# Recursive Partitioning

**Practical Machine Learning (with R)**

UC Berkeley

# Review

Model Goal → Produce a function

Given any point, p, where p is within the *span* of input value(s), make a prediction about the value of a response at point p.

- Span is defined by the variables used in the model.

# 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

# Linear Models

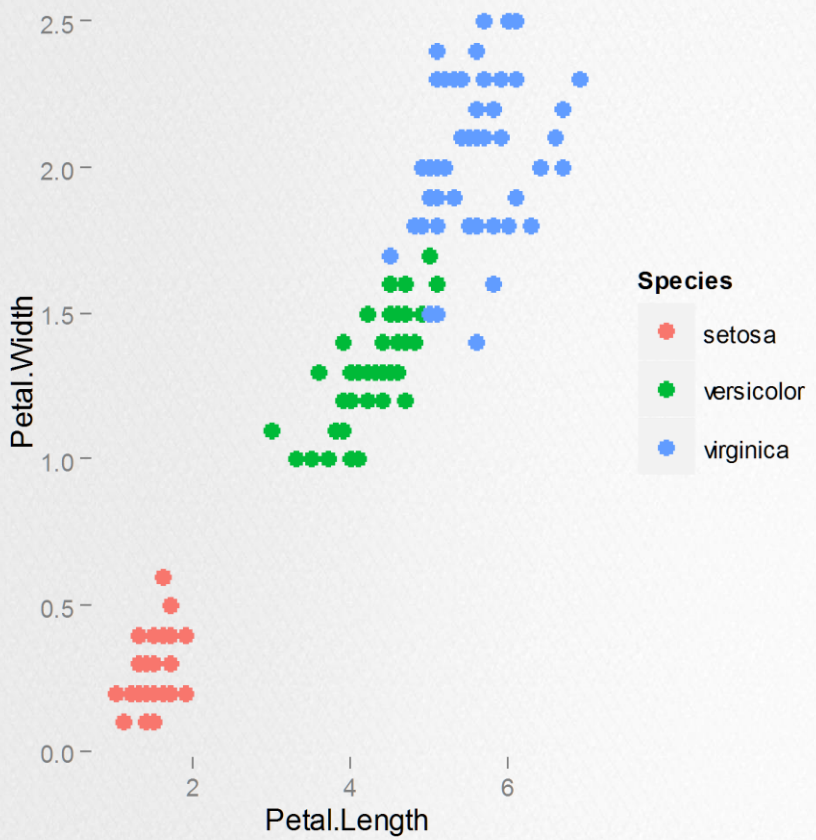
Assumes response is:

- ***linear*** wrt $x$'s
- ***continuous***
  - Obs w/ similar $x$'s →similar response (y)

- Real systems exhibit discontinuities

- Models that allow for discontinuities in response may yield better models.

# Goal




Find sub-regions of input space such that observations within each region are close/similar (*homogeneity*)

