Advanced Predictive Modeling Workshop

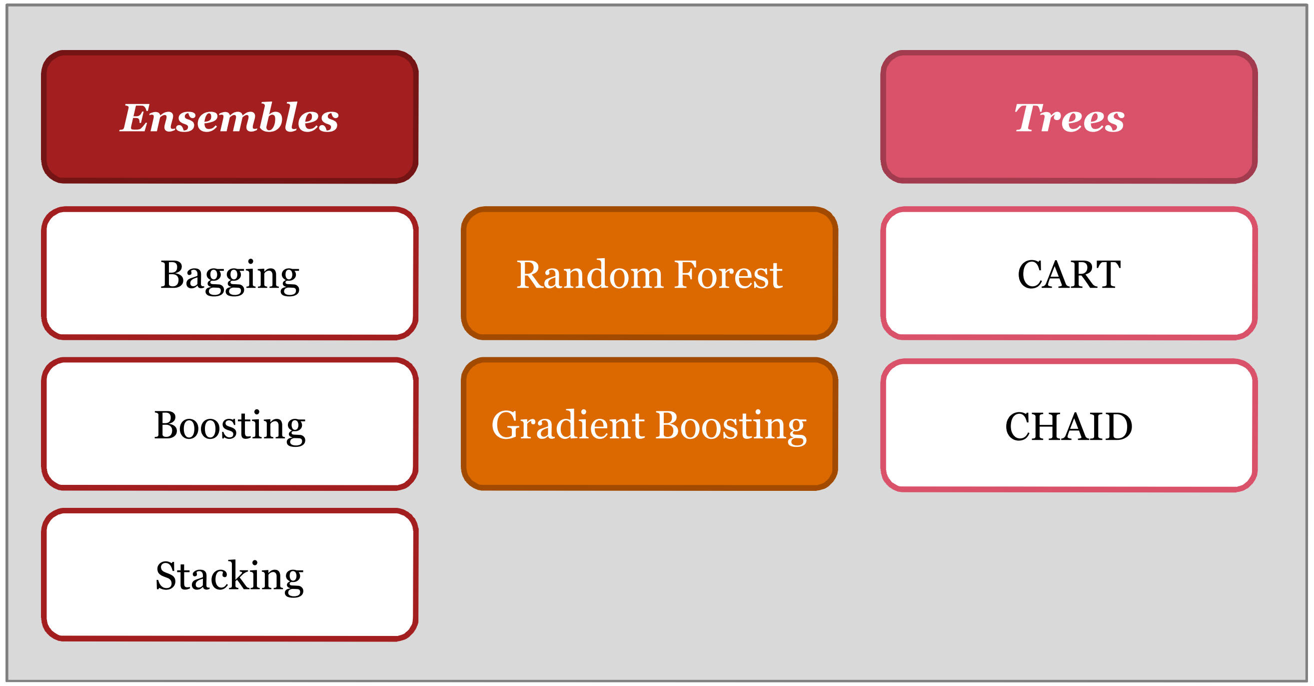
Tree-based Methods

March 27, 2017
San Diego, CA

Kudakwashe Chibanda, FCAS, MAAA
kudakwashe.chibanda@pwc.com
Jean-François Greeff, FASSA
jean.francois.greeff@pwc.com

