Lecture 7: Classifiers I

Intro to Data Science for Public Policy
Spring 2017

Jeff Chen + Dan Hammer

Roadmap

- A Surprise
- Motivation
- Classification Preliminaries
- Decision Trees
- <Break>
- Random Forests
- Homework #3
- Homework #4 – a head up

All completed class materials can now be downloaded here:

https://georgetownmccourt.github.io/data-science/

Roadmap

- A Surprise
- Motivation
- Classification Preliminaries
- Decision Trees
- <Break>
- Random Forests
- Homework #3
- Homework #4 – a head up

Isn’t this little guy just so **cute**?

Source; Wikimedia commons

Isn’t this little guy just so **cute**…?

Source; Wikimedia commons

Sea Lion Pup

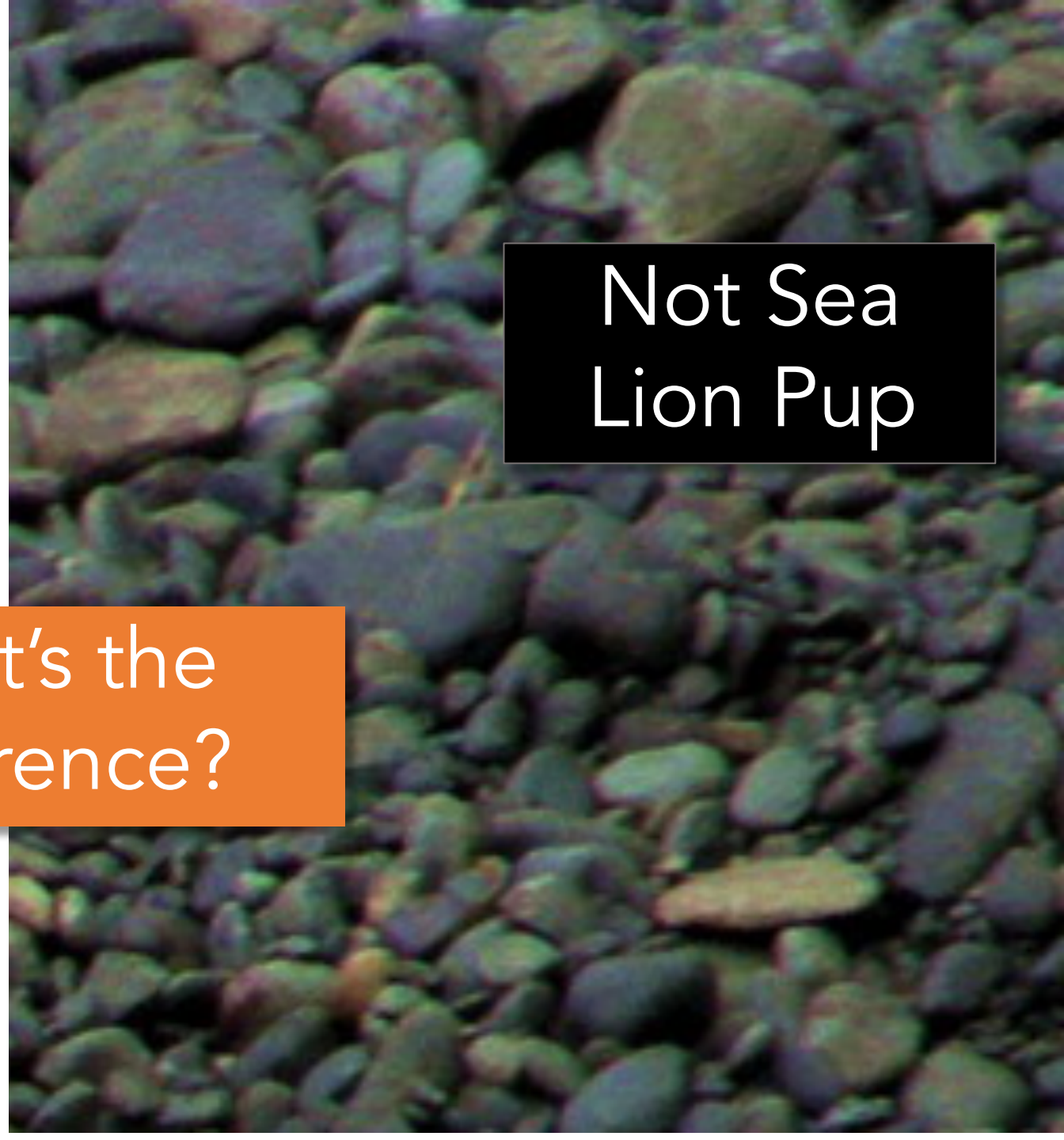
Source; Wikimedia commons

What's the difference?

Not Sea Lion Pup

Sea Lion Pup

Source; Wikimedia commons



Not Sea Lion Pup

What's the difference?

Sea Lion Pup

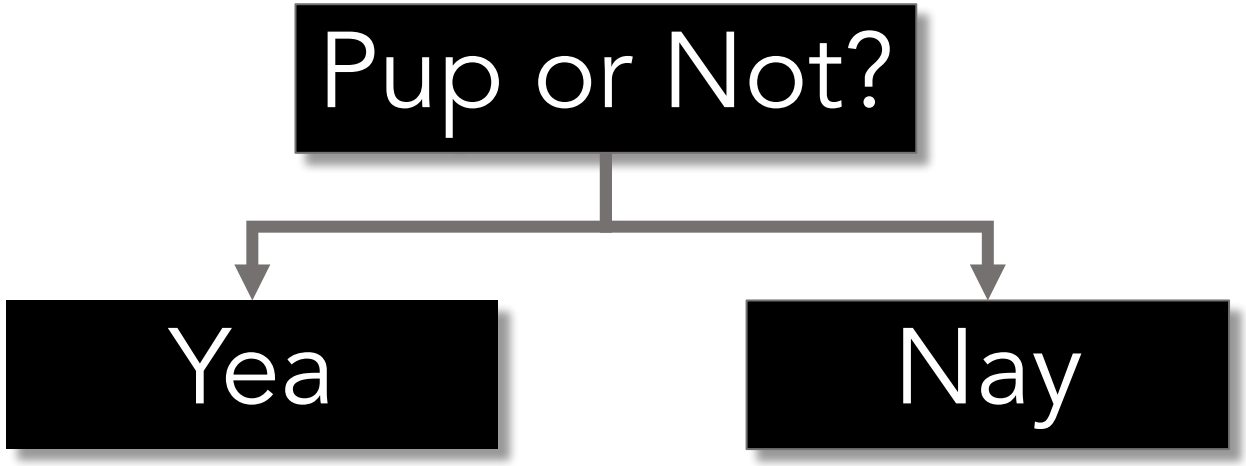
Source; Wikimedia commons

Not Sea Lion Pup

What's the difference?

