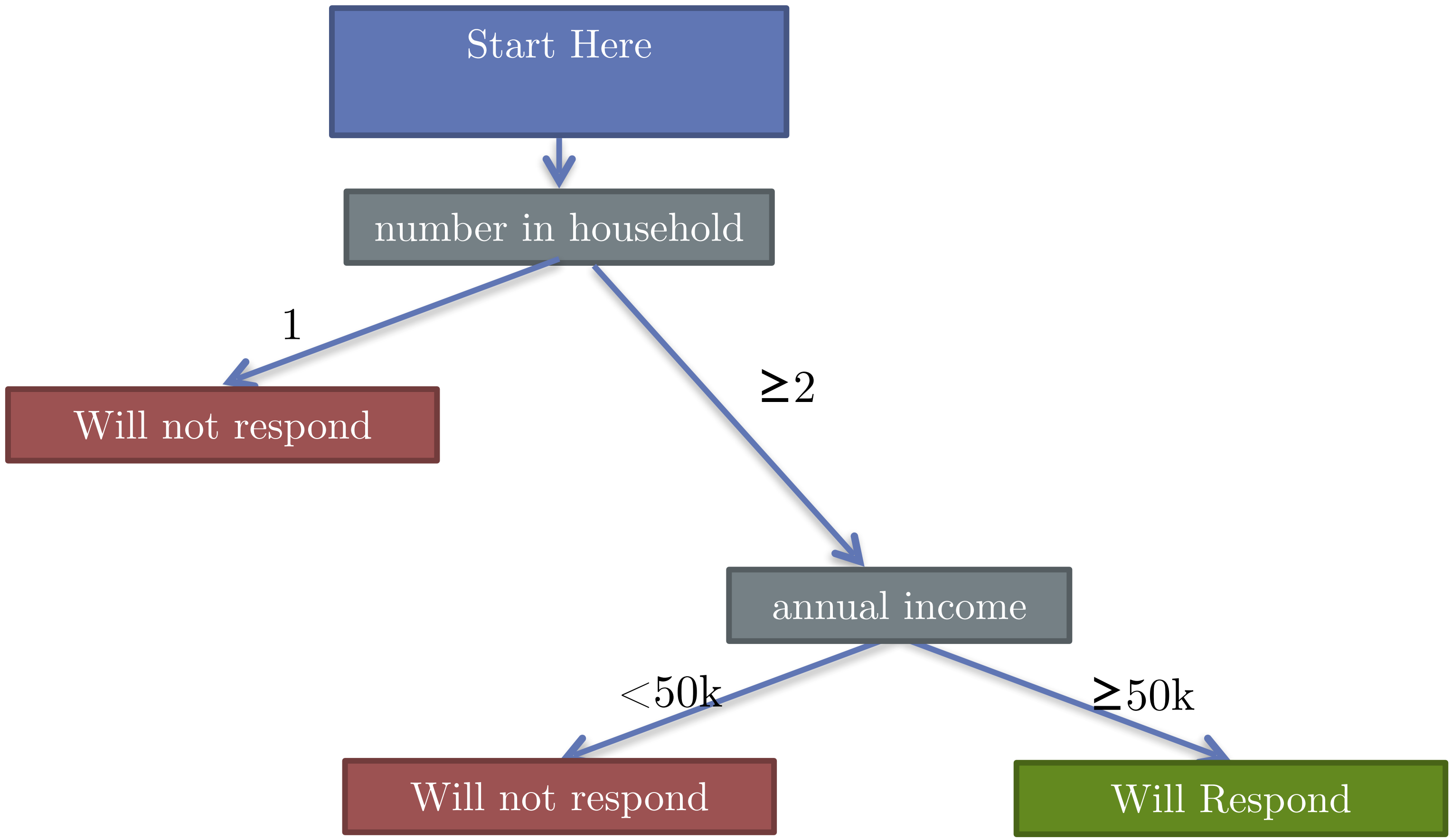
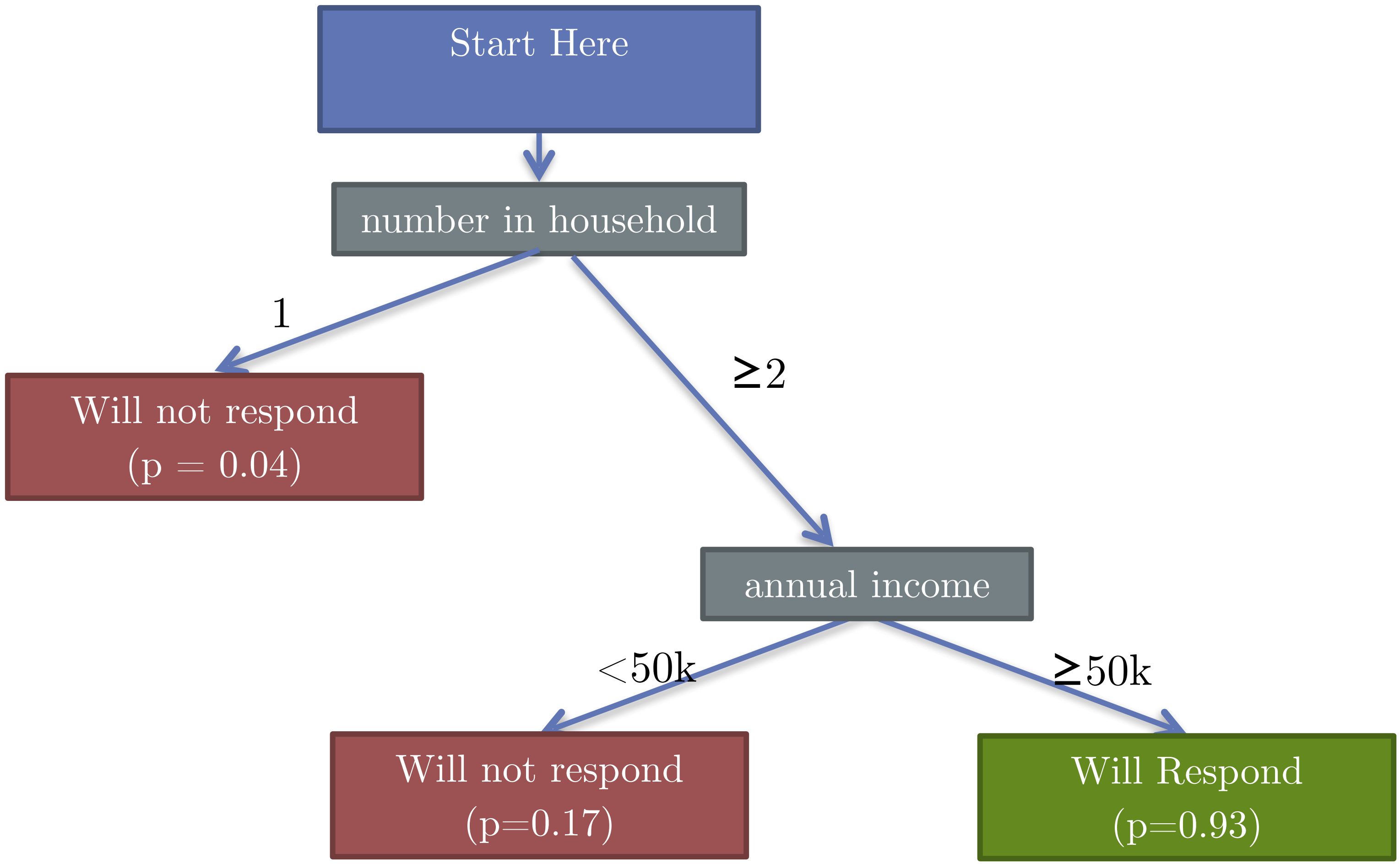
Classification And Regression Trees (CARTs)