Agenda

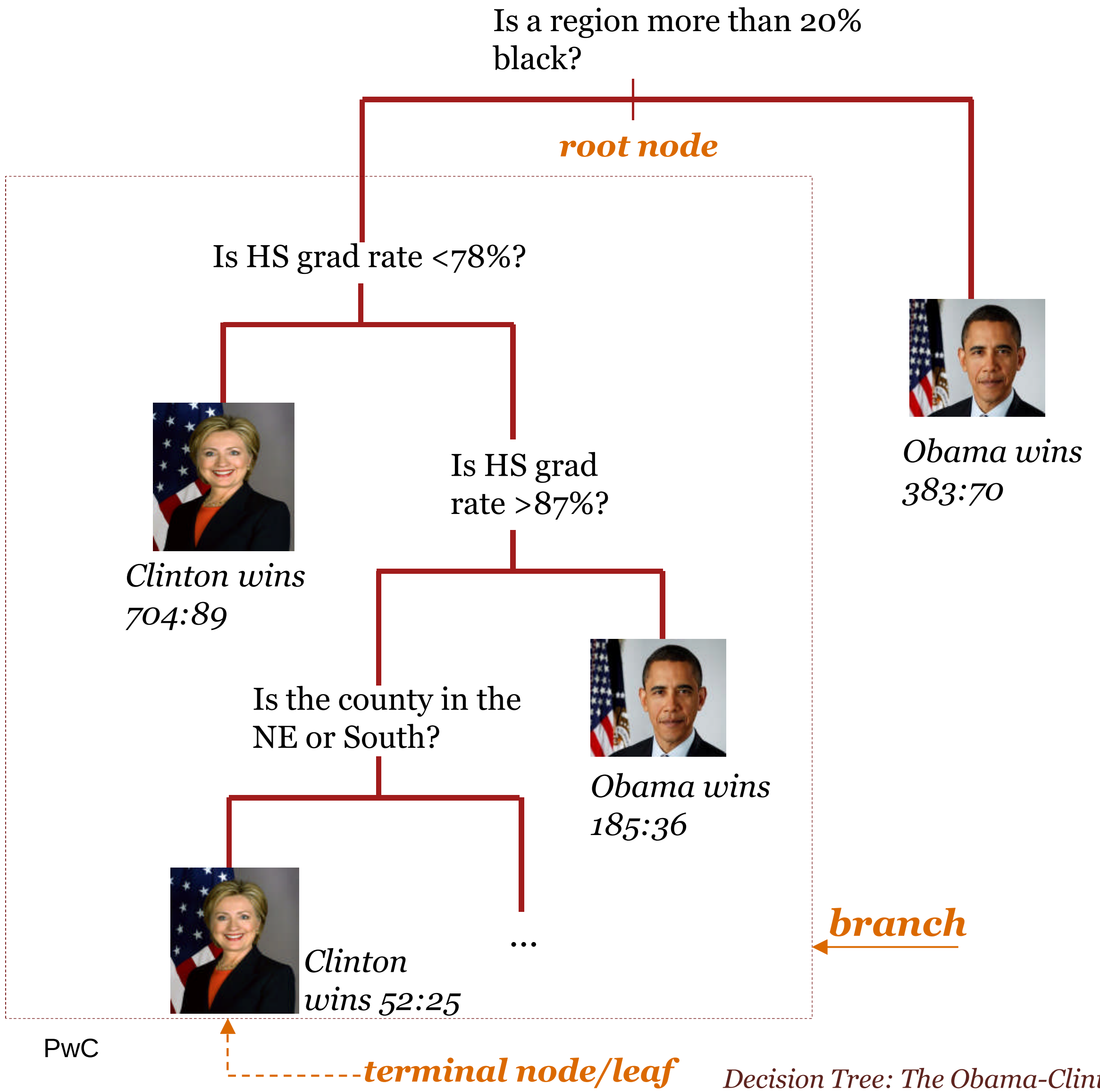
1. Decision trees
2. Ensembles
3. Classification examples
4. Regression examples



Decision trees

“If you dream of a forest, you better learn how to plant a tree”

Introduction



Decision Tree: The Obama-Clinton Divide (from NYT April, 2008)

Features

- Non-parametric classification/regression tools
- Create splits according to measures of homogeneity

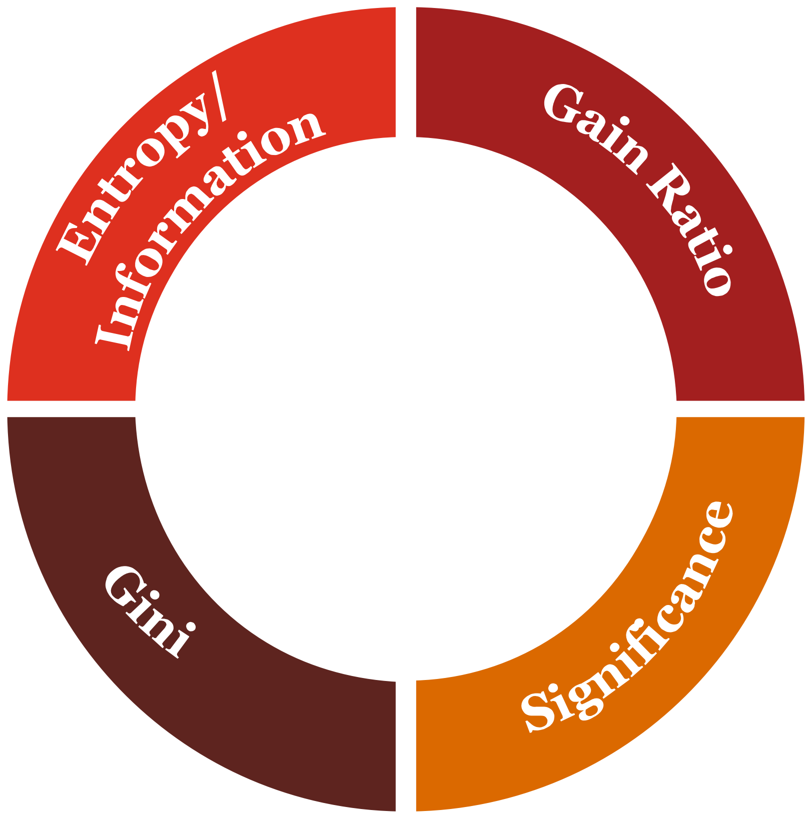
Advantages

- Simple to understand and interpret
- Flexible for non-linear or complex relationships

