Recursive Partitioning

Practical Machine Learning (with R)

UC Berkeley

Logistic Methods

Advantages

- …
- …

Disadvantages

- …
- …
- …

Linear Methods: Limitations

Advantages

- Interpretable
- Easy to train

Disadvantages

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

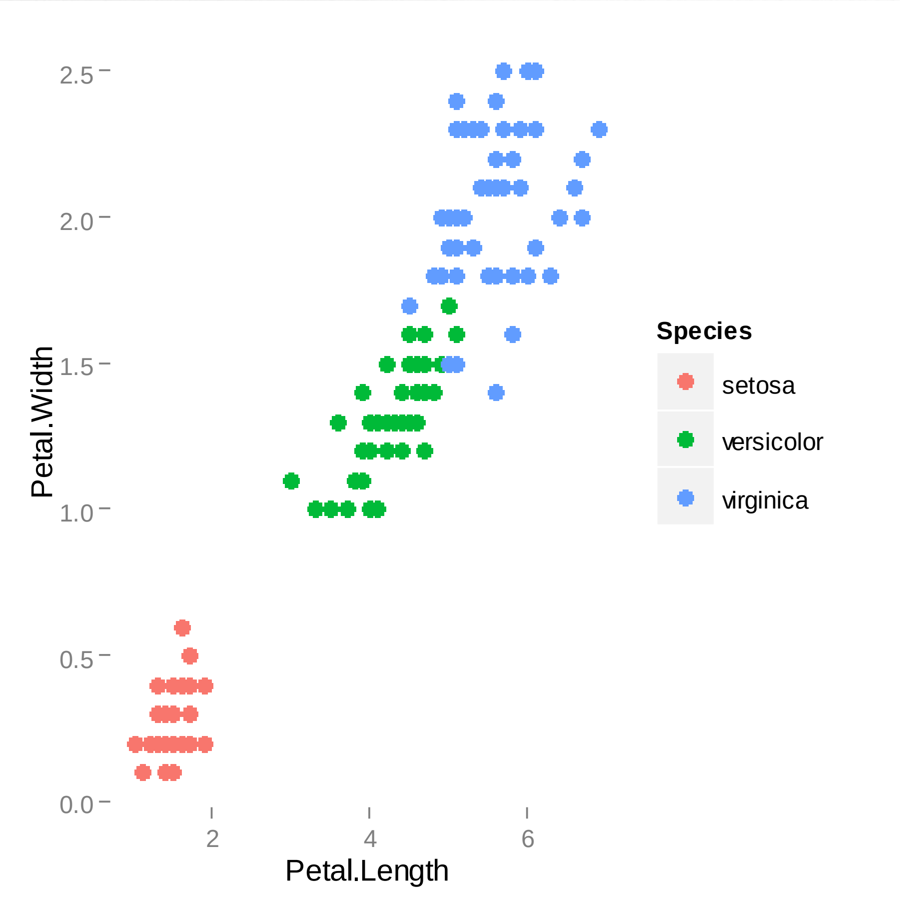
Partition Goal:

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

PROCEDURE

- Find best **univariate plane (aka axis parallel)** 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.
-

