DECISION PATTERNS

Decision Trees / Recursive Partitioning

Practical Machine Learning (with R)

UC Berkeley

Spring 2016

REVIEW AND EXPECTATIONS

Topics

- Administrativa
  - Role Call
  - Assignments due to github
  - Miscellaneous: BIDS Open House
- Review/Expectations
  - Last Lecture
- New Topics

Resampling

DO NOT ESTIMATE PERFORMANCE ON TRAINING DATA!

- Calculate ***unbiased*** performance:
  - Repeated resampling
  - K-Fold Cross Validation
  - Bootstrap

- Use additional hold out, if data permits

Binomial Performance

- Know where to look up formulas/definitions

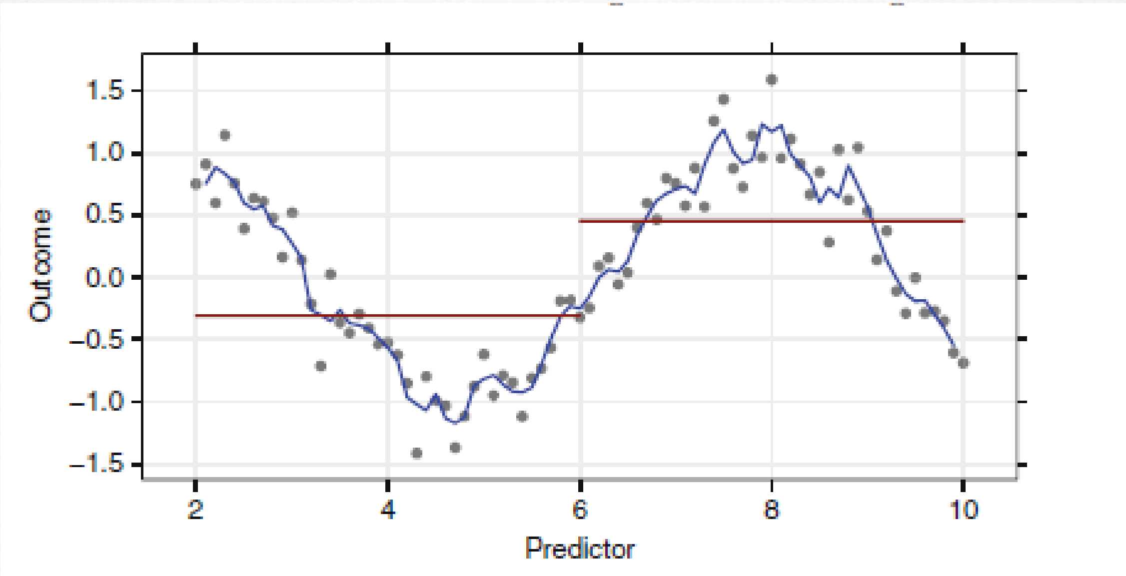
- Know
  - `Accuracy, Error Rate`
  - `[F|T][P|N]R`
  - Sensitivity, Specificity
  - Type I and Type II Errors

READING

Model Performance

- RMSE, MSE
- $R^2 \cong cor(y, \hat{y})$

Variance-Bias Trade-off



$$E[MSE] = \sigma^2 + (model\ bias)^2 + model\ variance$$

BIAS: How close the model comes to the true value. (High bias → poor fit)
VARIANCE: Stability of the model, susceptibility to new values

Classification Performance

- `predict` methods can provide
  - Classes
  - Class probabilities ⟸ Better!

- Class probs → Classes?
  - Apply ***softmax*** function

$$\hat{p}^*_\ell = \frac{e^{\hat{y}_\ell}}{\sum_{l=1}^{C} e^{\hat{y}_l}}$$

- Probabilities often need post predict → calibrations (talk about this with deployment)

Classification Performance

- Accuracy … problems?
- Confusion Matrix
  - `table`
  - `caret::confusionMatrix`
- `Cohen's Kappa:` $\kappa = \frac{O-E}{1-E}$
  - Kappa values within 0.30 to 0.50 → good fit
- ROC Curves / Lift Charts

CARET

Caret

"Misc functions for training and plotting classification and regression models."