Disadvantages

- Overfitting
- Unstable/Biased if certain classes of data dominate

Splitting Criteria



- Entropy measures the ***disorderliness*** for each variable level
- The ***purer*** the level for a given response, the more ***predictable*** the outcome
- The weighted average entropy across all levels of a variable gives us information

- A large number of observations in a level can ***bias*** the information towards the entropy of the concentrated level
- To compensate, ***Intrinsic Information*** is calculated
- II takes size and number of levels into account i.e. penalizes large values/splits
- $Gain\ Ratio = \frac{Information\ Gain}{Intrinsic\ Information}$

- Gini impurity is a purity measure that relies on ***misclassification***
- It measures the probability that a randomly selected observation will be placed in the wrong bucket (i.e. misclassified)

- ***p-values of Chi-Square*** statistics can be used to split nodes
- Measure statistical significance of a variable's levels and the response (i.e. test ***null hypothesis of independence***)
- Insignificant splits are merged while significant ones are tested for further splits

Purity Measures Calculations

Outlook (x variable)	Yes (y variable)	No	Total (by level)
Sunny (node i)	3	2	**5**
Overcast	4	0	**4**
Rainy	20	30	**50**
Total (t branch)	**27**	**32**	**59**

Entropy/ Information measure purity of outcomes at each node, taking number and size of nodes into account

Information Gain and Gini also take purity at the branch (regardless of splits) into account

IG measures increase in purity from having no splits [I(t)] to having c splits

- **Entropy** $\left[H_y(i)\right] = -\sum_y p(y|i)\, log(y|i)$

$$Entropy(sunny) = -\left(\frac{3}{5}\log\left(\frac{3}{5}\right) + \frac{2}{5}\log\left(\frac{2}{5}\right)\right) = 0.971$$

- **Information** $[H(t)] = -\sum_{i=0}^{c-1} p(i) H_y(i)$

$$H(Outlook) = \frac{5}{59} * 0.971 + \frac{4}{59} * 0 + \frac{50}{59} * 0.971 = 0.905$$

- **Information Gain** $[(IG(t)] = I(t) - H(t)$

$$IG(t) = -\left(\frac{27}{59}\log\left(\frac{27}{59}\right) + \frac{32}{59}\log\left(\frac{32}{59}\right)\right) - 0.905 = 0.09$$

- **Intrinsic Information** $[II(t)] =$

$$II(t) = -\left(\frac{5}{59}\log\left(\frac{5}{59}\right) + \frac{4}{59}\log\left(\frac{4}{59}\right) + \frac{50}{59}\log\left(\frac{50}{59}\right)\right) = 0.767$$

- **Gain Ratio** $[(GR(t)] = \frac{IG(t)}{II(t)}$

$$GR(t) = \frac{0.09}{0.767} = 0.117$$

- **Gini** $[(G(t)] = 1 - \sum_y p(i|t)^2$

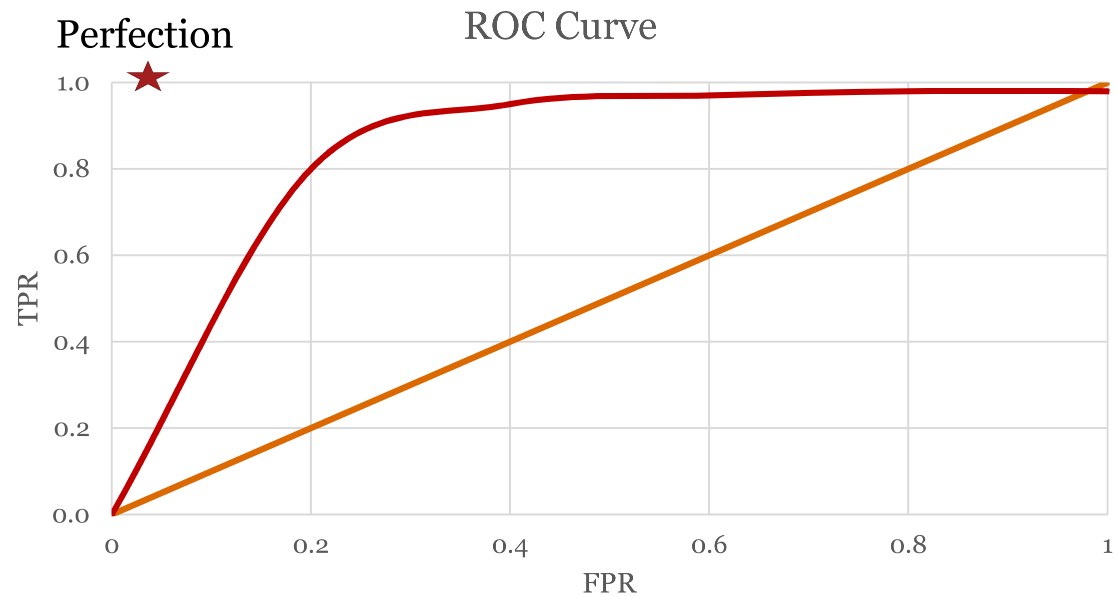
(prior to split)