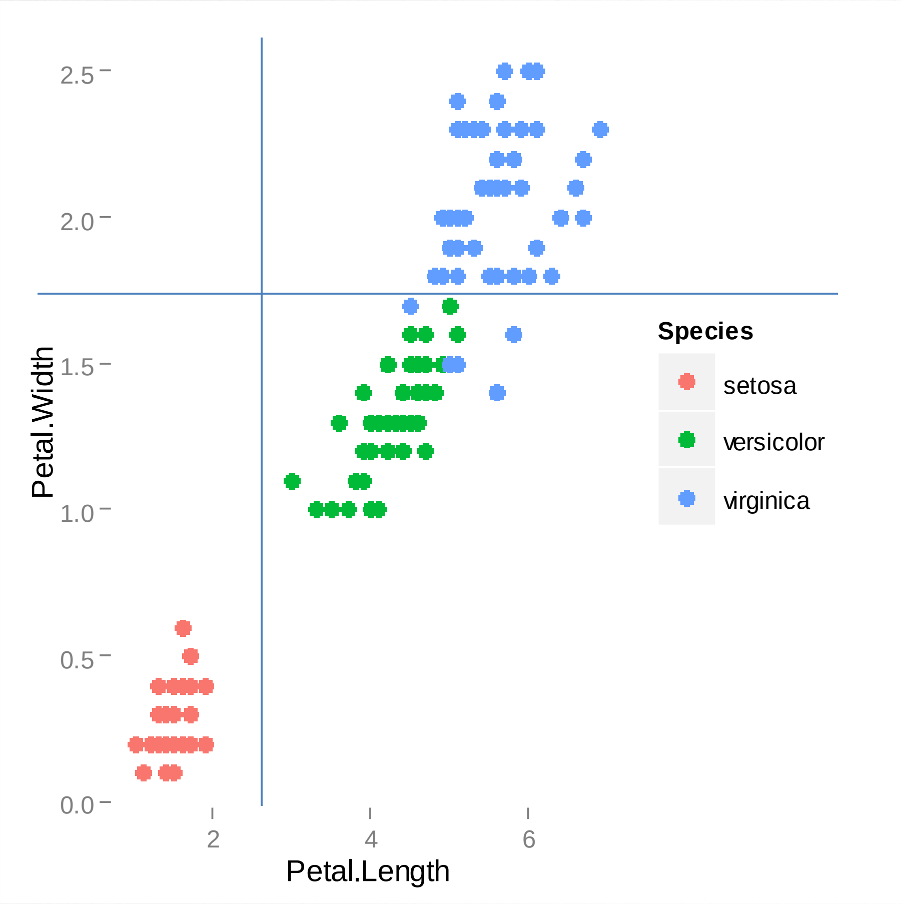
A Simple Example



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

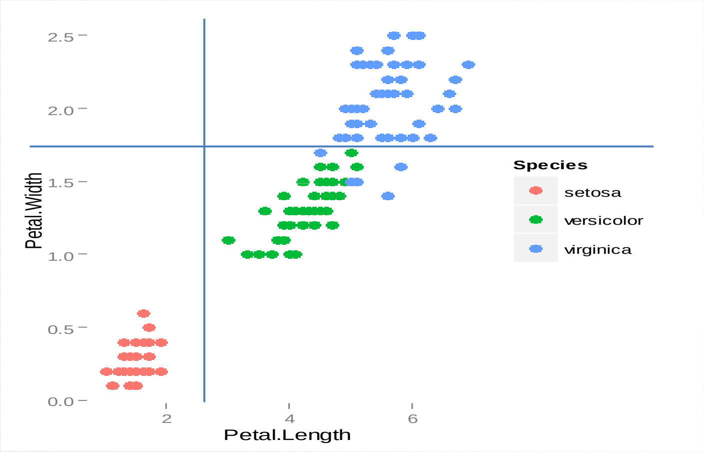


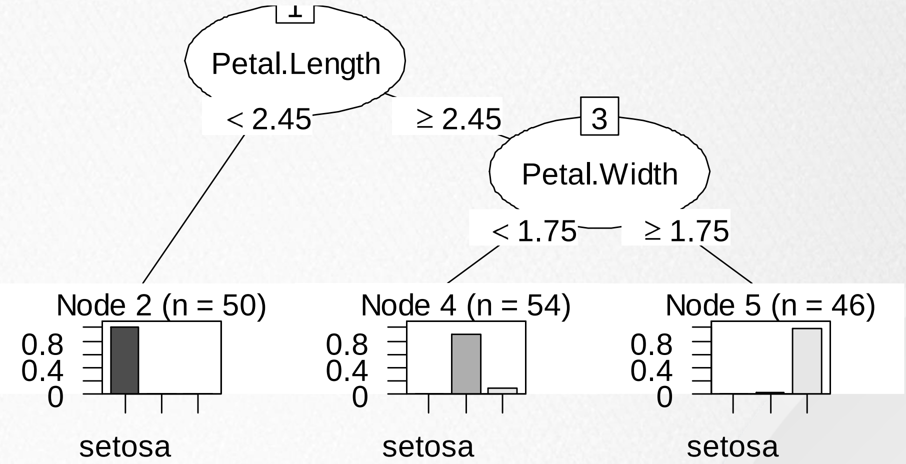
Partitioning Requirements

- **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

- Information Gain
- *Gini* index (Class)

- 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

Entropy (Two-Class Classification)

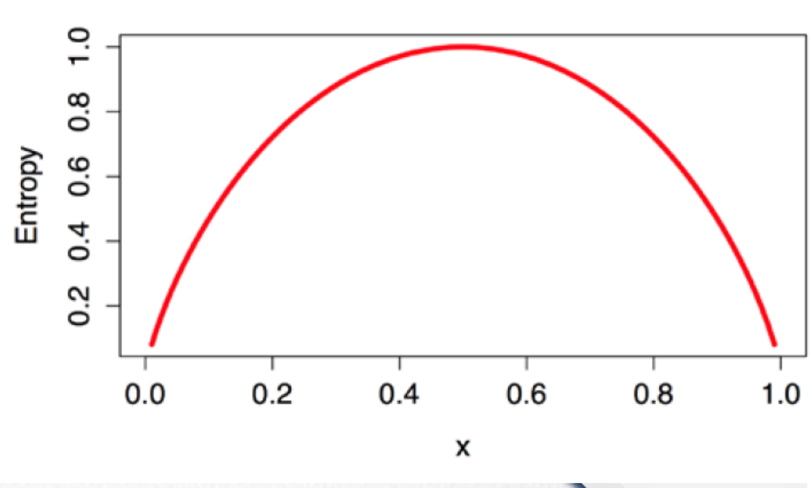
$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

`

$$p_1 + p_2 = 1$$

Entropy for node:

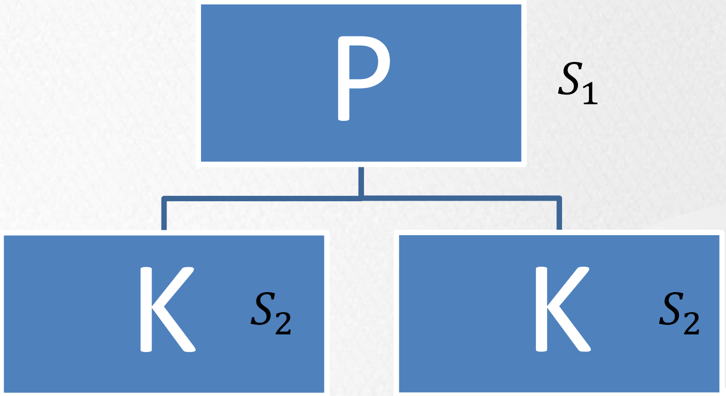
$$-p_1 \log_2(p_1) - p_2 \log_2(p_2)$$



Information Gain / Criteria

Change in Entropy of the System:

$$InfoGain = Entropy(S_1) - Entropy(S_2)$$



Change in system must consider all existing states. Splitting introduces an additional node.

$$Entropy(S) = \sum_{i=1}^{n} \omega_i Entropy(P_i)$$

$\omega_i \coloneqq node\ weight$
$(proportion\ of\ observations\ in\ node)$

Gini Index (Two-Class Classification)

- Measure node purity:

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

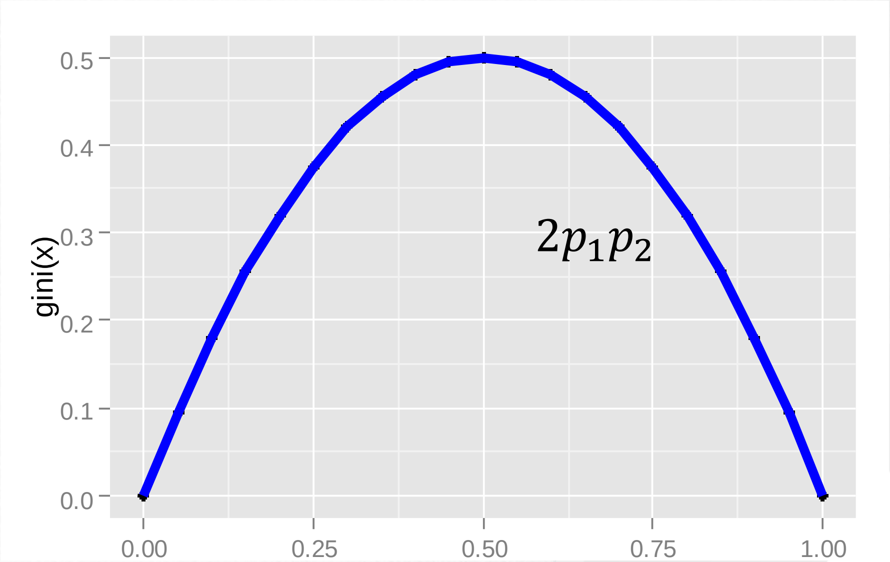
For two class:

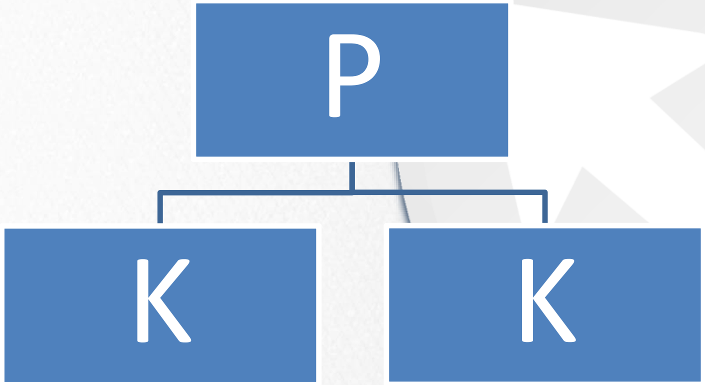
$$p_1+p_2=1$$

$$2p_1p_2$$

Minimize:

$$Gini(S)=\sum_{i=1}^{n}\omega_i 2p_1p_2$$





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

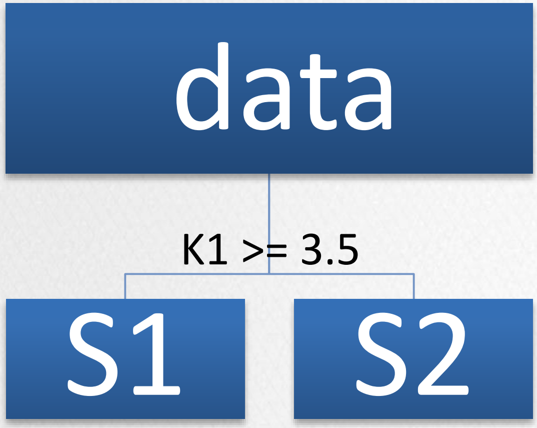
$$argmin_{x=n}(\sum_{i=1..2} err_i)$$

data

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}



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

- Now Adapt to Regression

APPENDIX

Linear Models