- Really:
  - Wraps 100's of modeling functions
  - Automates tediousness of model building
  - Manages a process

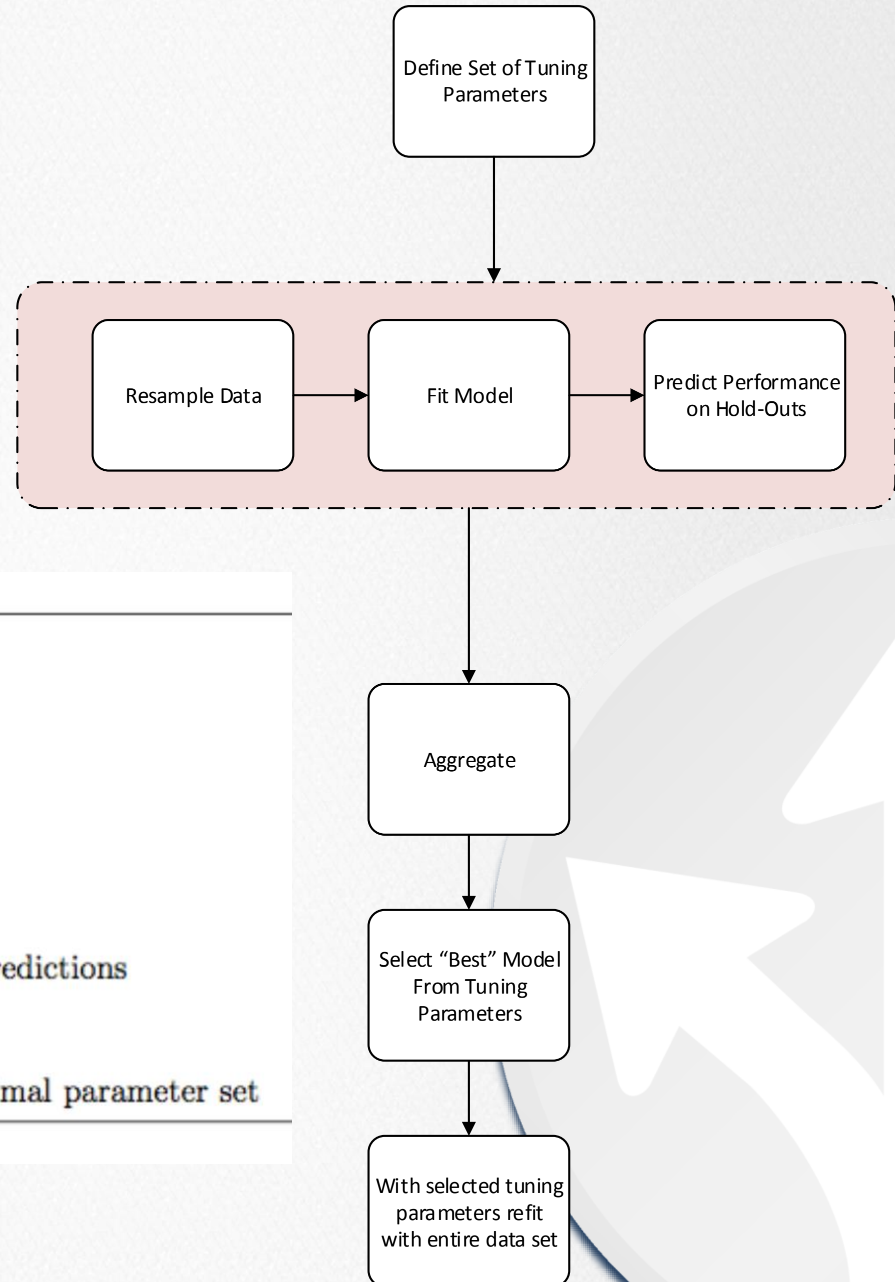
- Competitors:
  - mlr (machine learning with R): task focused
  - Rattle : Graham Williams et al. / Togaware.com
  - R Commander : Statistical workbench