$$G(t) = 1\ - [\left(\frac{27}{32}\right)^2 + \left(\frac{32}{59}\right)^2] = 0.496$$

Receiver Operating Characteristic (ROC)

- To measure predictive performance in binary classifier models, we rely on ***confusion matrices***
- Using a selected ***threshold***, we can bucket observations into each one of the four buckets as shown in the table
- Receiver Operator Curves (ROC) are commonly used to select a threshold
  - By plotting relationship between TPR and FPR, we can determine the point that *maximizes TPR while minimizing FPR*
  - We can also summarize the information by calculating ***Area Under Curve (AUC)***

	Predicted		
Actual	True Positive (TP)	False Negative (FN)	True Positive Rate (Sensitivity): $TPR = \frac{TP}{TP + FN}$
	False Positive (FP)	True Negative (TN)	False Positive Rate (Fall-out): $FPR = \frac{FP}{FP + TN}$



Types of Trees – ID3 and C4.5

ID3

- *Purity measure*: ***Entropy***
- *Methodology:* at each node, calculate entropy for all variables. Select variable with minimum entropy
- *Splits*: can have multiple splits
- *Continuous/missing data*: no
- *Risks*: does not ***prune***
    - *Fix*: use ***stopping criteria*** to avoid overfitting

C4.5

- *Purity measure*: ***Information Gain***
- *Methodology & splits:* similar to ID3
- *Continuous/missing data*: yes
- *Risks*: susceptible to ***outliers***
    - *Fix*: remove outliers

Classification And Regression Trees (CART)