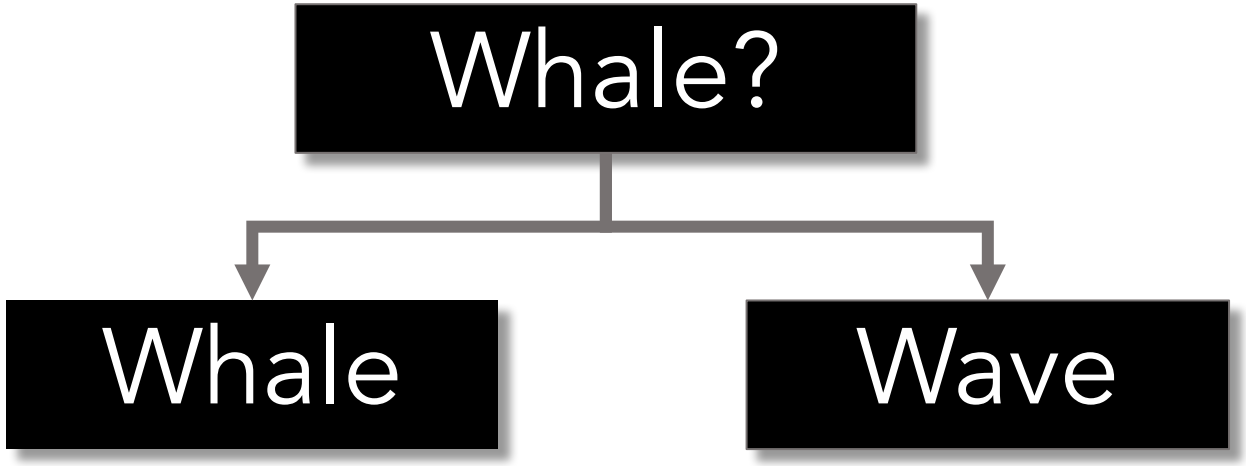
Binary variable

One feature, two classes



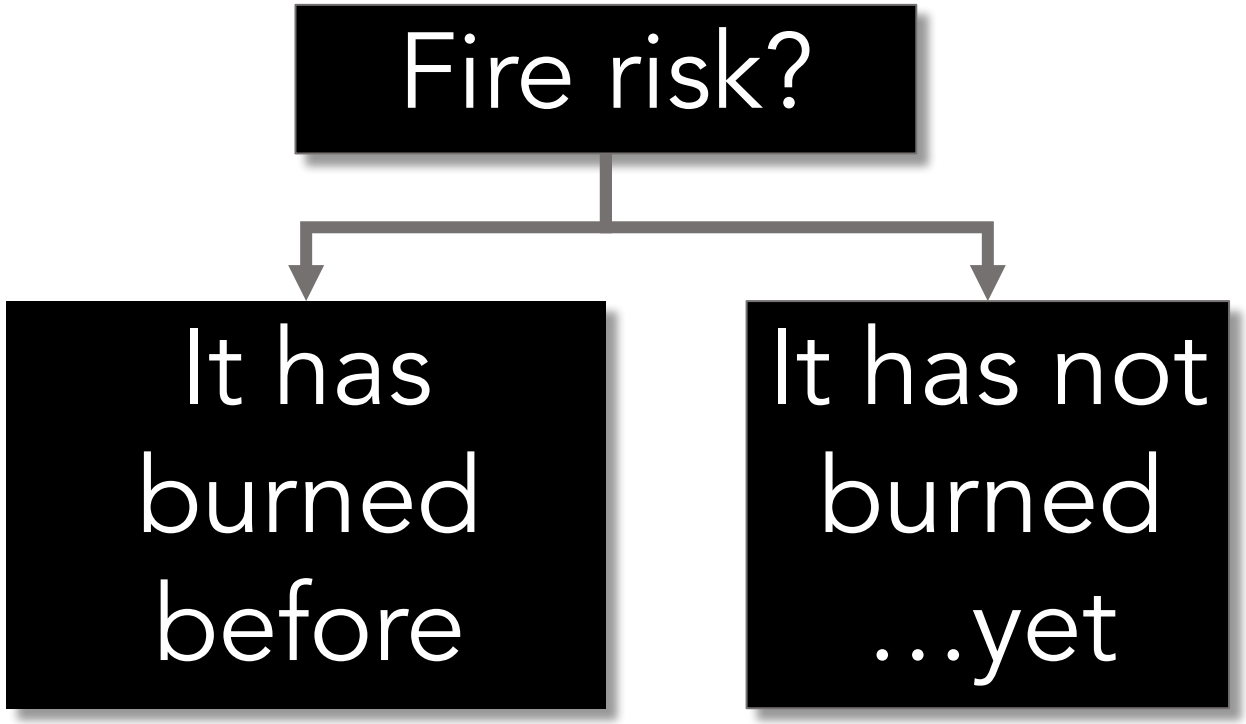
Binary variable

One feature, two classes



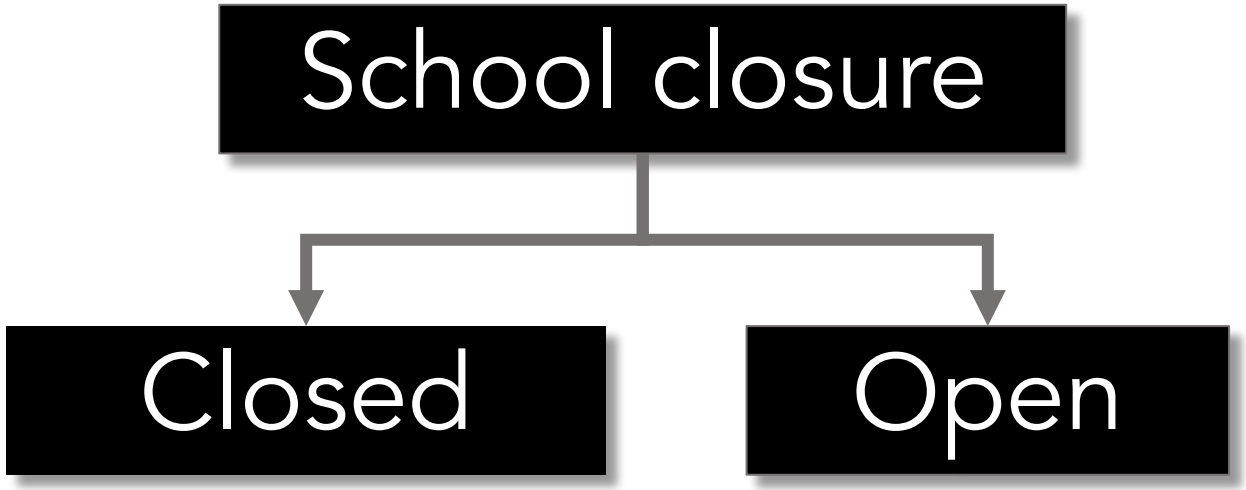
Binary variable

One feature, two classes



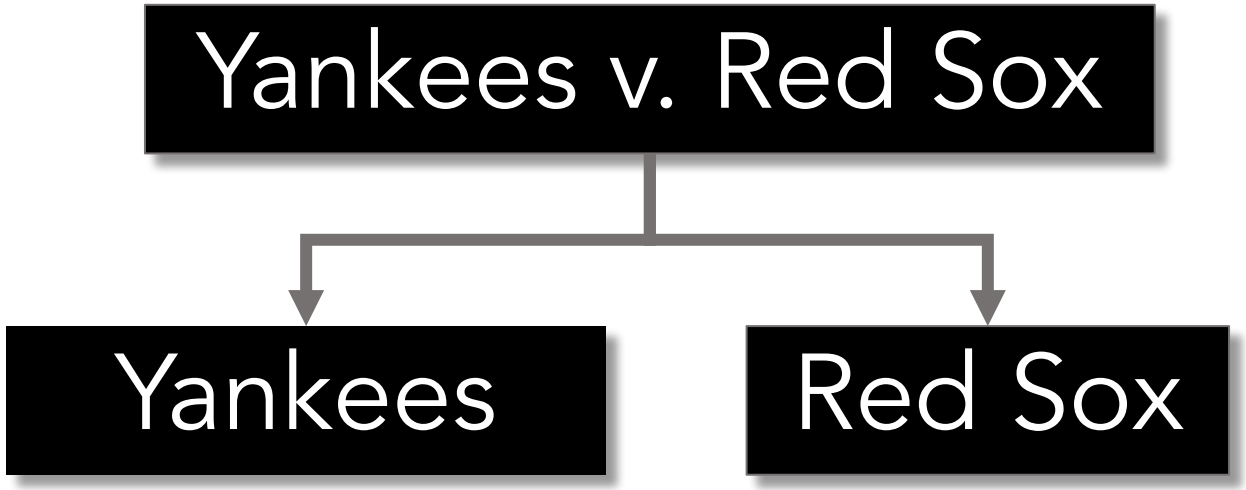
Binary variable

One feature, two classes



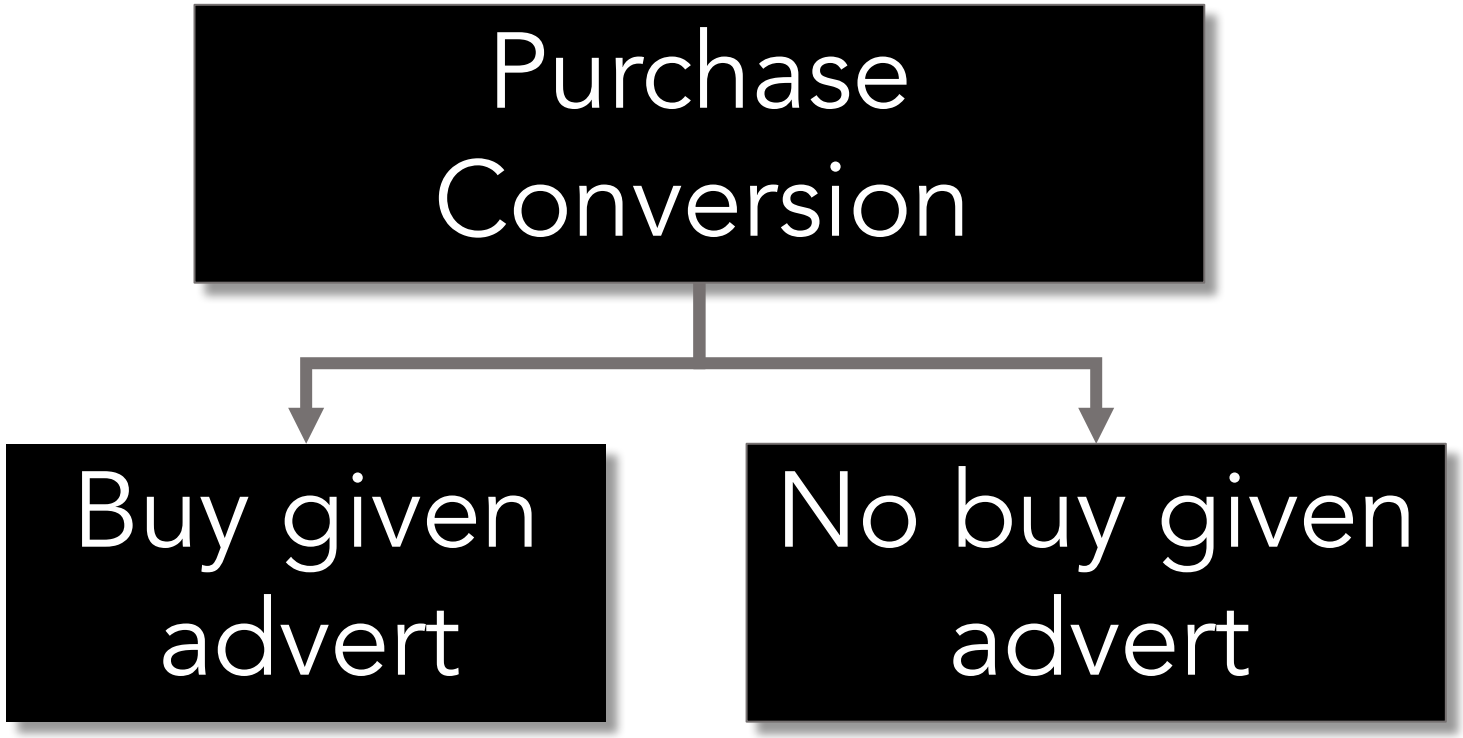
Binary variable

One feature, two classes



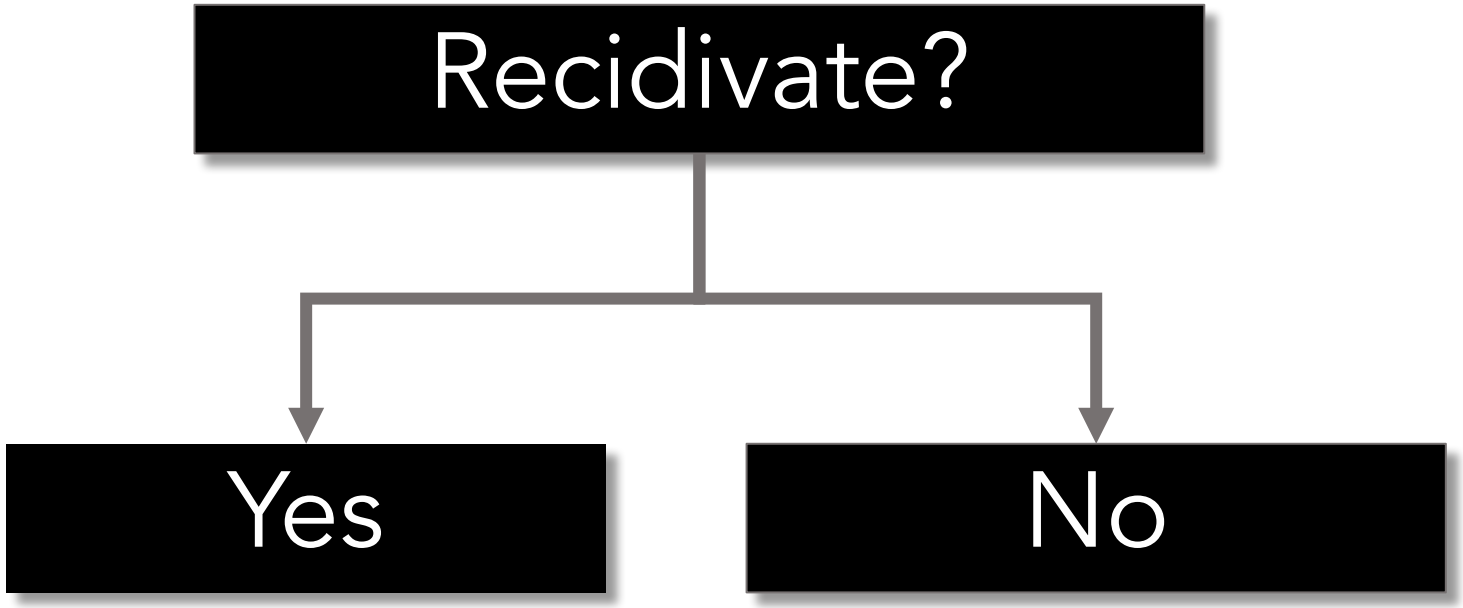
Binary variable

One feature, two classes



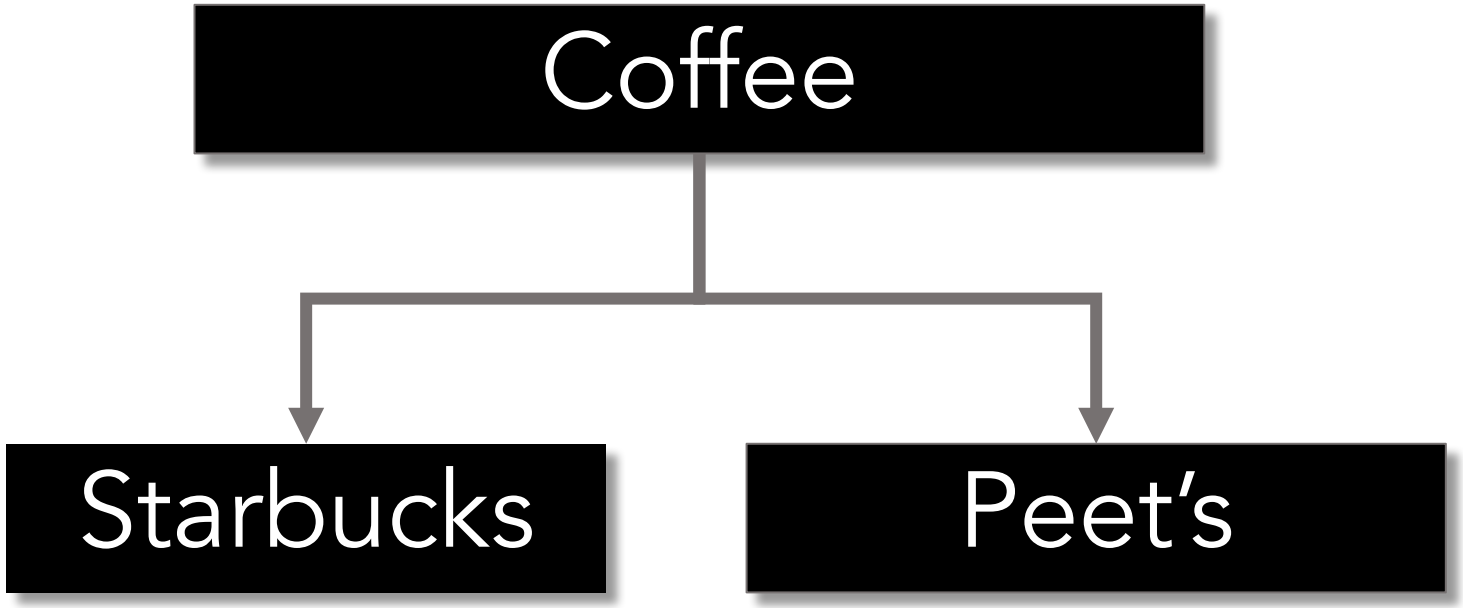
Binary variable

One feature, two classes



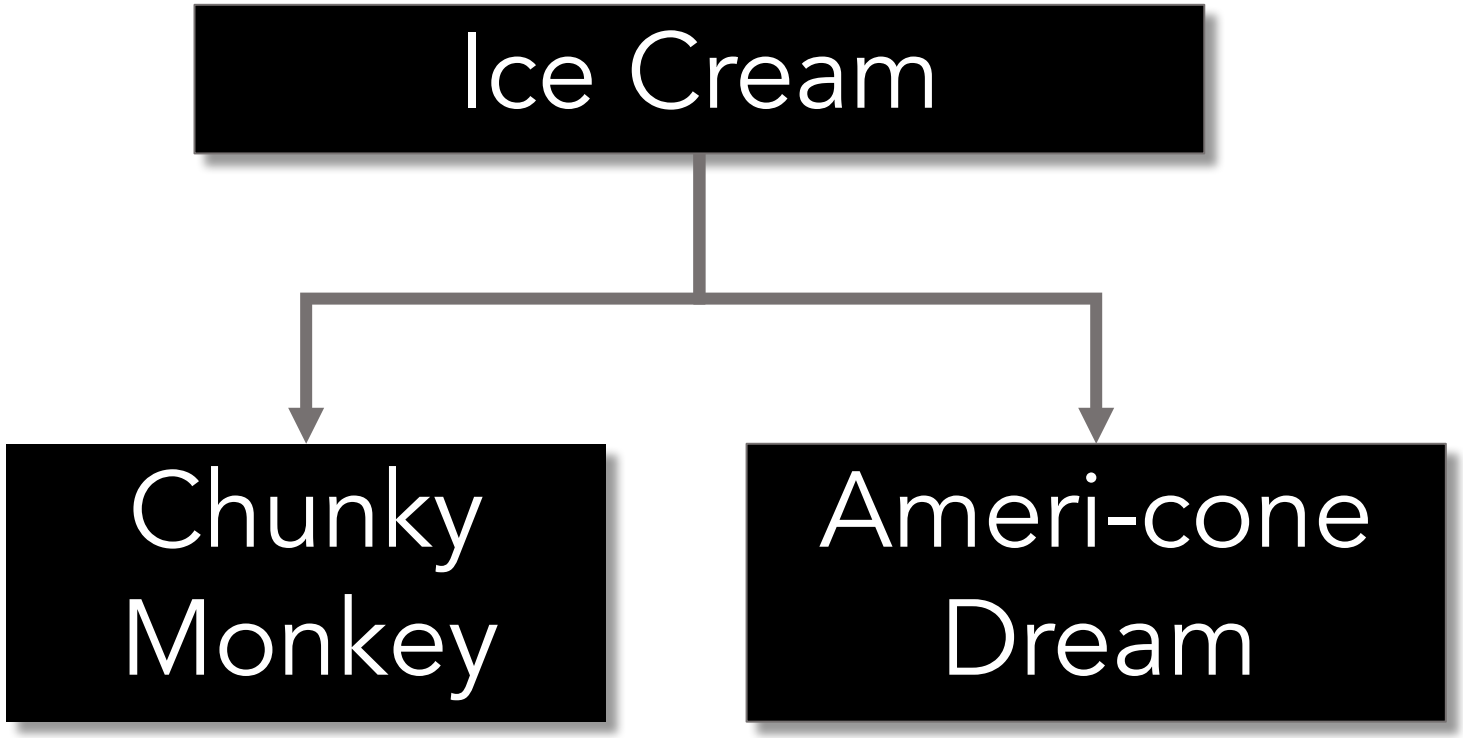
Binary variable

One feature, two classes



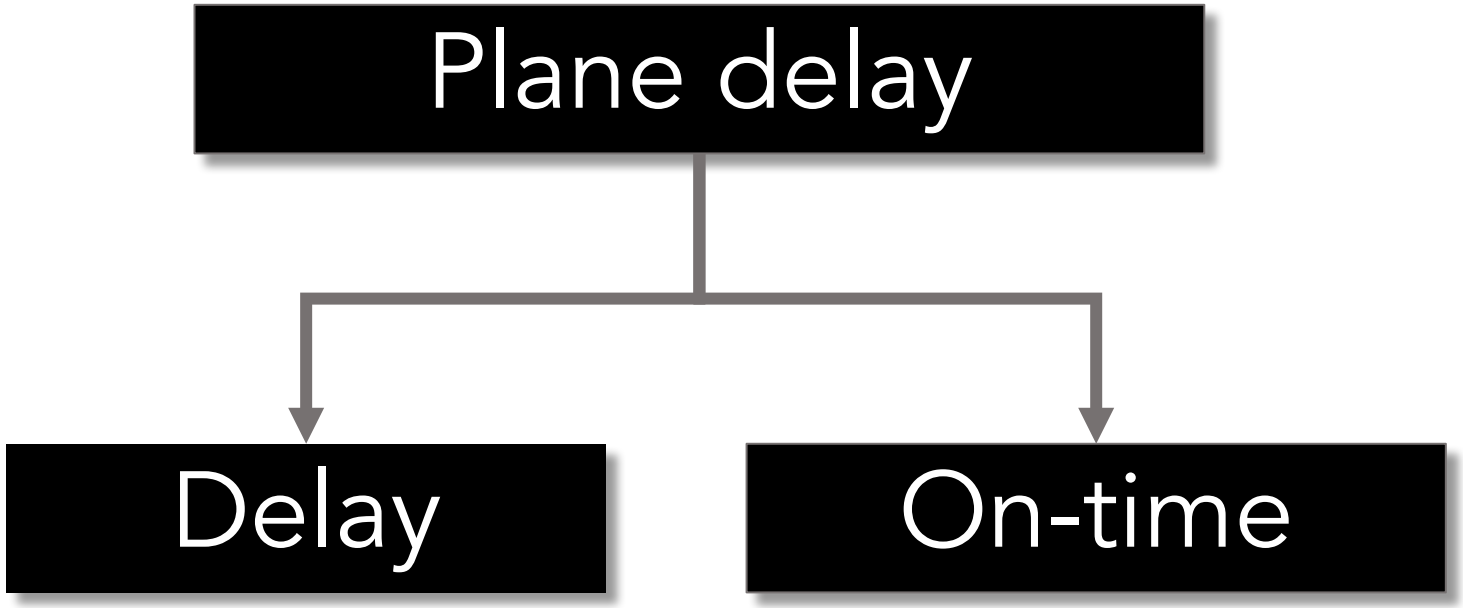
Binary variable

One feature, two classes



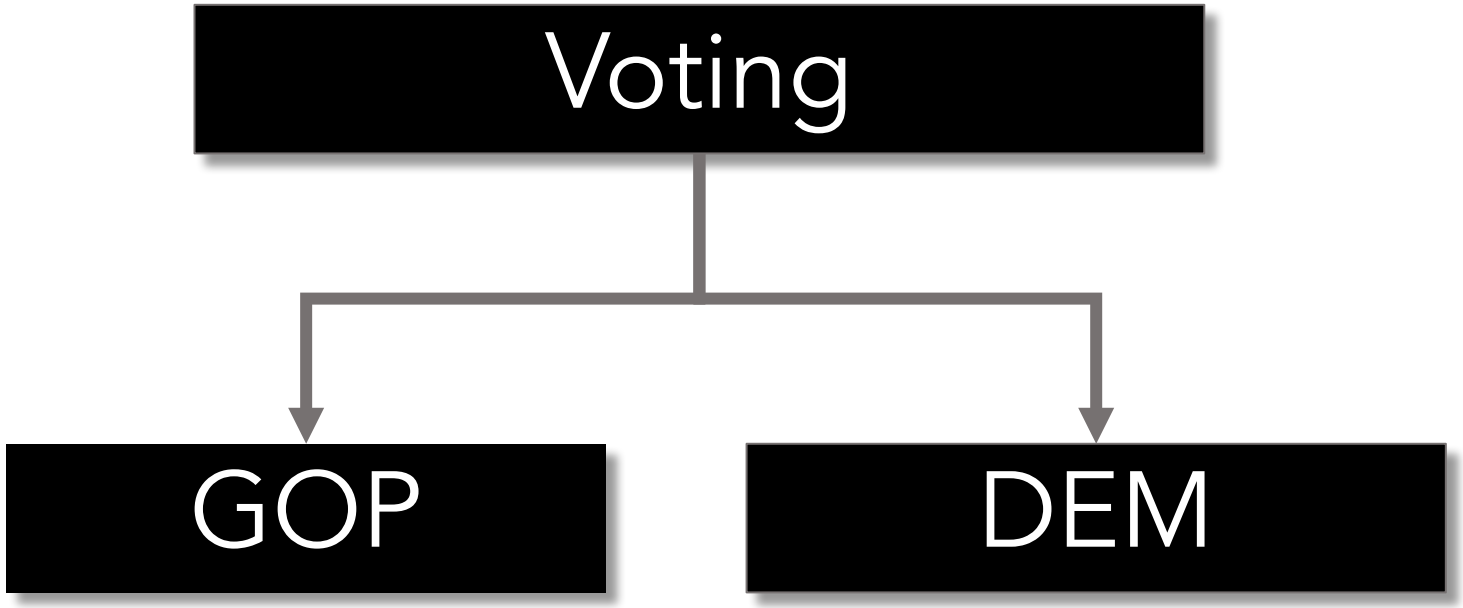
Binary variable

One feature, two classes



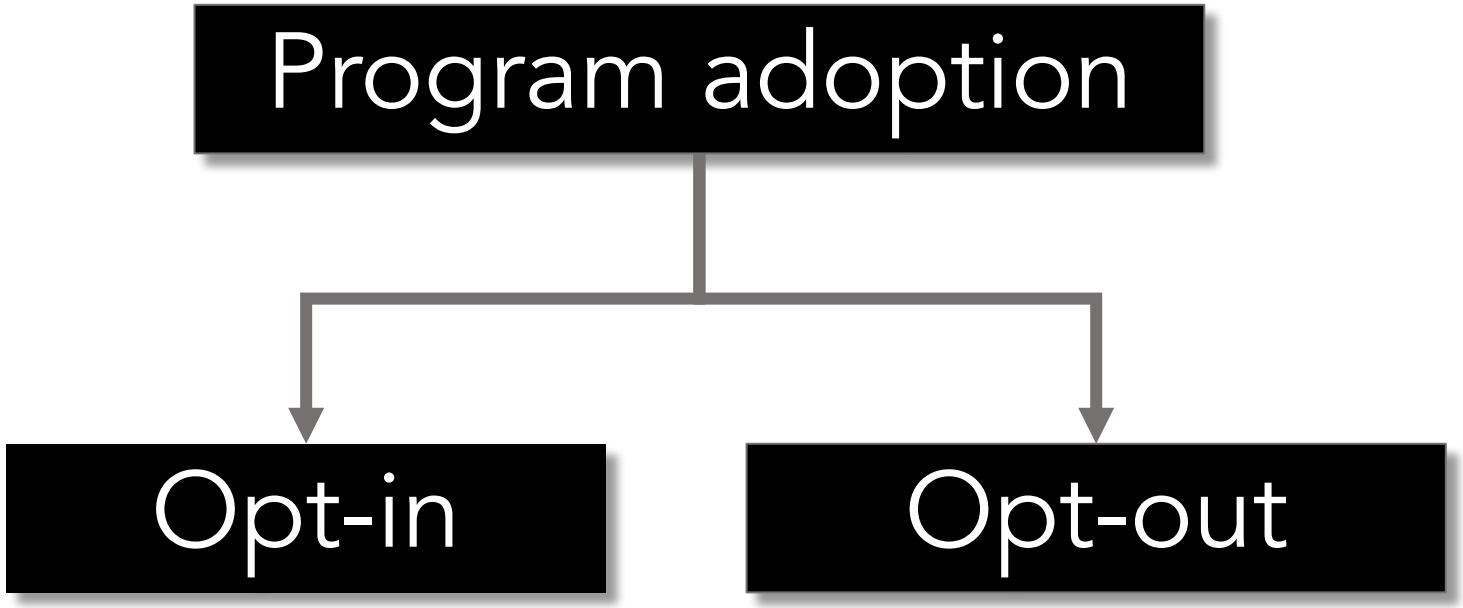
Binary variable

One feature, two classes



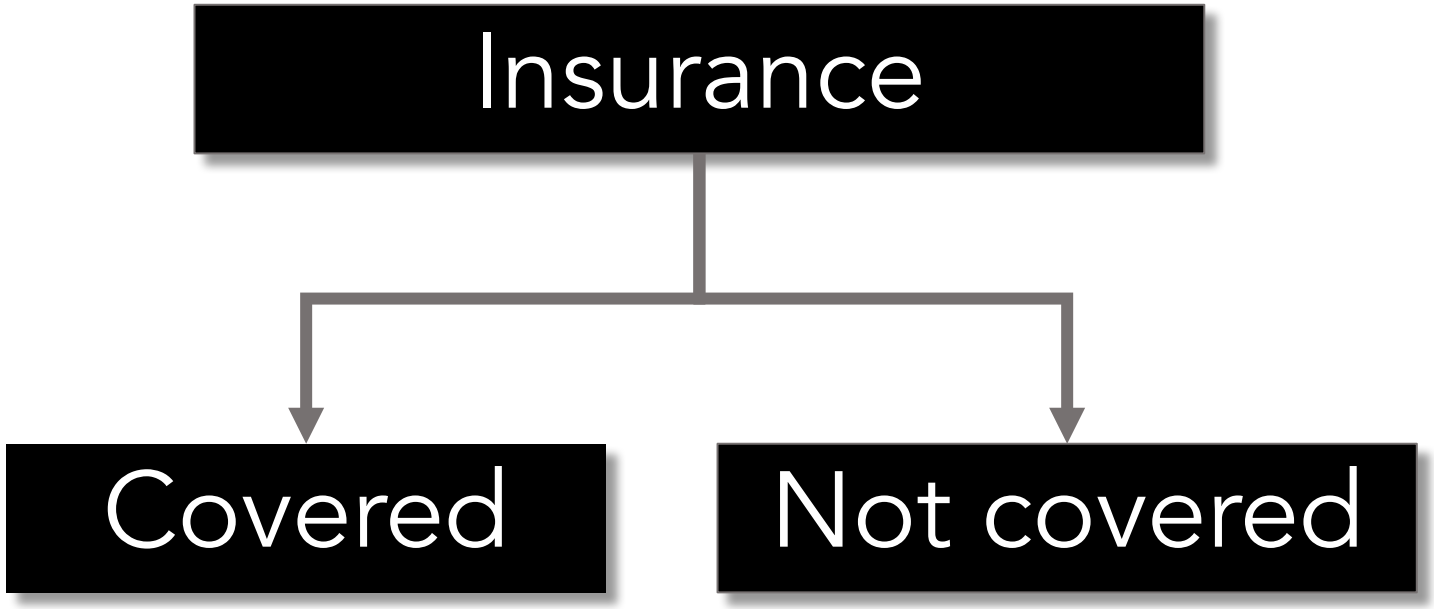
Binary variable

One feature, two classes



Binary variable

One feature, two classes



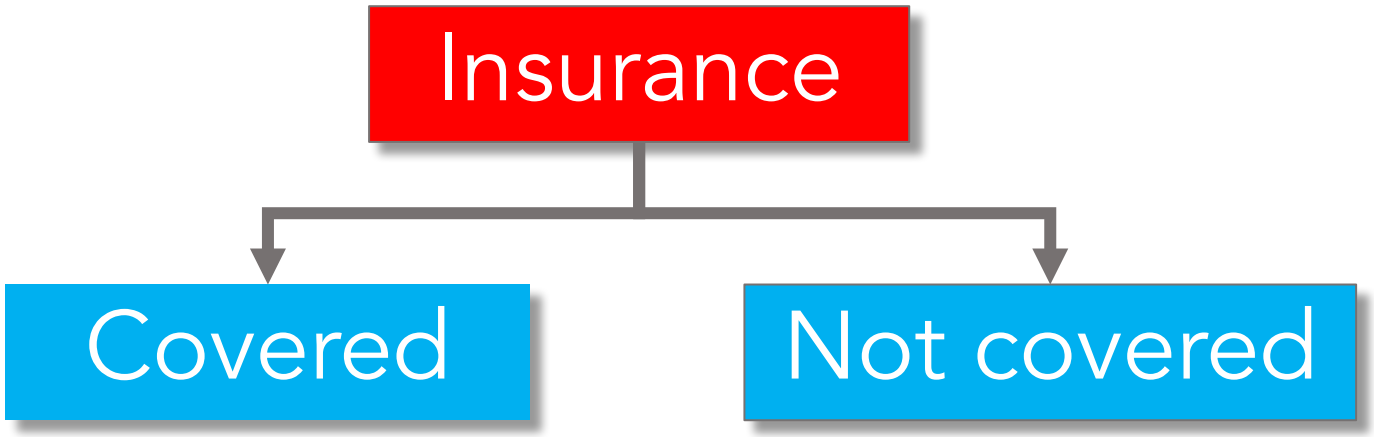
Some of the most pressing questions can be answered by simply understanding why people, things, events fall into one group but not another.

This is called **classification**.

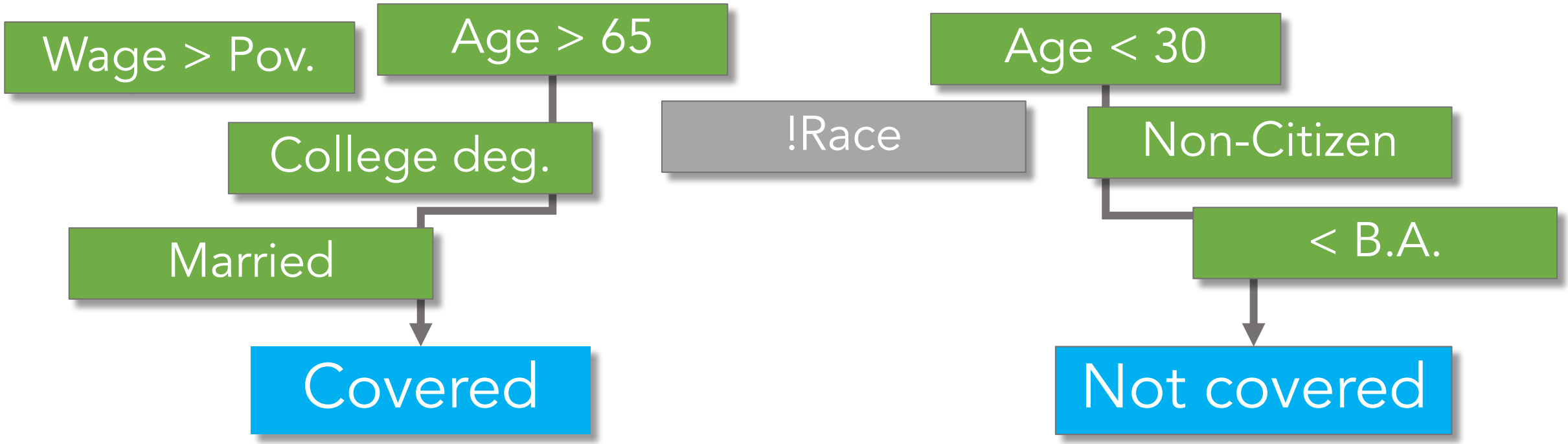
Roadmap

- A Surprise
- Motivation
- Classification Preliminaries
- KNN: Part Deux
- Decision Trees
- <Break>
- Random Forests
- Homework #3
- Homework #4 – a head up

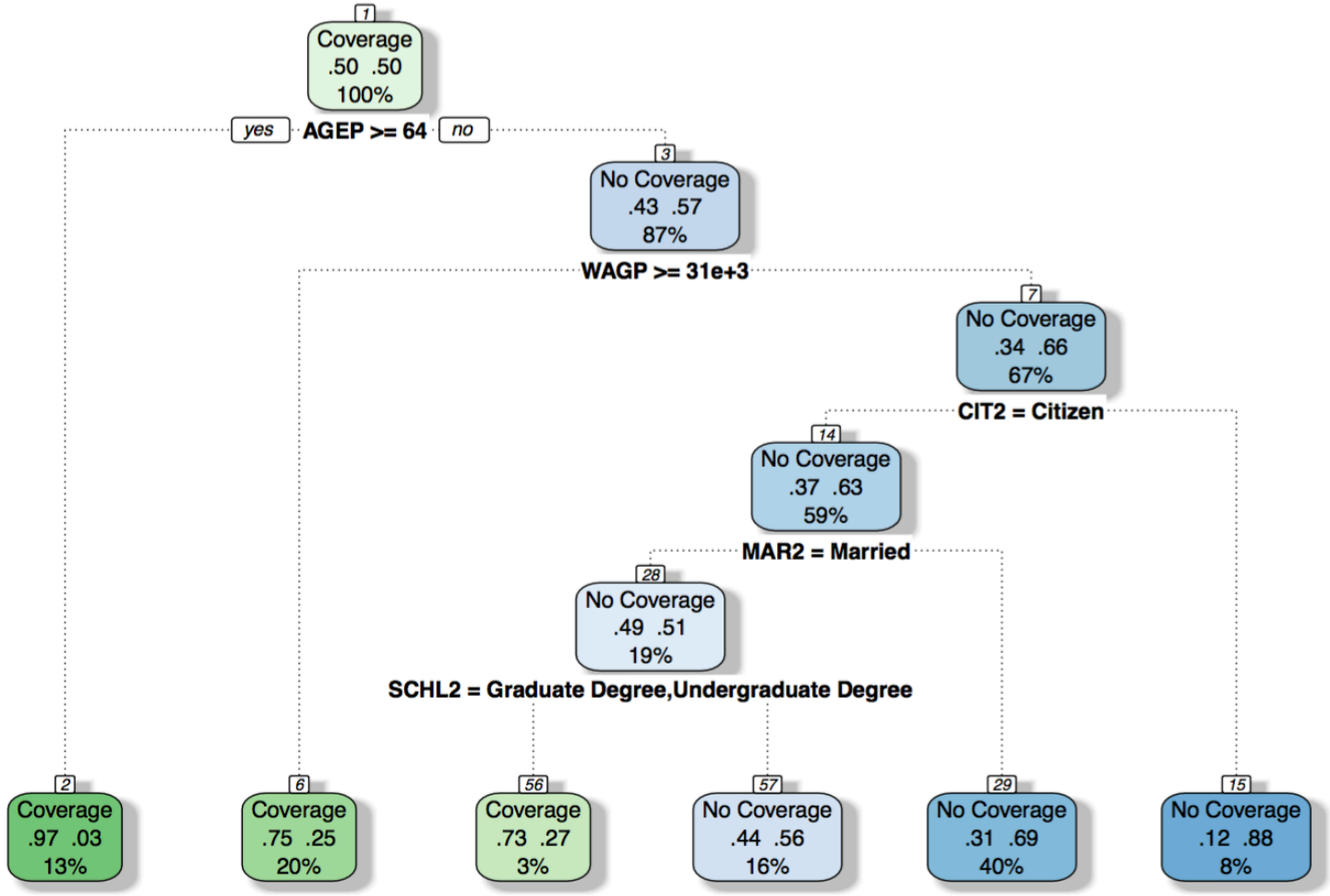
Classifiers are a type of supervised learning problem that handles **target features** that contain discrete labels, otherwise known as **classes**.



Given a training set, a classifier or classification algorithm learns how input features are associated with each class.



The algorithm may yield an equation, a series of equations or rules.



What are the differences between regression and classifiers?

What is the difference between
regressions and classifiers?

#1. The targets (Y) are different.

Regressions train on **continuous** features and return continuous predictions.