a.k.a. Decision Trees

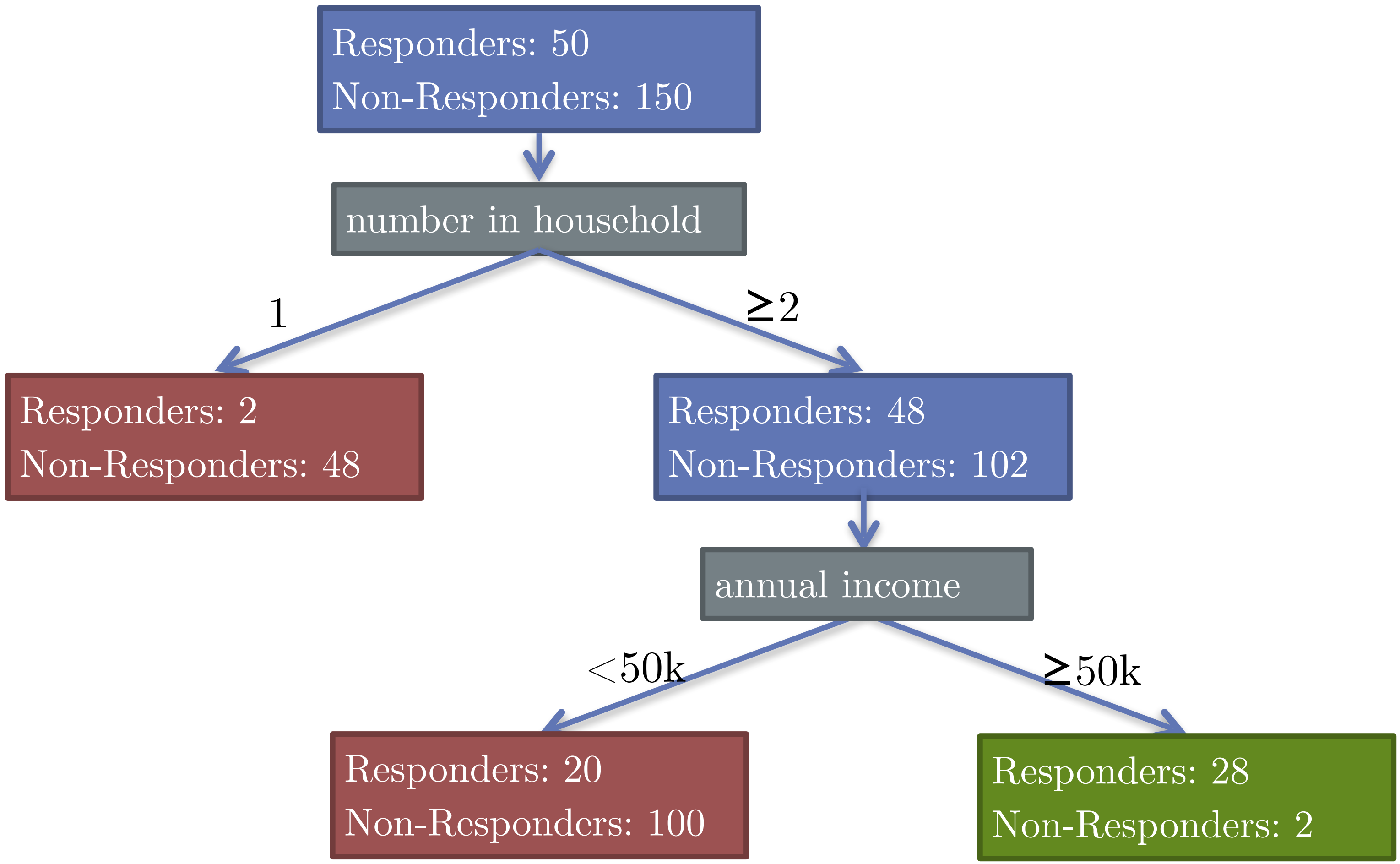
A Decision Tree Model



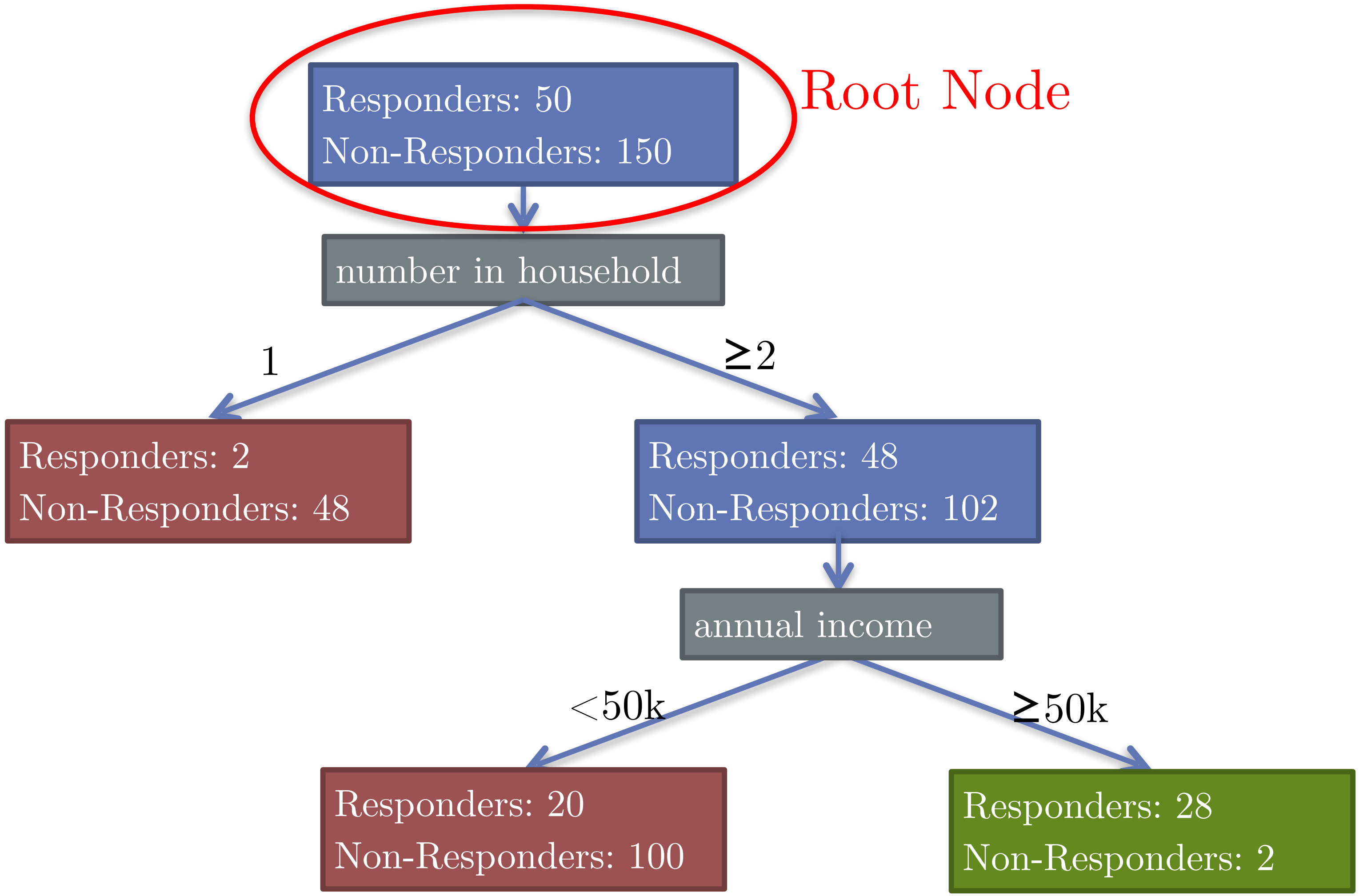
A Decision Tree Model



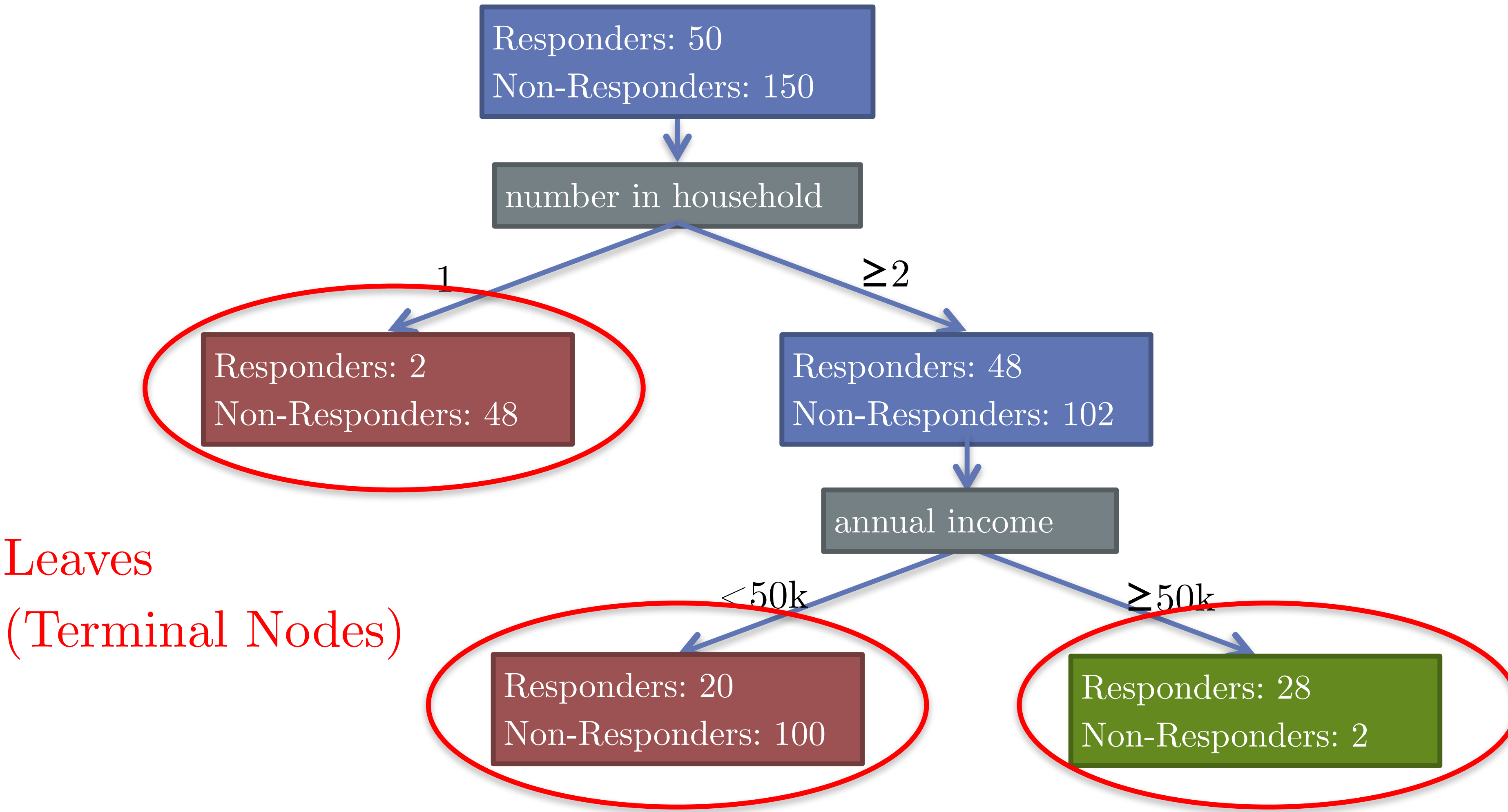
Decision Tree Model Creation



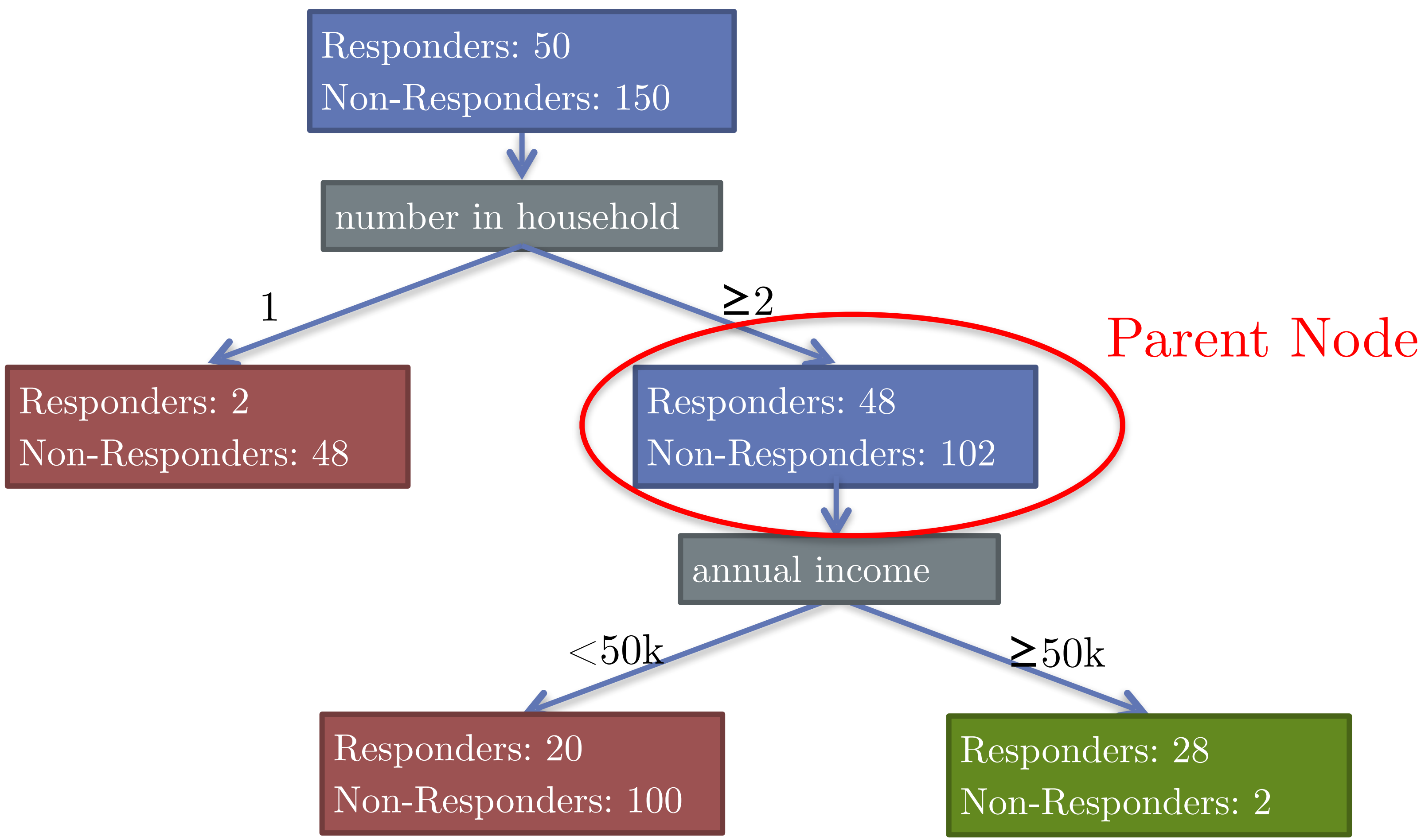
Decision Tree Model Creation



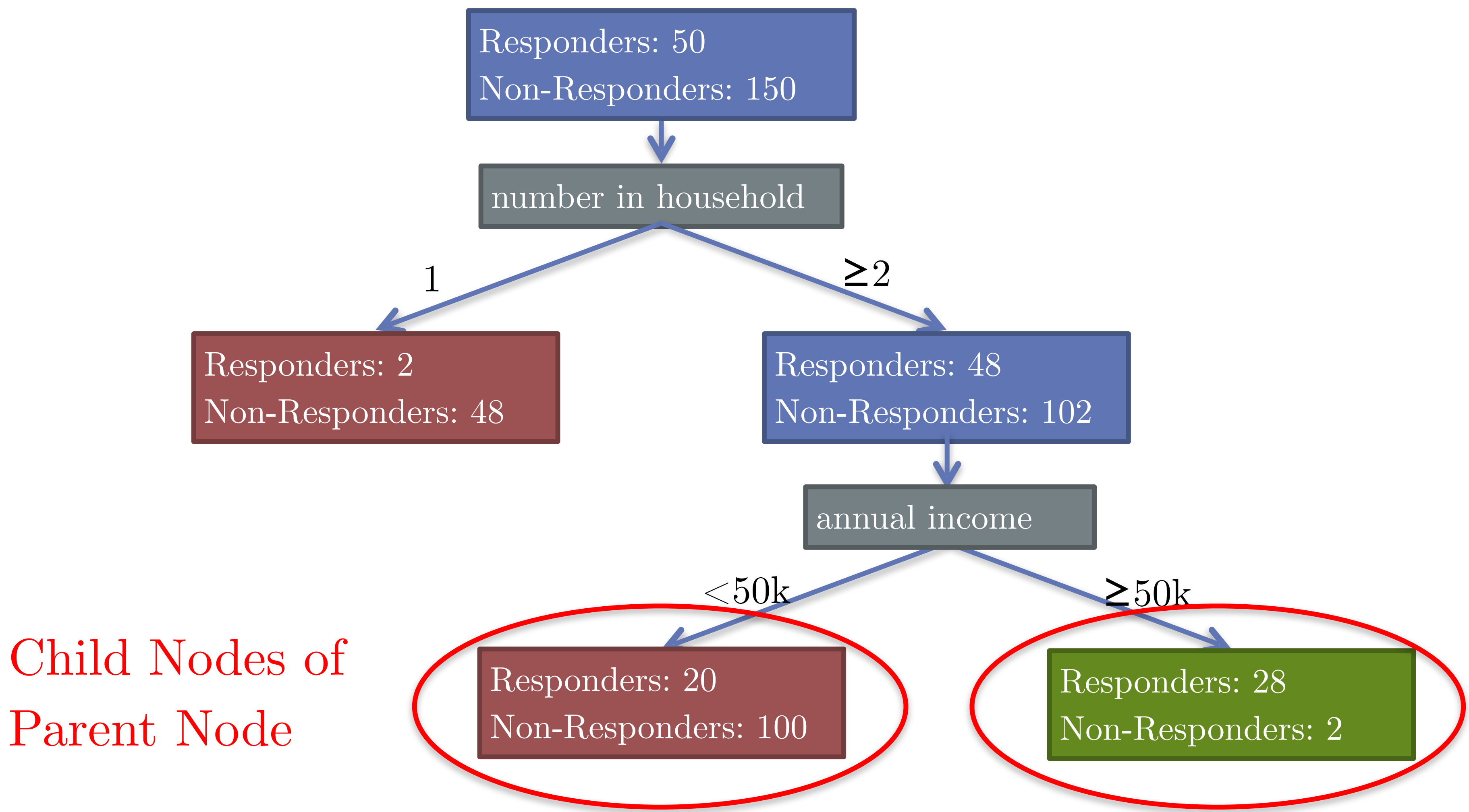
Decision Tree Model Creation



Decision Tree Model Creation



Decision Tree Model Creation



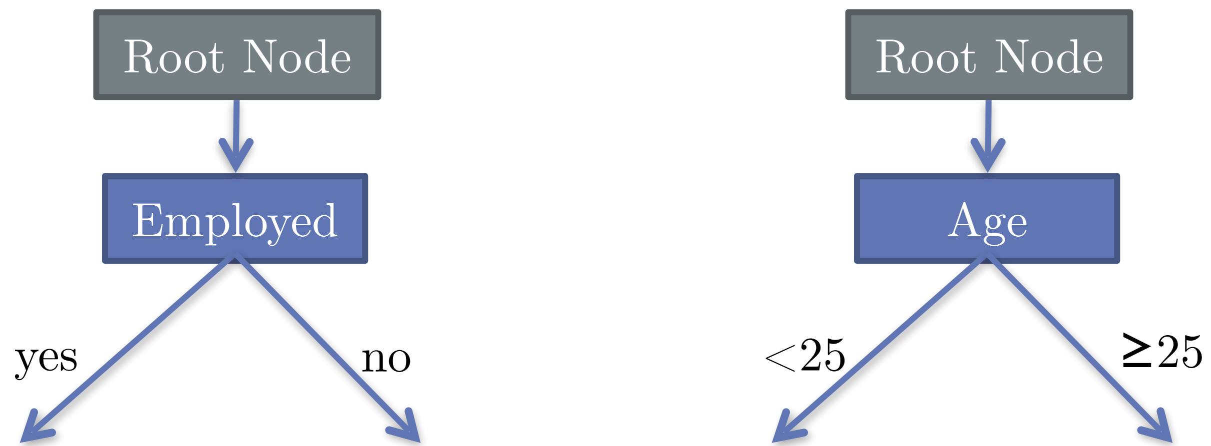
Classification Trees

• • •

Categorical/Ordinal Targets

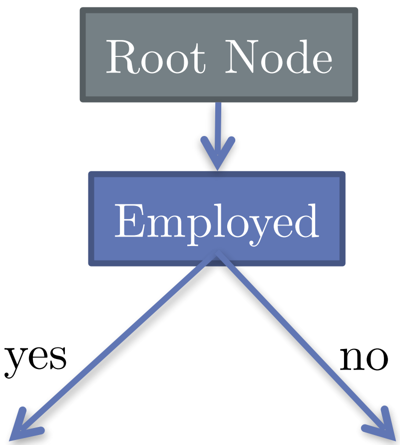
Building the model

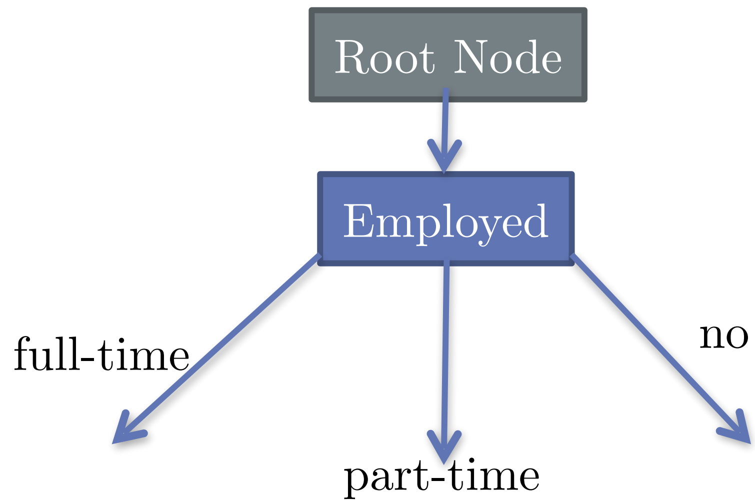
- A tree is built by recursively partitioning the training data into successively **purer** subsets.
  - (Having mostly No's **or** mostly Yes's for the target.)
- Partitioning is done according to some condition.



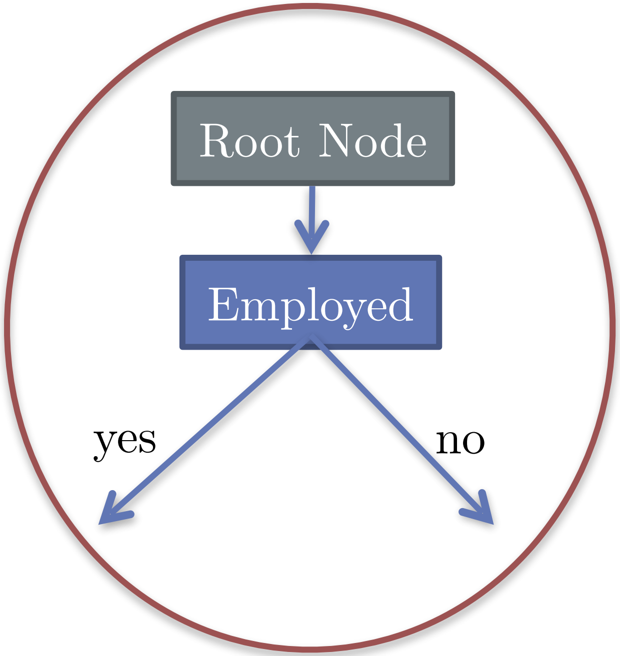
- How do we begin to assess these partitions?

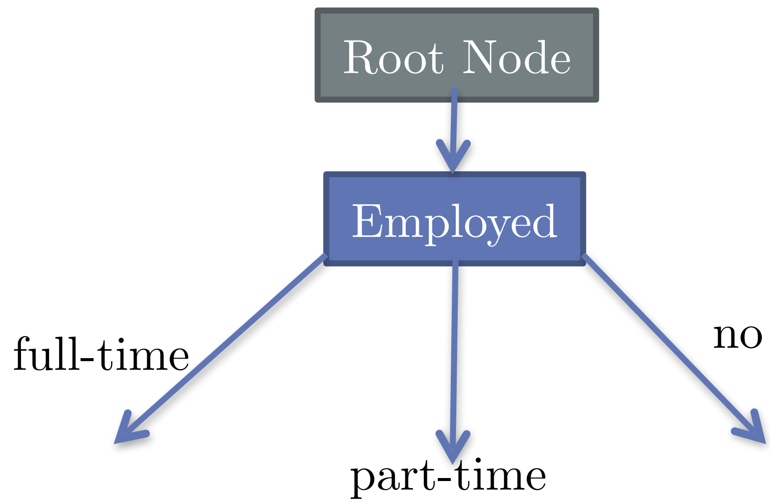
Binary Splits vs. Multi-way Splits





Binary Splits vs. Multi-way Splits





- We will primarily discuss binary splits
- Everything is easily extended to multiway splits
- Binary trees are far more common

Categorical Input Variables

- We consider **every possible way to separate** into two distinct groups.

- Example:

Marital Status= {Single, Married, Other}

Leaf 1	Leaf 2
Single	Married, Other
Married	Single, Other
Other	Single, Married

- There are $2^{L-1} - 1$ possible splits for a variable with L levels

Ordinal Input Variables

- **Only group together consecutive levels.**
- Example:

$$\text{Class} = \{\text{Lower, Middle, Upper}\}$$

Leaf 1	Leaf 2
Lower	Middle, Upper
Lower, Middle	Upper

- There are L-1 such splits for an ordinal variable with L levels.

Continuous Input Variables

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:
  Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:

Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 19	Age $\geq$ 19

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:
  Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 21	Age ≥ 21

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:

Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 23	Age $\geq$ 23

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:

Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 25	Age $\geq$ 25

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:

Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 29	Age $\geq$ 29

Binary Splits

- Continuous Attributes: We consider all possible splits between data points *or bins of* the variable.
- Example:

Age={18,18,19,21,21,23,25,29,35,37,40,40,41,43}

Leaf 1	Leaf 2
Age $<$ 35	Age $\geq$ 35

etc…

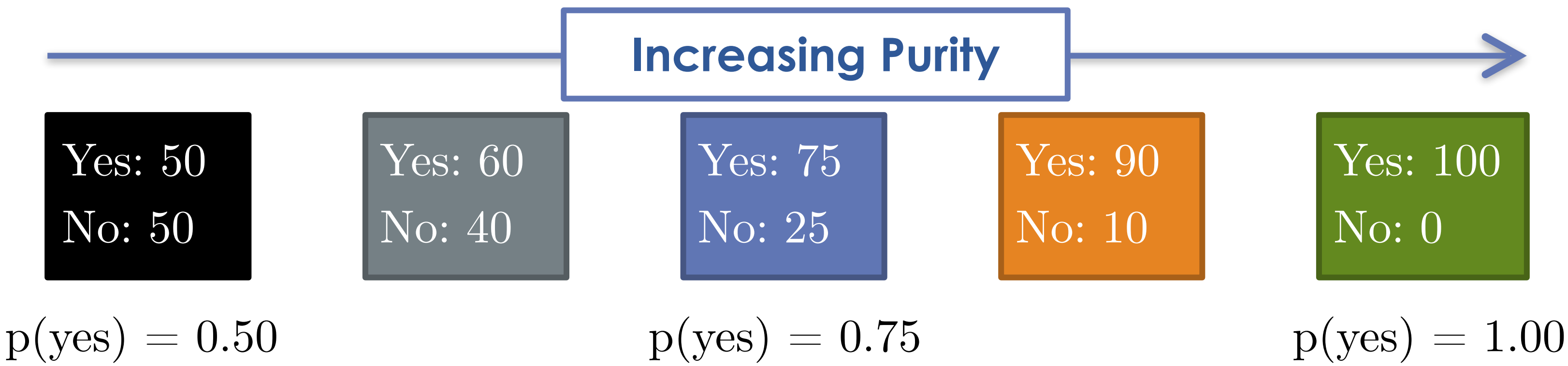
Missing Values

- One of the benefits of decision trees is their ability to handle missing values.

- Simply send missing values down one branch of the split (of course, it can get a lot fancier than that...)

Selecting the Best Split

- There are several measures used to select the best split.
- All are similar, but not identical
- All measure the **purity** of a node



- The more pure a leaf is, the less *training* error we make in that leaf.

Measures of Impurity

- Let $p(i|t) = p(class = i|node = t)$ be the fraction of records belonging to class i at a given node t. Let c be the number of classes in target variable.
- Entropy

$$Entropy(t) = -\sum_{i=1}^{c} p(i|t)\log_2 p(i|t)$$

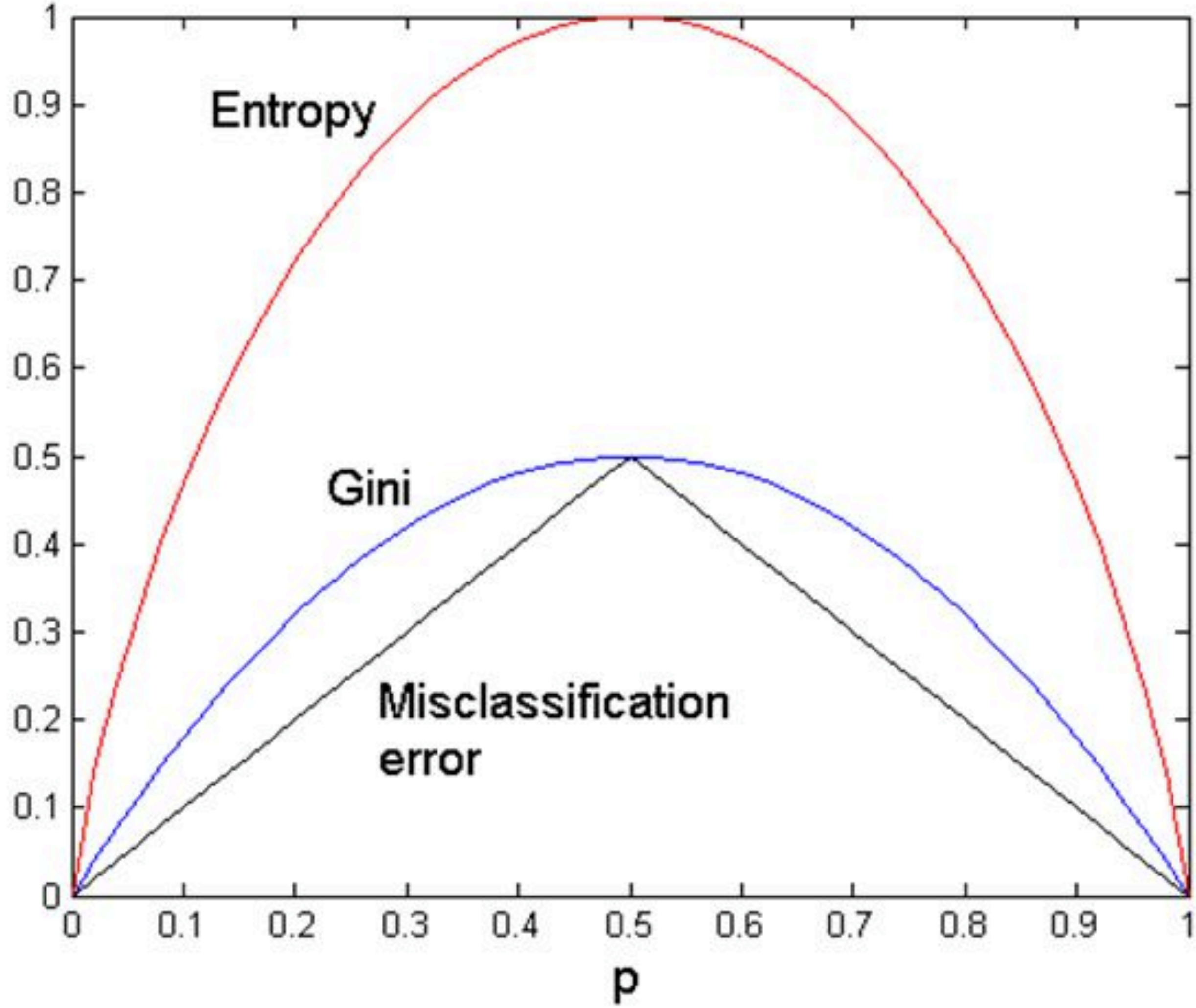
- Gini

$$Gini(t) = 1 - \sum_{i=1}^{c} [p(i|t)]^2$$

- Classification Error

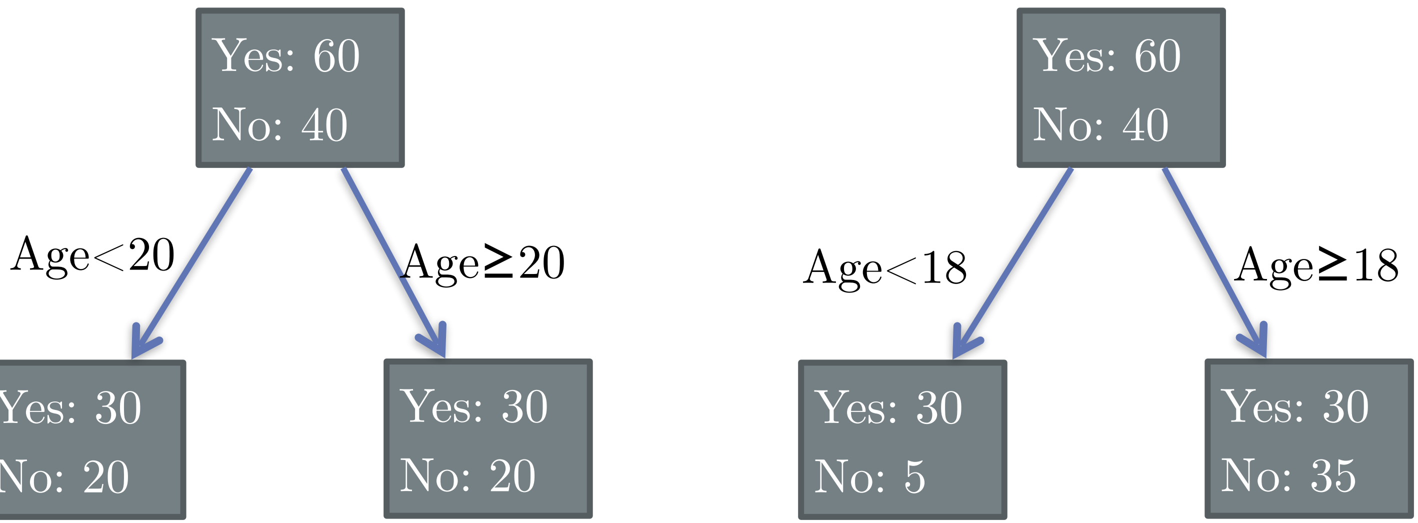
$$ClassificationError(t) = 1 - \max_i [p(i|t)]$$

