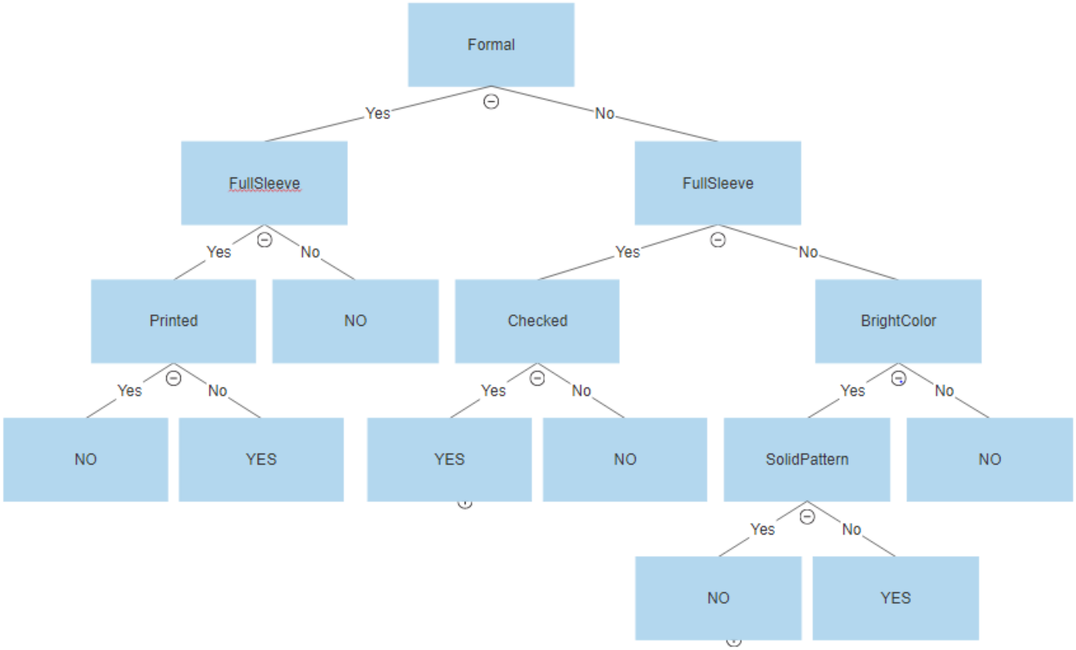
Overfitting, Bias and Variance

Sudeshna Sarkar

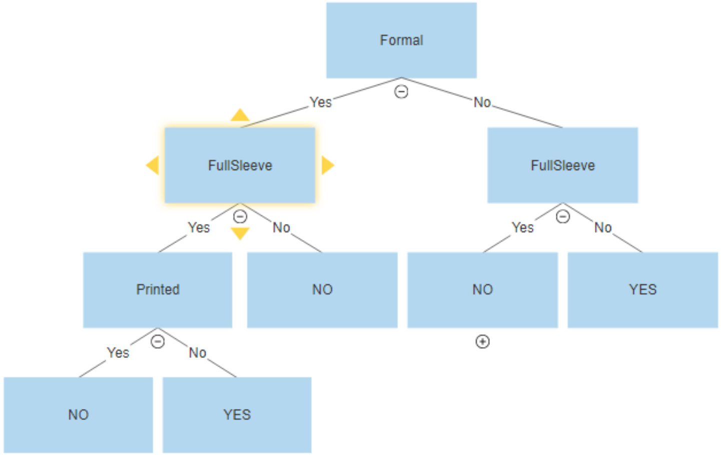
Centre of Excellence in Artificial Intelligence

Indian Institute of Technology Kharagpur

Which Decision Tree?