Caret Goals

Does a couple things:

- Preprocess data (transfroms, imputes)
- evaluate, using resampling, the effect of model tuning parameters on performance
- choose the "optimal" model across these parameters
- estimate model performance from a training set
- Variable Importance
- Aids feature selection

Process




```
Define sets of model parameter values to evaluate
for each parameter set do
    for each resampling iteration do
        Hold–out specific samples
        [Optional] Pre–process the data
        Fit the model on the remainder
        Predict the hold–out samples
    end
    Calculate the average performance across hold–out predictions
end
Determine the optimal parameter set
Fit the final model to all the training data using the optimal parameter set
```

Lots of Configurations

- Easy if you know what you are doing
- which `method`?

  [Caret Model List]*

- Controlled mostly through
  - `train (tuneLength, tuneGrid)`
  - `trainControl – supplied to train`

GERMANCREDIT / CARET EXAMPLE

NEW TOPICS

Logistic Regression: Multiple Classes (omit)

- If logistic regression predict 0-1 for a class, how do we get it to predict multiple classes?

- Create multiple models, one per class
- Get each prediction for each class

- Apply ***softmax*** function

$$\hat{p}^*_\ell = \frac{e^{\hat{y}_\ell}}{\sum_{l=1}^{C} e^{\hat{y}_l}}$$

- Select highest probability

KNN

ADVANTAGES	DISADVANTAGE

DECISION TREES / RECURSIVE PARTITIONING

Linear Methods

Advantages

- ...
- ...

Disadvantages

- ...
- ...
- ...

Linear Methods: Limitations

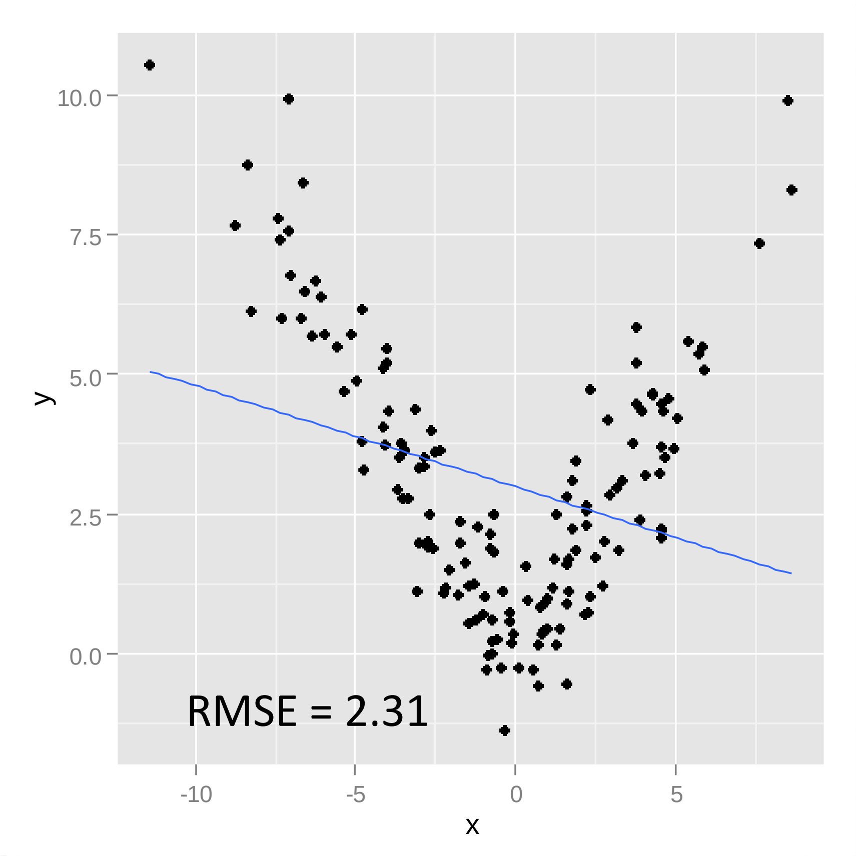
Advantages

- Interpretable
- Easy to train

Disadvantages

- Logistic regression: multiclass problems
- Highly sensitive to inputs
- Linear functions →
  do not model real data well