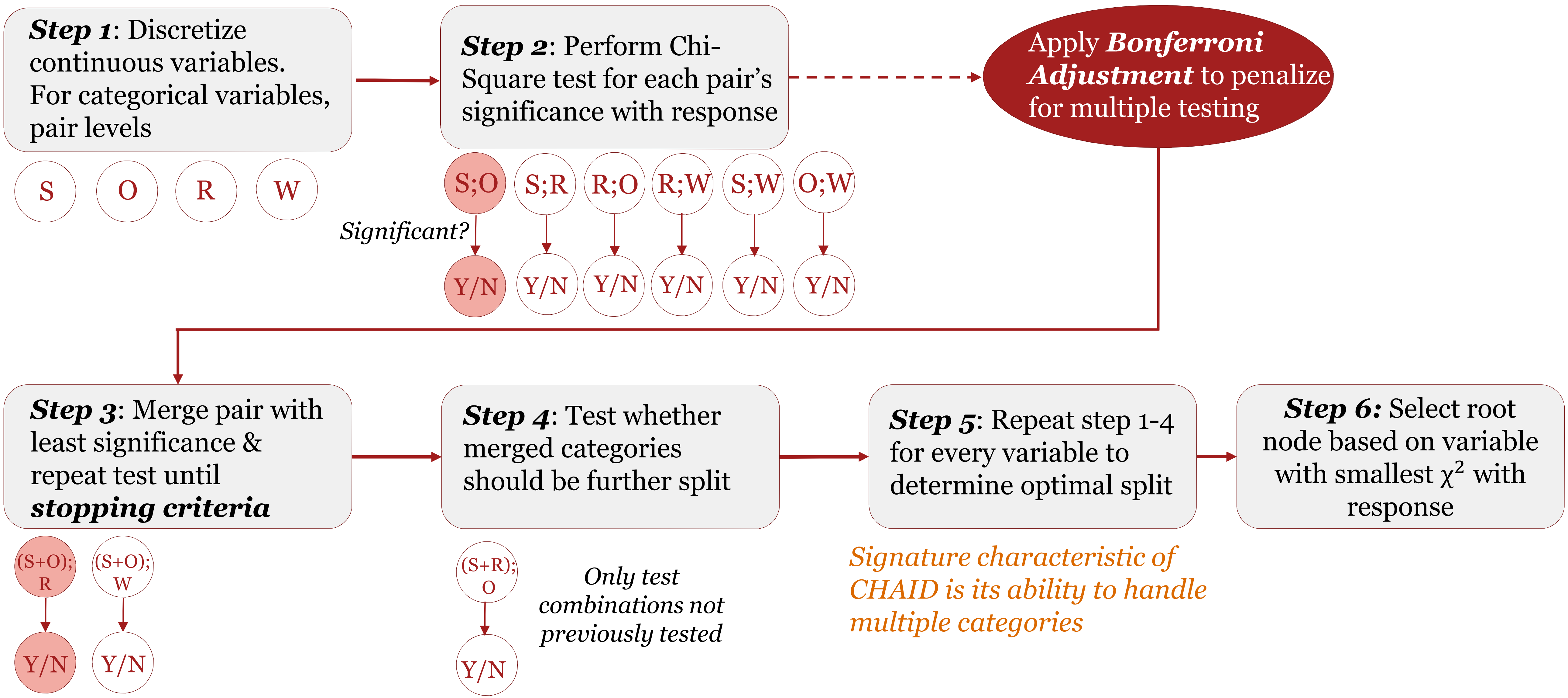
Classification Trees

- *Purity measure*: ***Gini impurity***
- *Methodology:* at each node, calculate gini for all variables. Select split with minimum gini
- *Splits*: ***binary***
- *Continuous data*: requires splitting
- *Risks*: does not work for ***multiple category*** data
  - *Fix*: use CHAID/ID3

Regression Trees

- *Purity measure*: ***Variance reduction***
- *Methodology:* For each variable, the split is determined by the point that ***minimizes SSE***
- *Continuous/missing data*: yes
- *Risks*: ***overfitting***
  - *Fix*: ***prune*** using Sum of Square Errors (SSE)

Chi-square Automatic Interaction Detector (CHAID)



Classification Example

Ensembles

Weak Learners and Strong Classifiers

Weak learners

- Performs well only on a subset of the domain
- May be unstable with small perturbations in data
- May be biased in its predictions

Typically what we **have**

Strong classifiers

- Performs well over the whole domain
- Stable across small changes in the data
- Unbiased in its predictions

What we **want**

?

Illustration of Ensembling (1)

Situation

- Transmit binary signal from A to B
- Ensure that signal uncorrupted

Ensemble approach

- Use 3 independent signal carriers
- Majority vote (Choose bits where 2+ of three carriers agree)

Consequence

- Reconstruct signal with reduced error

	Signal	Accuracy
Original signal	0100101001000110	
Signal 1	0100001001000110	93.75%
Signal 2	0100101001000111	93.75%
Signal 3	0100100101000110	87.5%
Combined Signal	**0100101001000110**	100%

	Signal	Accuracy
Original signal	0100101001000110	
Signal 1	010000101010101	60.00%
Signal 2	000011100011110	60.00%
Signal 3	010000000010010	66.67%
Combined Signal	**010000000010110**	73.33%

Illustration of Ensembling (2)

- 3 signals with probability of corruption 30% per bit
  - $P(All\ correct) = 0.7^3 = 34.29\%$
  - $P(2\ correct) = 3 \times (0.7^2 \times 0.3) = 44.09\%$
  - $P(1\ correct) = 3 \times (0.7^1 \times 0.3^2) = 18.90\%$
  - $P(None\ correct) = 0.3^3 = 2.70\%$
- Correction made for 44.09% of the bits
- Expected accuracy of 78.38% per bit