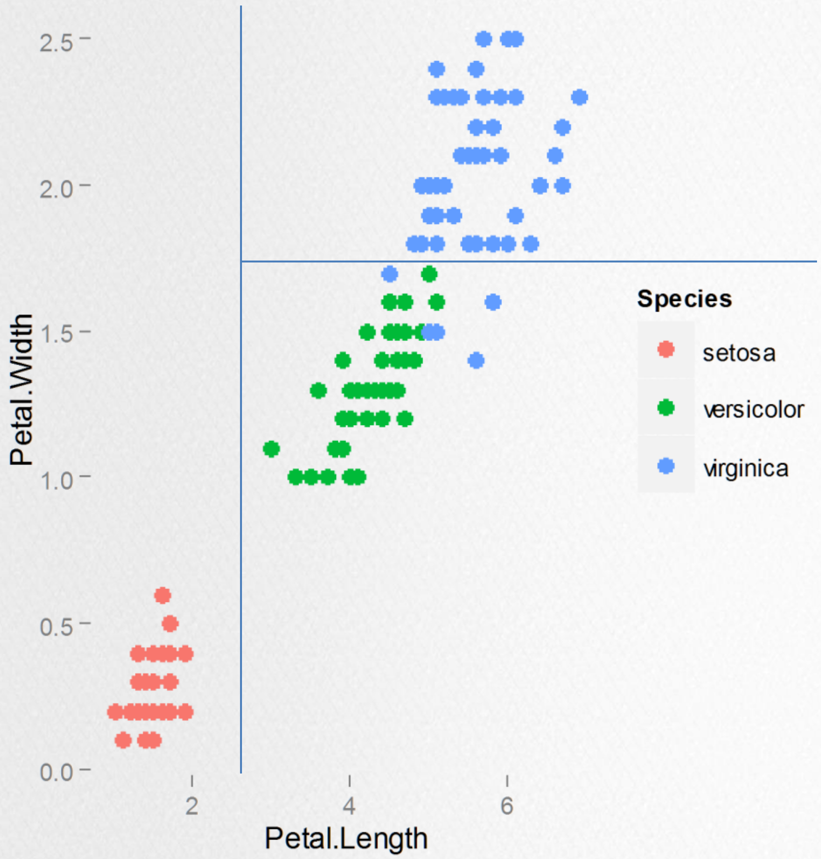
Associate a response $\hat{y}$ within each sub-region (and possibly each point within the sub-region.

There are many algorithms that apply this strategy, they vary by:

- Loss function(s)
- Restricted class of functions
- Search Methodology

# DECISION TREES

# CART / RPART Example



# Procedure

1. Evaluate all possible **univariate** split points in the data.

2. Split data using **best split point** to split the data into two subsets ("node", "leaf").

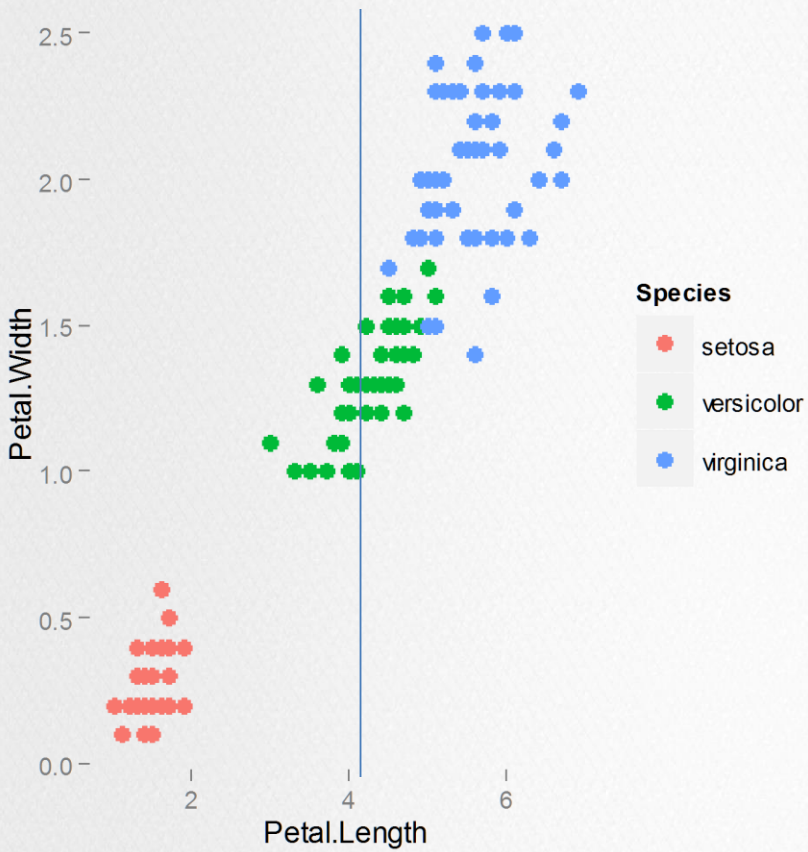
3. For each resulting subset determine the best split point as in step 1.

4. Split only the subset with the best split

5. Repeat 3&4 until stopping condition is met.

# THE BEST SPLIT?

# THE BEST SPLIT




- Each (potential) split produces two sub-regions …
- Are the sub-regions more similar (homogeneous) than the entire region? How much?