Classifiers train on **discrete** variables and can return a probability and/or a discrete prediction.

What is the difference between
regressions and classifiers?

#1. The targets (Y) are different.

Y = [0, 1.23, 525.2, 210, 0.53]
Y-hat = [0.4, 1.2, 450, 200, 0.4]

Y = ['yes','no','yes','no','no','yes'] = [1,0,1,0,0 ,1]
Y-hat = [0.89, 0.1, 0.6, 0.2, 0.4, 0.99]

What is the difference between
regressions and classifiers?

#2. The goals are different.

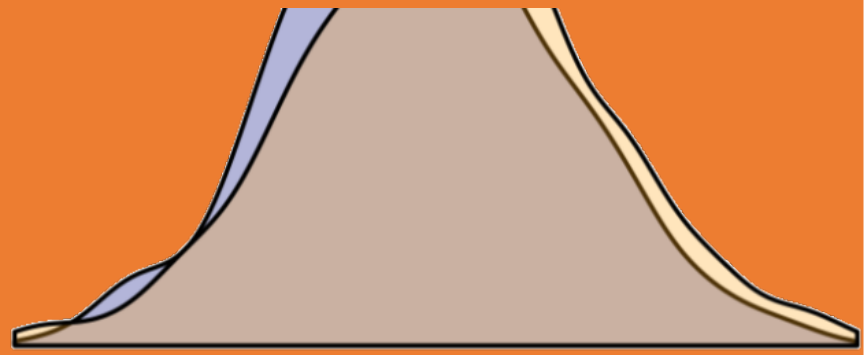
Regressions try to find which features move together.

Classifiers look for opportunities to maximize separability.

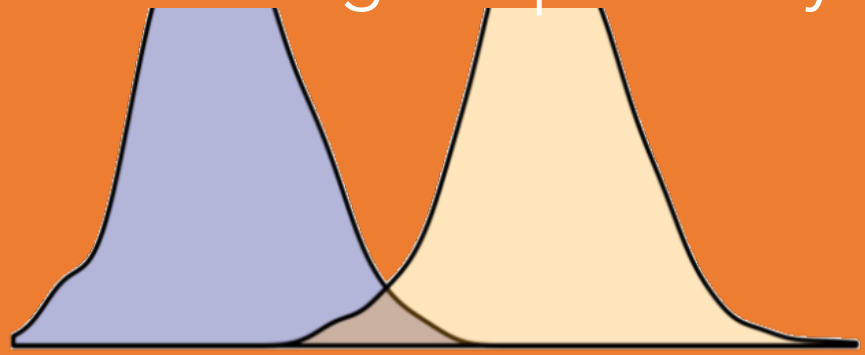
What is the difference between
regressions and classifiers?

Side bar: Separability is the extent to which two classes can be clearly identified using quantitative variables. Greater the separability, greater the accuracy of prediction.

Low Separability



High Separability



What is the difference between
regressions and classifiers?

#3. Accuracy metrics are different

Regressions are measured in terms of R-squared, F-statistics, RMSE, and MAPE.

Classifiers rely on measures that can be derived from a confusion matrices.

What is the difference between
regressions and classifiers?

Side bar: What's a confusion matrix?

	PREDICT: TRUE	PREDICT: FALSE	
ACTUAL: TRUE	TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)	True Positive Rate (TPR) Aka Recall or Sensitivity TP/(TP+FN)
ACTUAL: FALSE	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)	True Negative Rate (TNR) Aka Specificity TN/(TN + FP)
	Positive Predicted Value (PPV) or Precision TP/(TP+FP)	Negative Predicted Value (NPV) TN/(TN+FN)	% Accuracy (TP + TN)/ALL

What is the difference between
regressions and classifiers?

Question: If a value is predicted to be TRUE, what should the cutoff threshold be?

$$Pr(Y) = [0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1.0]$$

What is the difference between
regressions and classifiers?

Question: If a value is predicted to be TRUE, what should the cutoff threshold be?

Pr(Y) = [0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9, 1.0]

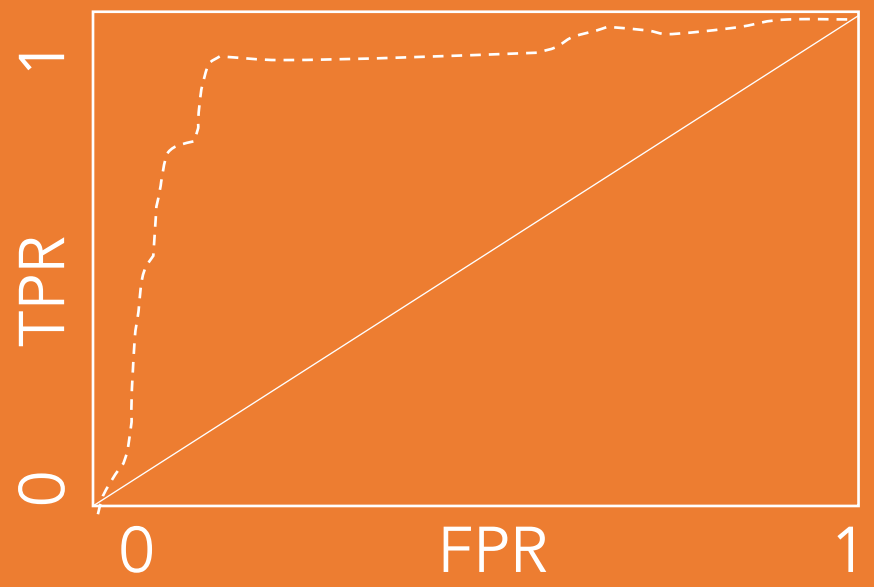
Threshold can be at any point!