Linear Models



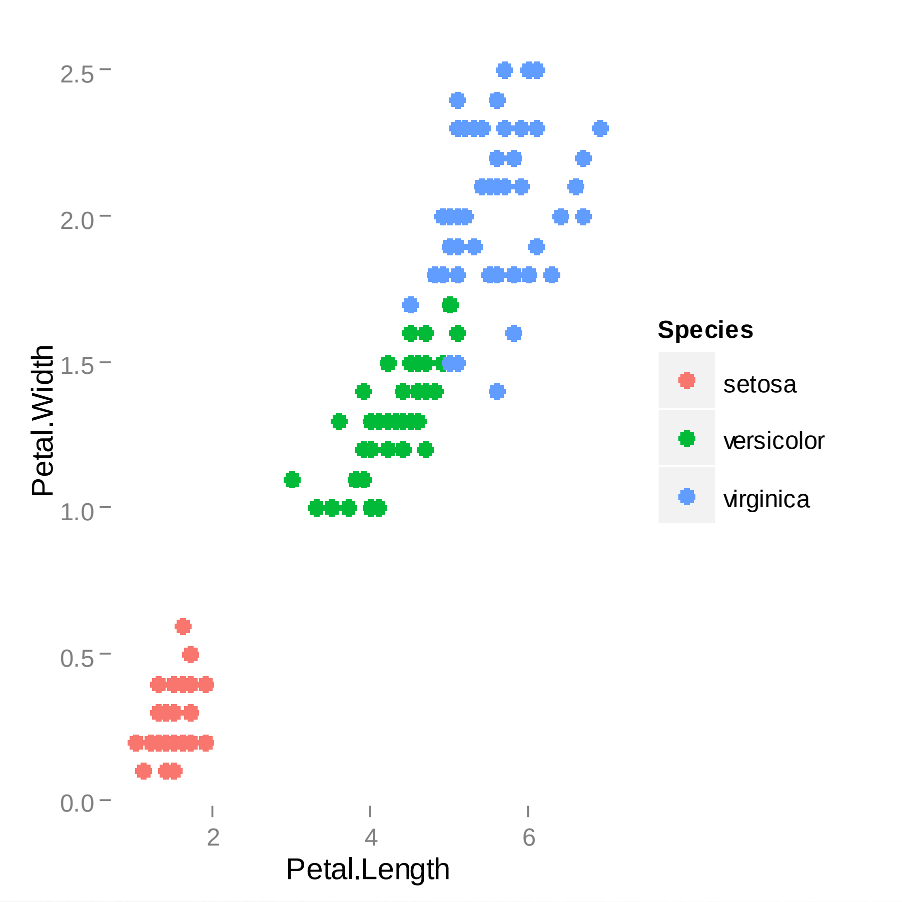
Partition Goal:

PARTITION INPUT SO THAT THE RESULTING SMALLER GROUPS ARE MORE HOMOGENEOUS THAN THE PARENT.

PROCEDURE

- Find best univariate plane to split the data into two subsets, such that the subsets are more alike than there parents
- 1. In each resulting subset ("node", "leaf") find the best univariate split, but only split the best one of these.
- 2. Repeat until stopping condition is met.