# HOW HOMOGENEITY IS QUANTIFIED DEPENDS ON THE ***TYPE OF RESPONSE***

# CONTINUOUS RESPONSE (REGRESSION)

# Continuous Response

- For continuous responses, we can use a standard error ***measure***

$$err = y - \hat{y}$$

- This error must be evaluated for every point … how is $\hat{y}$ determined?

$$\hat{y} = mean(y)\,|\,\mathrm{x} \in s$$

# Continuous Response

- Use standard ***metric***, e.g. **RMSE**, **MAE**, **MAPE**

$$\sum_{i=1}^{S} err(y, \hat{y})$$

→Evaluate over all possible split points

→Split IFF metric is reduced by some threshold amount.

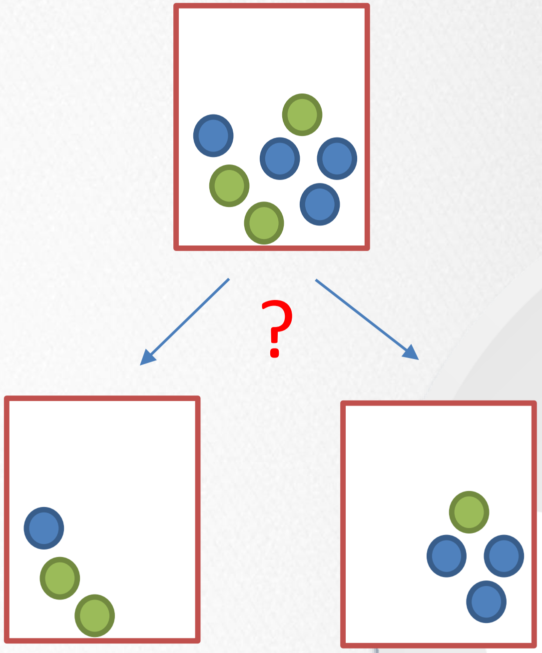
# CLASSIFICATION

# Classification

Still need to answer the questions:

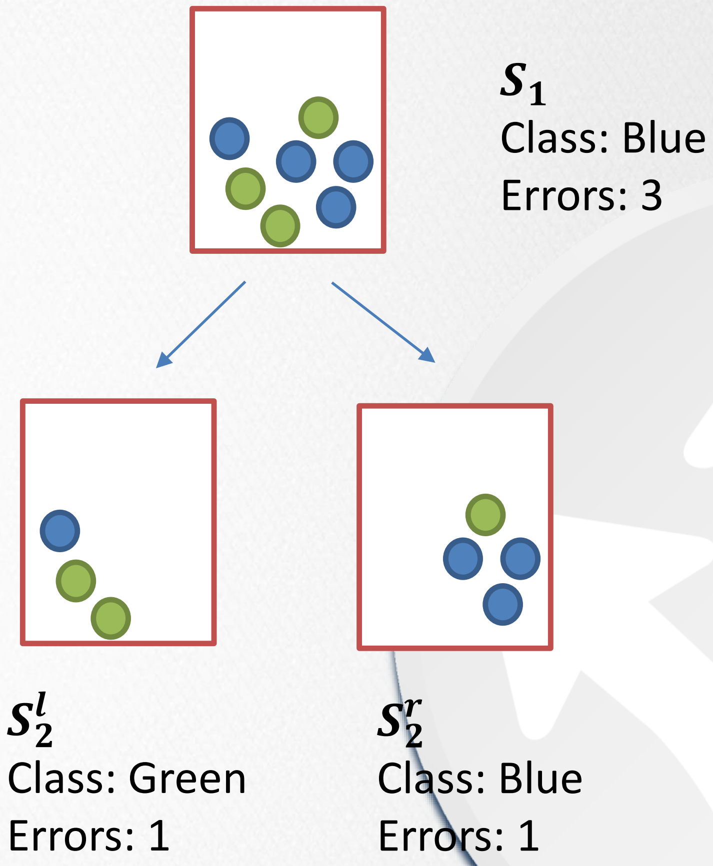
- What is the best split point? /
  What is the value of a potential splits?