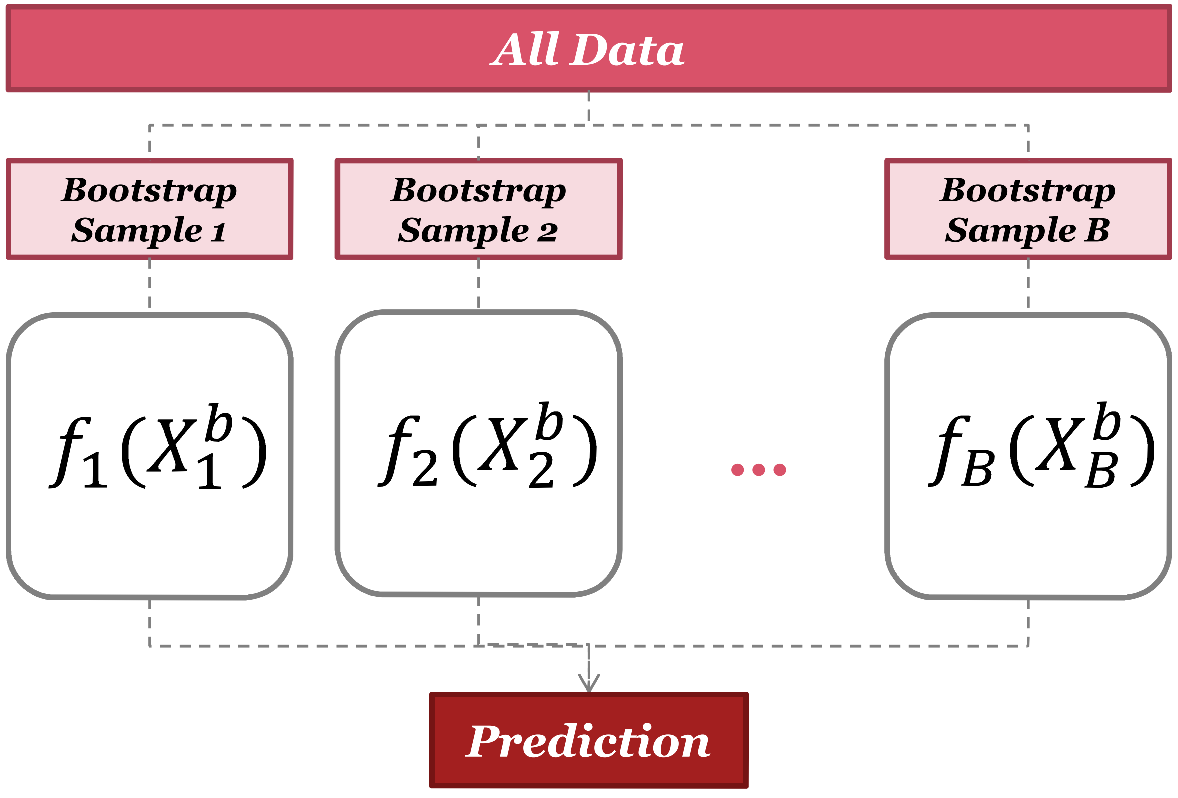
Only if signals ***uncorrelated***

Bagging

Bootstrap Aggregation

Algorithm

1. Create bootstrap resample of data
2. Fit model on each resample
3. Scoring:
   - Classification: Majority vote
   - Regression: Mean/Median score



Advantages

- Produces more stable predictions – i.e. reduces variance
- Less likely to over-fit data