Comparing Measures For a 2-class Problem



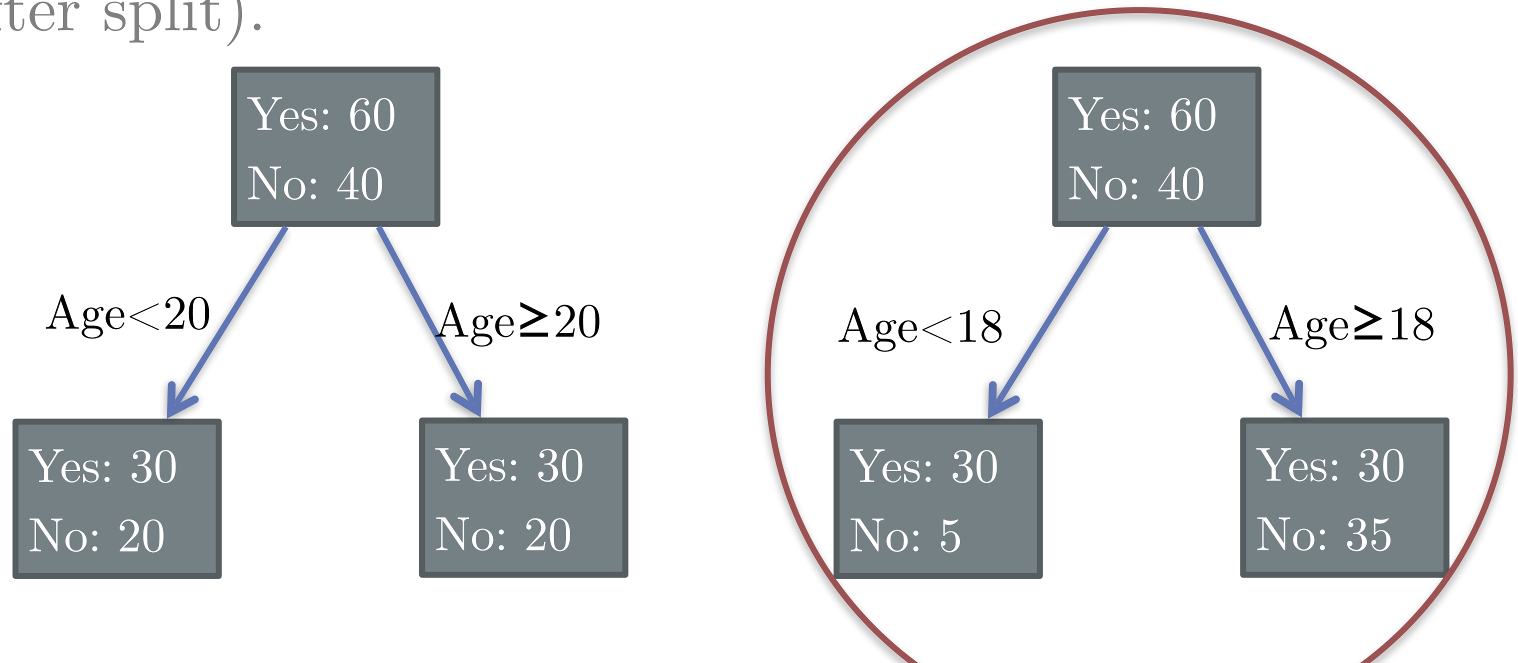
Selecting the best split

To assess a given test condition, we compare the impurity of the parent node (before split) with impurity of child nodes (after split).



Selecting the best split

To assess a given test condition, we compare the impurity of the parent node (before split) with impurity of child nodes (after split).



Split on the right has the best GAIN in purity.
(i.e. Reduction of impurity)

Gain (Worth)

$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$

Δ:= Gain

$I(t)$:= Impurity of parent node

$I(t_L)$ and $I(t_R)$:= Impurity of left/right child nodes

n:= Number of observations in parent

n_L and n_R := Number of observations in left/right child

Gain (Worth)

$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$

weighted avg. of child node impurity

Δ:= Gain

$I(t)$:= Impurity of parent node

$I(t_L)$ and $I(t_R)$:= Impurity of left/right child nodes

n:= Number of observations in parent

n_L and n_R := Number of observations in left/right child

Gain (Worth)

$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$

Larger Gain ➔More pure branches

Δ:= Gain

$I(t)$:= Impurity of parent node

$I(t_L)$ and $I(t_R)$:= Impurity of left/right child nodes

n:= Number of observations in parent

n_L and n_R := Number of observations in left/right child

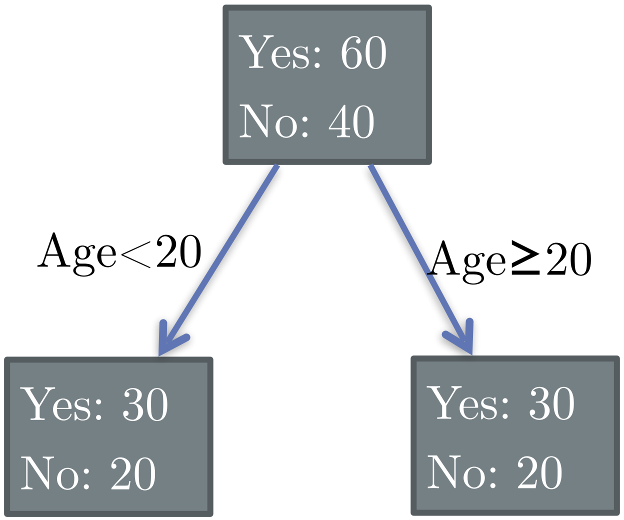
Gain (Worth)

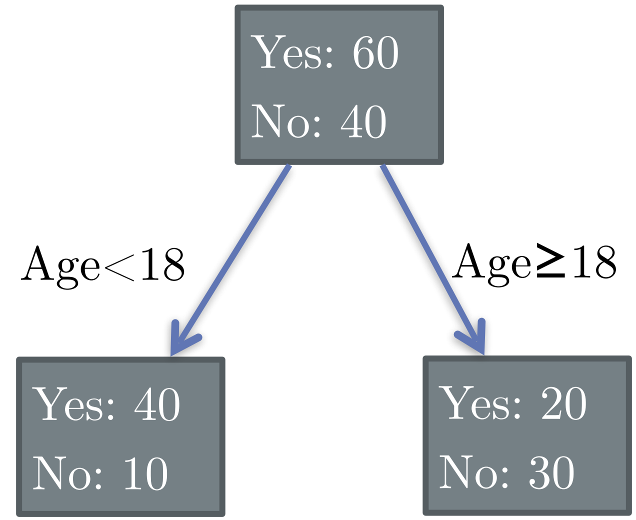
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$

When entropy is used, this difference in entropy is called *Information Gain.*

(For more information, see Tom Carter's slides at
http://astarte.csustan.edu/~tom/SFI-CSSS/2005/info-lec.pdf)

Example: Comparing 2 splits with Gain, Impurity Measure Gini

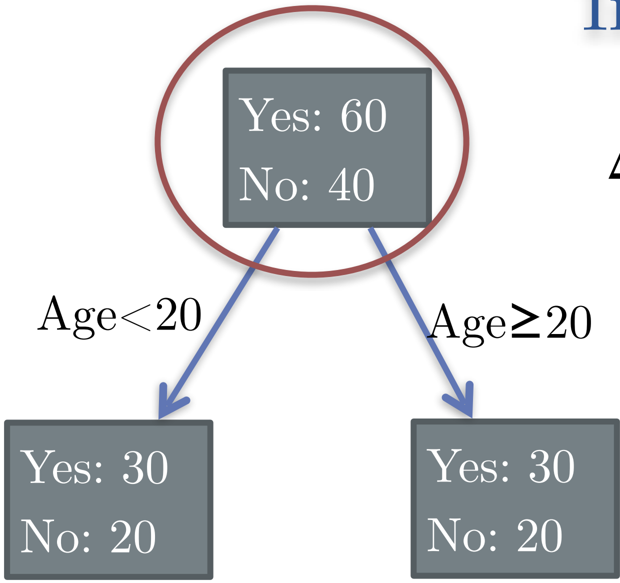




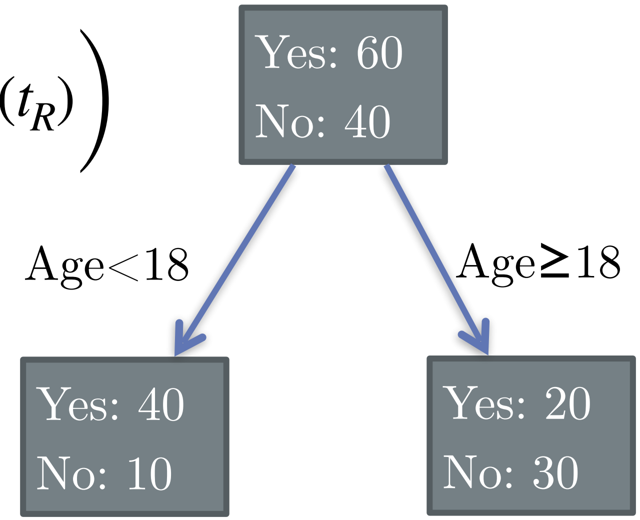
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$

$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



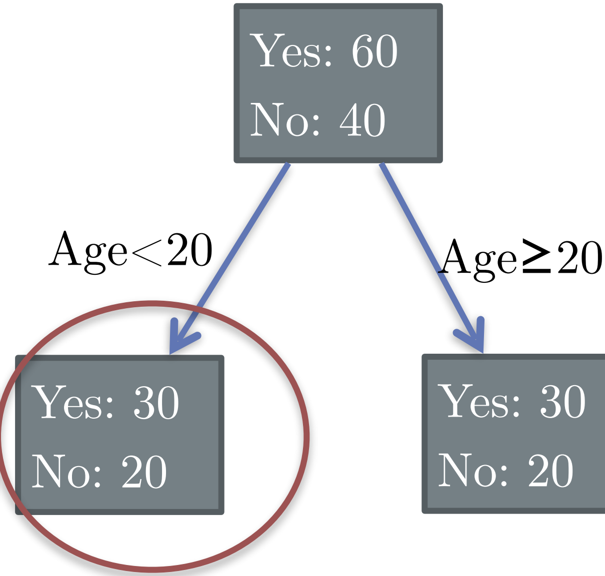
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$



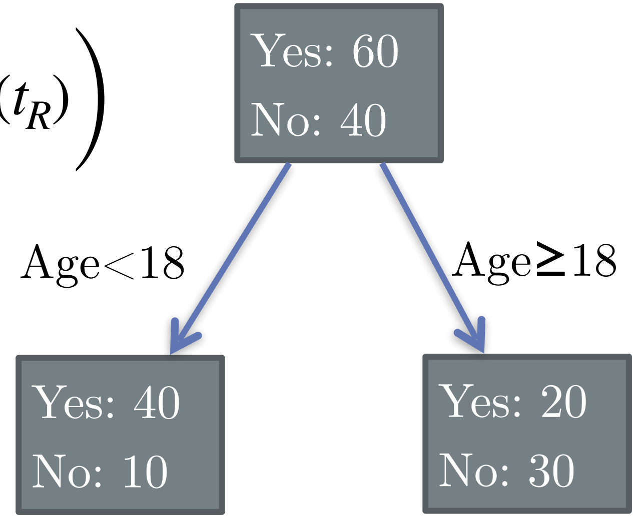
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t) = 1 - \left[\left(\frac{60}{100} \right)^2 + \left(\frac{40}{100} \right)^2 \right] = 0.48$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



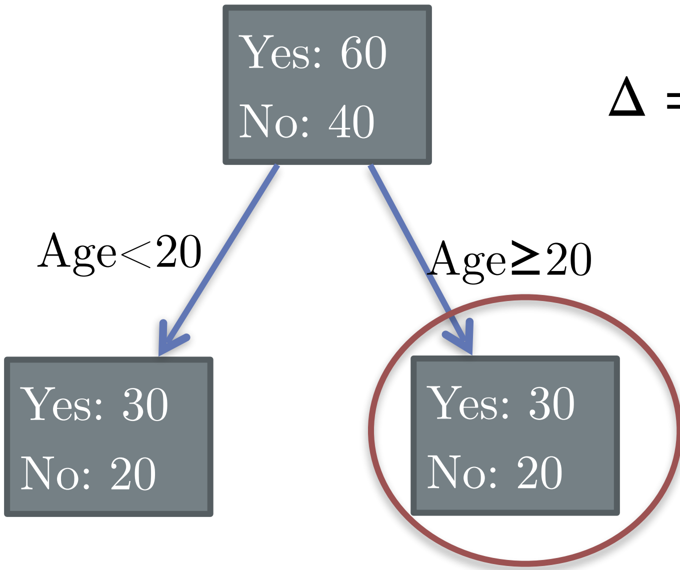
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$



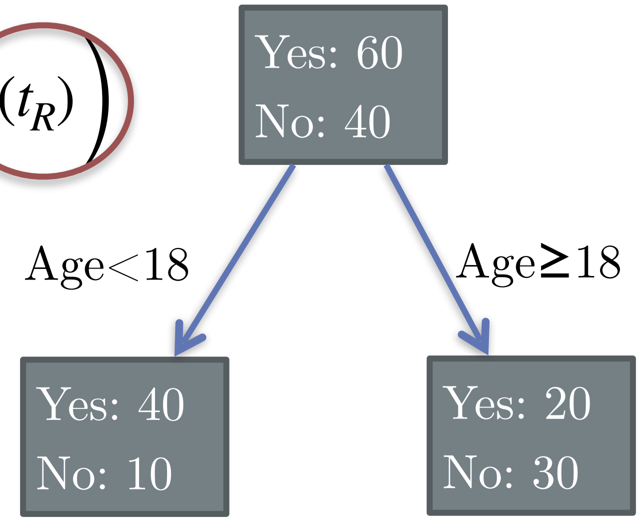
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t_L) = 1 - \left[\left(\frac{30}{50} \right)^2 + \left(\frac{20}{50} \right)^2 \right] = 0.48$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



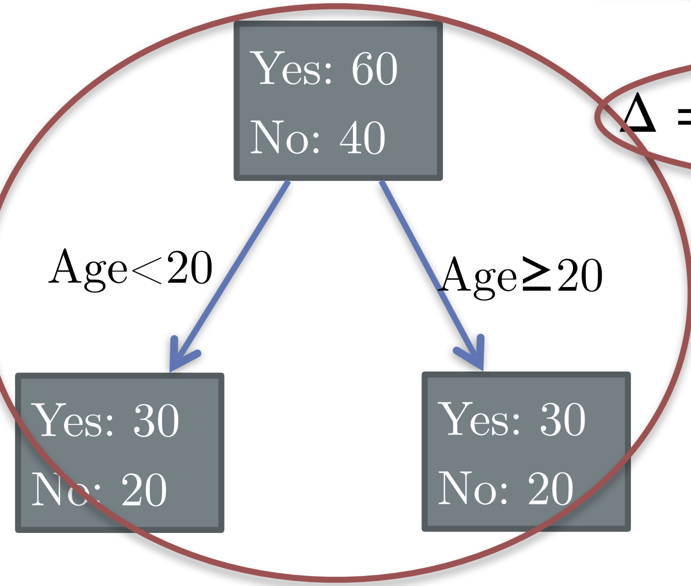
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$



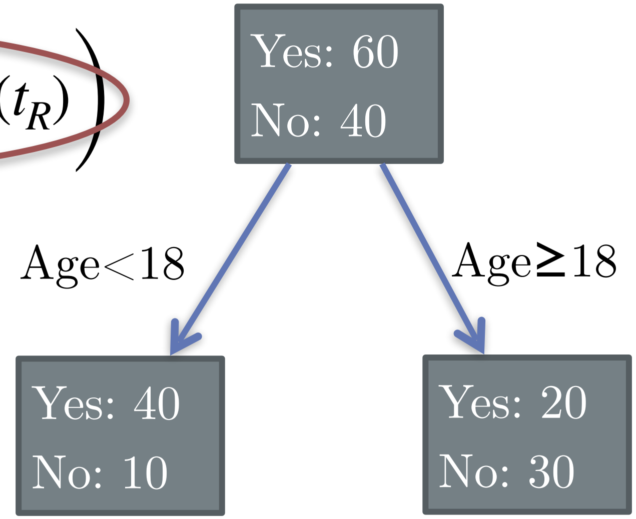
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t_R) = 1 - \left[\left(\frac{30}{50} \right)^2 + \left(\frac{20}{50} \right)^2 \right] = 0.48$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



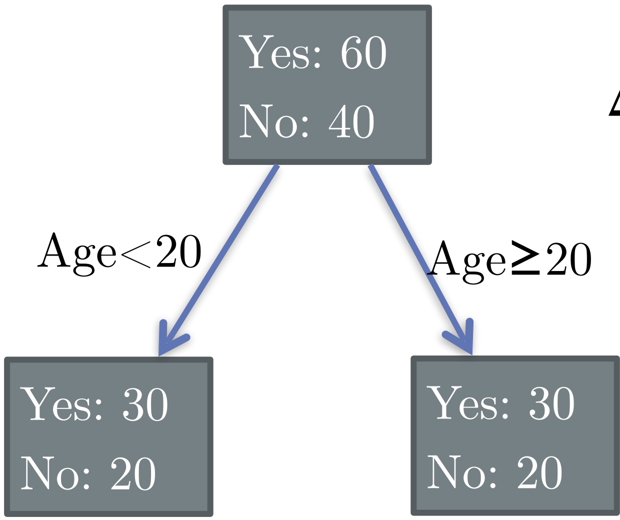
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$



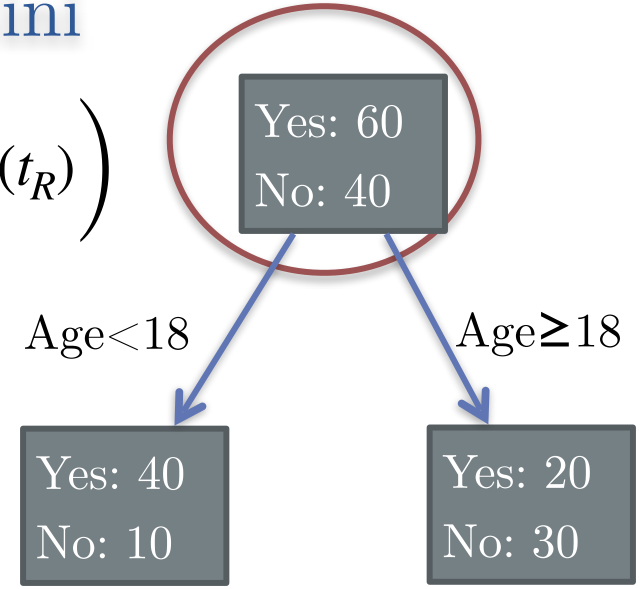
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$\Delta = 0.48 - \left(\frac{50}{100} 0.48 + \frac{50}{100} 0.48\right) = 0$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



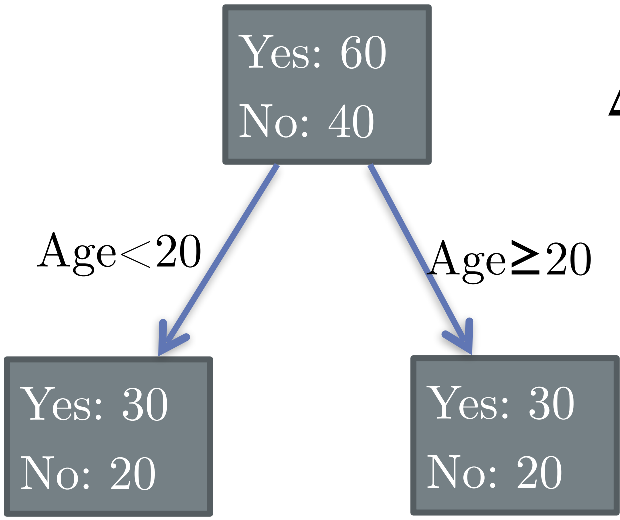
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$



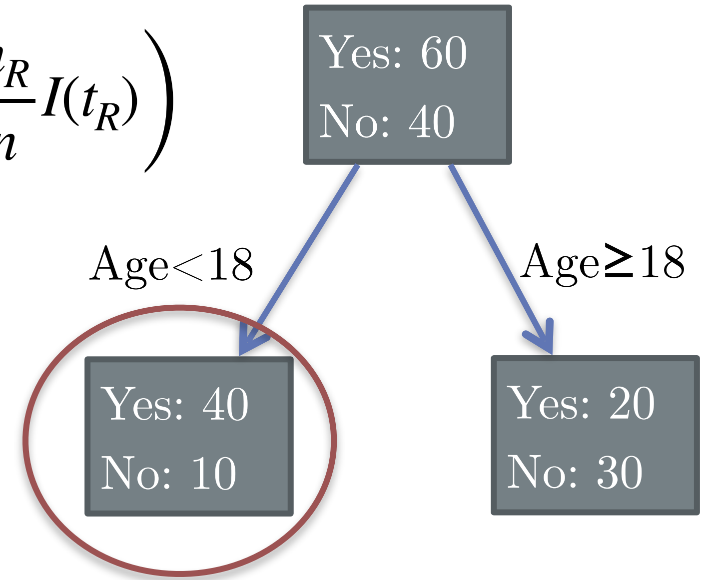
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t) = 1 - \left[\left(\frac{60}{100} \right)^2 + \left(\frac{40}{100} \right)^2 \right] = 0.48$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