Training Error = 0.05
Test Error = 0.2



Training Error = 0.1
Test Error = 0.15

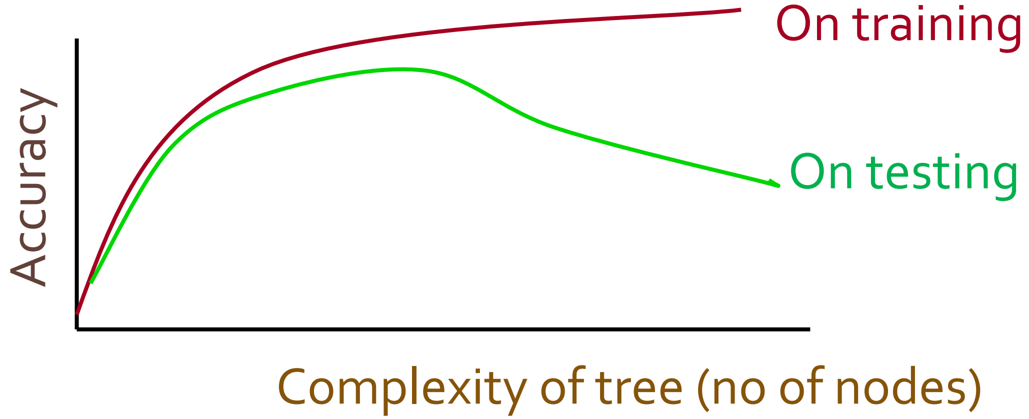
Overfitting

Overfitting :