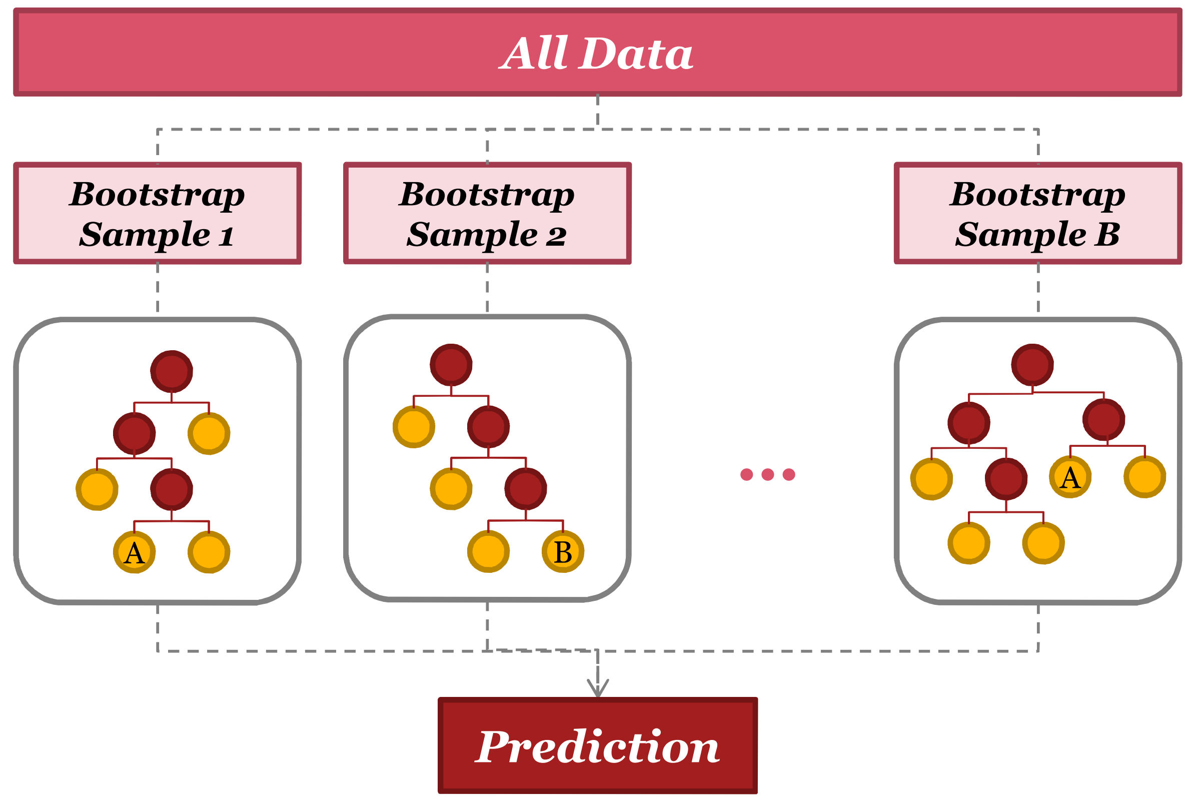
Disadvantages

- Generates a "black box"

Random Forests

Bagging Decision Trees

- Introduced by Leo Breiman (2001)
- Uses bagging to improve decision trees
- De-correlates trees by sampling
  - o Data with replacement
  - o Columns/features at each node
- Produces out-of-bag error rates
- Produces variable importance measure
- Parameters to tune*:
  1. Number of trees
  2. Number of features to select at each node



* There are other parameters such as the sampling rate and maximum depth of the tree.

Boosting

Algorithm

- Rather than fitting models to bootstrap samples of the data – boosting fits sequential models focusing on areas of poor performance
- Subsequent models correct errors of previous models

Advantages

- Decrease bias in predictions

Disadvantages

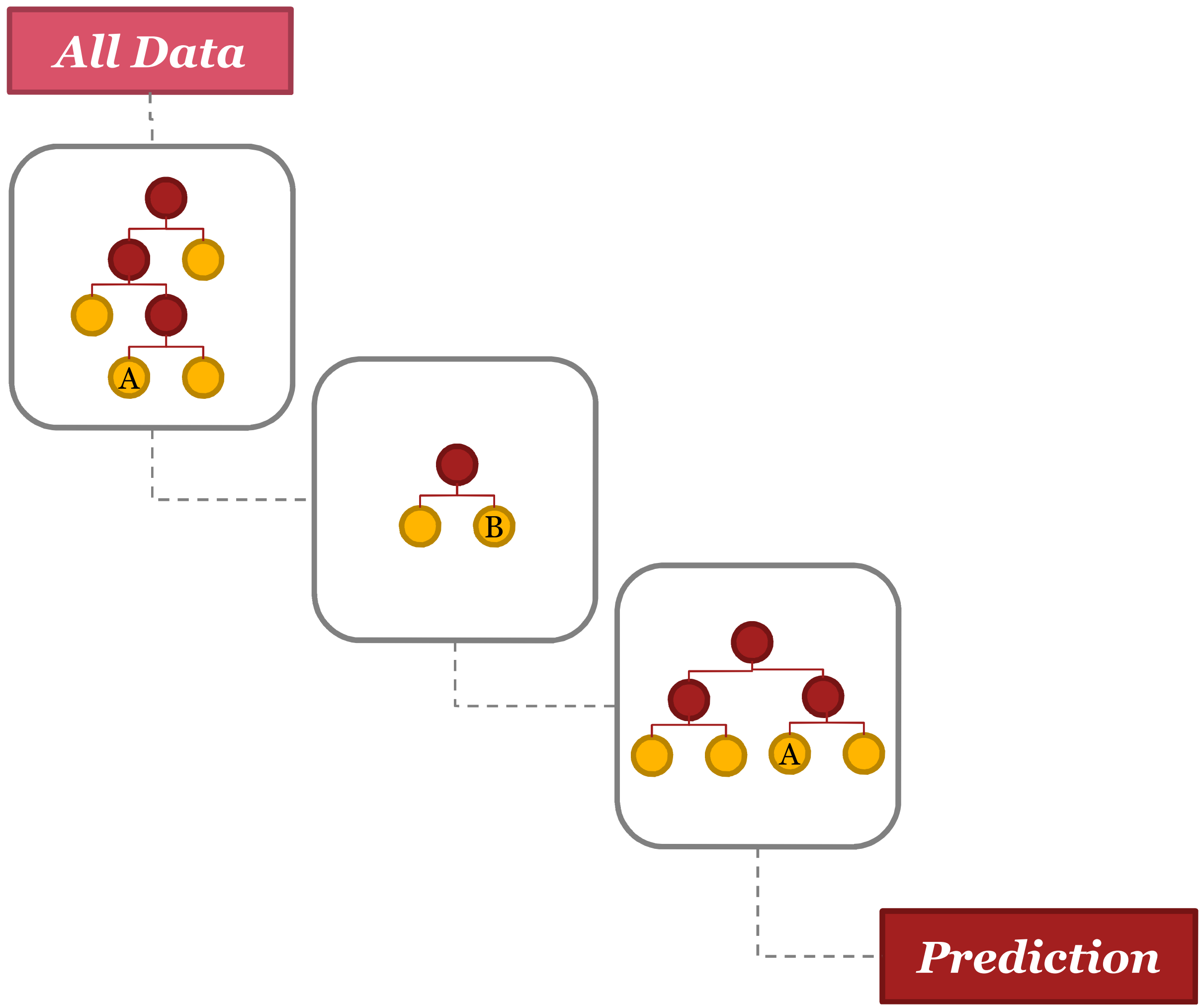
- Generates a “black box”
- May be sensitive to outliers and noise