- When do we stop? /
  Is this a split that we want to take?



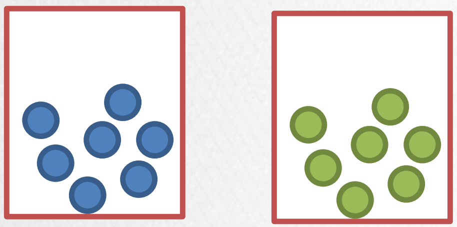
# Misclassification Error

- Works the same as before misclassification.
- $err = \begin{cases} 1 \mid y \neq \hat{y} \\ 0 \mid y = \hat{y} \end{cases}$
- $\hat{y}$ is the majority class for the sub-region
- Compare state $\boldsymbol{S_1}$ to $\boldsymbol{S_2}$

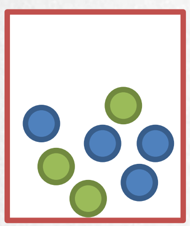


# Entropy

The entropy of a system is determined by its ***homogeneity***:



Low Entropy – complete homogeneity, all responses are the same class



High Entropy – homogeneous each class is equally represented

# Entropy (Binary Classification)

## Entropy

$$-p_i \log_2(p_i)$$

Each class contributes to the entropy:

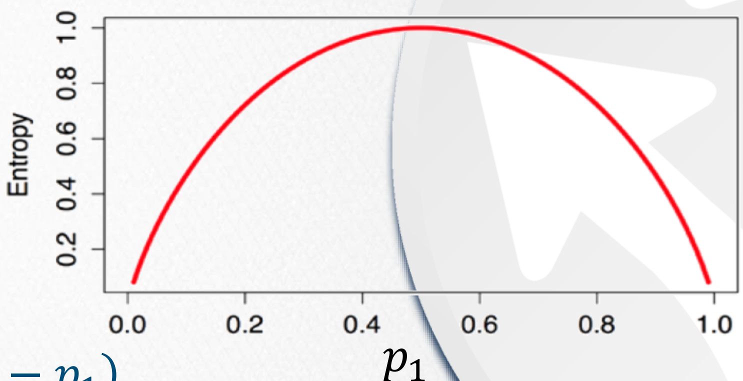
$$Entropy(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$

**For binary classes**

$$p_1 + p_2 = 1$$

$$Entropy(S) = -p_1 \log_2(p_1) - p_2 \log_2(p_2)$$

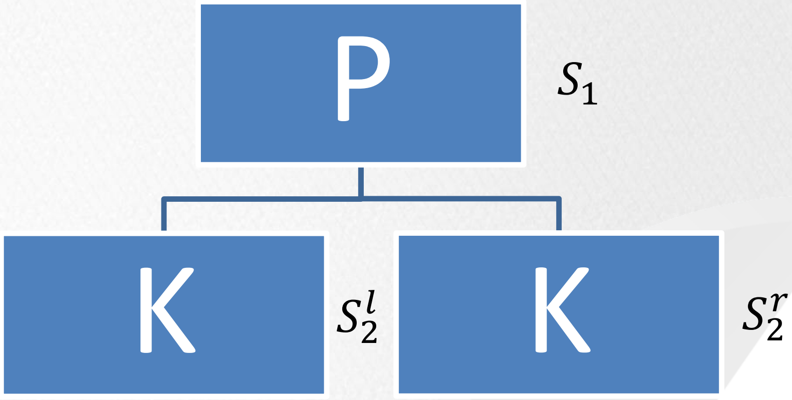
$$Entropy(S) = -p_1 \log_2(p_1) - (1 - p_1) \log_2(1 - p_1)$$