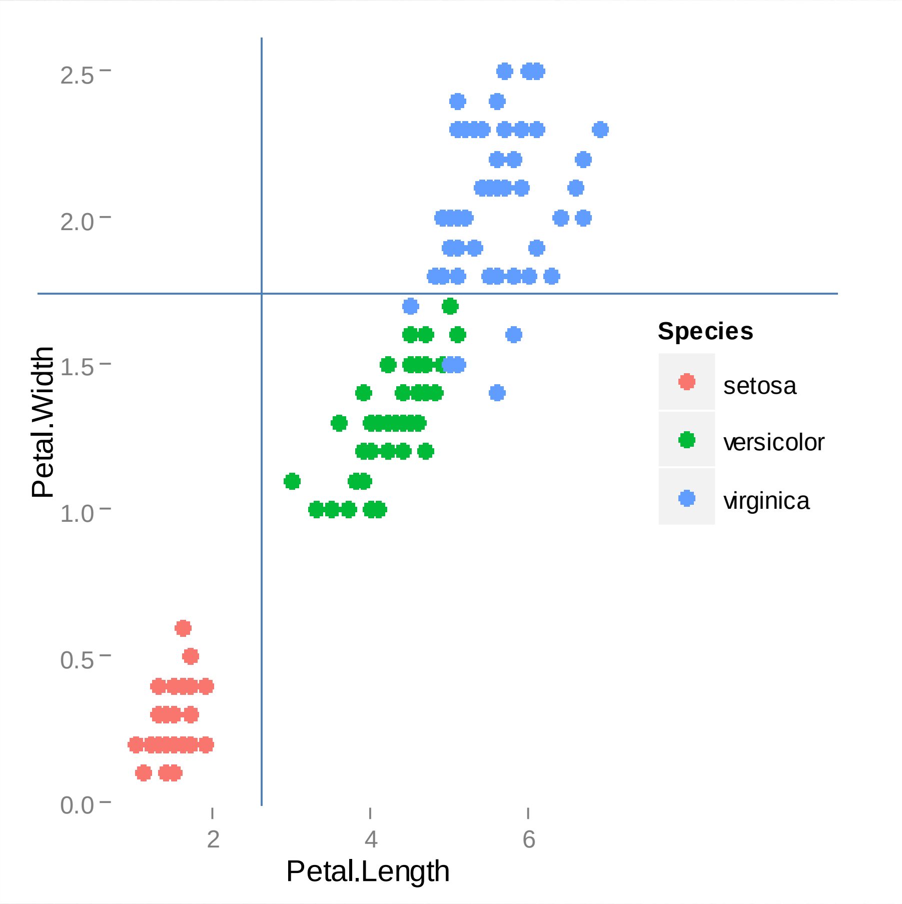
A Simple Example



Partitioning Requirements

- Restricted Class of Functions:
  - First order propositional logic (for partitions)
  - Aggregation (for outcomes)
- Error Methods
  - Normal error calculations
- Search Methods
  - Recursive Partitioning