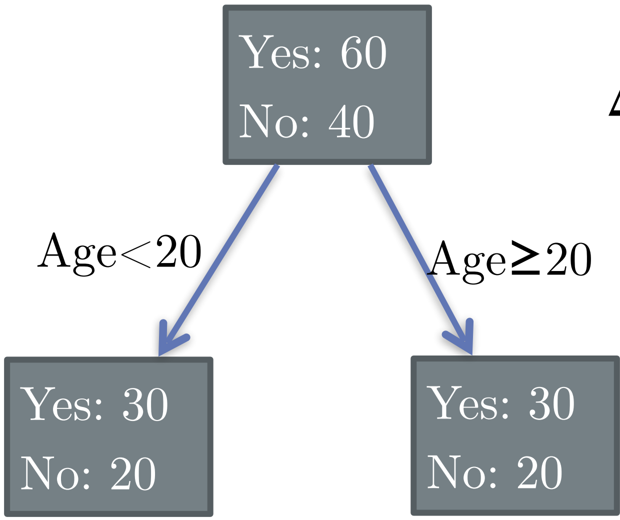
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$



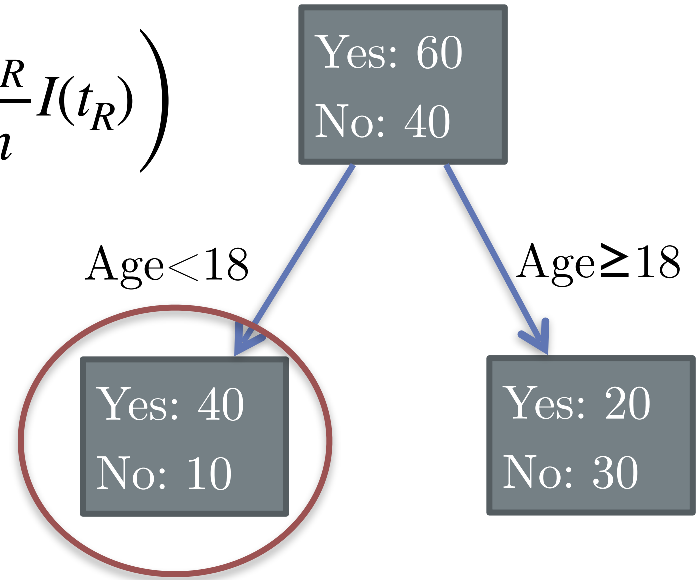
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t_L) = 1 - \left[\left(\frac{40}{50}\right)^2 + \left(\frac{10}{50}\right)^2\right] = 0.32$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



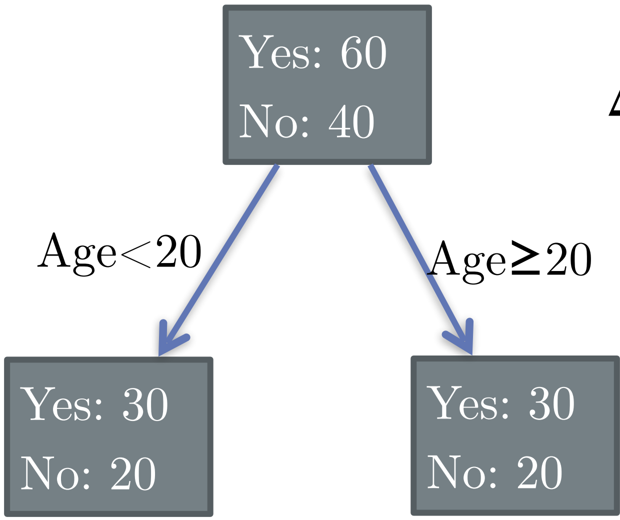
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$



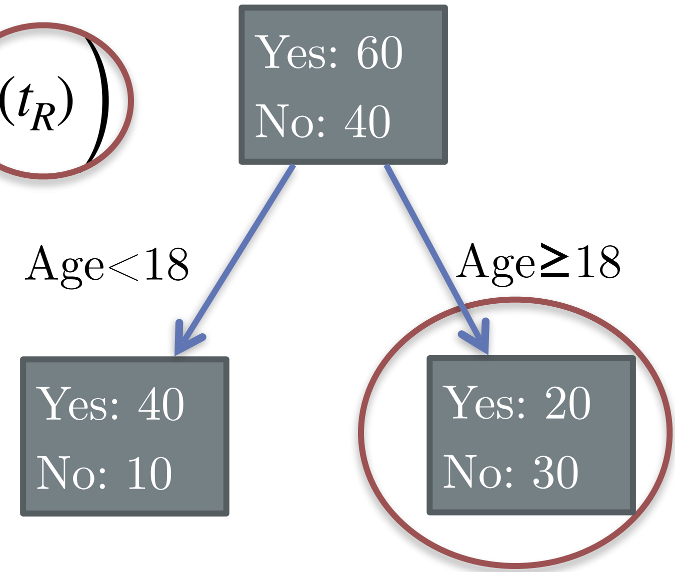
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t_L) = 1 - \left[\left(\frac{40}{50}\right)^2 + \left(\frac{10}{50}\right)^2\right] = 0.32$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



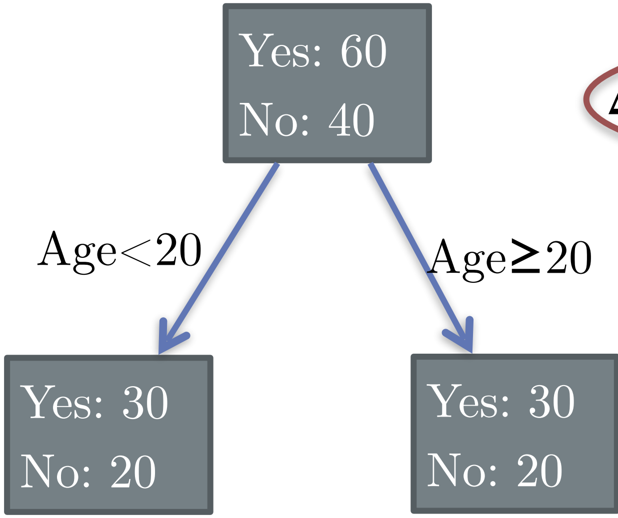
$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R) \right)$$



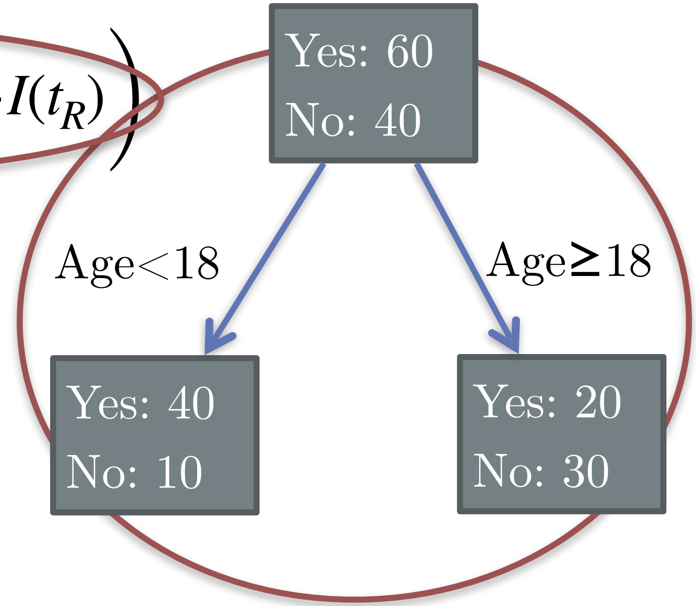
$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$I(t_R) = 1 - \left[\left(\frac{20}{50} \right)^2 + \left(\frac{30}{50} \right)^2 \right] = 0.48$$

Example: Comparing 2 splits with Gain, Impurity Measure Gini



$$\Delta = I(t) - \left(\frac{n_L}{n} I(t_L) + \frac{n_R}{n} I(t_R)\right)$$



$$I(t) = Gini(t) = 1 - \sum_{i=1}^{c} [p(i \mid t)]^2$$

$$\Delta = 0.48 - \left(\frac{50}{100} 0.32 + \frac{50}{100} 0.48\right) = 0.08$$

So the split on the right has a higher gain and is thus the better split

Creating the tree

- Compute the gain for all possible splits and select the best one.
- Repeat process recursively until some stopping condition is met
  - No splits meet some minimum Gain
  - All leaves have some minimum number of observations
  - A stopping condition is a way of *prepruning* the tree
- Prune Tree
  - Generally difficult to choose the right thresholds in prepruning
  - Can grow a larger tree and prune back branches in supervised fashion. (Essentially picking the threshold after the fact.)

Pruning a Decision Tree

- Simplifies the model
  - Occam's razor – law of parsimony
  - "Plurality is not to be posited without necessity" (Duns Scotus 1290)
- Prevents overfitting the training data
  - An accurate model on training: one bin for each leaf! #TerribleIdea
- **Simply remove leaves/nodes** in a bottom-up fashion, cutting splits with lowest gain first, while **optimizing performance on validation data**

Viya Demo 1

Telco Customer Churn

• • •

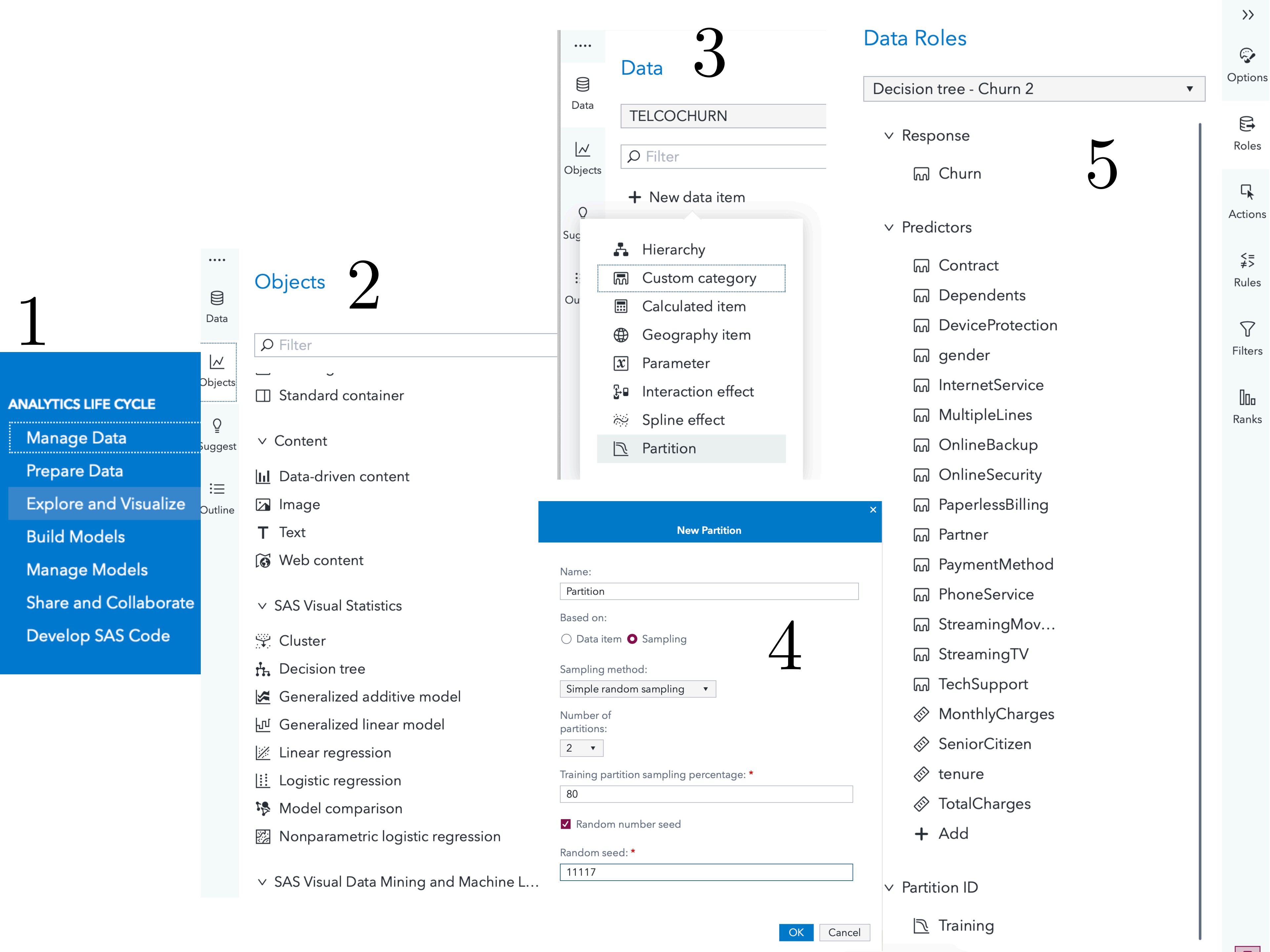
https://www.kaggle.com/blastchar/telco-customer-churn

Problem Introduction

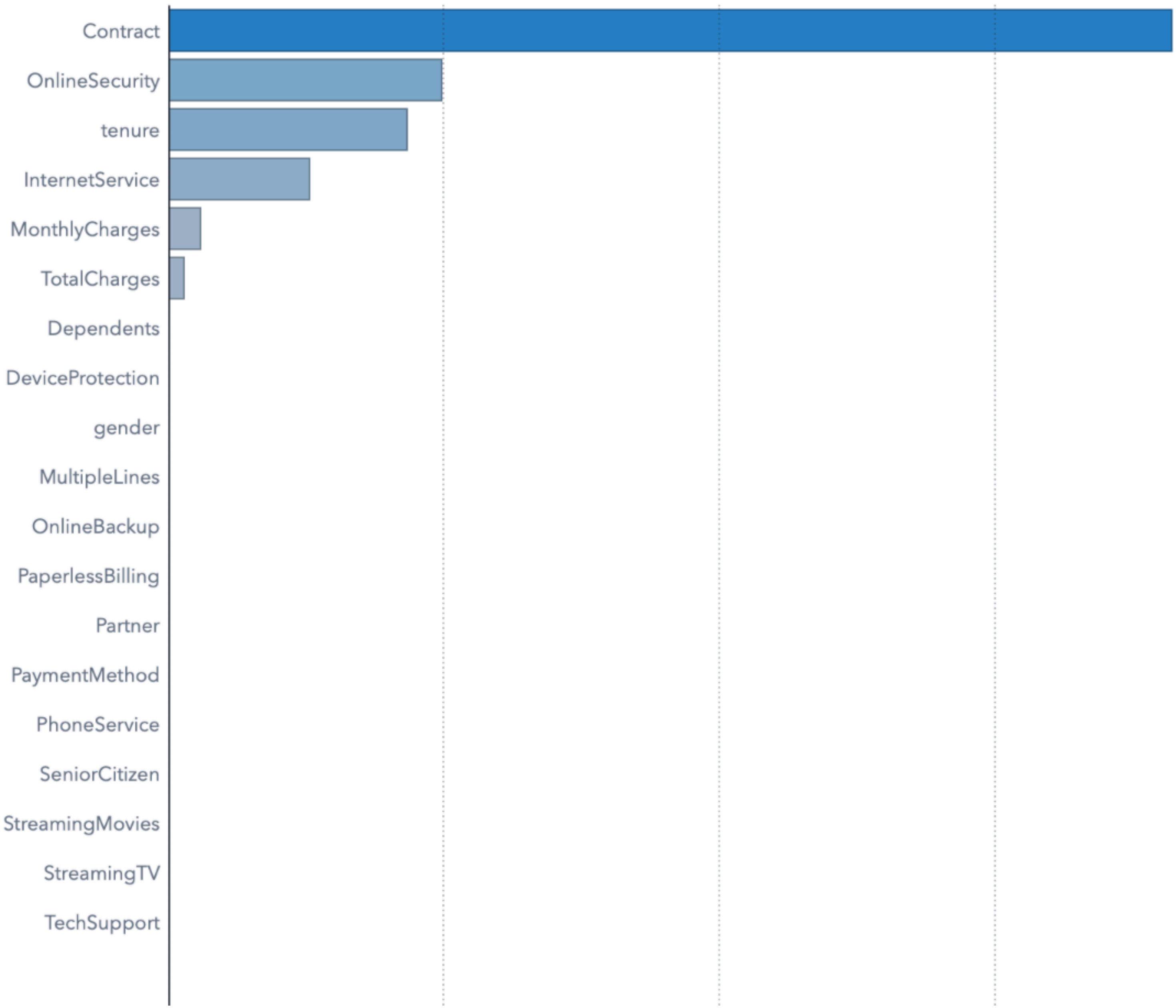
Goal: Predict behavior to retain customers. Analyze all relevant customer data and develop focused customer retention programs.

The data set includes information about:

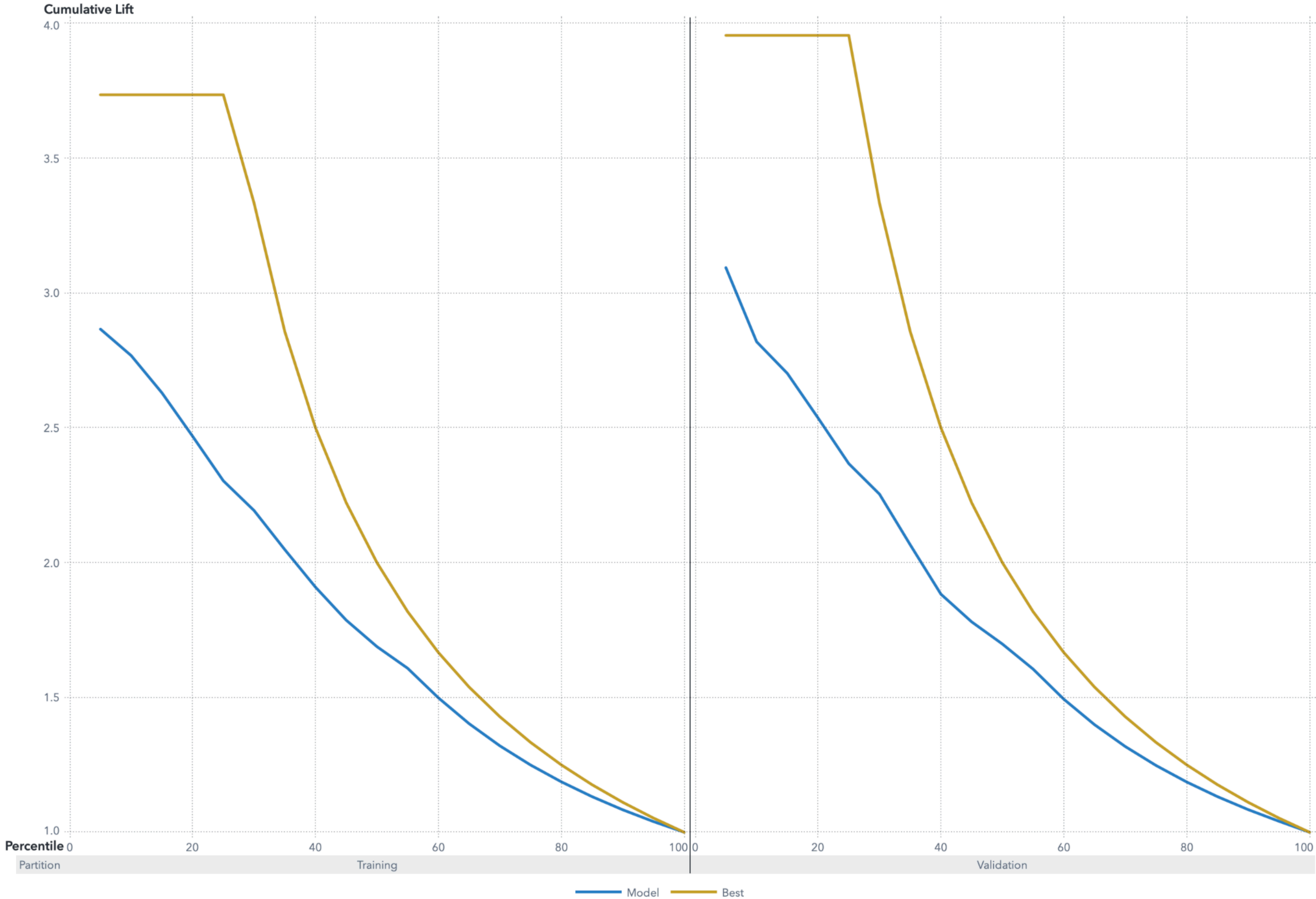
- Customers who left within the last month (and customers who did not) – the **target column** is called **Churn**
- **Services that each customer has signed up for** – *phone, multiple lines, internet, online security, online backup, device protection, tech support*, and *streaming TV and movies*
- Customer **account information** – *tenure* as a customer, *contract, payment method, paperless billing, monthly charges*, and *total charges*
- **Demographic info** about customers – *gender, age range*, and if they have *partners* and *dependents*