What is the difference between
regressions and classifiers?

Question: What do we do with that fine nugget of confusing wisdom?

#A. Receiving Operating Characteristic Curve (ROC)

A plot of TPR and FPR tested at many thresholds between 0 and 1. The area under the dotted line ranges from 0.5 (coin toss accuracy) to 1.0 (perfect accuracy).



What is the difference between regressions and classifiers?

Question: What do we do with that fine nugget of confusing wisdom?

#B. F1-Score

A ratio of recall and precision where 0 is a poor prediction and 1 is a perfect prediction. Best used if the sample is balanced.

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

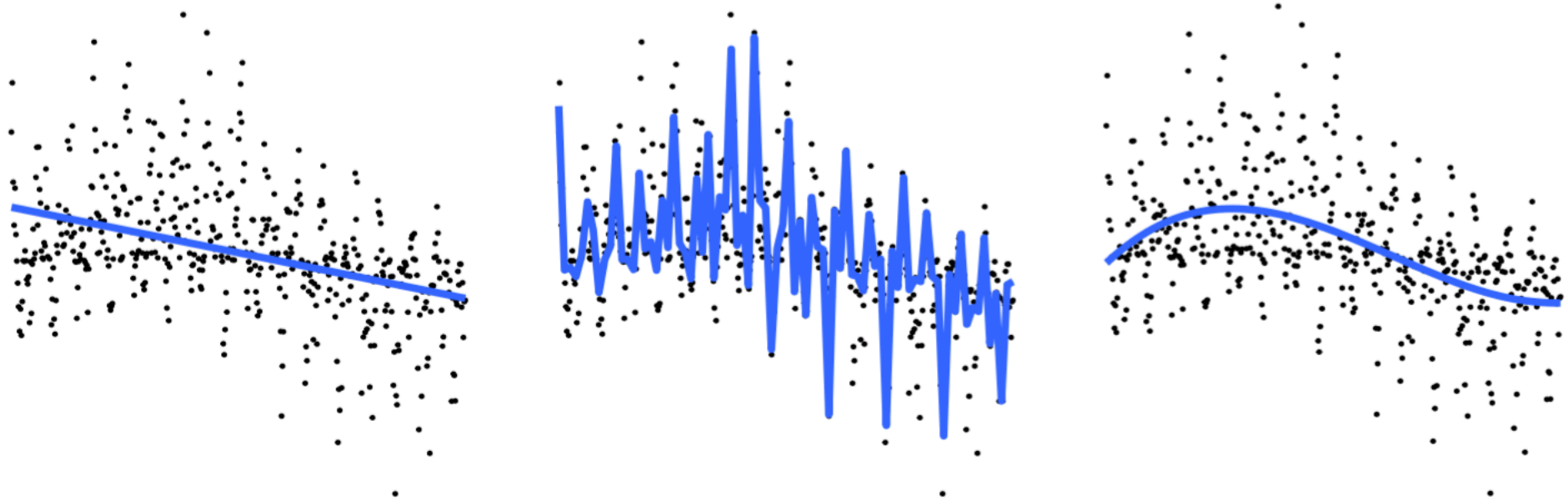
#C. True Positive Rate

$$\text{TPR} = \frac{TP}{TP+FN}$$

What is a common consideration for both regressions and classifiers?

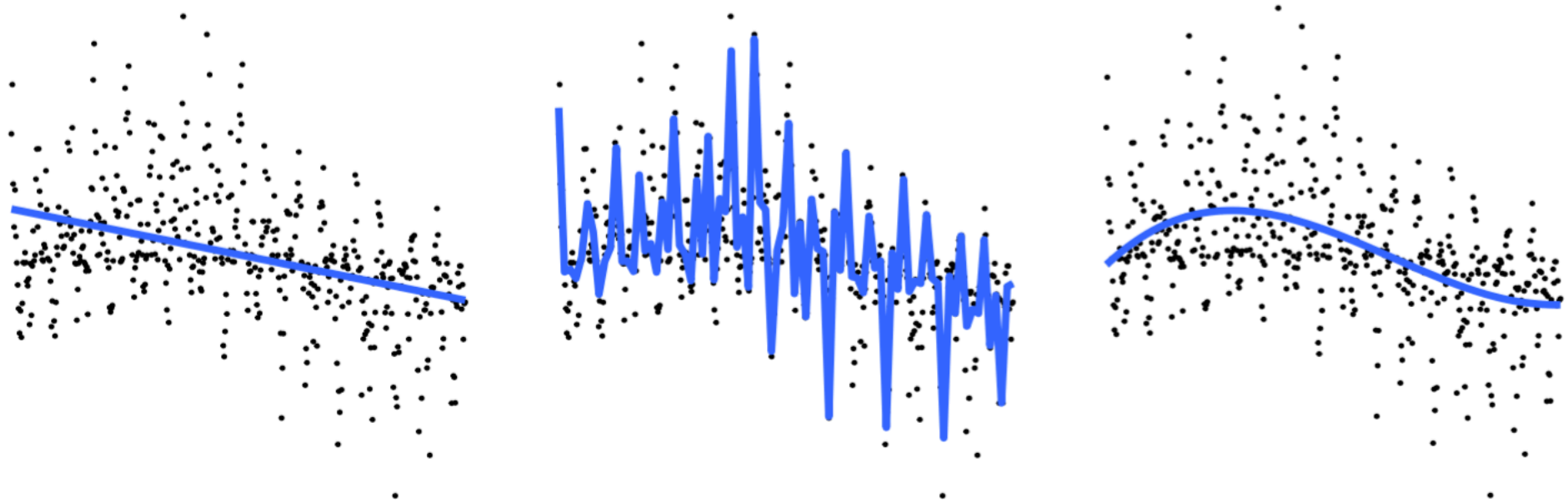
#1. The Bias-Variance Tradeoff

Bias is defined as the erroneous assumption of a model (e.g. overly simplistic, overly complicated). Variance is the error. In short user assumptions need to be gut checked.



What is a common consideration for both regressions and classifiers?

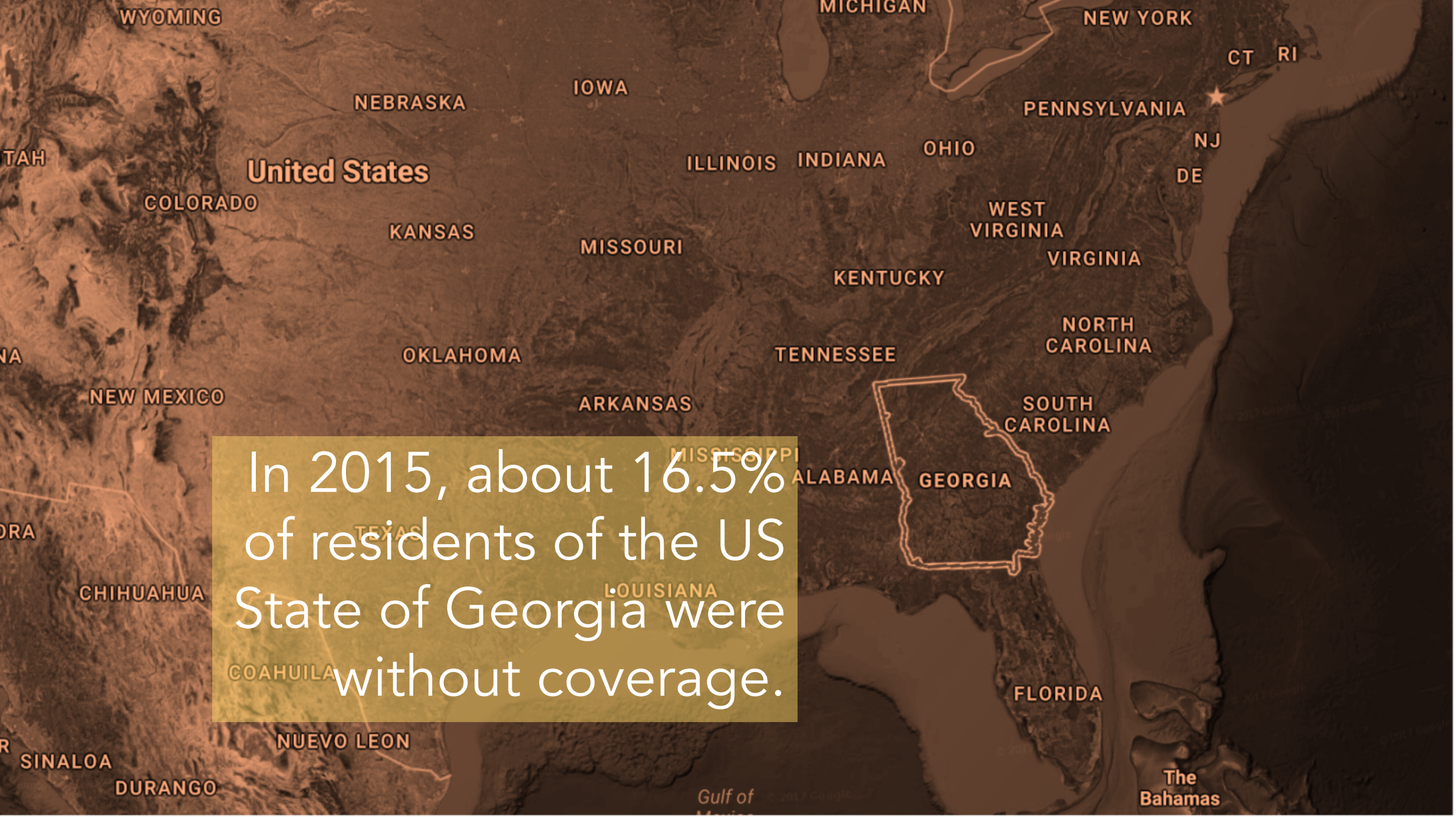
#2. The Bias-Variance Tradeoff type of the model fits is likely to be found in a public policy setting and which in a tech startup?



Roadmap

- A Surprise
- Motivation
- Classification Preliminaries
- Decision Trees
- <Break>
- Random Forests
- Homework #3
- Homework #4 – a head up

Health insurance coverage is a problem in the United States. In some states, there are double-digit percentages of residents without coverage.



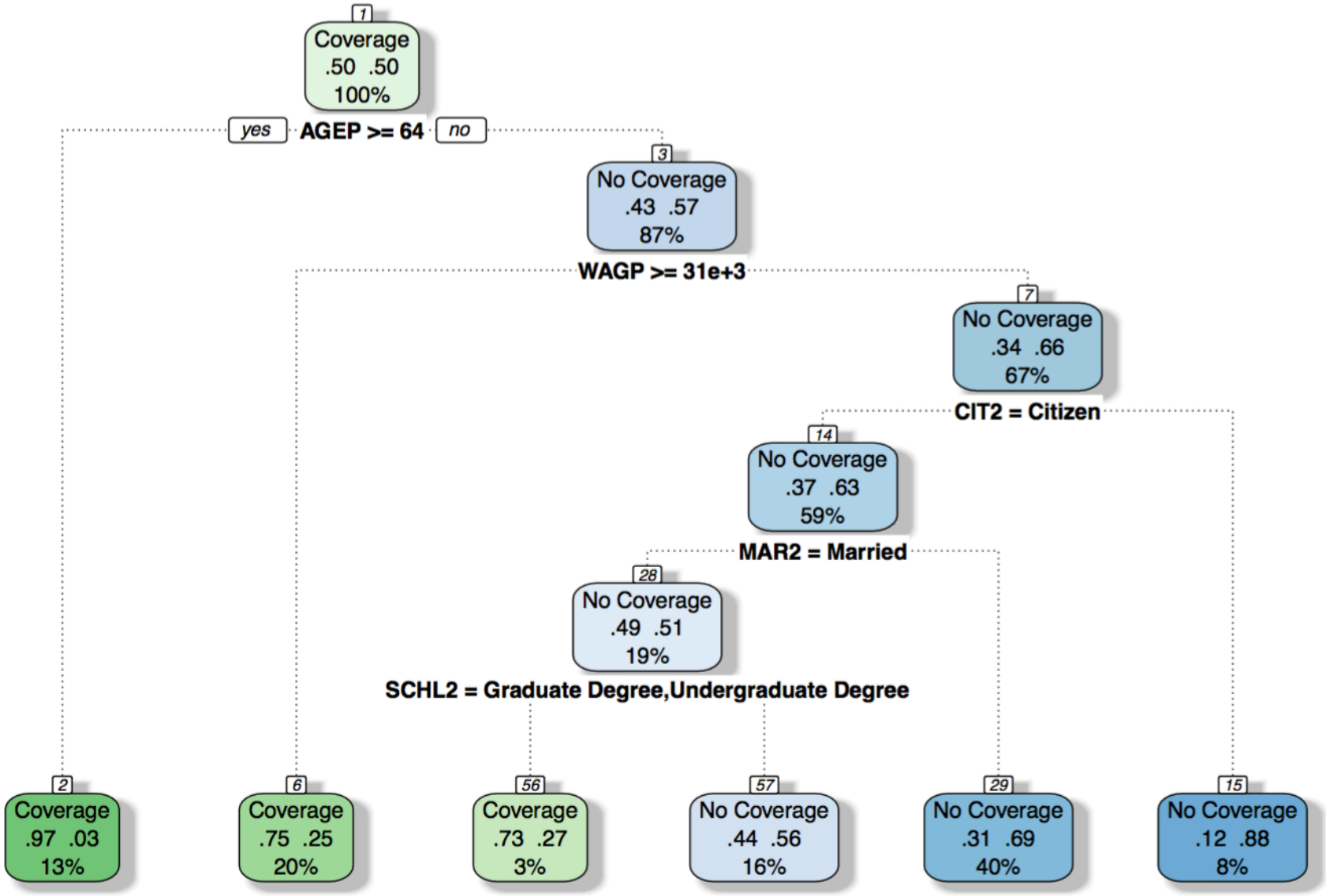
In 2015, about 16.5% of residents of the US State of Georgia were without coverage.

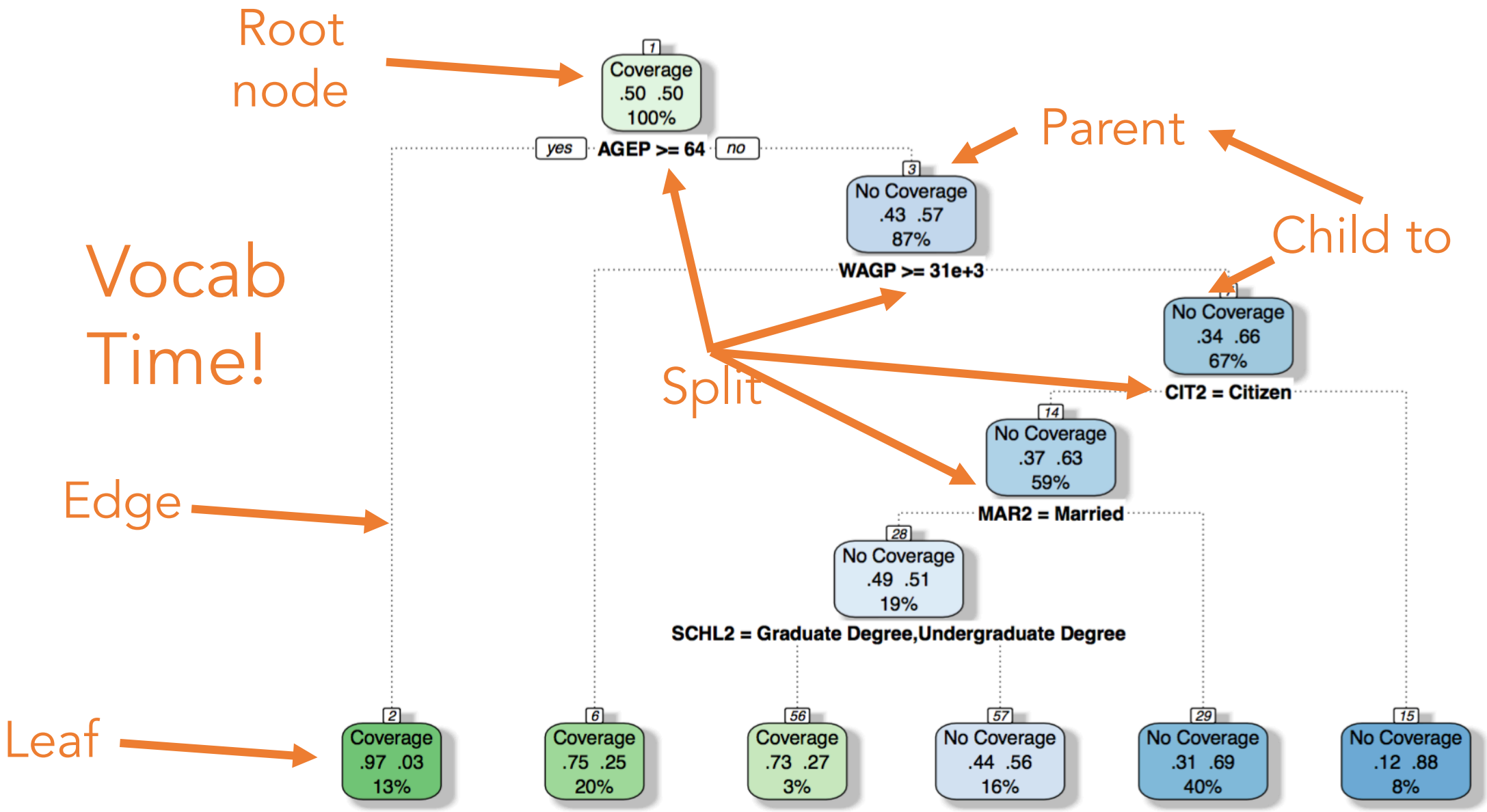
To get people registered and enrolled, 16.5% needs to be decomposed into targetable populations. But which?