A Simple Example

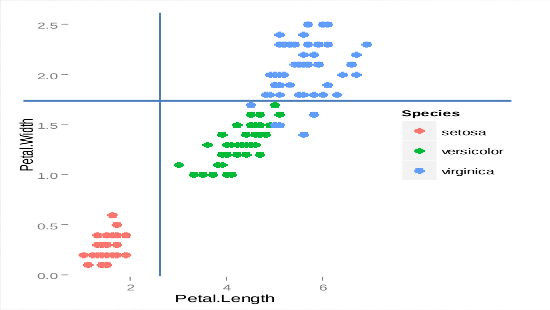


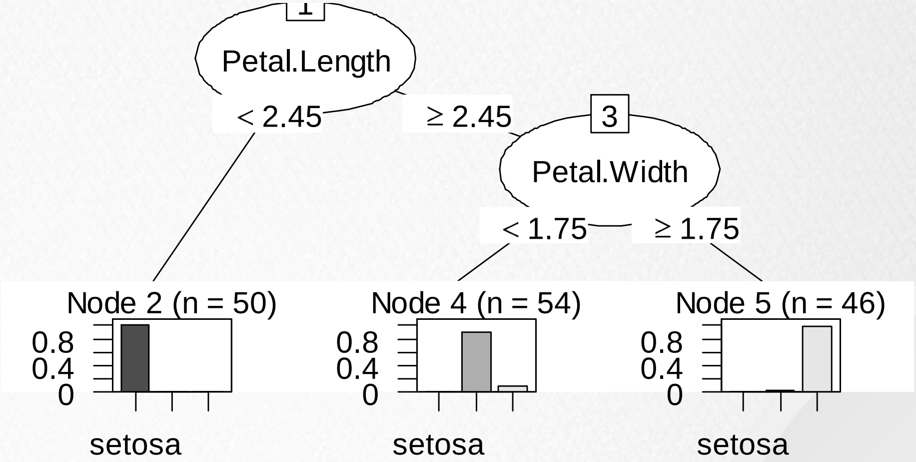
Partitioning Requirements

- Restricted Class of Functions
  - First Order Propositional Logic (for partitions)
  - Aggregation (for outcomes)
- Error Methods
  - Standard Error Methods
  - Regression: SSE, etc.
  - Class.: Misclassification Rate, etc
- Search Methods
  - Recursion and Exhaustive

Some notes

Splitting by planes is the same as a tree





Partitions define a rule*

- Rules can be associated with outcomes → aggregation method

Trees always partition "all of of space"

Splitting on Categorical Variable

- RMSE
- Accuracy/Error rate
- *Gini* index (Class)
- Information Crit.

- Select "metric"
- For each categorical variable
  - Find $argmin_{s \in S}(\sum_{S_i} err_i)$, i = 1..2
- Calculate: $\sum_{S_i} err_i$

Treatment of Categorical Variables

- Grouped Categories
  - Value treated as related

- Independent Categories
  - Values Treated as Independent

Gini Index (Two-Class Classification)

- Measure node purity:

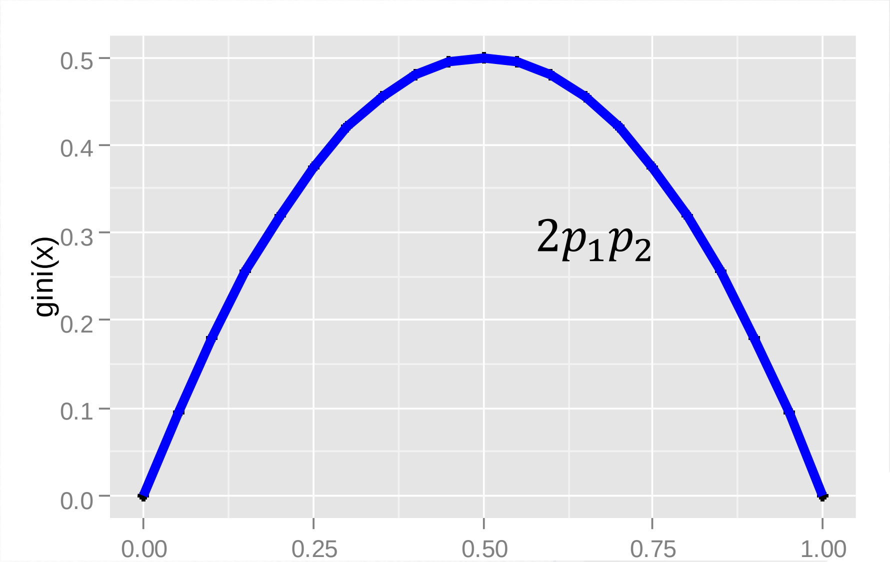
$$p_1(1-p_1)+p_2(1-p_2)$$

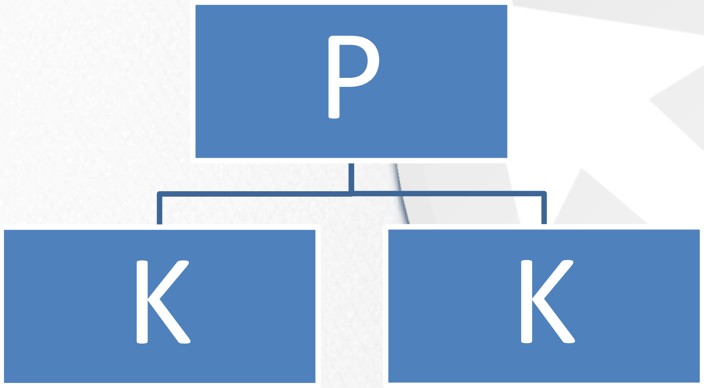
For two class:

$$p_1+p_2=1$$

$$2p_1p_2$$

Minimize! Is the weighted sum Gini index smaller than that of the parent?