# 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:

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

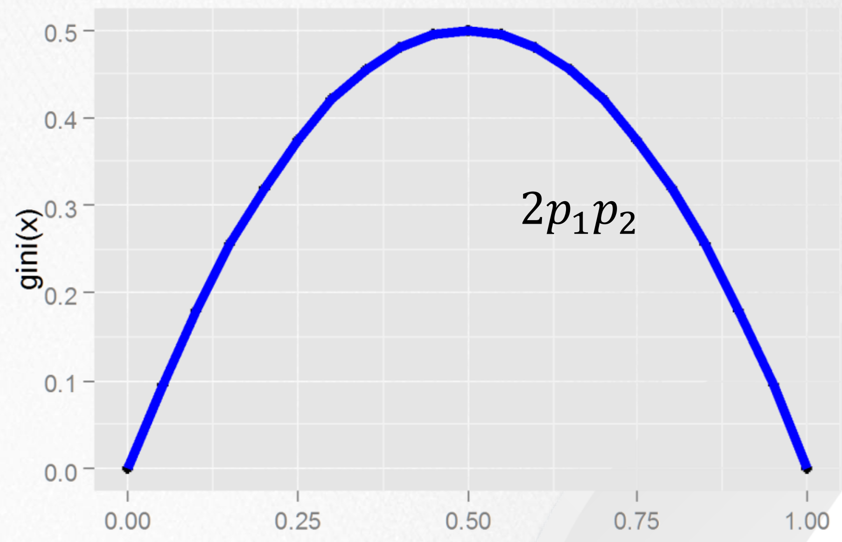
For two class:

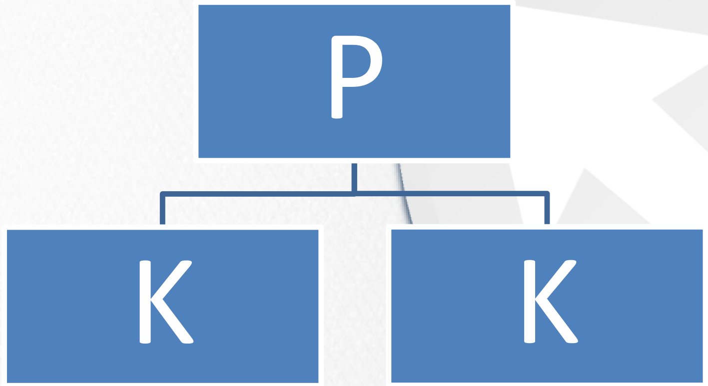
$$p_1+p_2=1$$

$$2p_1p_2$$

Minimize:

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



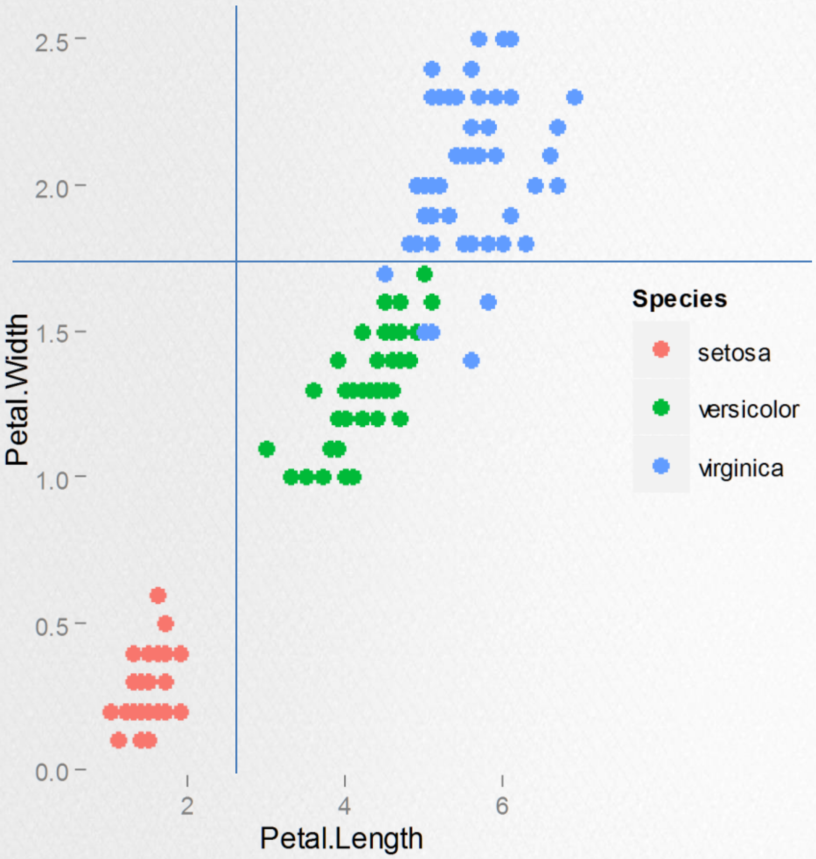



# Stopping Criteria

There are several stopping criteria that can be used:

- Minimum observations in sub-region
- Threshold no split decreases improves the model more than a threshold amount

# A Simple Example



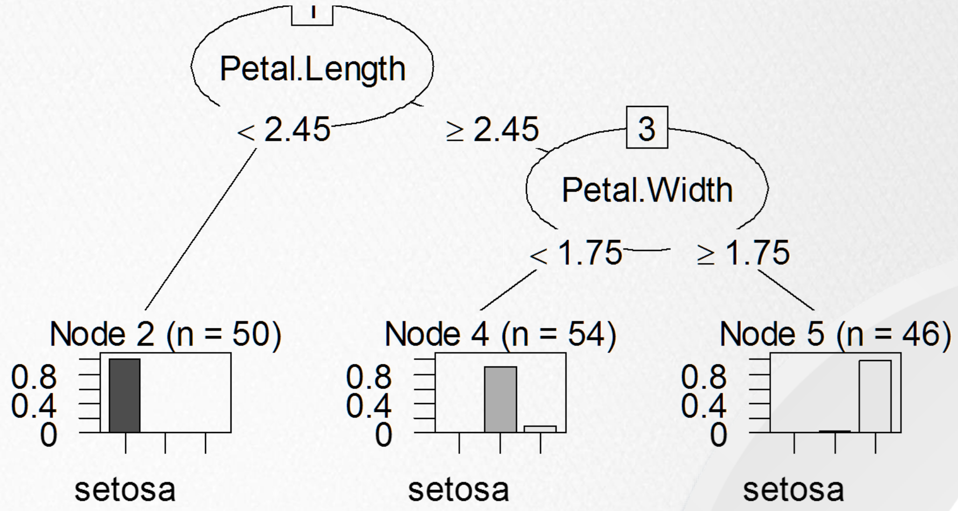

## Partitioning Requirements

- **Loss/Error Methods**
  - Reg.: SSE, RMSE,MAE, etc.
  - Class.: entropy, gini, IG, Misclassification rate
- **Restricted Class of Functions**
  - First Order Propositional Logic (for partitions)
  - Aggregation (for outcomes)
- **Search Methods**
  - Recursion and Exhaustive
  - Stopping Criteria

# NOTES

# SPLITTING BY PLANS IS THE SAME AS A TREE

## Splitting by planes is the same as a tree


## Partitions define a rule*

- Rules can be associated with outcomes → aggregation method

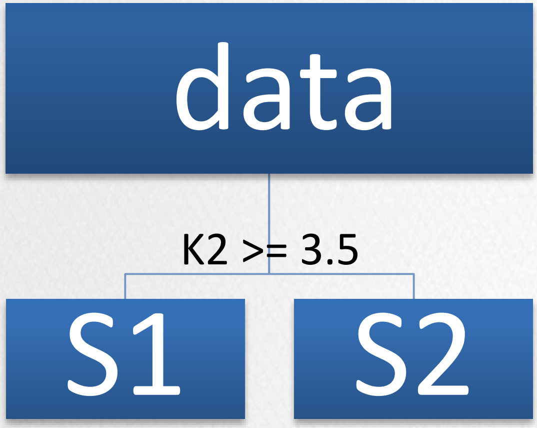
## Trees always partition "all of of space"

# 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.

# Treatment of Categorical Variables

- Grouped Categories
  - Value treated as related

- Independent Categories
  - Values Treated as Independent

# 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 II

- 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

# APPENDIX