Let's say that:

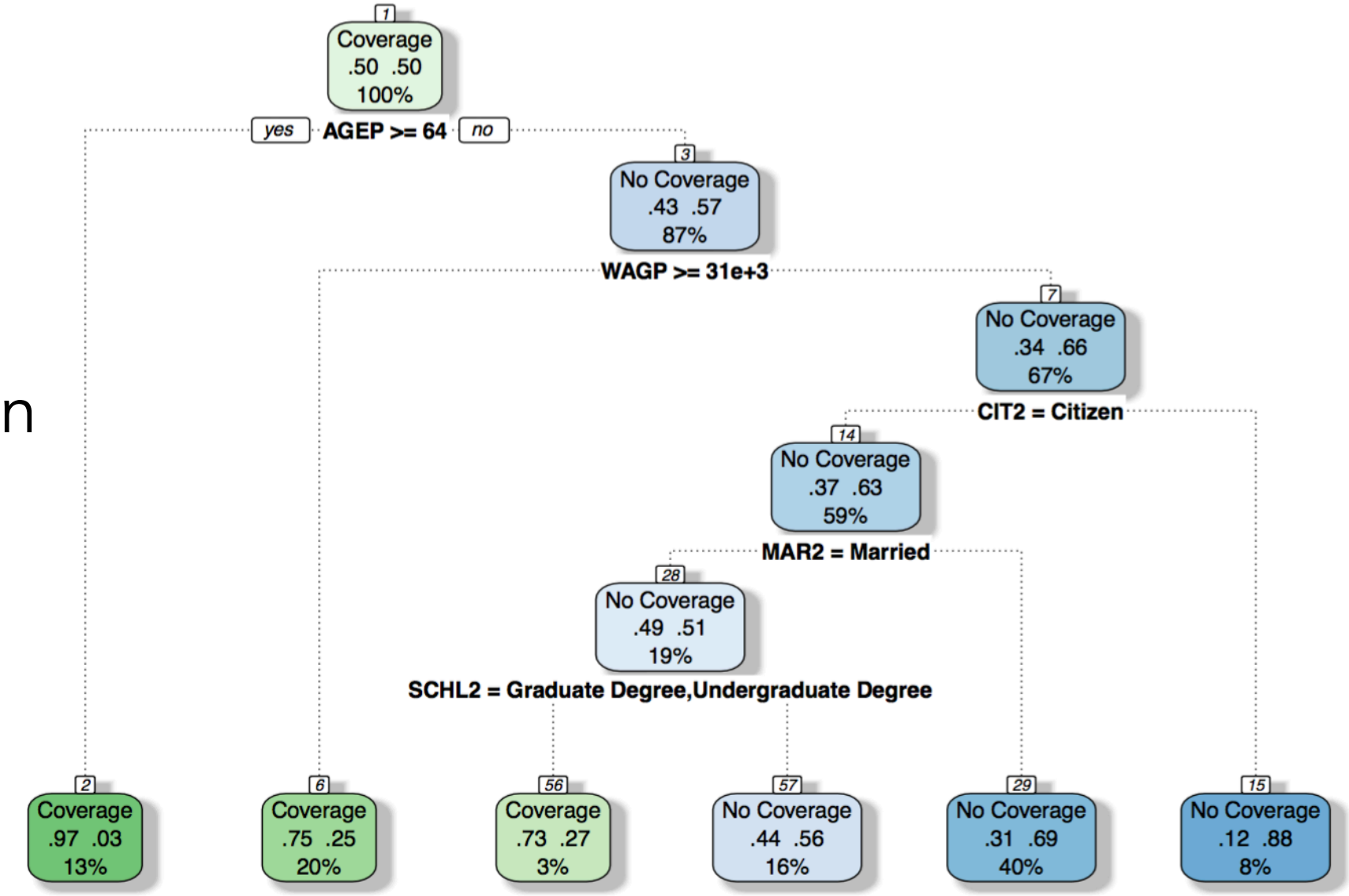
Y(Coverage) = f(Marital Status, Age, Wage, Race, Education, Citizenship)

A decision tree subsets a population into smaller more homogeneous subpopulations using input features.



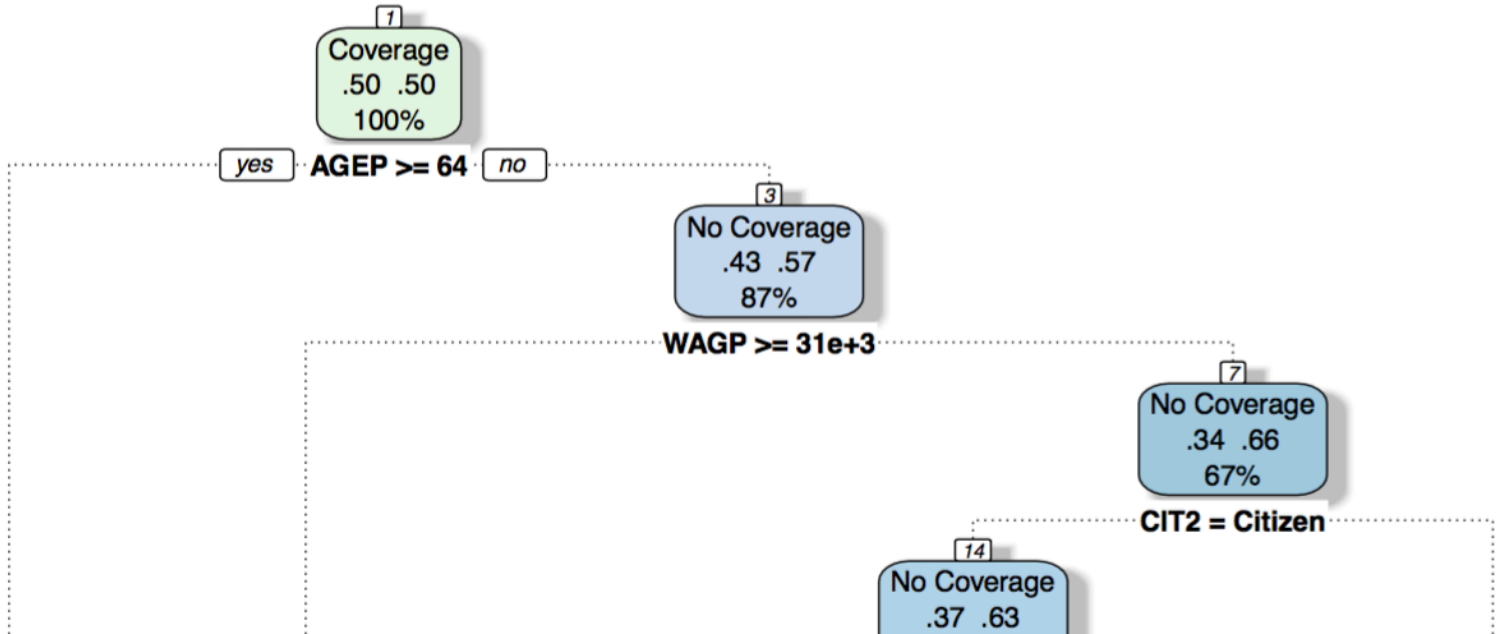

Root node
Vocab Time!
Edge
Leaf
Parent
Child to
Split
1
Coverage
.50 .50
100%
yes
AGEP >= 64
no
3
No Coverage
.43 .57
87%
WAGP >= 31e+3
No Coverage
.34 .66
67%
CIT2 = Citizen
14
No Coverage
.37 .63
59%
MAR2 = Married
28
No Coverage
.49 .51
19%
SCHL2 = Graduate Degree,Undergraduate Degree
2
Coverage
.97 .03
13%
6
Coverage
.75 .25
20%
56
Coverage
.73 .27
3%
57
No Coverage
.44 .56
16%
29
No Coverage
.31 .69
40%
15
No Coverage
.12 .88
8%

Question: How does each variable get chosen? This can be as good as using a pivot table, right??



Decision tree algorithm

1. Let Sample = S, Target = Y, Input Features = X
2. Screen records for cases that meet termination criteria.
   If each base case that is met, partition sample to isolate homogeneous cases.
3. For each X:
   Calculate the attribute test comparing all X's and Y
4. Compare and identify Xi that yields the greatest separability
5. Split S using input feature that maximizes separability
6. Iterate process on steps 3 through 5 until termination criteria is met



Decision tree algorithm

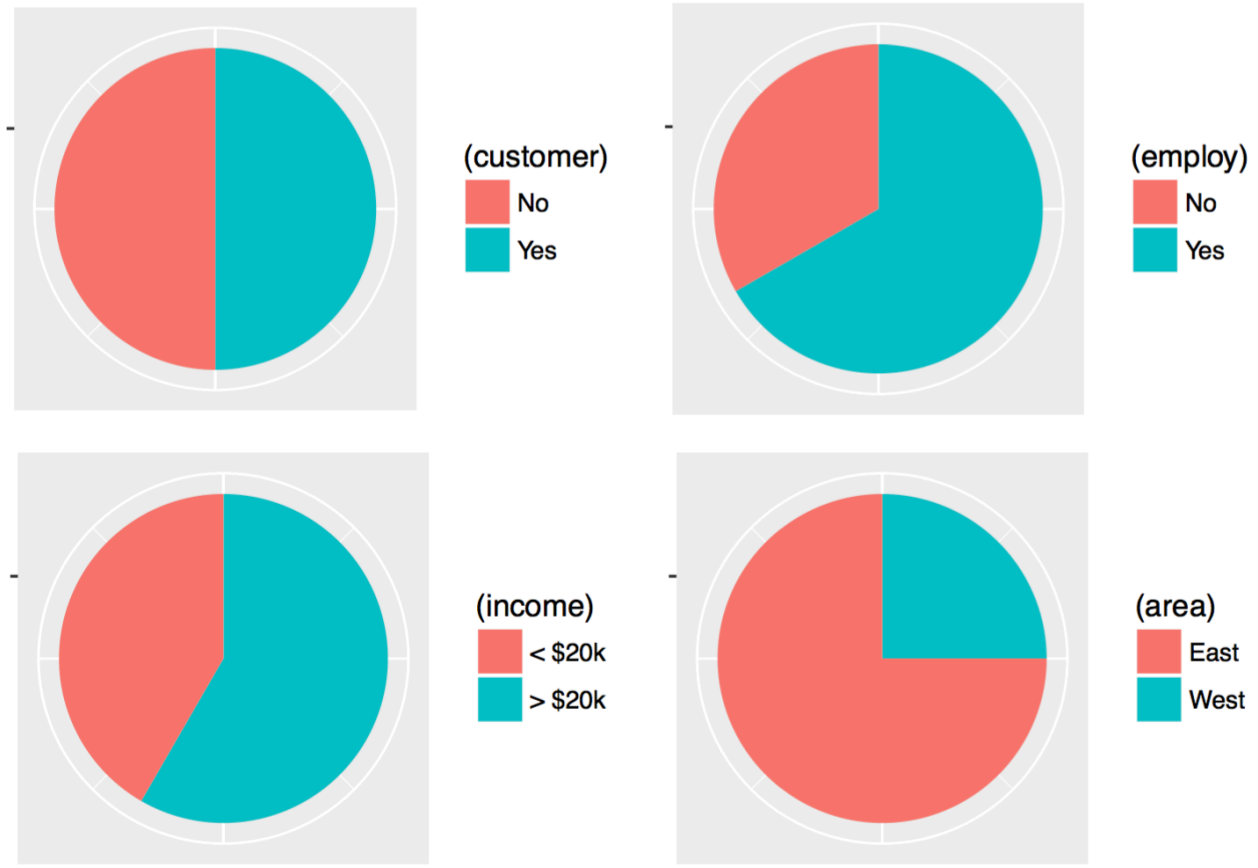
3. Starting from the root node

- For each input feature:
  - Split node into two parts based on some threshold of a given input feature
  - Calculate an "attribute test" for each input feature.
  - Store the result of the test
- After all variables are tested, choose the variable with the best attribute test
- Split node into two

If termination criteria is not met, repeat process on each node.

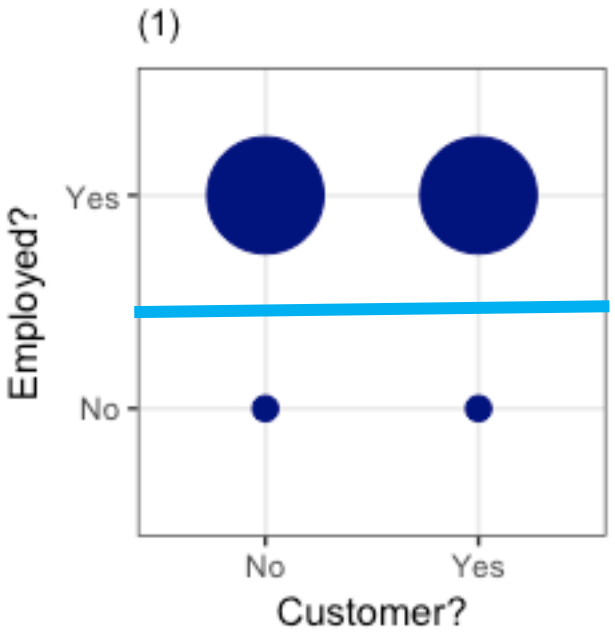
What's an attribute test?

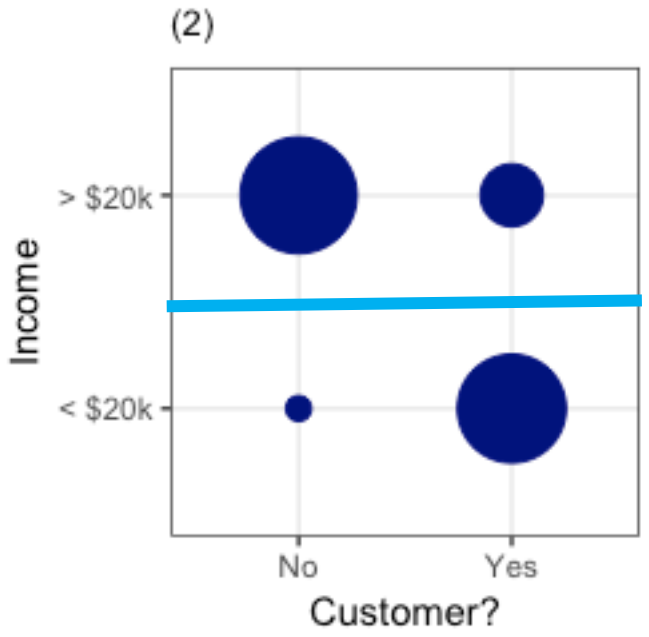
Scenario: Which feature is most associated with being a customer?

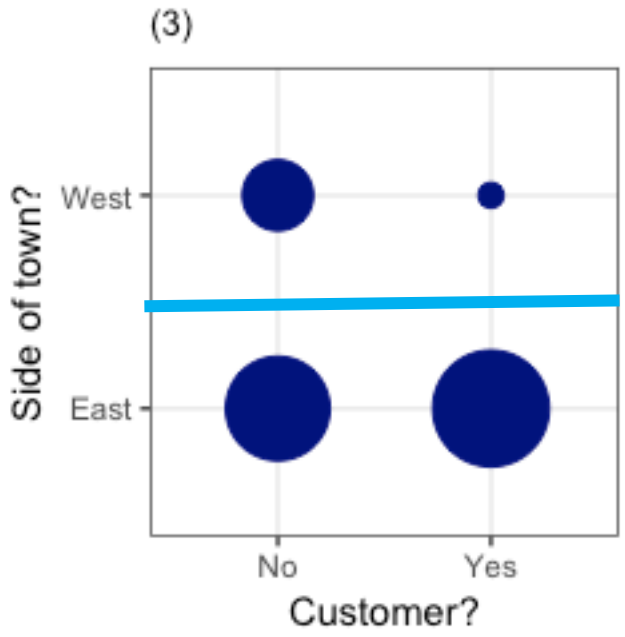


What's an attribute test?

Upon calculating a cross-tab, which of the three plots provides the most information?