1
ANALYTICS LIFE CYCLE
Manage Data
Prepare Data
Explore and Visualize
Build Models
Manage Models
Share and Collaborate
Develop SAS Code
2
Objects
Data
Objects
Suggest
Outline
Filter
Standard container
Content
Data-driven content
Image
Text
Web content
SAS Visual Statistics
Cluster
Decision tree
Generalized additive model
Generalized linear model
Linear regression
Logistic regression
Model comparison
Nonparametric logistic regression
SAS Visual Data Mining and Machine L...
3
Data
TELCOCHURN
Filter
New data item
Hierarchy
Custom category
Calculated item
Geography item
Parameter
Interaction effect
Spline effect
Partition
4
New Partition
Name:
Partition
Based on:
Data item
Sampling
Sampling method:
Simple random sampling
Number of partitions:
2
Training partition sampling percentage: *
80
Random number seed
Random seed: *
11117
OK
Cancel
5
Data Roles
Decision tree - Churn 2
Response
Churn
Predictors
Contract
Dependents
DeviceProtection
gender
InternetService
MultipleLines
OnlineBackup
OnlineSecurity
PaperlessBilling
Partner
PaymentMethod
PhoneService
StreamingMov...
StreamingTV
TechSupport
MonthlyCharges
SeniorCitizen
tenure
TotalCharges
Add
Partition ID
Training
Options
Roles
Actions
Rules
Filters
Ranks

Variable Importance

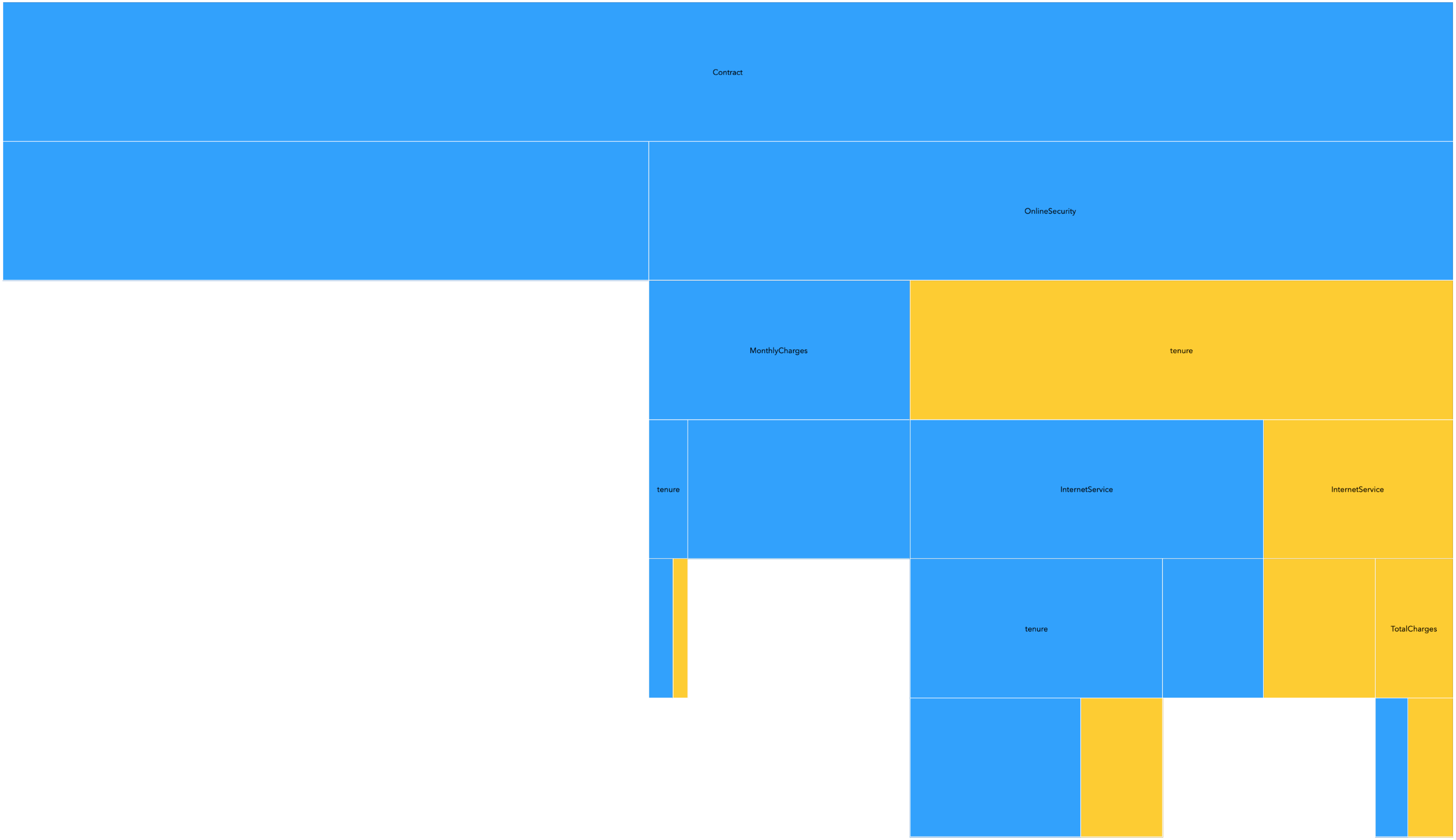


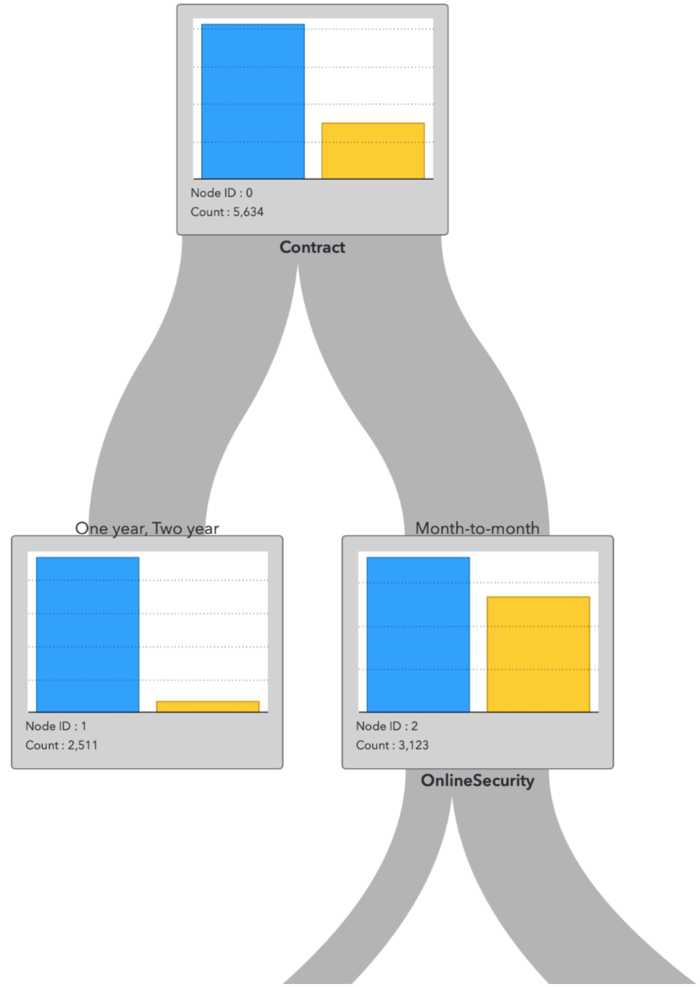
Lift



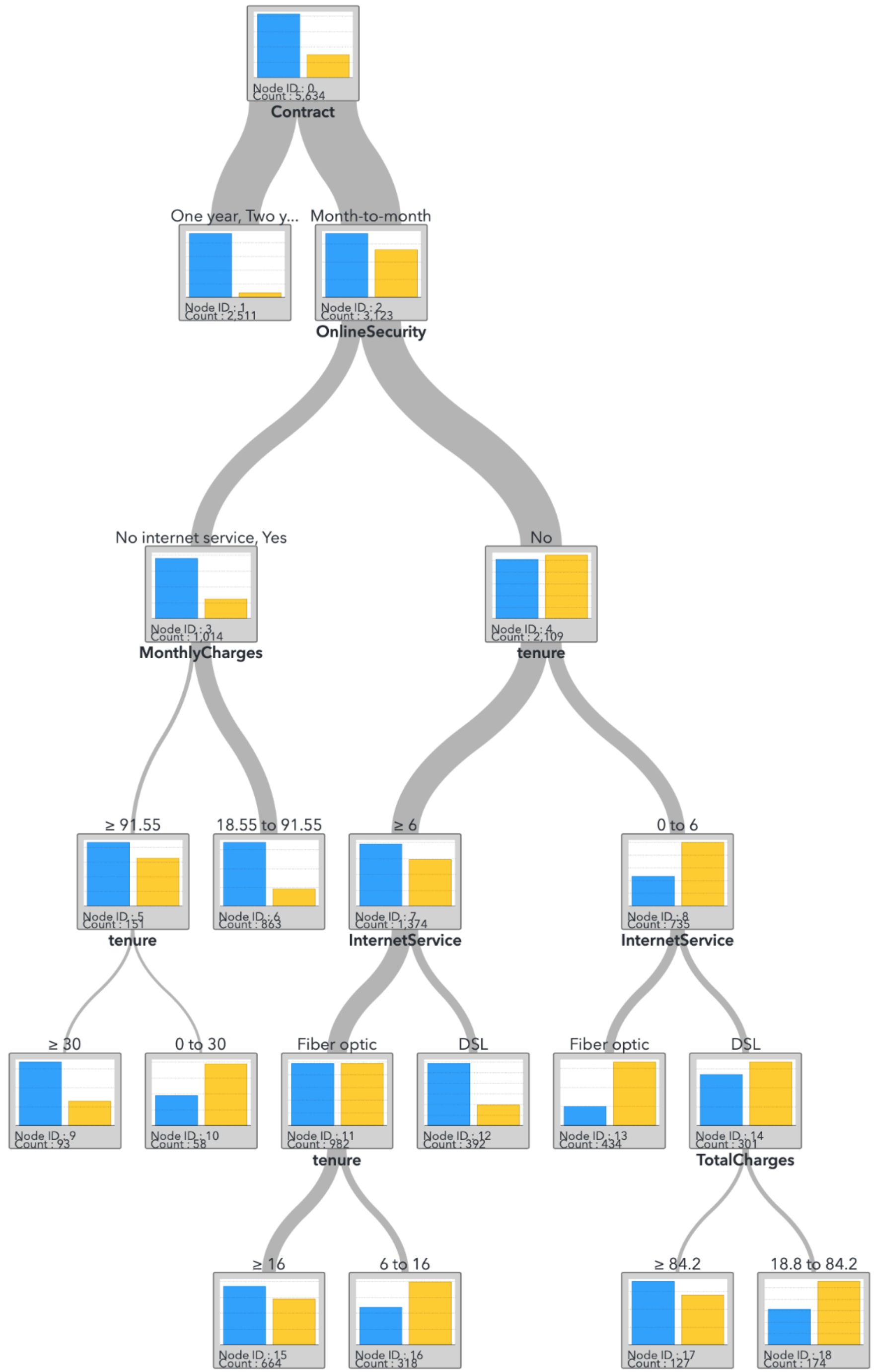
Cumulative Lift
4.0
3.5
3.0
2.5
2.0
1.5
1.0
Percentile
0
20
40
60
80
100
0
20
40
60
80
100
Partition
Training
Validation
Model
Best

‹ Decision Tree | Icicle | Variable Importance | Assessment ›




Node ID : 0
Count : 5,634
Contract
One year, Two year
Month-to-month
Node ID : 1
Count : 2,511
Node ID : 2
Count : 3,123
OnlineSecurity

≥ 91.55
Node ID : 5
Count : 151
tenure
18.55 to 91.55
Node ID : 6
Count : 863
≥ 6
Node ID : 7
Count : 1,374
InternetService
0 to 6
Node ID : 8
Count : 735
InternetService
≥ 30
Node ID : 9
Count : 93
0 to 30
Node ID : 10
Count : 58
Fiber optic
Node ID : 11
Count : 982
tenure
DSL
Node ID : 12
Count : 392
Fiber optic
Node ID : 13
Count : 434
DSL
Node ID : 14
Count : 301
TotalCharges
≥ 16
Node ID : 15
Count : 664
6 to 16
Node ID : 16
Count : 318
≥ 84.2
Node ID : 17
Count : 127
18.8 to 84.2
Node ID : 18
Count : 174
Churn
No
Yes



Node ID : 0
Count : 5,634
Contract
One year, Two y...
Month-to-month
Node ID : 1
Count : 2,511
Node ID : 2
Count : 3,123
OnlineSecurity
No internet service, Yes
No
Node ID : 3
Count : 1,014
MonthlyCharges
Node ID : 4
Count : 2,109
tenure
≥ 91.55
18.55 to 91.55
≥ 6
0 to 6
Node ID : 5
Count : 151
tenure
Node ID : 6
Count : 863
Node ID : 7
Count : 1,374
InternetService
Node ID : 8
Count : 735
InternetService
≥ 30
0 to 30
Fiber optic
DSL
Fiber optic
DSL
Node ID : 9
Count : 93
Node ID : 10
Count : 58
Node ID : 11
Count : 982
tenure
Node ID : 12
Count : 392
Node ID : 13
Count : 434
Node ID : 14
Count : 301
TotalCharges
≥ 16
6 to 16
≥ 84.2
18.8 to 84.2
Node ID : 15
Count : 664
Node ID : 16
Count : 318
Node ID : 17
Count : 127
Node ID : 18
Count : 174

Part II

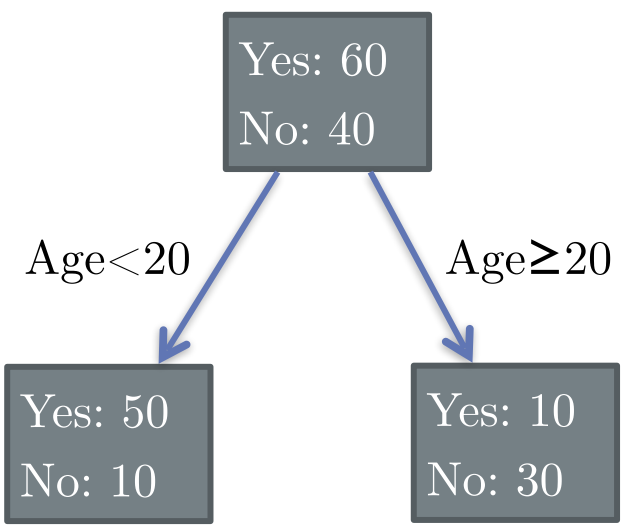
• • •

CHAID and Regression Trees

CHAID

CHi-squared Automatic Interaction Detection

- 1980 PhD thesis by Gordon Kass
- Rather than using gain to determine splits, use chi-square tests!
- Analyze decision tree splits like we do contingency tables:

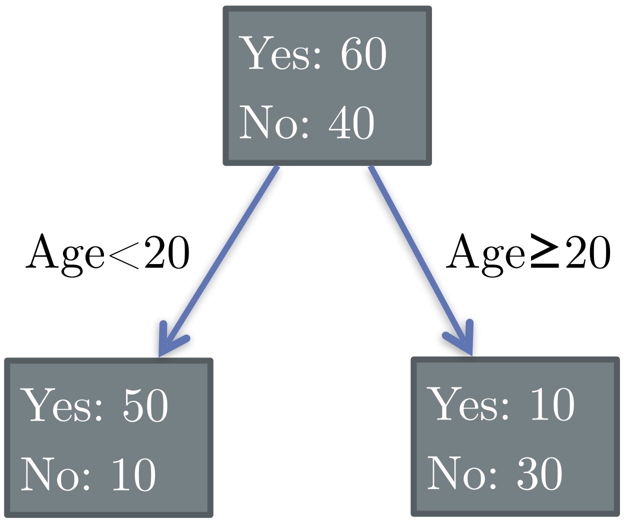


	Yes	No	Total
Age<20	50	10	60
Age≥20	10	30	40
Total	60	40	100

$$\chi^2 = \sum_{cells} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

CHAID

CHi-squared Automatic Interaction Detection



	Yes	No	Total
Age<20	50	10	60
Age≥20	10	30	40
Total	60	40	100

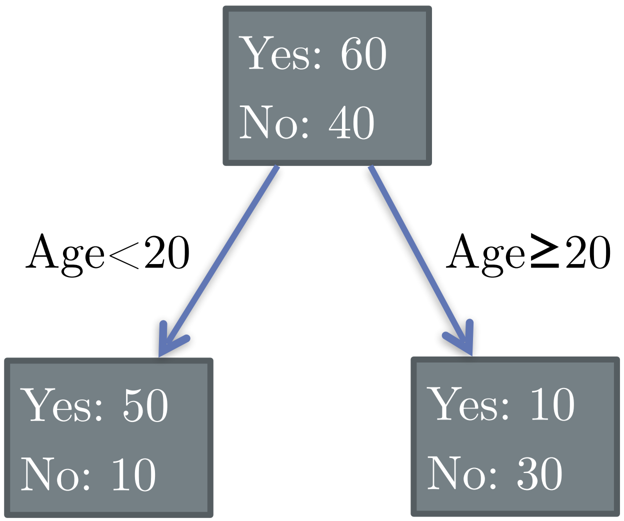
$$\chi^2 = \sum_{cells} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

Larger χ^2 statistic ➔ Smaller p-value ➔ Stronger relationship

only b/c sample size is constant in comparison at a given parent node!

CHAID

CHi-squared Automatic Interaction Detection



	Yes	No	Total
Age<20	50	10	60
Age≥20	10	30	40
Total	60	40	100

$$\chi^2 = \sum_{cells} \frac{(\text{observed} - \text{expected})^2}{\text{expected}}$$

Larger χ^2 statistic → Smaller p-value → Stronger relationship

Uses **logworth** to choose a split: $\text{logworth}(p) = -\log_{10}(p)$

Logworth

$$\text{logworth}(p) = -\log_{10}(p)$$

Tells us approx # of decimal places of our p-value.

Examples:

- $\text{logworth}(0.001) = -\log_{10}(0.001) = -(-3) = 3.$
- $\text{logworth}(0.0001) = 4$
- logworth(0.0004) is between 3 and 4
    - $0.\underline{000}1 < 0.0004 < 0.\underline{00}1$
    - $\log_{10}(0.0001) < \log_{10}(0.0004) < \log_{10}(0.001)$
    - $-\log_{10}(0.0001) > -\log_{10}(0.0004) > -\log_{10}(0.001)$
    - $4 > -\log_{10}(0.0004) > 3$

LARGER LOGWORTH => BETTER SPLIT.

Kass Adjustments (i.e. Bonferroni Adjustments)

- Hypothesis testing to compare many variables at many potential splits. (Could be thousands of comparisons!)
- Beware the family-wise error rate!!
- **Adjust the test significance to (α/m)** where α is your desired significance level and m is number of tests.
- **Equivalent to multiplying p-values by m** and keeping α unchanged.

Kass Adjustments (i.e. Bonferroni Adjustments)

Suppose we compare *Age* (interval) with *Insurance Status* (binary).

No Adjustment

- best p-value for *Age* is **0.01** and occurs when splitting at Age<20, Age≥20
- p-value for *Insurance Status* is **0.05**

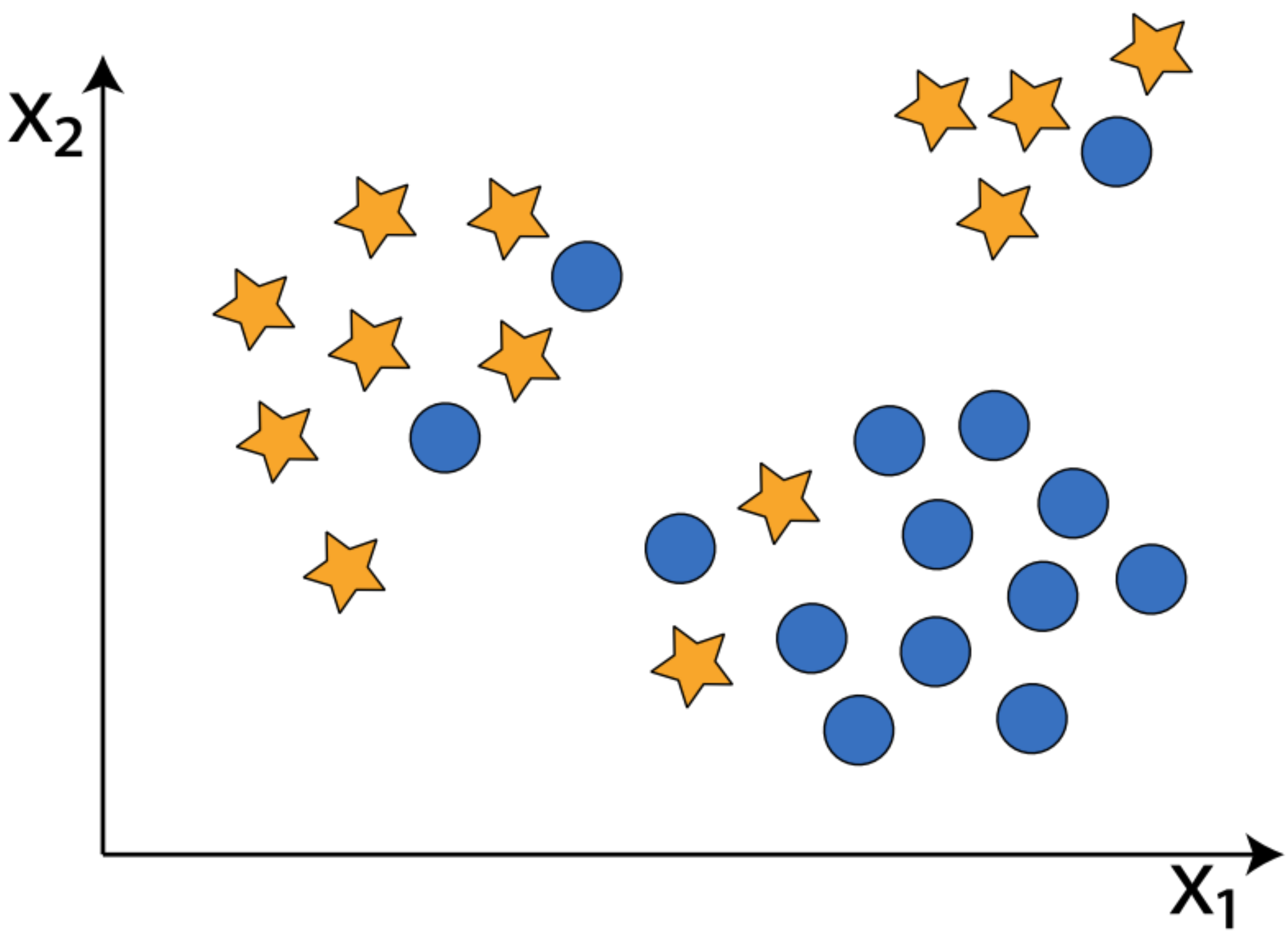
Pick *Age*<20, *Age*≥20 as the splitting criterion.