Splitting on Continuous Variable

- Determine Metric
- Order data
  - If metric is a "cumulative" function calculate as cumulative function:

    e.g. $FPR = cumsum(FP)/cumsum(TN + FP)$

  - Otherwise calculate at all possible split points or subset of split points

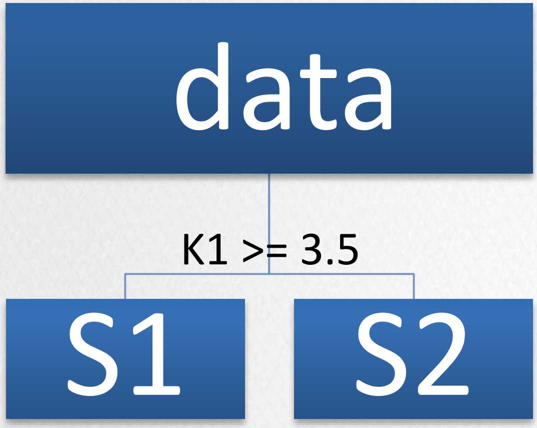
$$argmin_{x=n}(\sum_{i=1..2} err_i)$$

data

Ordinal							
Categorical			Continuous				
K1	**K2**	**K3**	**V1**	**V2**	**V3**	**V4**	**V5**
E_{K1}	E_{K2}	E_{K3}	E_{V1}	E_{V2}	E_{V3}	E_{V4}	E_{V5}

Choose the split that minimizes the error

$$argmin_S(Error)$$



Ordinal							
Categorical			Continuous				
K1	K2	K3	V1	V2	V3	V4	V5
E_{K1}	E_{K2}	E_{K3}	E_{V1}	E_{V2}	E_{V3}	E_{V4}	E_{V5}

Choose the split that minimizes the error

$$argmin_S(Error)$$

REPEAT WITH S1 AND S2

* Very often predictor will be used again.

Missing Data

- Missing values in predictors are common
- A split determines which observations go to the LHS and RHS. How to Handle `Nas`?
- `NA_Categorical`
  - Treat as separate category
- `NA` (in general)
  - Use **Surrogate Splits**

Surrogate Splits

- Tree is built ignoring missing data
  - Any record with incomplete data (response or predictor) is rejected -or-
  - Missing data is rejected from determined the split
- Variables are often collinear → splits are similar and send variables down the same path.
  - Choose a surrogate split that best approximates the chosen split (accuracy)
  - Very often this is also a good split.

Tree Method Advantages I

- Highly interpretable
- Predict easy to implement (even in SQL)
- Handle many predictors (sparse, skewed, continuous, categorical) --> little need to pre-process them
- Non-parametric: do not require specification of predictor-response relationship

Tree Method Advantages I

- Inherent method for handling missing data
- Trees insensitive to monotonic (order-preserving) transformation of inputs
  - 2*x
  - No use in scaling and centering
- Intrinsic feature selection
- Computational simple and quick

Tree Disadvantages

- High Model Variance(sensitive to data)
  - Derives from each subsequent split is dependent on prior splits
- Less than optimal predictive performance
  - Rectangular regions!!!
- Limited number of outcome values
- Selection bias toward predictors with higher number of distinct values
- Tuning parameter, C_p

Tree Variants

- There are many tree variants

- Tweaks
  - change how splits are determined ? How many splits?
  - when to stop growing the tree
  - how the node value is determined

Rules

- As derived from trees often have repeated conditions

```
NumCarbon > 3.777 &
SurfaceArea1 > 0.978 &
SurfaceArea1 > 8.404 &
FP009 <= 0.5 &
FP075 <= 0.5 &
NumRotBonds > 1.498 &
NumRotBonds > 1.701
```

Rules and their conditions live on their own, conditions can be adjusted to help bias-variance trade-off

RPART EXAMPLE

APPENDIX