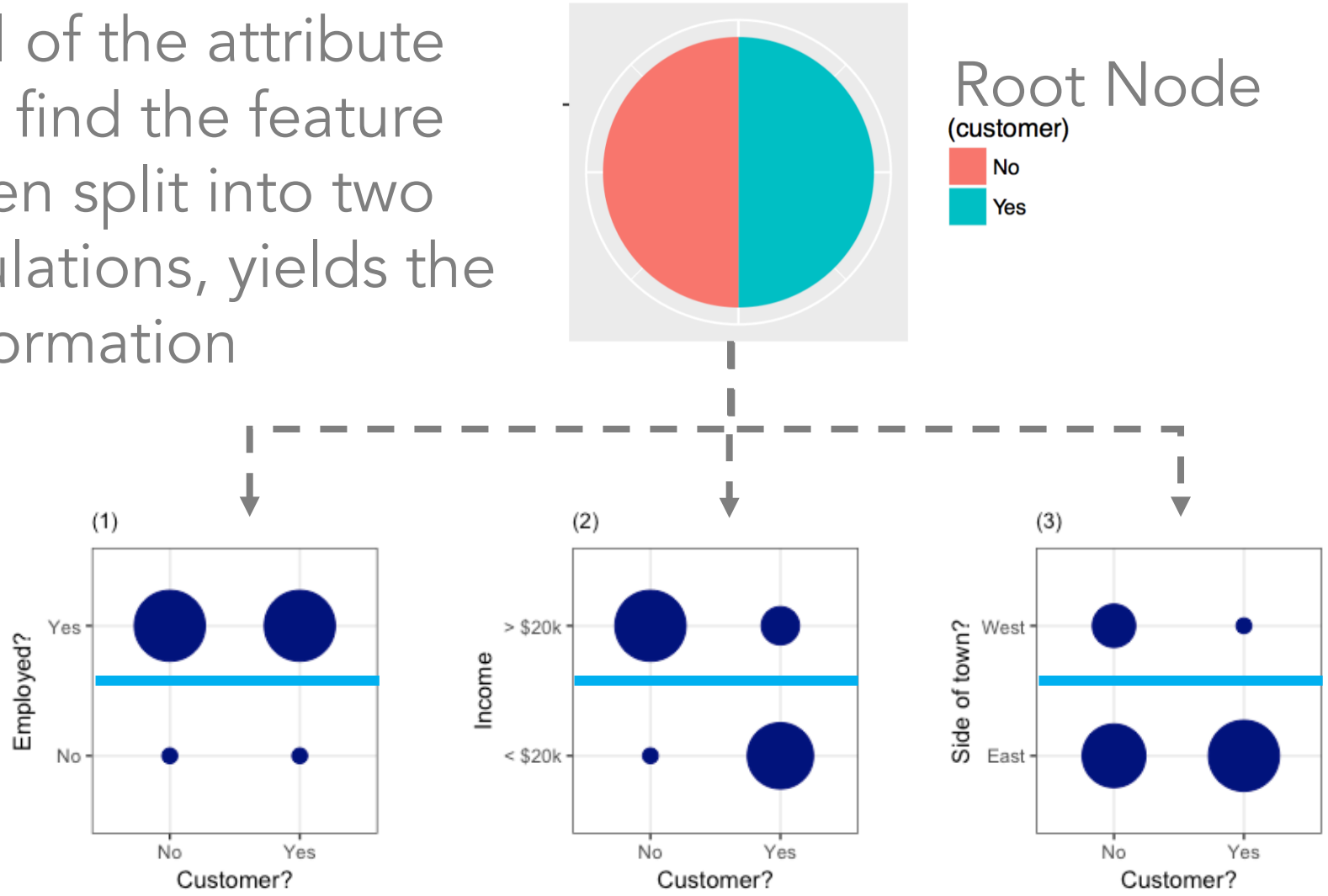






What's an attribute test?

The goal of the attribute test is to find the feature that, when split into two subpopulations, yields the most information



What's an attribute test?

Entropy can be used to calculate consistency of information.

$$\text{Entropy} = \sum -p_i log_2(p_i)$$

Where p is the proportion of a sample that is of class *i*.

What's an attribute test?

Example Calculation.

$$\text{Entropy} = \sum -p_i log_2(p_i)$$

	$< \$20k$	$> \$20k$
No	0	6
Yes	5	1
Total	5	7

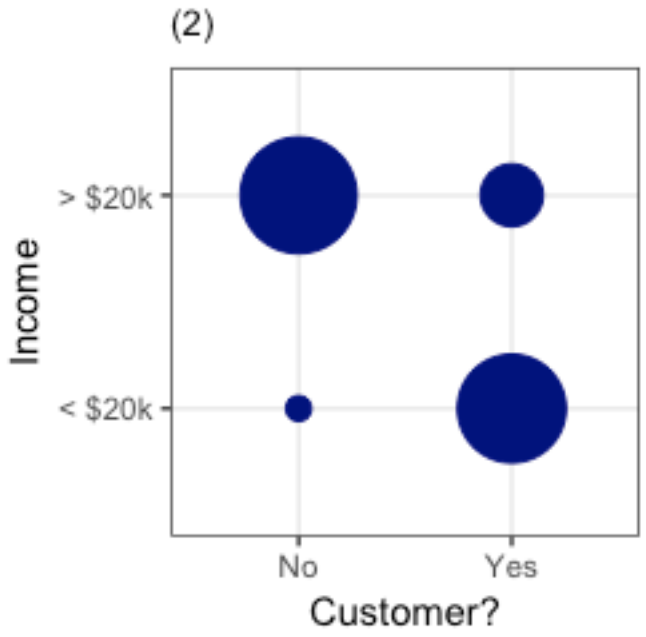
$$\text{Entropy}_{\text{income} < 20k} = (-p_{user} log_2(p_{\text{user}})) - (-p_{\text{non-user}} log_2(p_{\text{non-user}}))$$
$$= -\tfrac{5}{5} log_2(\tfrac{5}{5}) = 0$$

$$\text{Entropy}_{\text{income} < 20k} = (-p_{user} log_2(p_{\text{user}})) - (-p_{\text{non-user}} log_2(p_{\text{non-user}}))$$
$$= -\tfrac{6}{7} log_2(\tfrac{6}{7}) + -\tfrac{1}{7} log_2(\tfrac{1}{7}) = 0.5916728$$

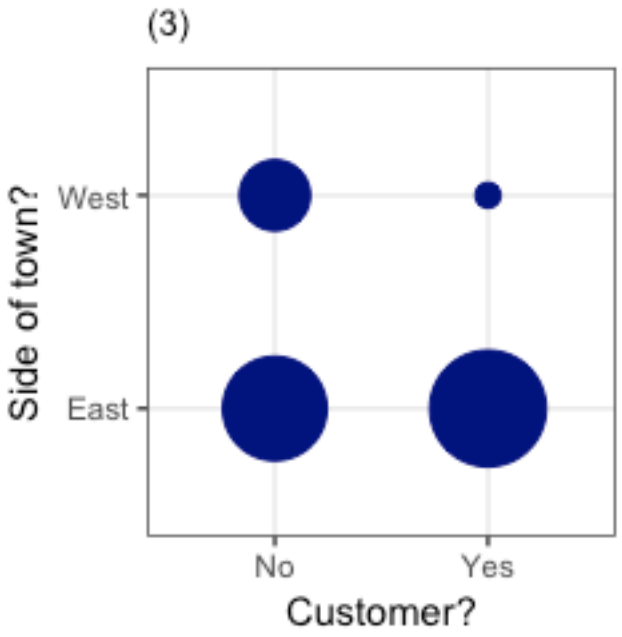
What's an attribute test?

Entropy of each cross-tab

Entropy = 1



Entropy = 0.35



Entropy = 0.97

The actual attribute test is "Information Gain"?

Information Gain is a variant of entropy that compares the entropy of the root to the average entropy of child nodes

$$\text{IG} = \text{Entropy}_{\text{root}} - \text{Avg Child Entropy}$$

The actual attribute test is "Information Gain"?

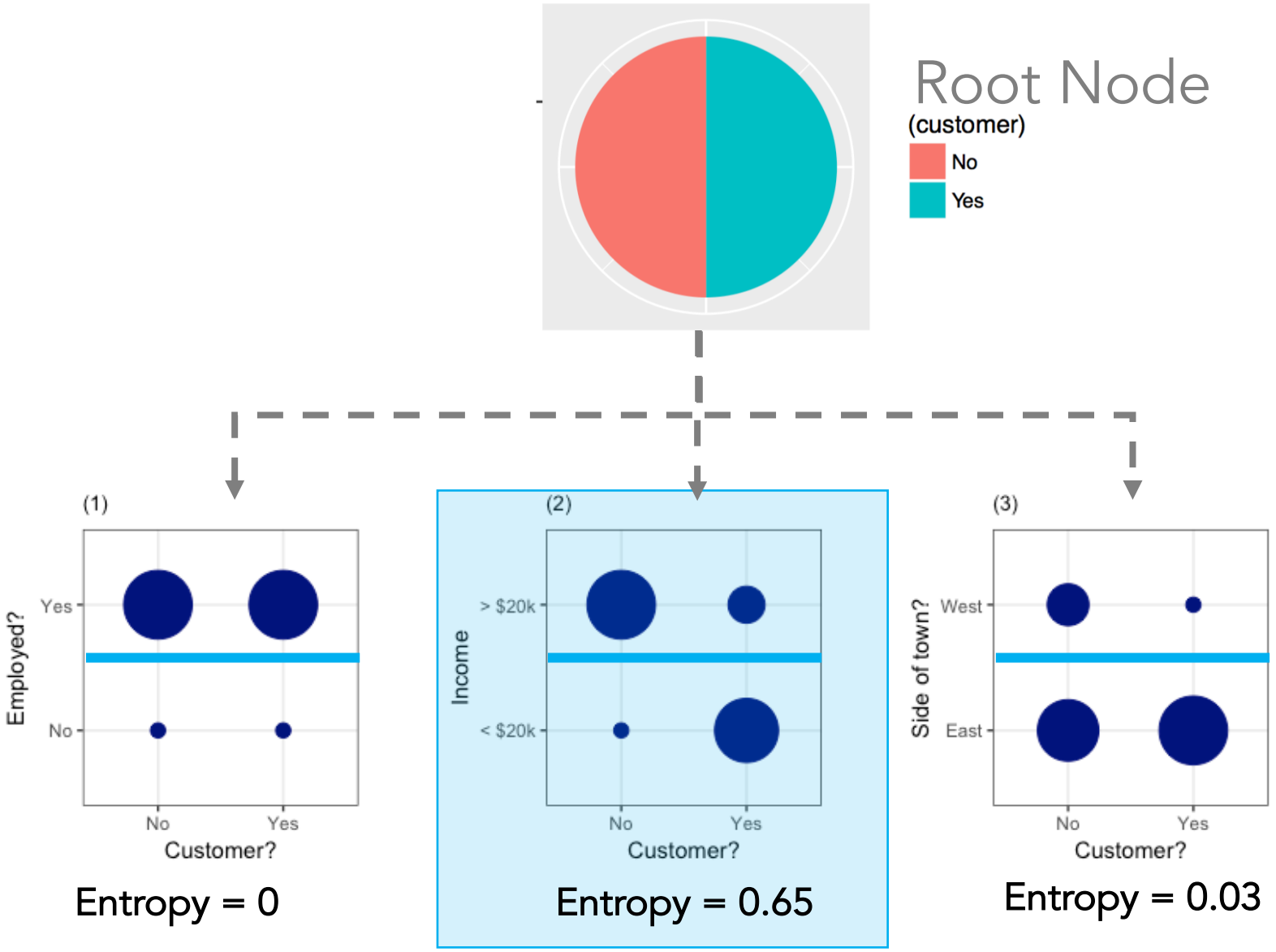
Information Gain is a variant of entropy that compares the entropy of the root to the average entropy of child nodes

$$\text{IG} = \text{Entropy}_{\text{root}} - \text{Avg Child Entropy}$$

$$\begin{aligned}\text{Entropy}_{\text{usership}} &= (-p_{user} log_2(p_{\text{user}})) - (-p_{\text{non-user}} log_2(p_{\text{non-user}})) \\ &= (-\tfrac{6}{12} log_2(\tfrac{6}{12})) + (-\tfrac{6}{12} log_2(\tfrac{6}{12})) \\ &= 1.0\end{aligned}$$

$$\text{Entropy}_{\text{income split}} = \tfrac{5}{12}(0) + \tfrac{7}{12}(0.5916728) = 0.3451425$$

Income was the best option.

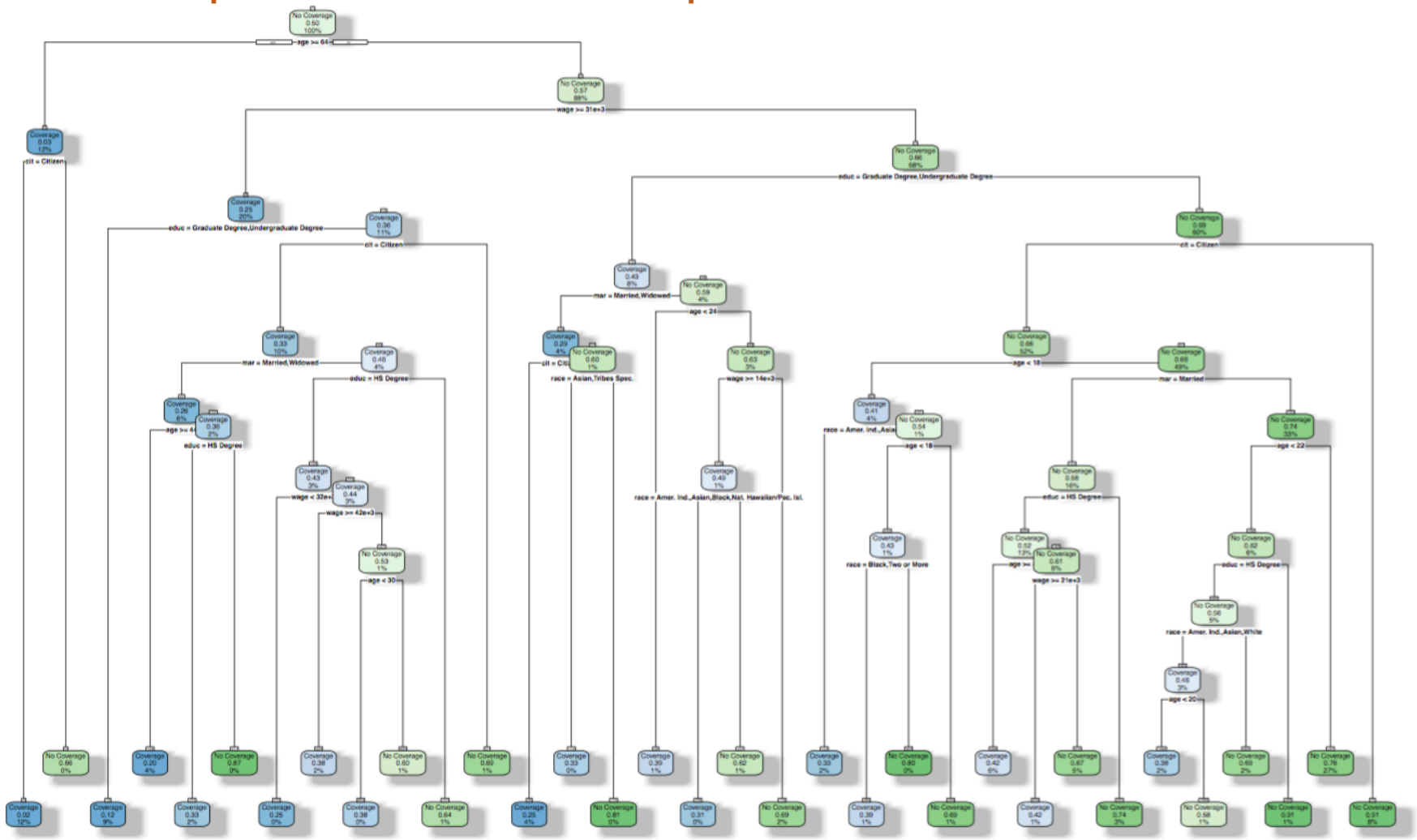


Other attribute tests can also include:

$$\text{IG} = \text{Entropy}_{\text{root}} - \text{Avg Child Entropy}$$

$$\text{Gini Impurity} = \sum p_i(1 - p_i) = 1 - \sum p_i^2$$

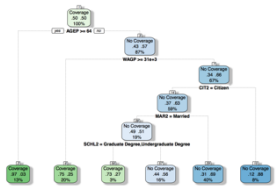
Trees can be grown until there are no more data points to split. What's the problem with that?



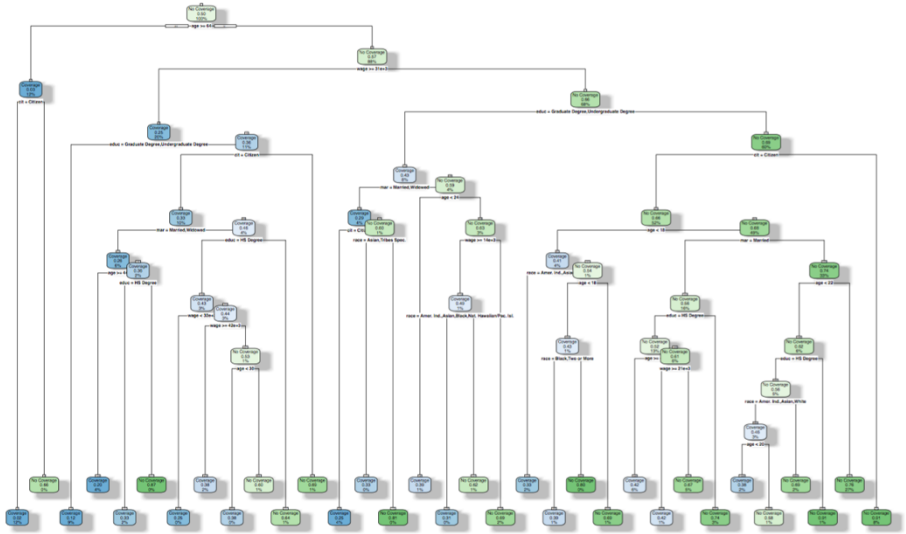
Decision trees can over fit easily.

The Bias-Variance Tradeoff

Trees that are built on too few observations may be built on noise. Trees that are overly simplified suffer from omitted variable bias.