Bonferroni Adjustment

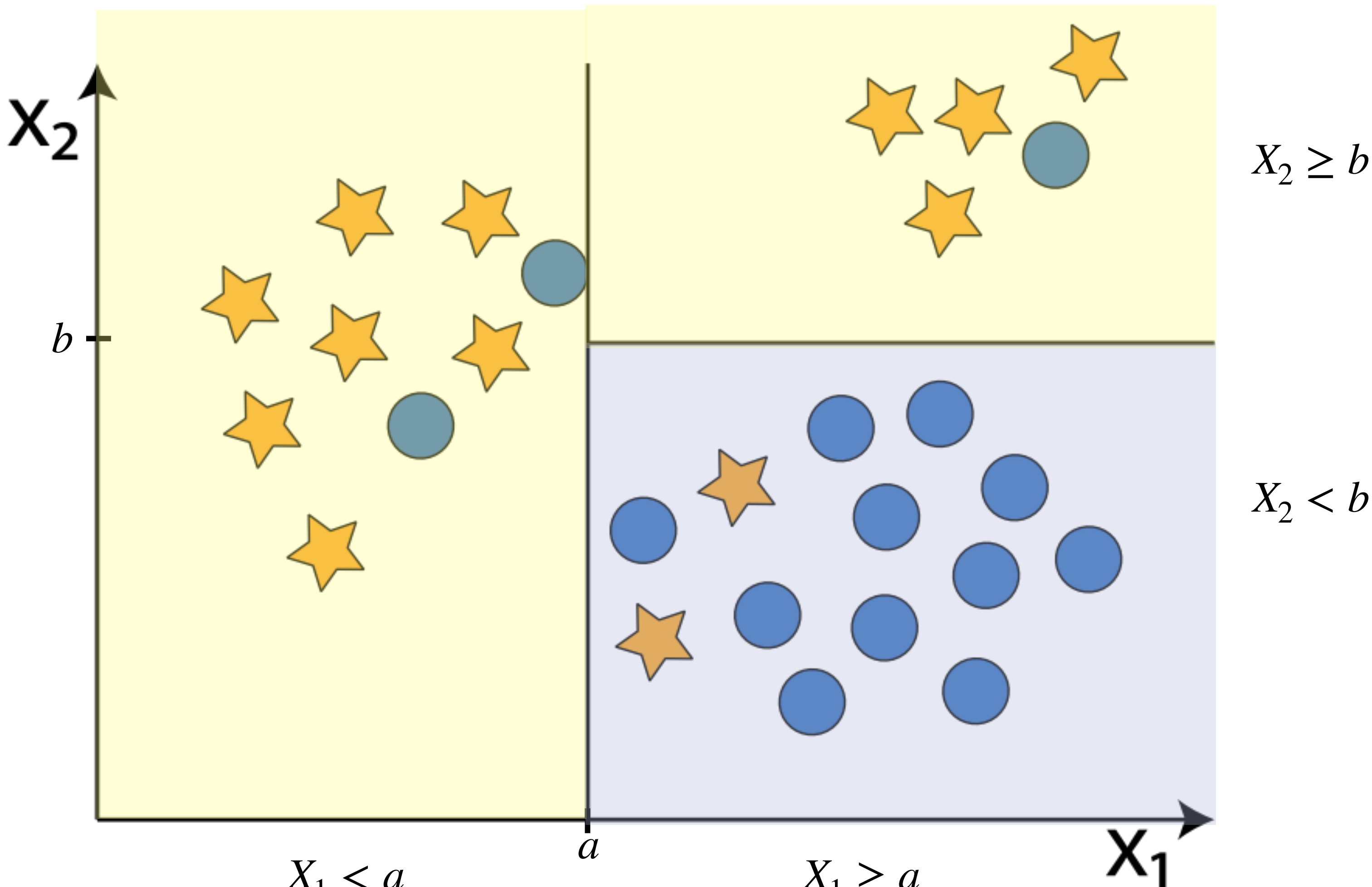
- Age had 51 unique values (50 possible splits)
- Insurance Status had 1
- Not fair to compare these p-values! In 50 tests, using **one** with a p-value of 0.01 is not convincing!
- Adjust p-values by multiplying by number of tests:
  - Age: (0.01)*50 = **0.5**
  - Insurance Status: (0.05)*1 = **0.05**

Pick *Insurance Status* as splitting criterion.

Decision Tree Boundaries

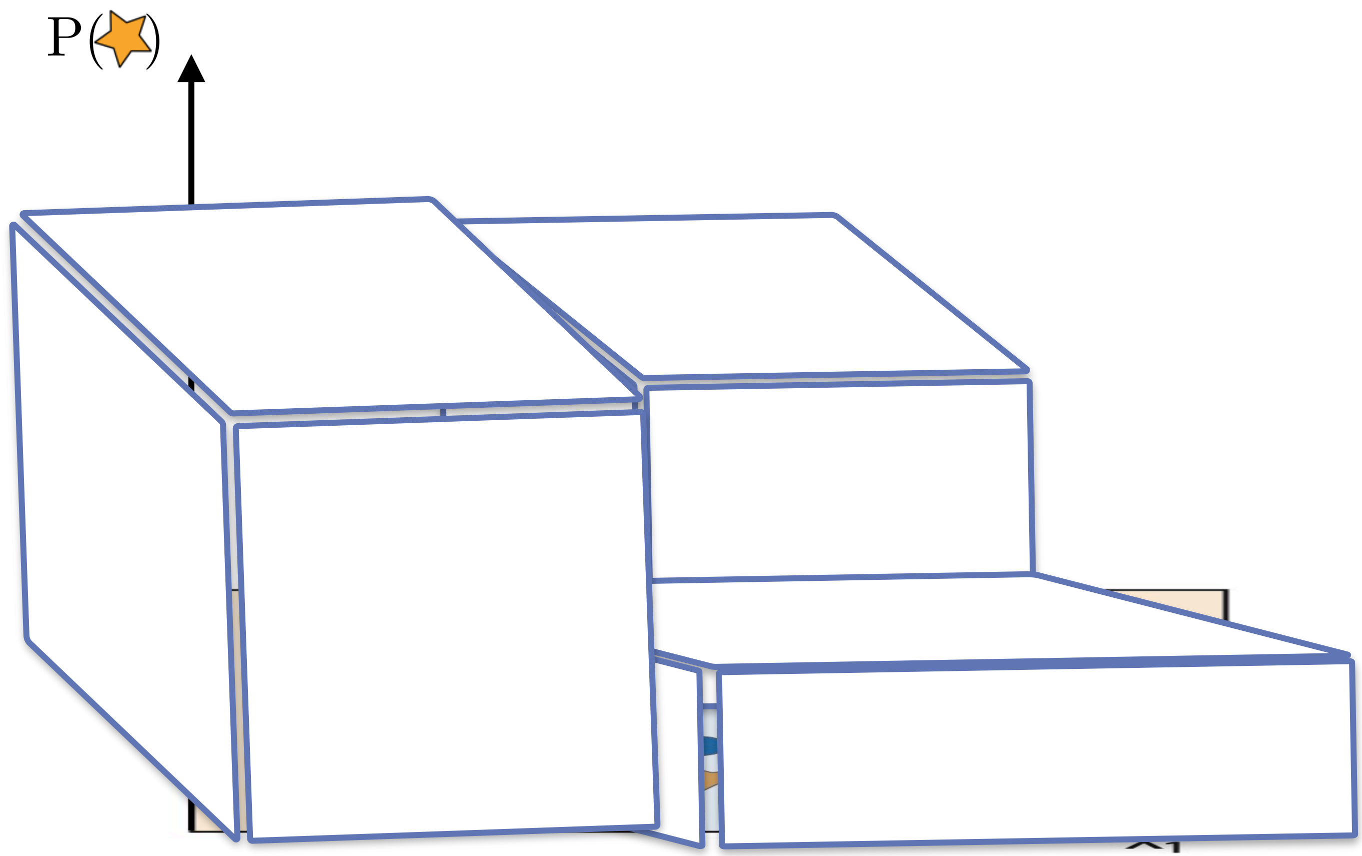


Decision Tree Boundaries



Decision Tree Response Surface

(Building with legos - no diagonals!)

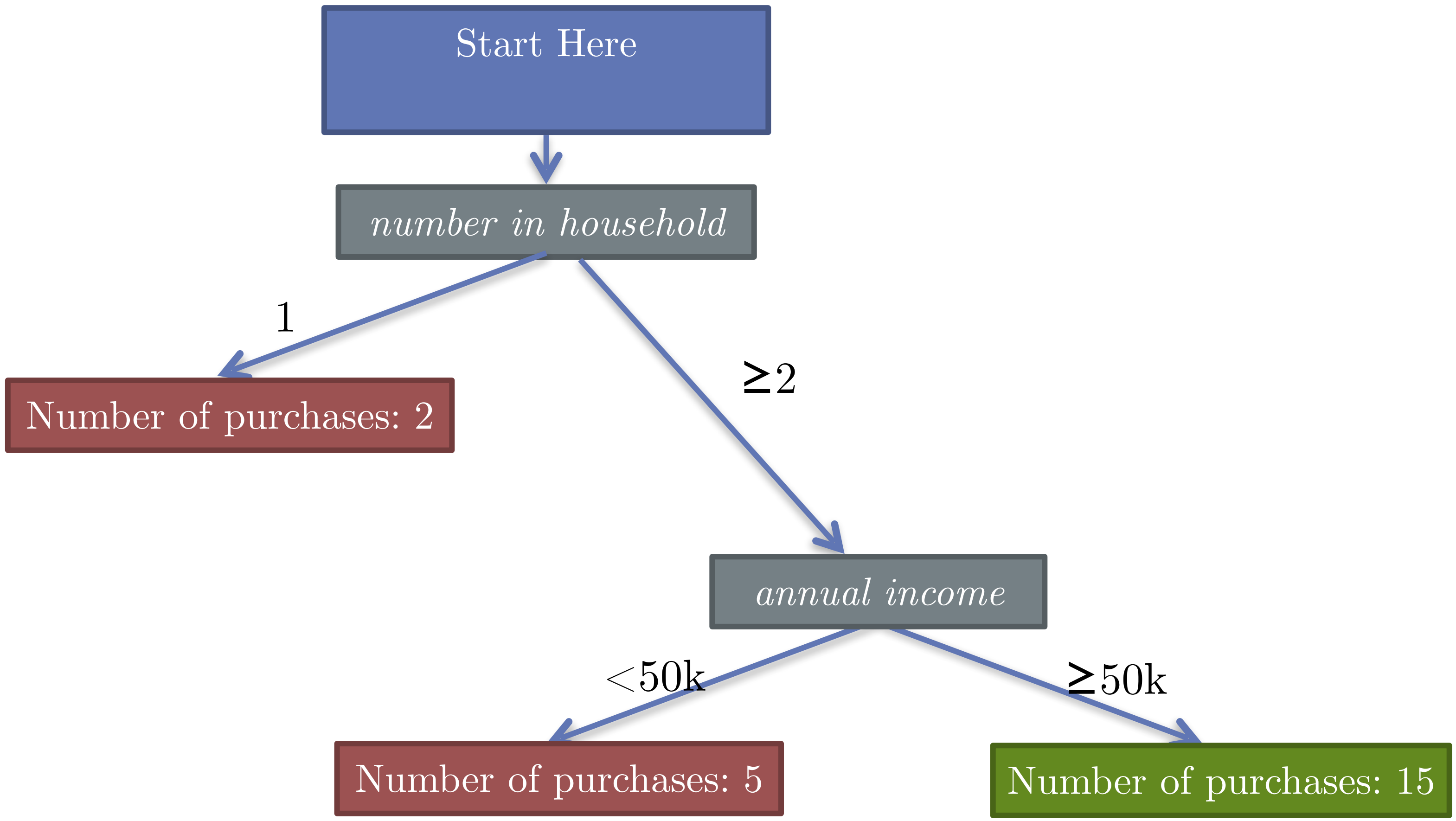


Regression Trees

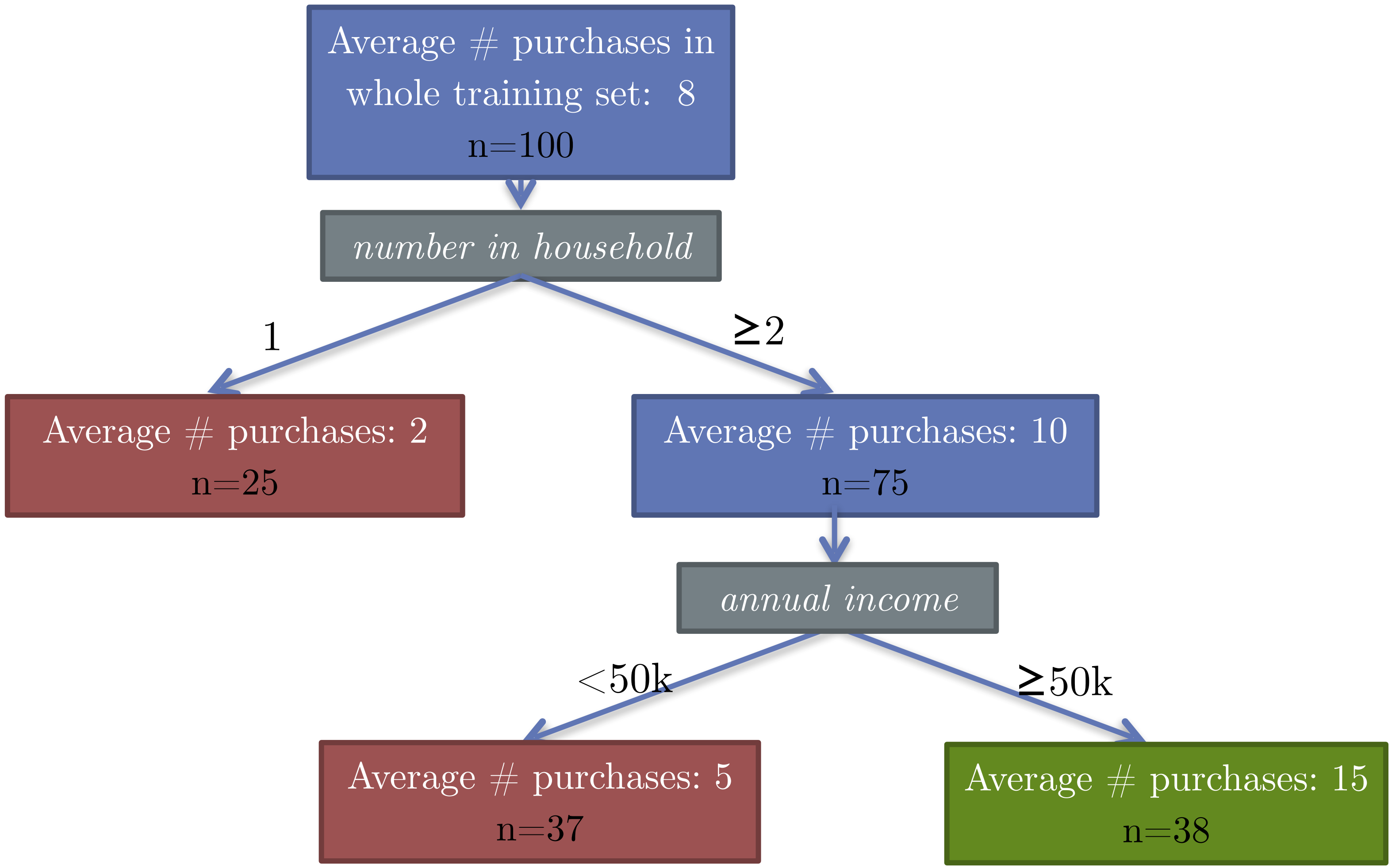
• • •

Same thing, but with **continuous target variables**

Regression Tree Model



Regression Tree Model Creation



Determining Splits

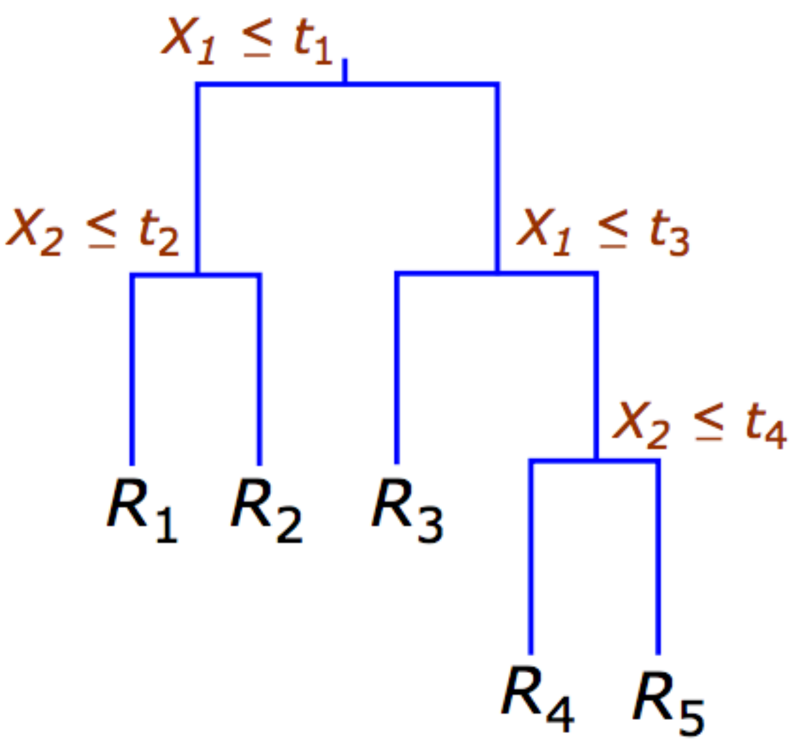
- Entropy/Gini no longer make sense for continuous target
- Instead:
  - Reduce *Average Squared Error* (i.e. variance since prediction is mean of observations in leaf)

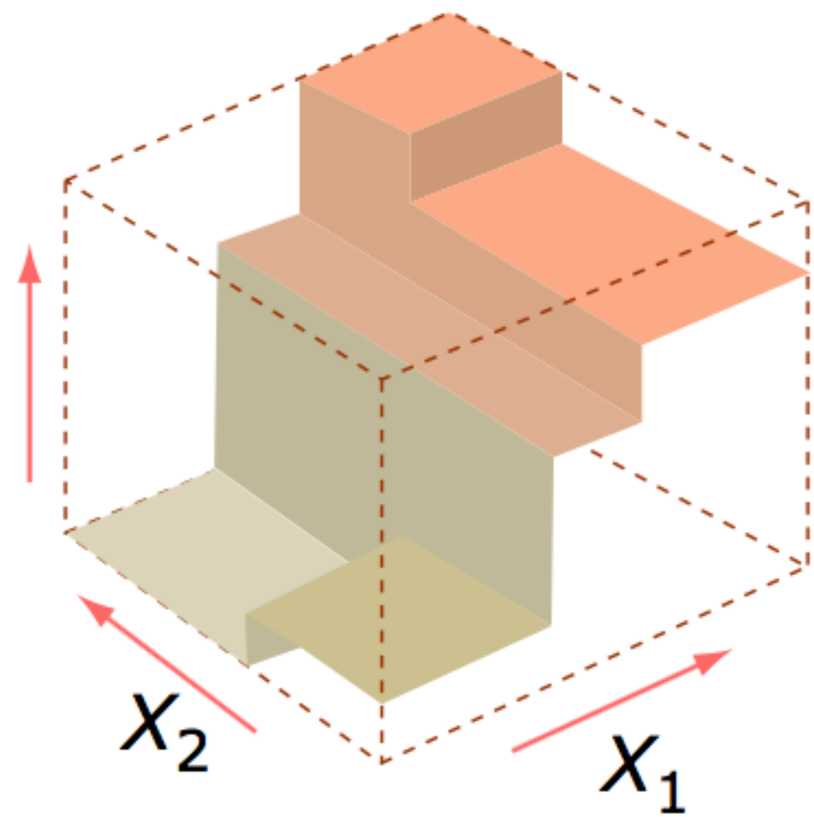
$$\sum_{i=1}^{N_t} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{N_t} (y_i - \bar{y}_i)^2 = Var(\mathbf{y}) \text{ within node}$$

  - Or Maximize *logworth* using p-value from an F-test
    - Testing whether means (predicted value) of leaves is different
    - (Same as a t-test for difference of means in binary case)
    - Think ANOVA overall F-test: are any of these means different?