AdaBoost	GBM
Adaptive Boosting	Gradient Boosted Machines / Models
Fits model to weighted distribution of the data. More weight is given to observations that have the highest error rate.	Fits model to the residual of the prior models.

Gradient Boosted Trees

Boosting Decision Trees

- Introduced by Jerome Friedman (1999)
- Uses boosting to improve decision trees
- XGBoost algorithm most common
  - o Stochastic gradient descent
  - o Feature sub-sampling
- Parameters to tune*:
  1. Number of trees
  2. Depth of trees
  3. Learning rate



Stacking

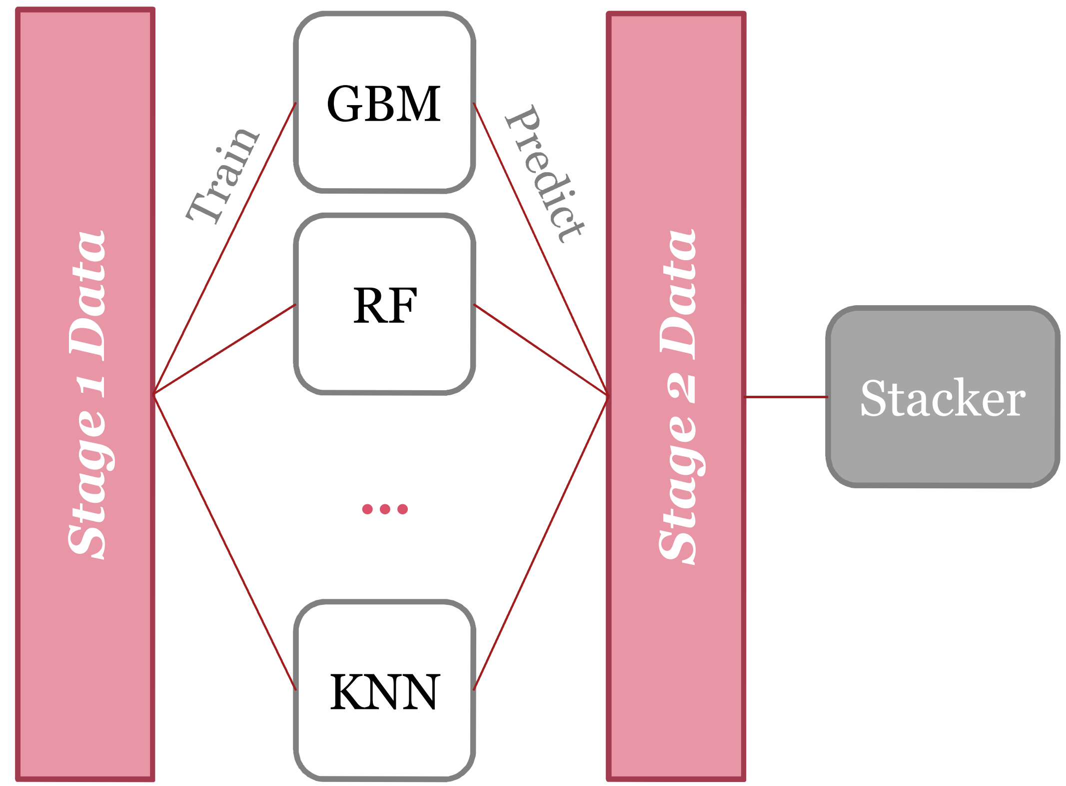
Stacked generalization & Blending

Algorithm

- Two stages of model fitting
  1. First Stage: Fit base learners to data
  2. Second Stage: Fit meta-learner to predictions of base learners

Considerations

- Different approaches to how the stacking is performed
- Careful consideration needs to be given to what data is used at what stages
- Need diverse models



Classification Example

Predictive Modeling Applications

Advanced Predictive Modeling Workshop

Tree-based Methods

Q&A

Kudakwashe Chibanda, FCAS, MAAA
kudakwashe.chibanda@pwc.com
Jean-François Greeff, FASSA
jean.francois.greeff@pwc.com