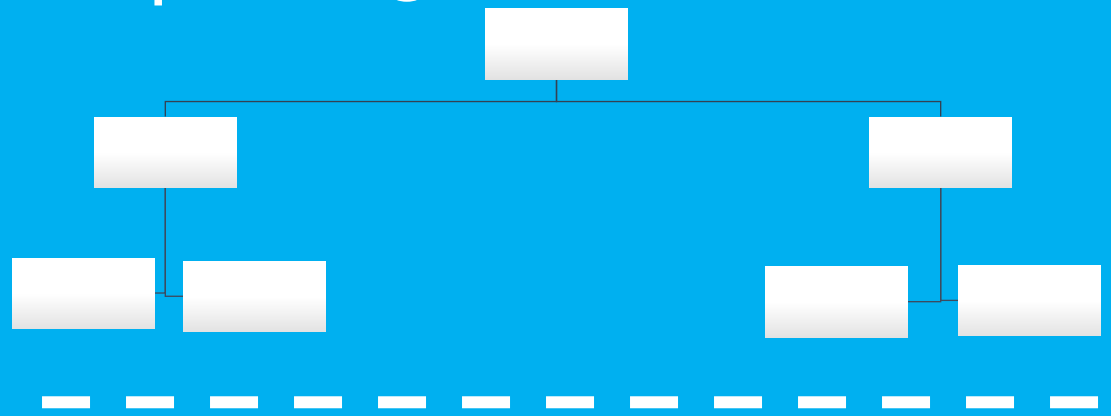
Vs.



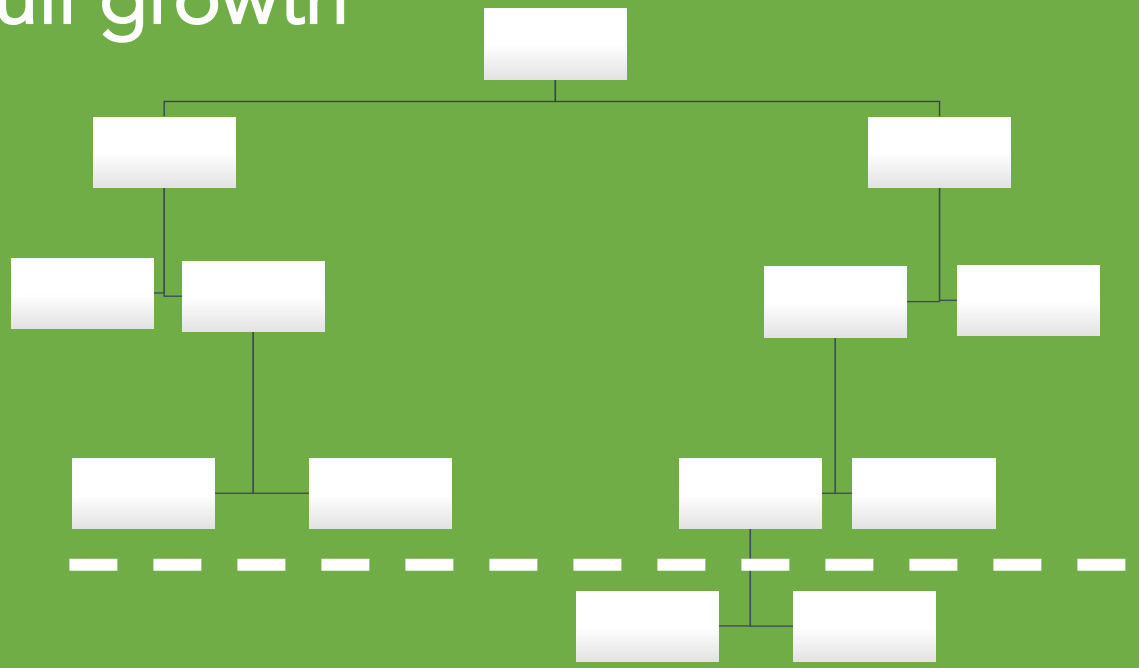
Tuning is the process of calibrating the model to minimize overfitting

Stop criteria vs. Pruning

Stop tree growth if criteria met



Prune or cut down the tree after full growth



Preventing [or minimizing] overfitting

Stop criteria vs. Pruning

Stop tree growth if criteria met

- A node has fewer records than a pre-specific threshold;
- The purity or information gain falls below a pre-specified level or is equal to zero;
- The tree is grown to n-number of levels

Prune or cut down the tree after full growth

- Calculate the tree's error after each split
- Prune tree at optimal number of nodes

There are issues though.

Stop criteria vs. Pruning

Stop criteria tend to under fit the tree.

Pruning is computationally costly (Complex trees take forever)

How do we know which variable matters the most?

$$\text{Variable Importance}_k = \sum \text{Goodness of Fit}_{\text{split, k}} + (\text{Goodness of Fit}_{\text{split,k}} \times \text{Adj. Agreement}_{\text{split}})$$

Variable's importance is the sum of all the contributions variable k makes towards predicting the target.

Basically the sum of all Gini purity values for whenever variable k caused a node split.

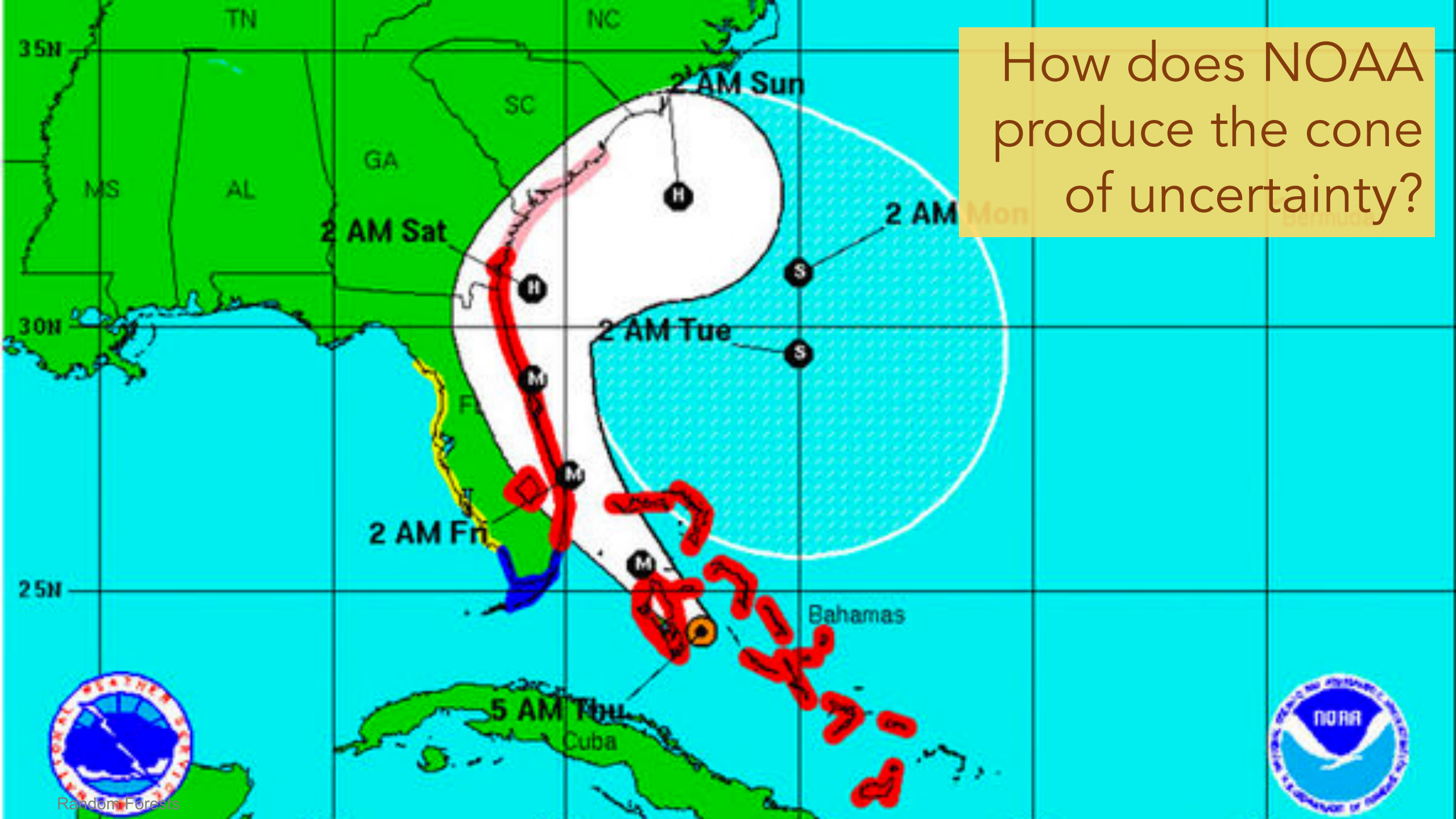
Decision Trees: Strengths and Weaknesses

Strengths	Weakness
- Rules (e.g. all the criteria that form the path from root to leaf) can be directly interpreted. - Method is well-suited to capture interactions and non-linearities in data. - Technique can accept both continuous and continuous variables without prior transformation. - Feature selection is conducted automatically	- Data sets with large number of features will have overly complex trees that, if left unpruned, may be too voluminous to interpret. - Trees tend to overfitted at the terminal leafs when samples are too small.

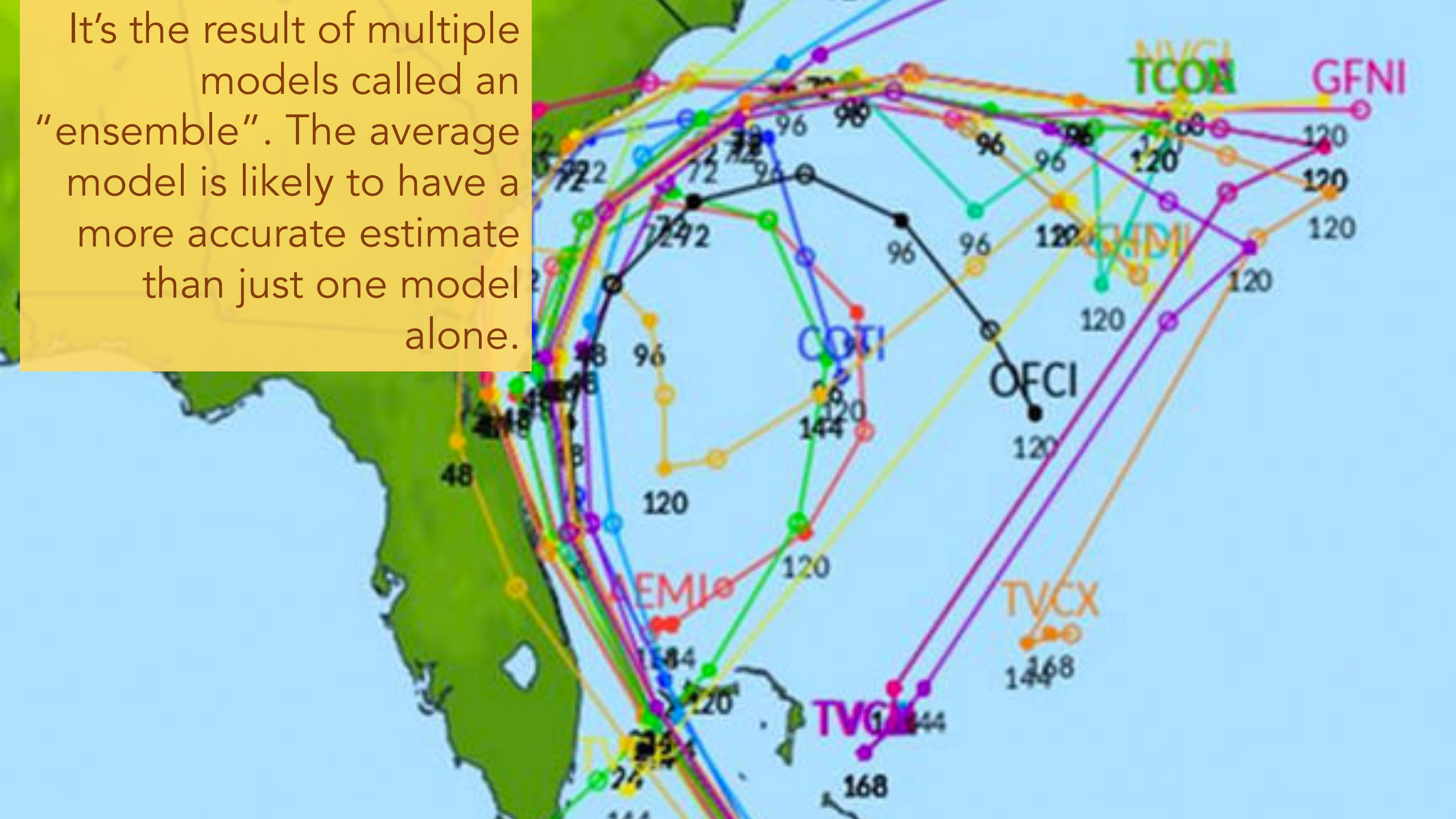
CODE-ALONG

Roadmap

- A Surprise
- Motivation
- Classification Preliminaries
- Decision Trees
- <Break>
- Random Forests
- Homework #3
- Homework #4 – a head up


How does NOAA produce the cone of uncertainty?
TN
NC
SC
GA
MS
AL
35N
30N
25N
2 AM Sun
2 AM Sat
2 AM Mon
2 AM Tue
2 AM Fri
5 AM Thu
Bahamas
Cuba
Random Forests

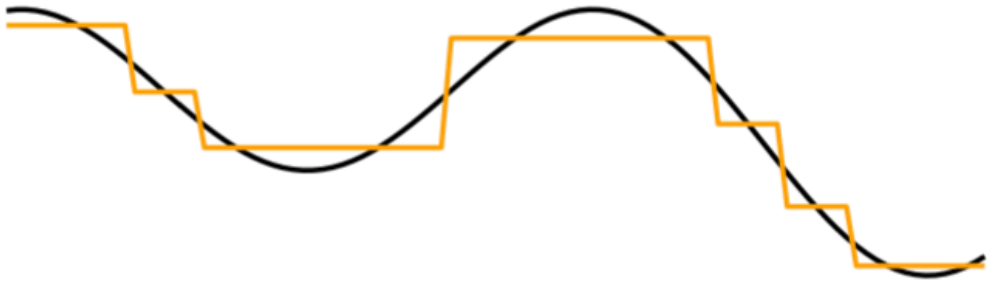
It's the result of multiple models called an "ensemble". The average model is likely to have a more accurate estimate than just one model alone.




ensemble
Decision tree – ensemble –
~~ensemble – pen~~

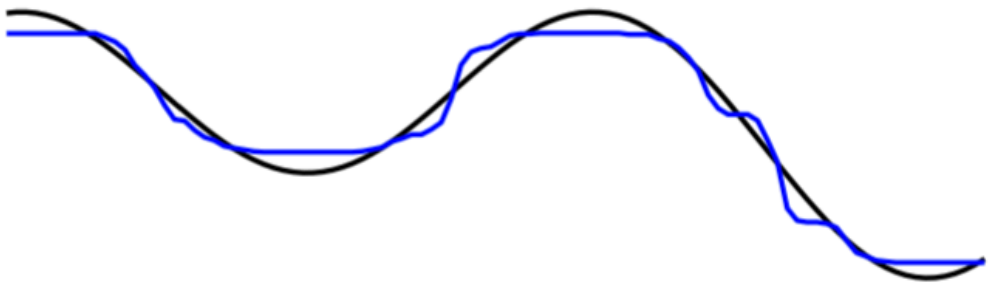
(1) Actual

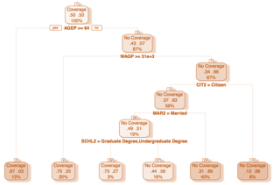
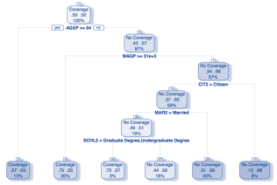
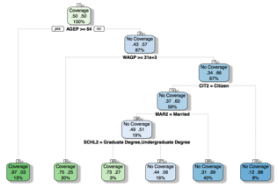
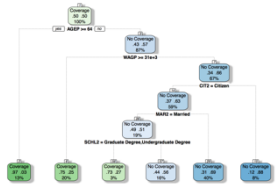
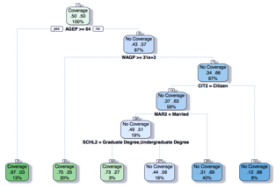
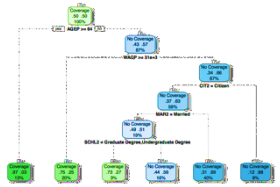
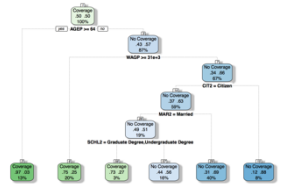
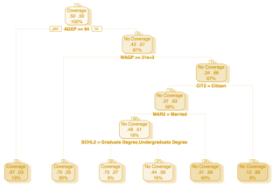
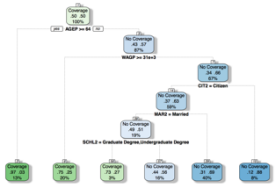
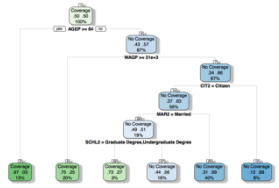
(2) Single Tree



(3) 50 Models

(4) Ensemble Average



Random Forests are an extension of decision tree learning that relies on an ensemble of models to asymptotically predict targets.