Regression Tree Response Surface

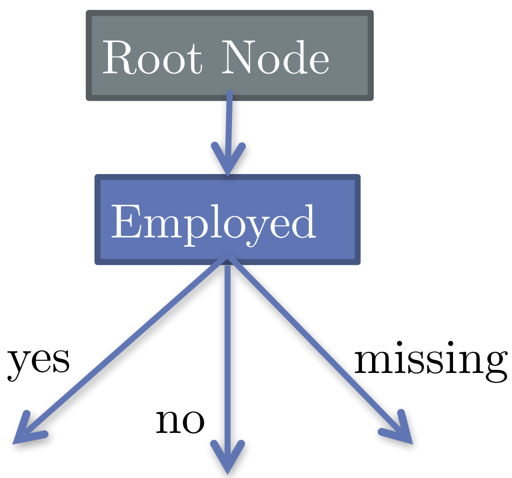
(Building with legos - no diagonals!)

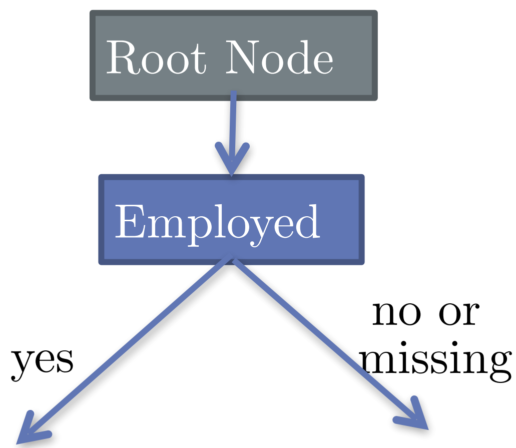




Advantages of tree models

1. **Explainability**
2. Predicted probability/response has **meaning** in training set
3. **Can handle missing values**





Alternatively via **surrogate splits**: designate an alternative variable split if the given variable is missing. Surrogate splits are chosen in a way that they split the population in the most similar fashion to the current split (often use a highly correlated variable).

Advantages of tree models

1. **Explainability**
2. Predicted probability/response has **meaning** in training set
3. **Can handle missing values**
4. Can be used for **variable selection**
5. Great for **ensembles**
   (basis for Random Forests and Gradient Boosting)
6. **No assumptions** to verify
7. Generally **immune to scale of input vars**/standardization
   (less effort in data pre-processing)
8. Generally **immune to the effect of outliers** or high leverage observations

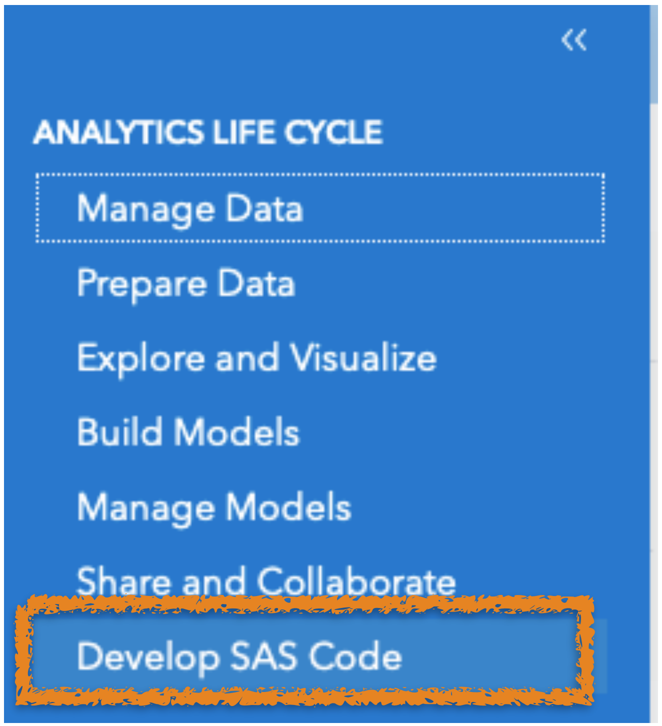
Disadvantages of tree models

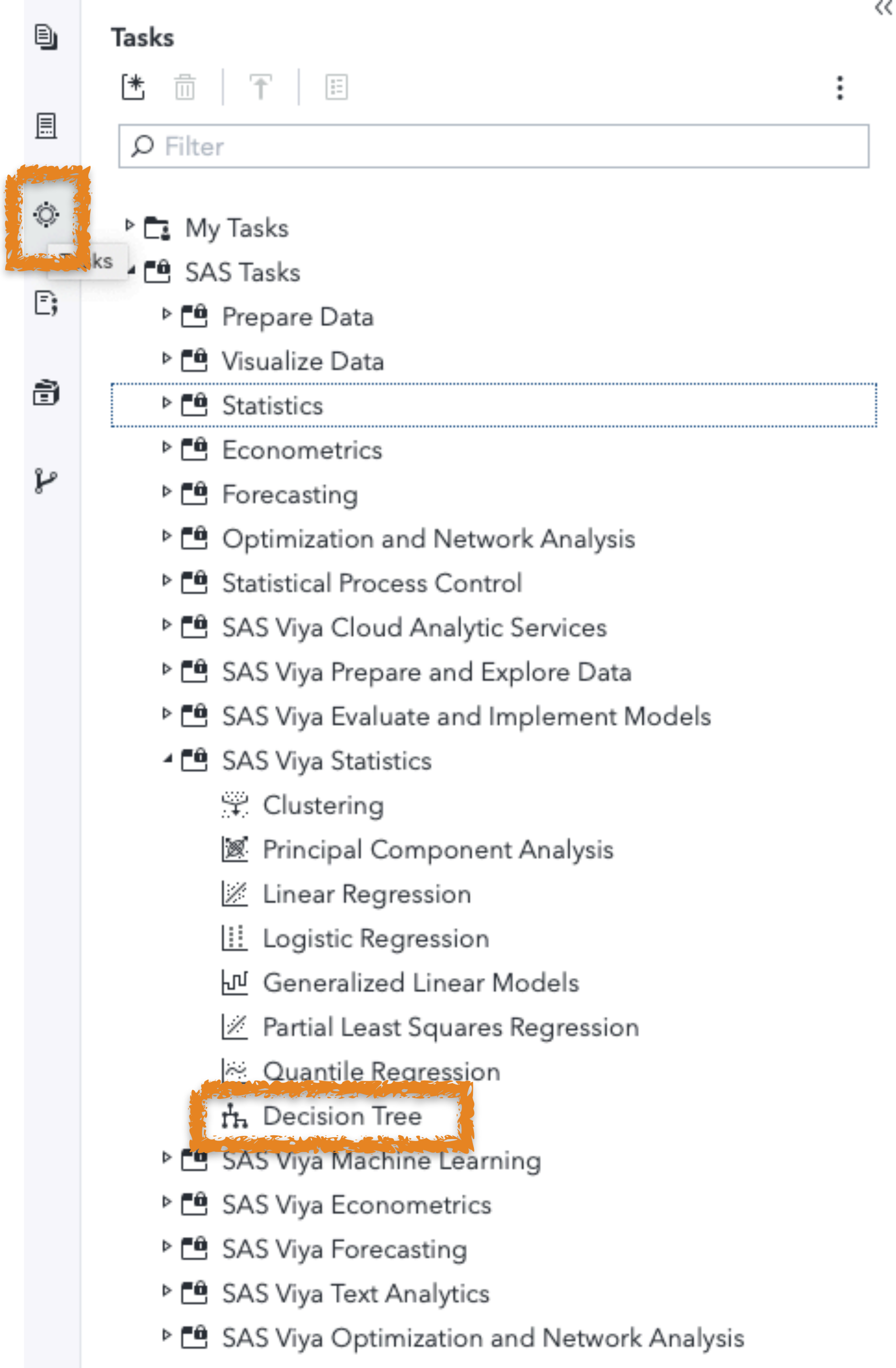
1. **Simplistic** Regression/Decision Surface (non-smooth)
2. All **variables forced to interact**
   a. Only the top split acts independently
   b. Inefficient
3. **Greedy** Algorithms
   a. Struggle in the presence of many variables
   b. Cannot return the globally optimal tree
4. Can be **unstable** (sensitive to small changes in input) - both when training the model *and* when making predictions. (*think*: sides of 'lego buildings' on the response surface)

Viya Demo 2

• • •

TelcoChurn using Tasks in SAS Studio


ANALYTICS LIFE CYCLE
Manage Data
Prepare Data
Explore and Visualize
Build Models
Manage Models
Share and Collaborate
Develop SAS Code

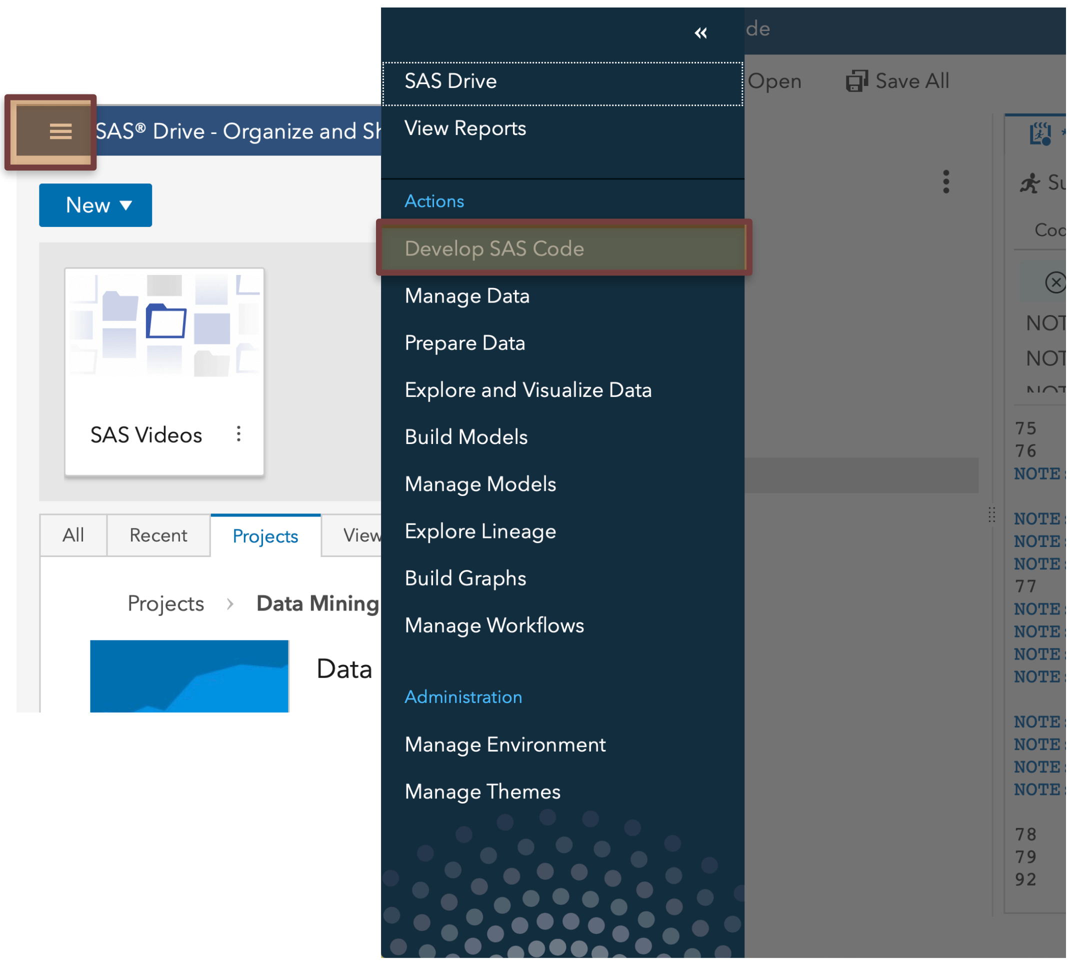

Tasks
Filter
My Tasks
SAS Tasks
Prepare Data
Visualize Data
Statistics
Econometrics
Forecasting
Optimization and Network Analysis
Statistical Process Control
SAS Viya Cloud Analytic Services
SAS Viya Prepare and Explore Data
SAS Viya Evaluate and Implement Models
SAS Viya Statistics
Clustering
Principal Component Analysis
Linear Regression
Logistic Regression
Generalized Linear Models
Partial Least Squares Regression
Quantile Regression
Decision Tree
SAS Viya Machine Learning
SAS Viya Econometrics
SAS Viya Forecasting
SAS Viya Text Analytics
SAS Viya Optimization and Network Analysis

Viya Demo 3

• • •

Breast Cancer Malignancy

Viya Demo

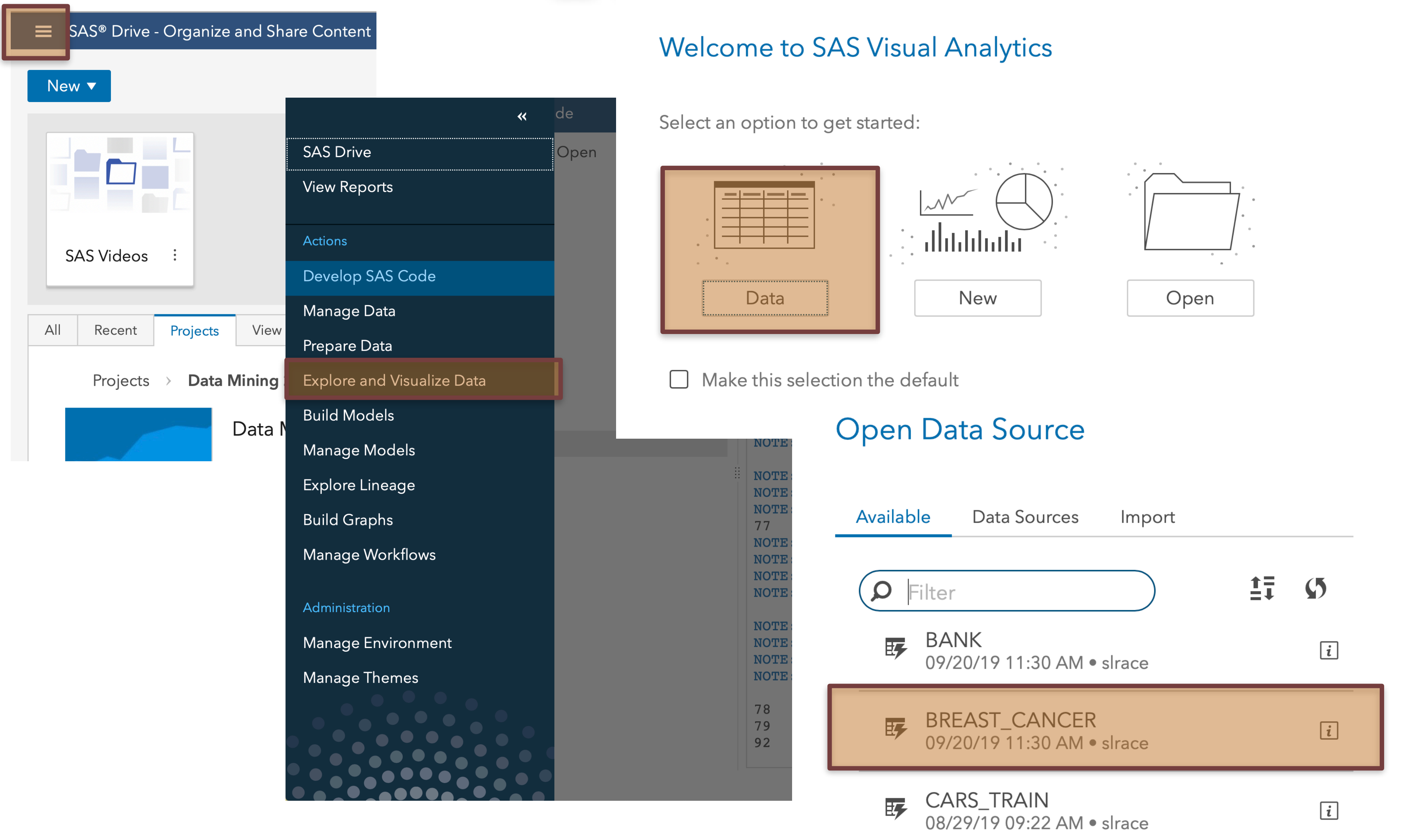


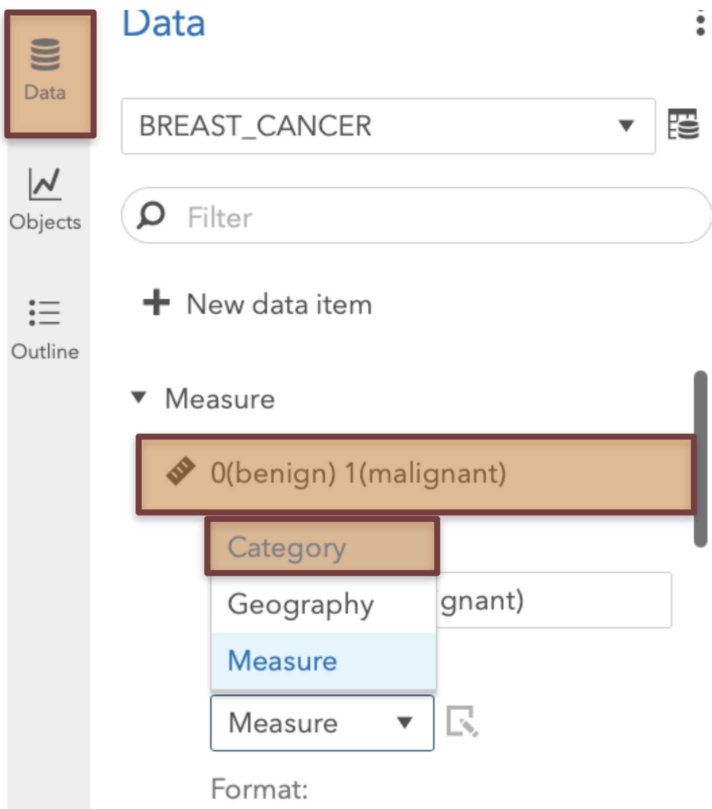
Submit Code:

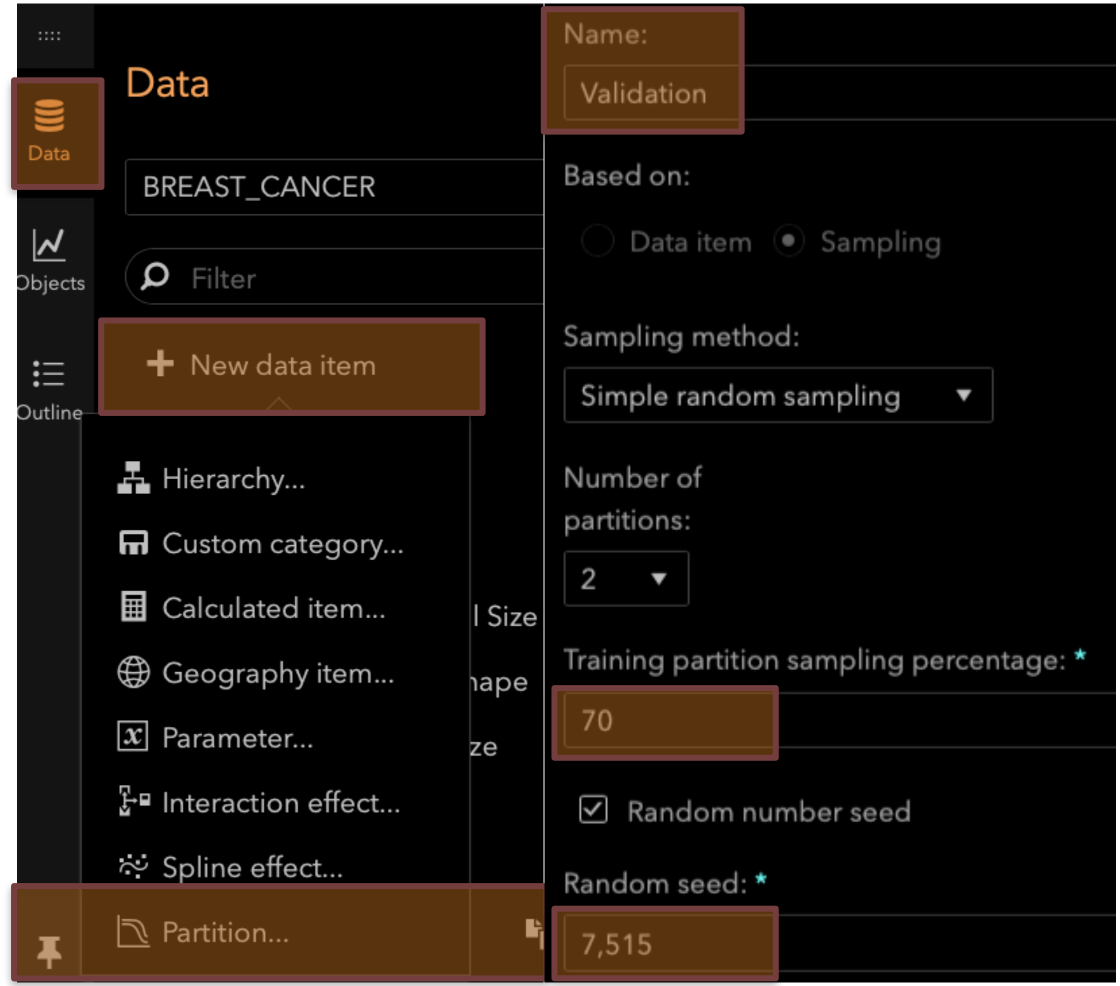
```
cas;
caslib _all_ assign;
```

You will repeat this step EVERY time you use Viya to load the Public library!

