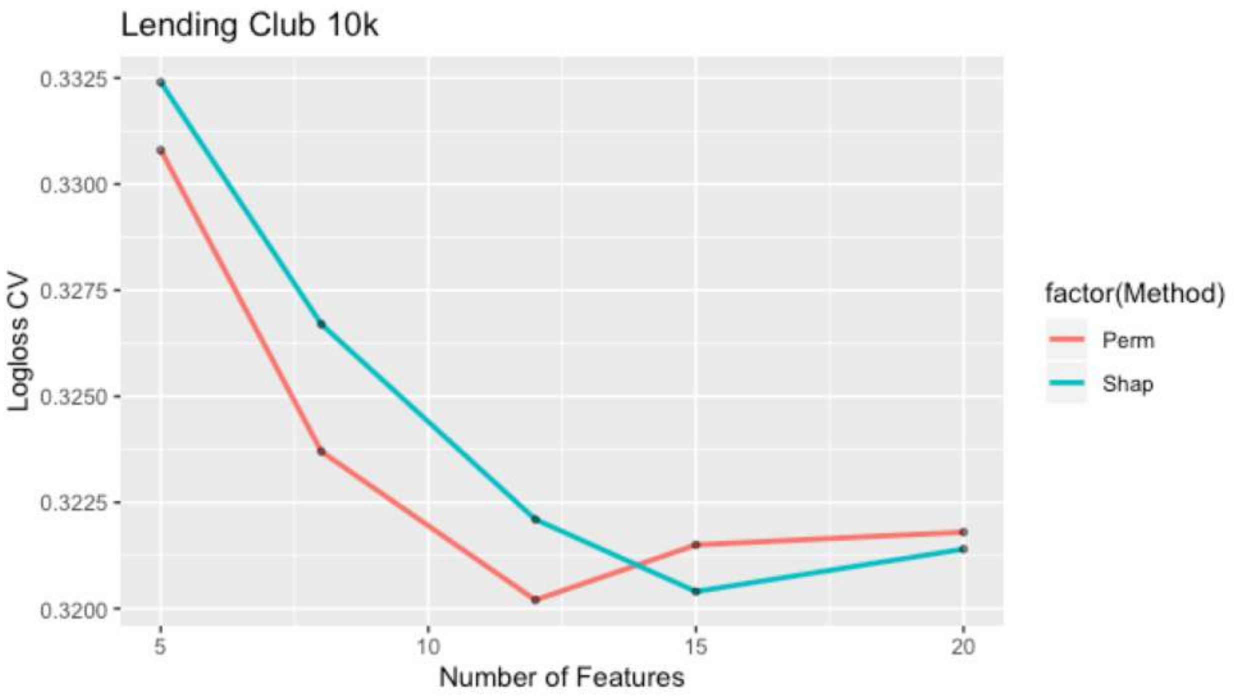
Aggregating Shapley Values

1. Feature Importance
2. **Feature interactions**
3. Feature Selection
4. Supervised Clustering



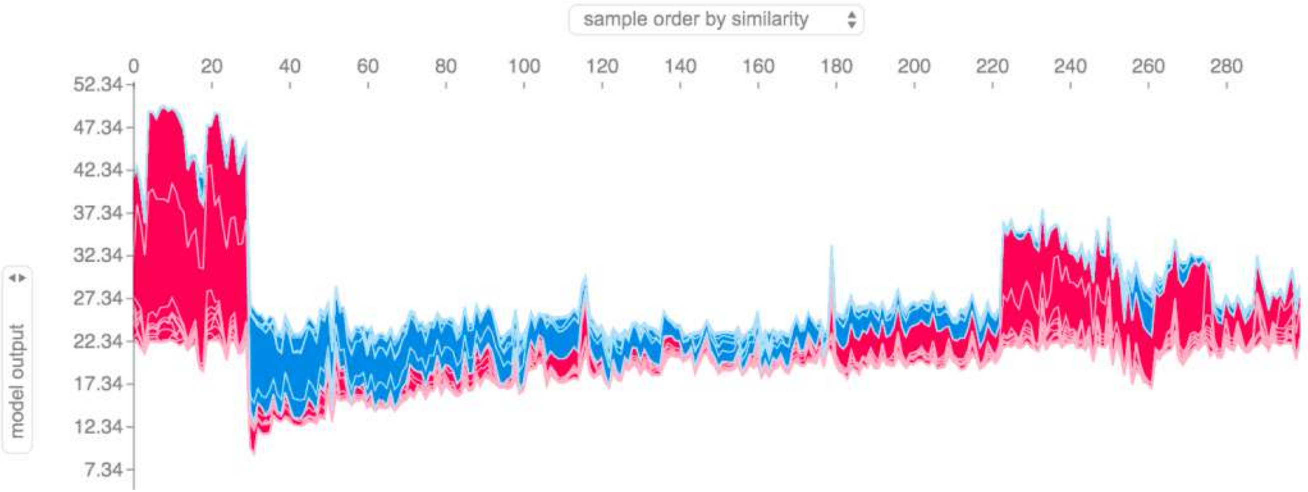
Feature Selection

1. Feature Importance
2. Feature interactions
3. **Feature Selection**
4. Supervised Clustering



Aggregating Shapley Values

1. Feature Importance
2. Feature interactions
3. Feature Selection
4. **Supervised Clustering**



Use This!
Model Agnostic Explanation Tools

What are features driving the model? Feature importance

How is a specific feature driving? Partial dependence

Let's explain some examples? Prediction explanation techniques (LIME, SHAP, XEMP . . .)

Question time

rajiv.shah@datarobot.com
@rajcs4

https://bit.ly/inter_workshop