Concepts: Random Forests rely on fundamentals of **Bootstrapping --** any process that involves sampling with replacement. This can help with characterizing uncertainty and improving the mean estimate.

```
 y <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
 sample(y, replace=T)
[1] 10  9  4  3  2  1 10  3  6  9
 sample(y, replace=T)
[1]  7  2  4  4  9 10 10  7  1  4
 sample(y, replace=T)
[1]  5  1  5  3 10  1  5  8  9  5
 sample(y, replace=T)
[1]  9  3  9  2  1  2  9 10  3  8
```

Aggregating the results of each bootstrap is called "Bagging" or "Bootstrap Aggregating". Bagging is known as a "meta-algorithm" that relies on other models.

```
 y <- c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
 sample(y, replace=T)
[1] 10  9  4  3  2  1 10  3  6  9
 sample(y, replace=T)
[1]  7  2  4  4  9 10 10  7  1  4
 sample(y, replace=T)
[1]  5  1  5  3 10  1  5  8  9  5
 sample(y, replace=T)
[1]  9  3  9  2  1  2  9 10  3  8
```

Variable sampling involves sampling from variables. At scale, a model where the specification is randomly selected has the same bias as any other model from the draw.

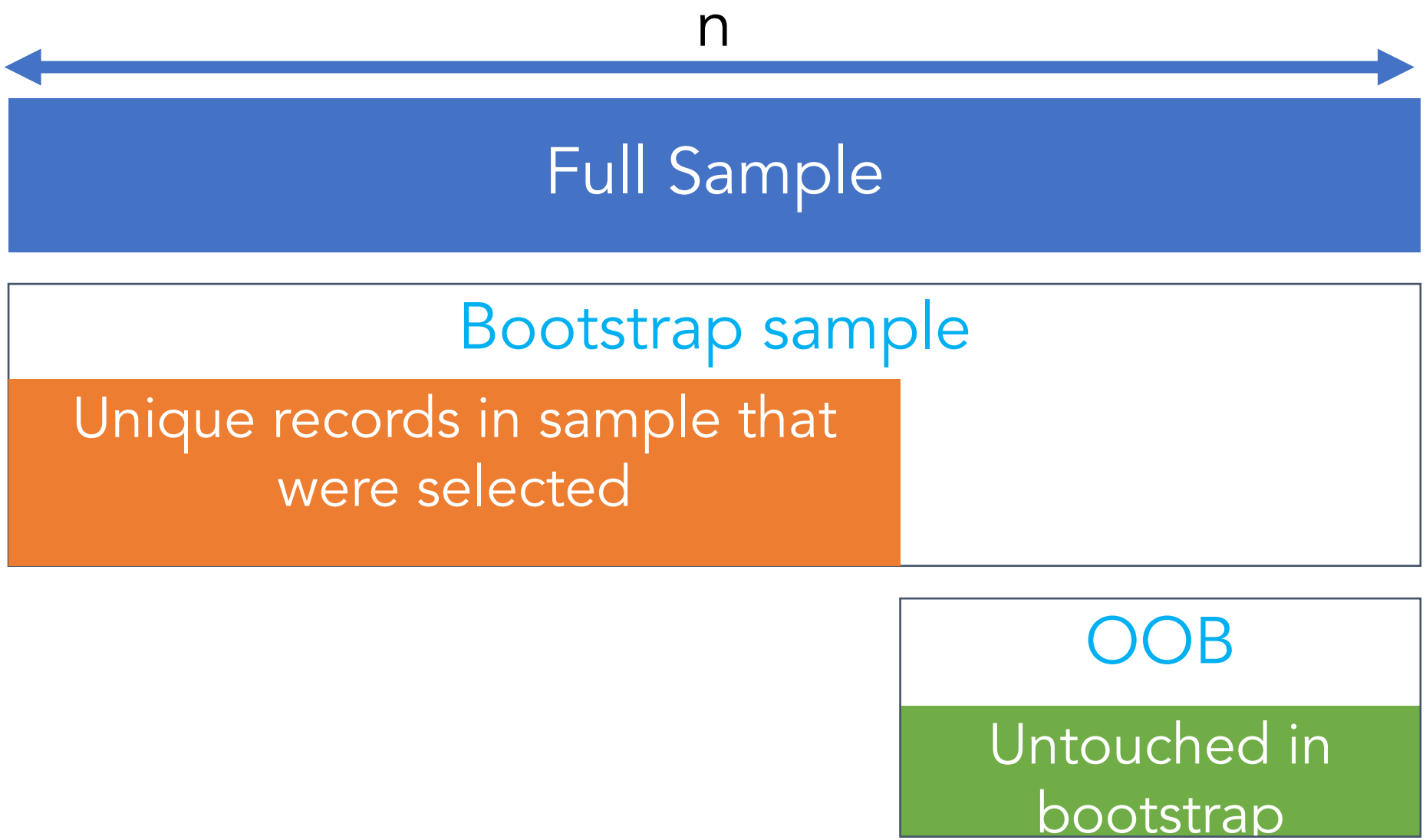
```
> colnames(df)
[1] "x1" "x2" "x3" "x4" "x5" "x6" "x7" "x8" "x9"
> sample(colnames(df), 2, replace=F)
[1] "x5" "x6"
> sample(colnames(df), 2, replace=F)
[1] "x1" "x2"
> sample(colnames(df), 2, replace=F)
[1] "x4" "x1"
> sample(colnames(df), 2, replace=F)
[1] "x3" "x1"
```

Random Forests involve growing a forest of decision trees where observations are bootstrapped and variables are sampled. Each tree is run unpruned.

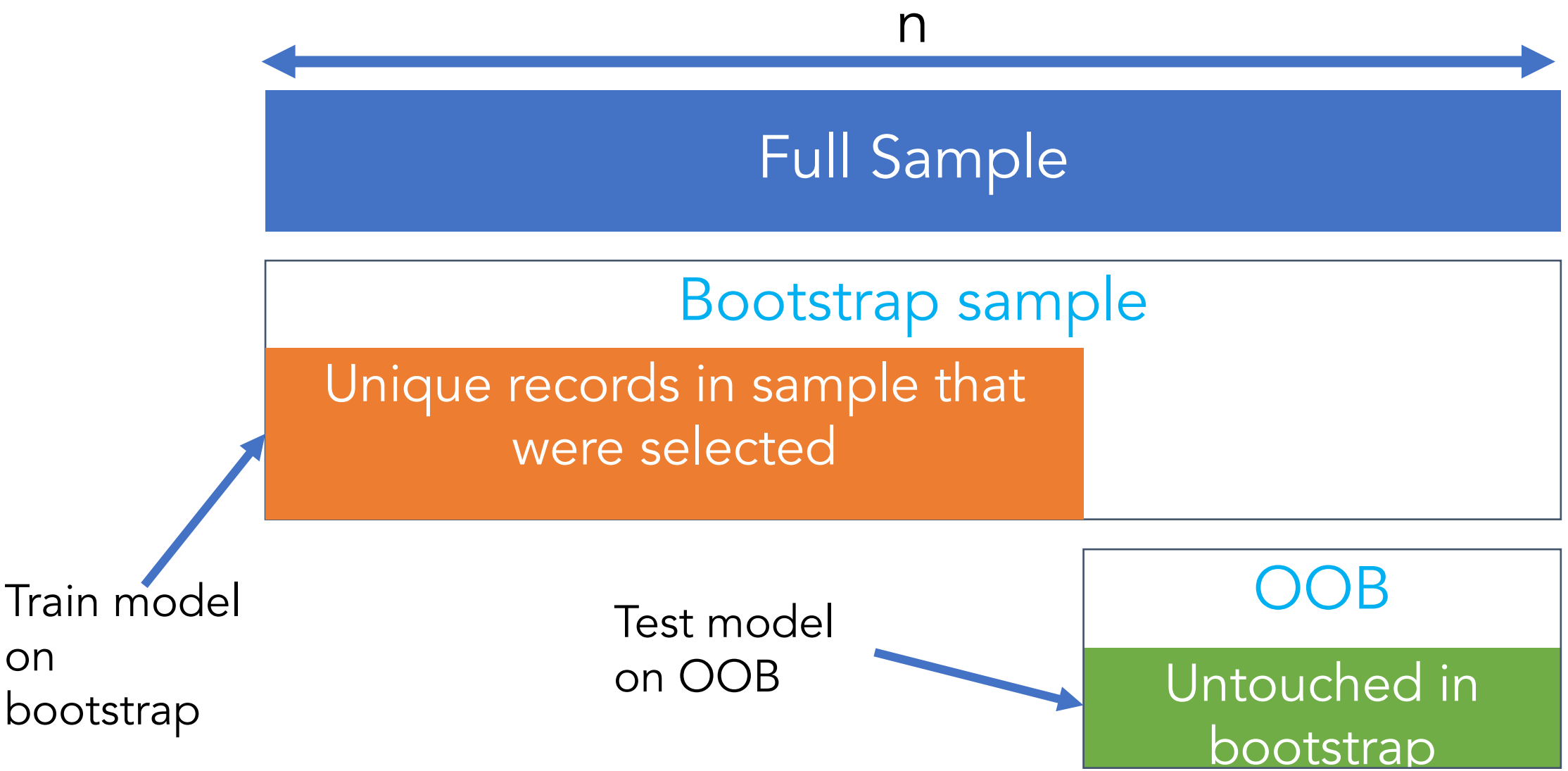
The Pseudo code

```
Let S = training sample, K = number of input features
  1. Randomly sample S cases with replacement from the original data.
  2. Given K features, select k features at random where k < K.
  3. With a sample of s and k features, grow the tree to its fullest complexity.
  4. Predict the outcome for all records.
  5. Out-Of-Bag (OOB). Set aside the predictions for records not in the s cases.
Repeat steps 1 through 5 for a large number of times saving the result after each tree.
Vote and average the results of the tree to obtain predictions.
Calculate OOB error using the stored OOB predictions.
```

What is OOB?


n
Full Sample
Bootstrap sample
Unique records in sample that were selected
OOB
Untouched in bootstrap

Why OOB?


n
Full Sample
Bootstrap sample
Unique records in sample that were selected
Train model on bootstrap
Test model on OOB
OOB
Untouched in bootstrap

Graphical Example: Iteration 0

	X1	X2	X3	X4	X5	X6
1						
2						
3						
4						
5						

Graphical Example: Iteration 1

Selected

OOB

	X1	X2	X3	X4	X5	X6
1						
2						
3						
4						
5						

Graphical Example: Iteration 2

Selected

OOB

	X1	X2	X3	X4	X5	X6
1						
2						
3						
4						
5						

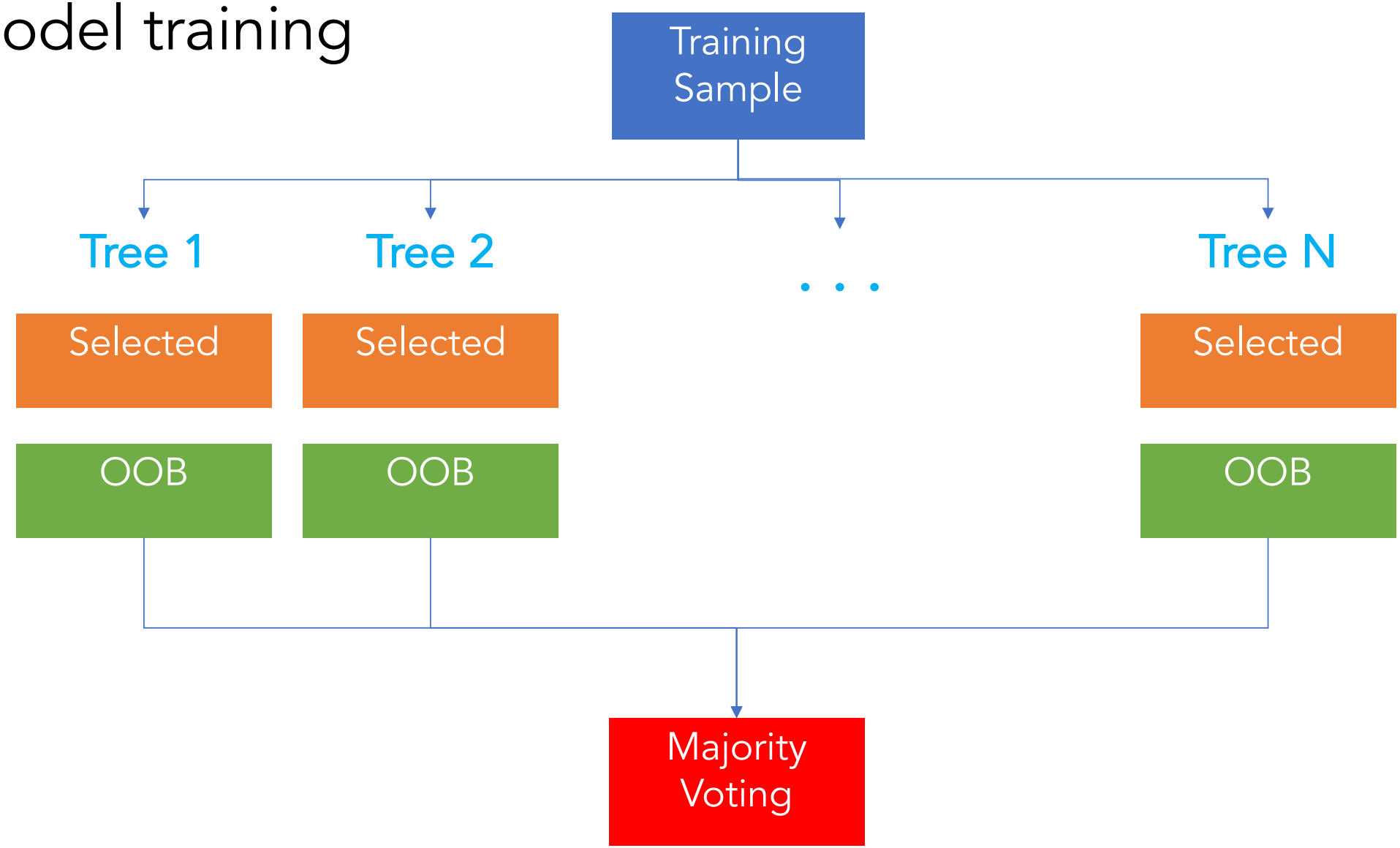
Graphical Example: Iteration N

Selected

OOB

	X1	X2	X3	X4	X5	X6
1						
2						
3						
4						
5						

Model training



Tuning is the process of calibrating the model to minimize overfitting

of trees vs. # of variables

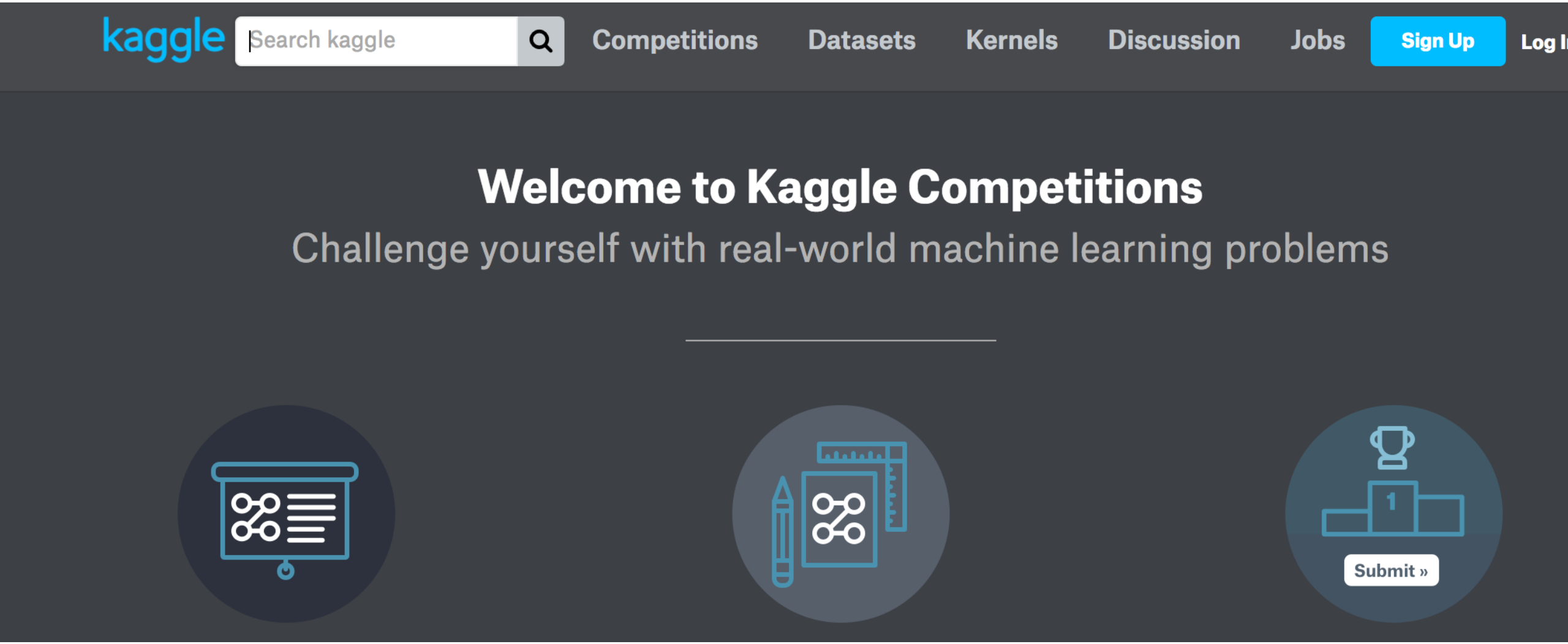
The number of trees is important for deriving stable estimates of variable importance.

Beyond a certain threshold, more trees doesn't improve accuracy.

# of variables influences the depth and richness of models

Variable importance is interpreted using the Mean Decrease Gini

Random Forests are the technique of choice for most Kaggle competitions.



CODE-ALONG