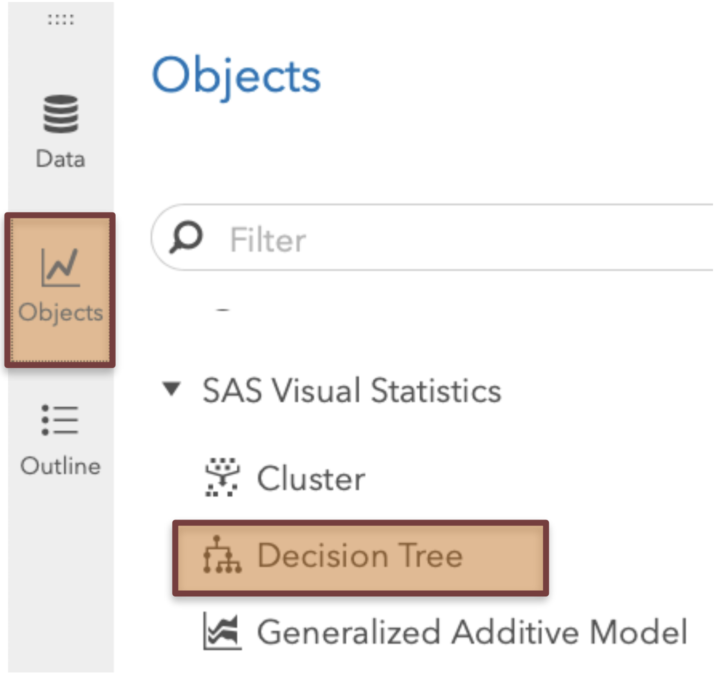
Identifying Malignant Tumors

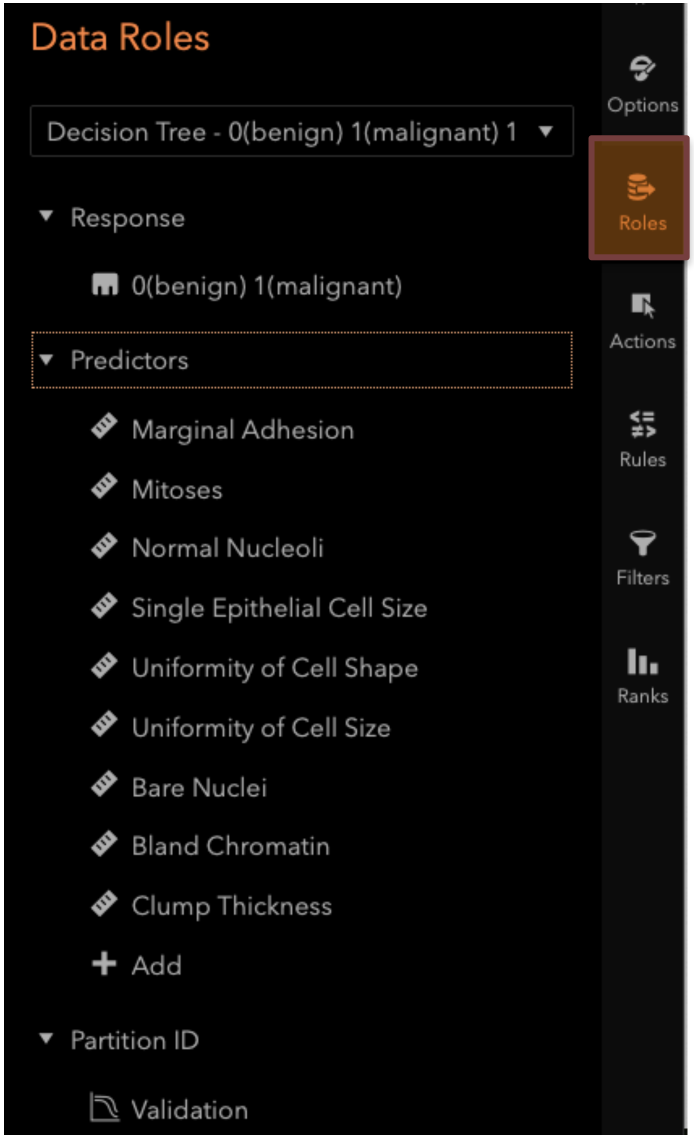






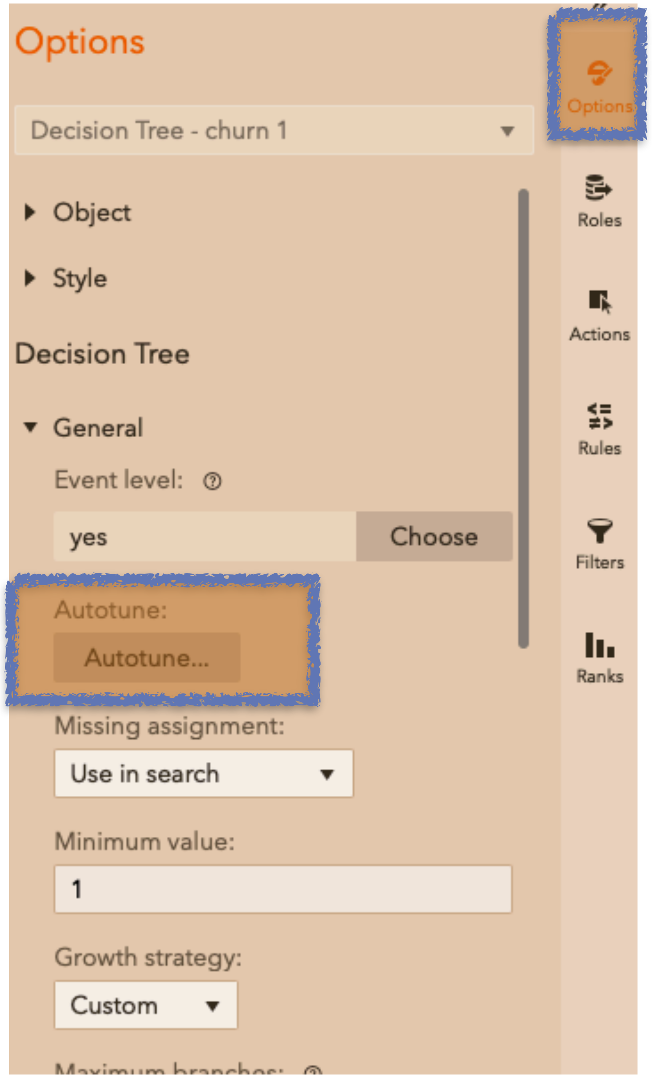
Change target attribute to categoric
variable (split into training/validati



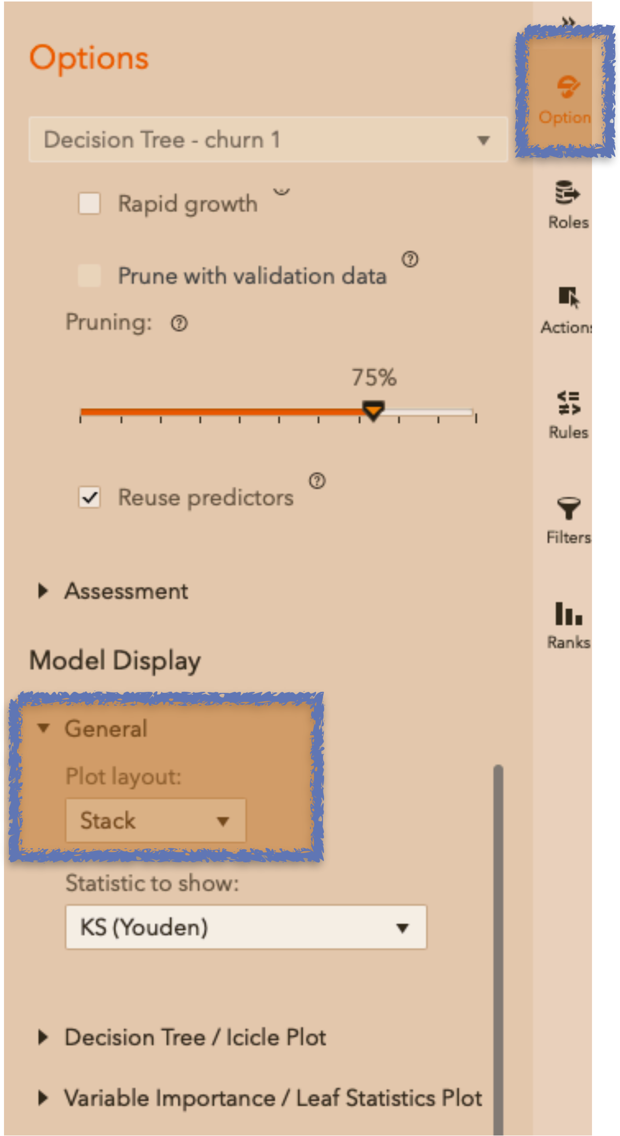


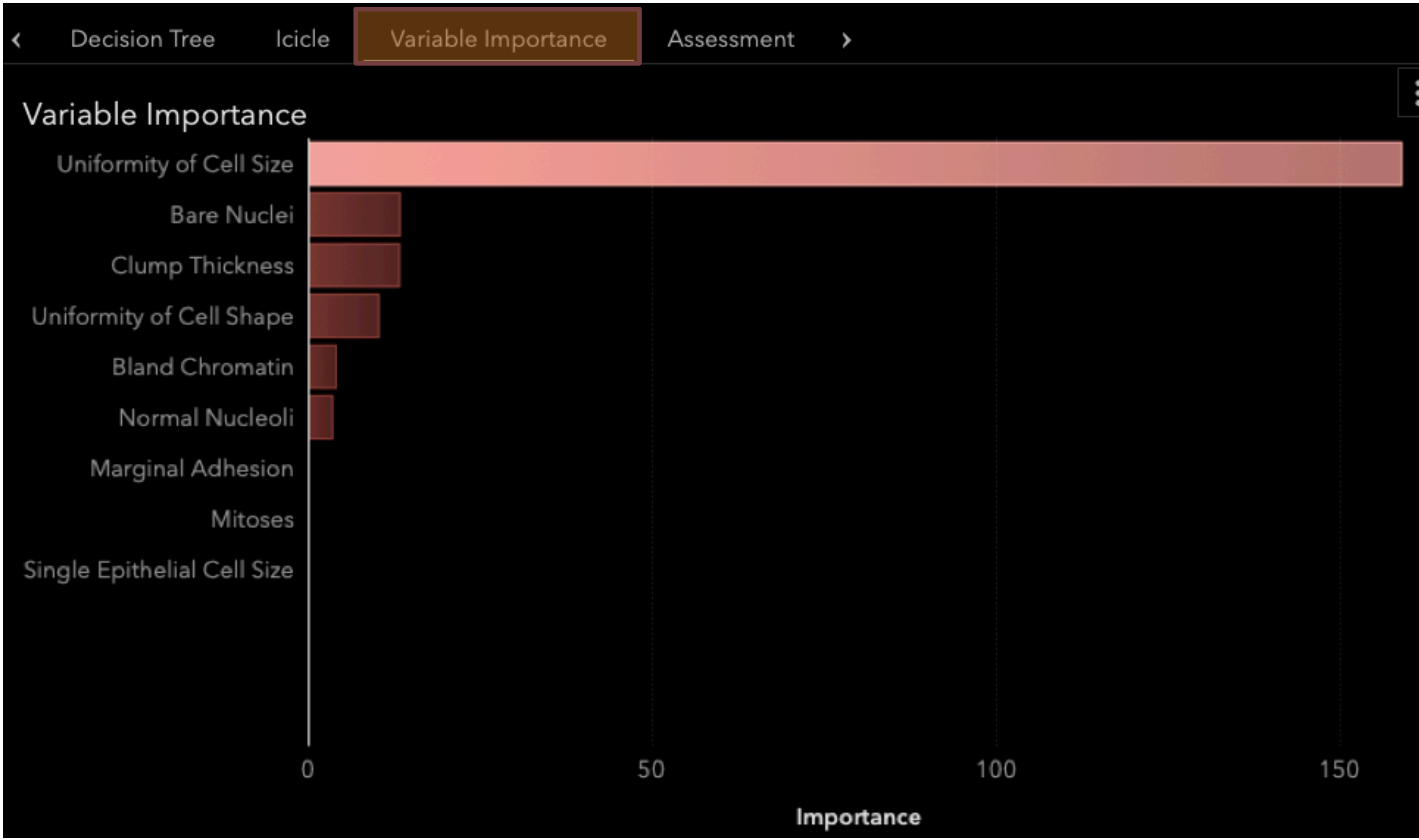
Create a decision tree and set the

Autotune Function

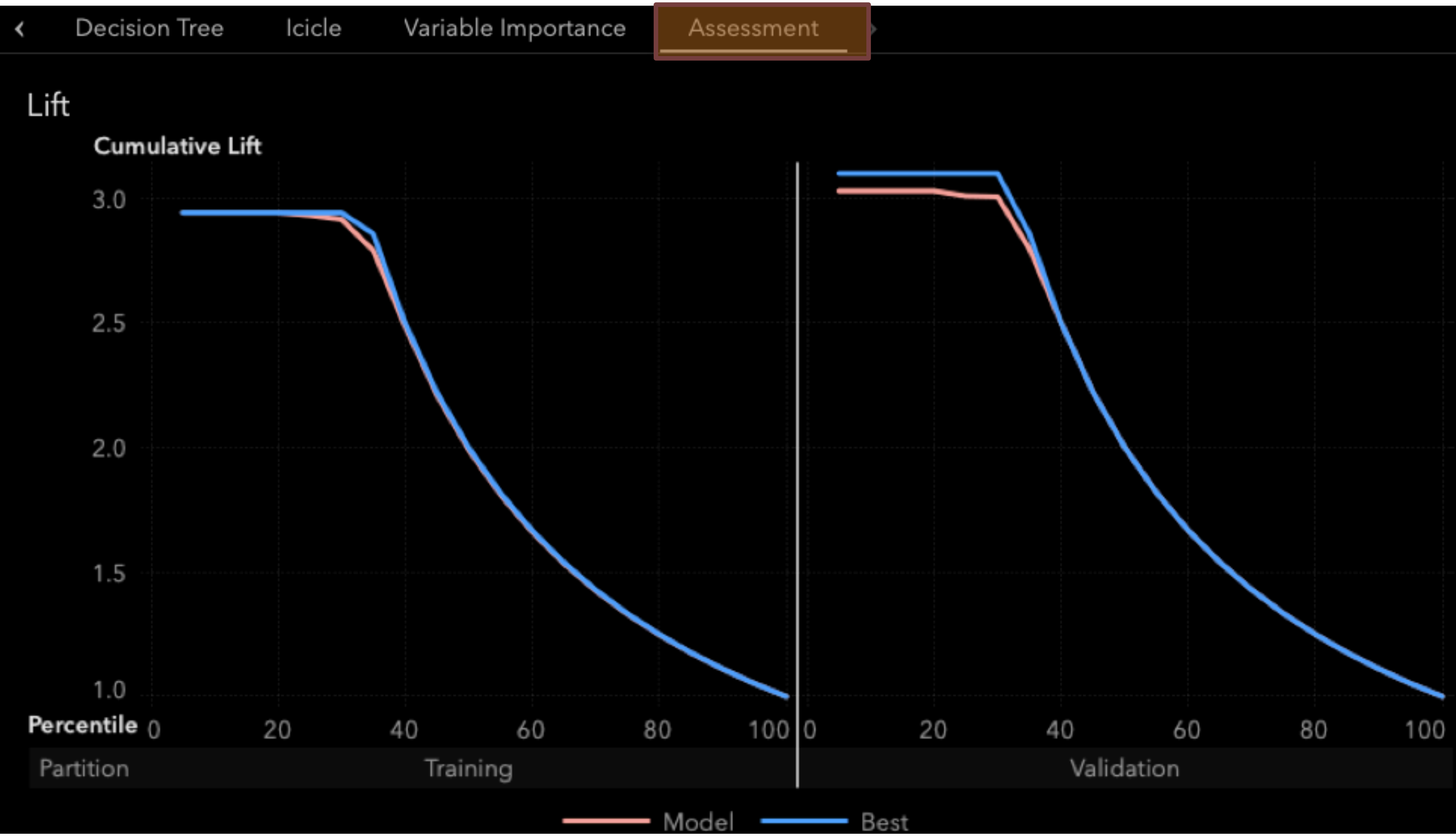


Stack Display





Decision Tree
Icicle
Variable Importance
Assessment
Variable Importance
Uniformity of Cell Size
Bare Nuclei
Clump Thickness
Uniformity of Cell Shape
Bland Chromatin
Normal Nucleoli
Marginal Adhesion
Mitoses
Single Epithelial Cell Size
0
50
100
150
Importance



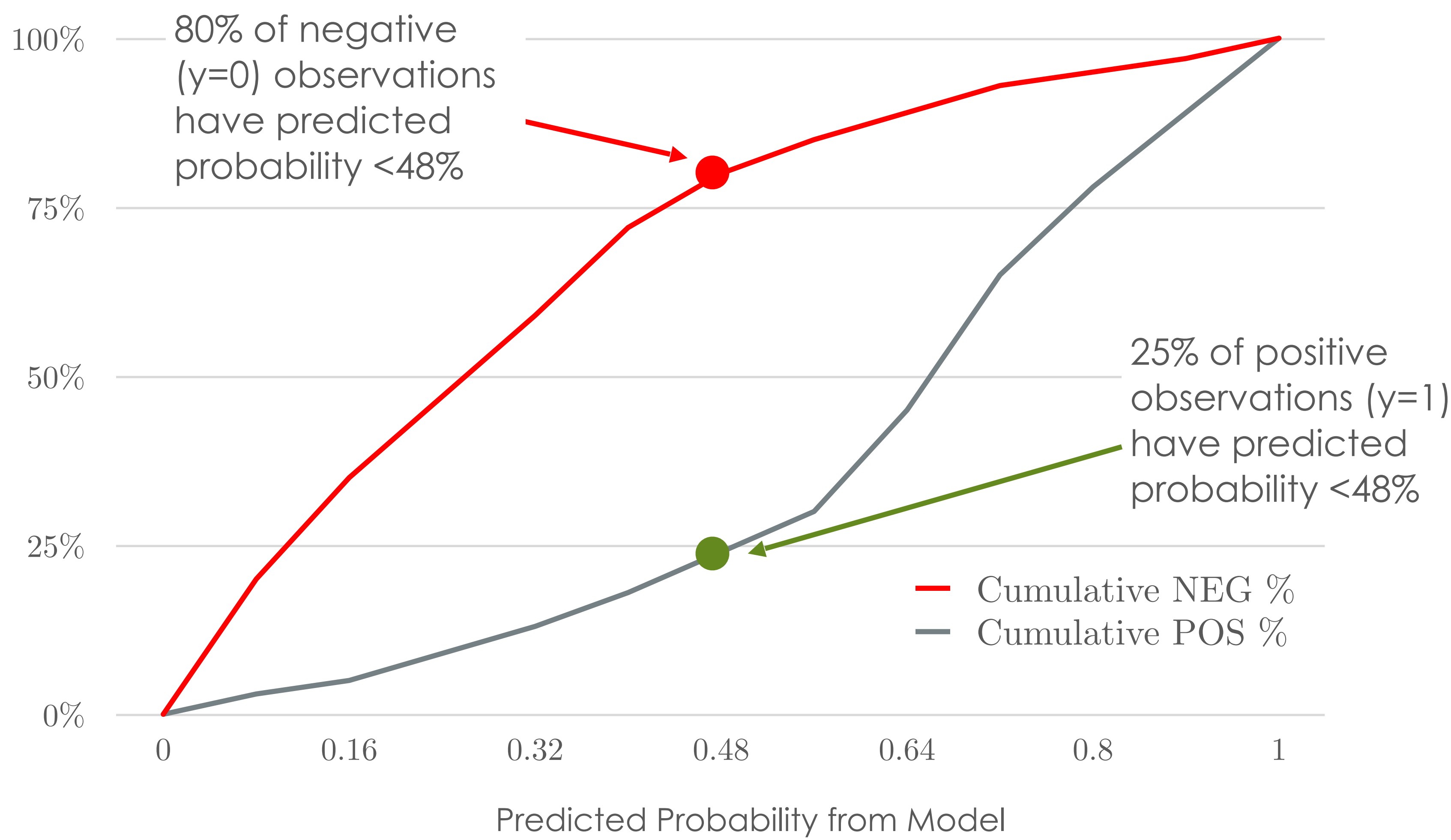
Decision Tree
Icicle
Variable Importance
Assessment
Lift
Cumulative Lift
3.0
2.5
2.0
1.5
1.0
Percentile
0
20
40
60
80
100
Partition
Training
Validation
Model
Best

Additional Reference Slides

• • •

The K-S Statistic

Kolmogorov-Smirnov (KS) Statistic



Kolmogorov-Smirnov (KS) Statistic