- Fit the training data too well
- But fail to generalize to new examples

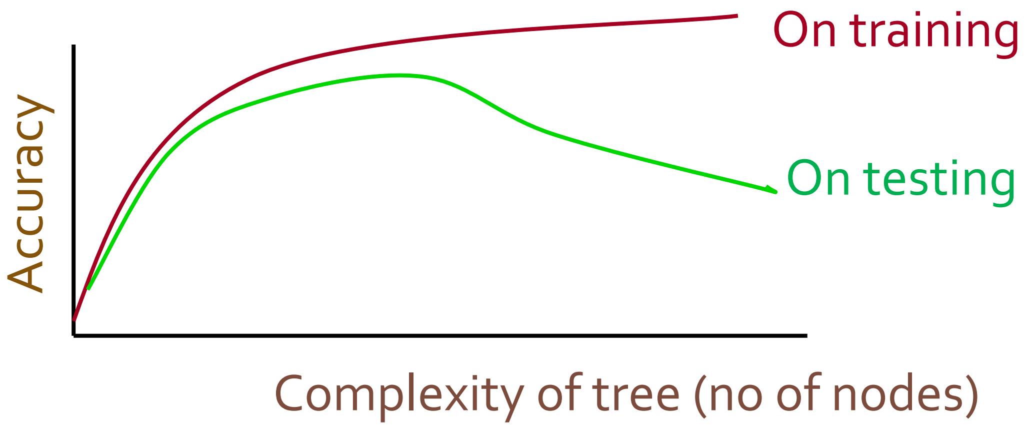
Causes

- Noise
- Irrelevant Features
- Insufficient Data

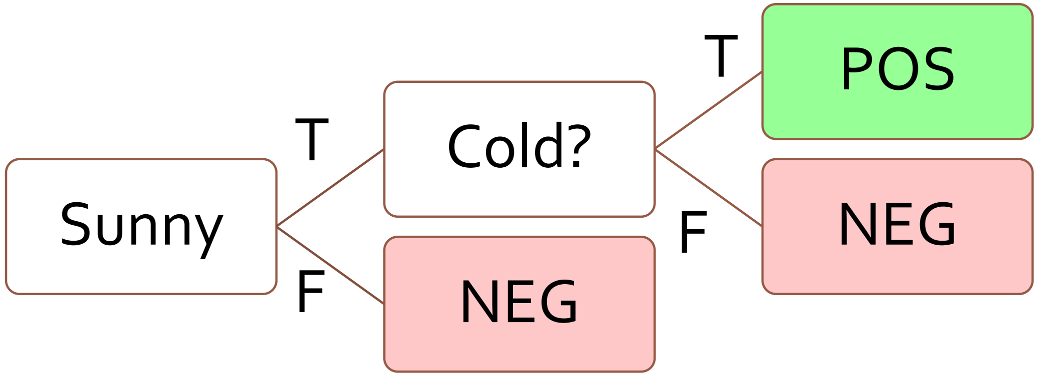


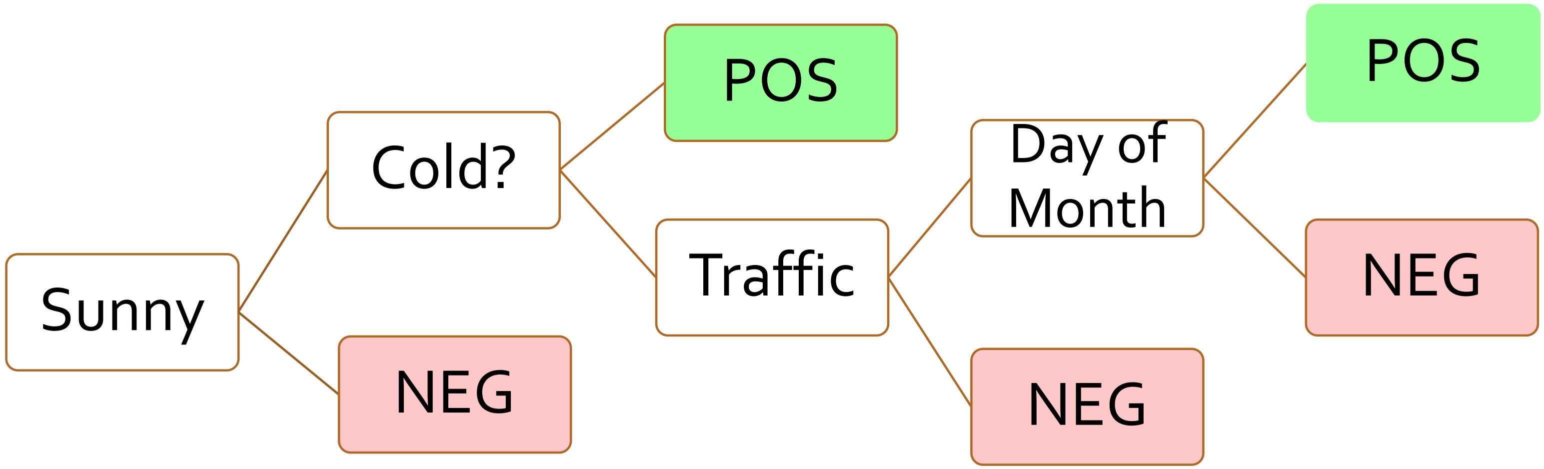
Overfitting

A hypothesis h is said to overfit the training data if there is another hypothesis h' such that h has smaller error than h' on the training data but h has larger error on the test data than h' .



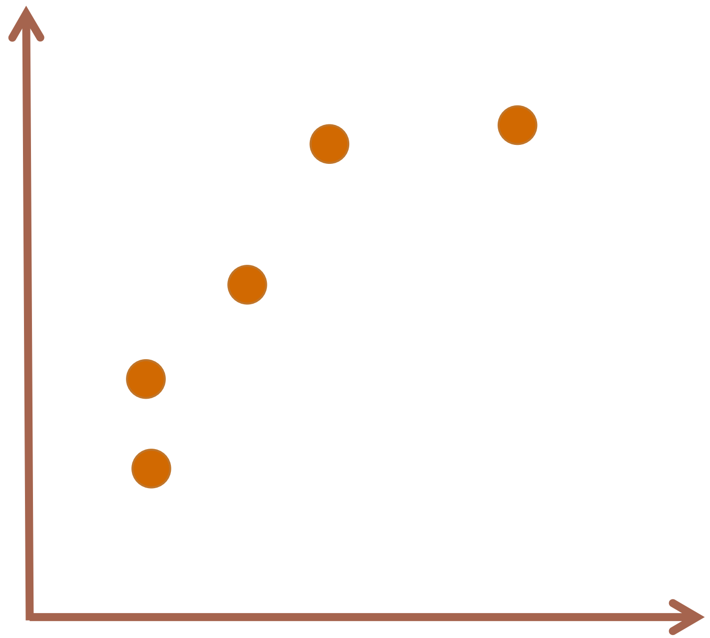
Overfitting with noisy data



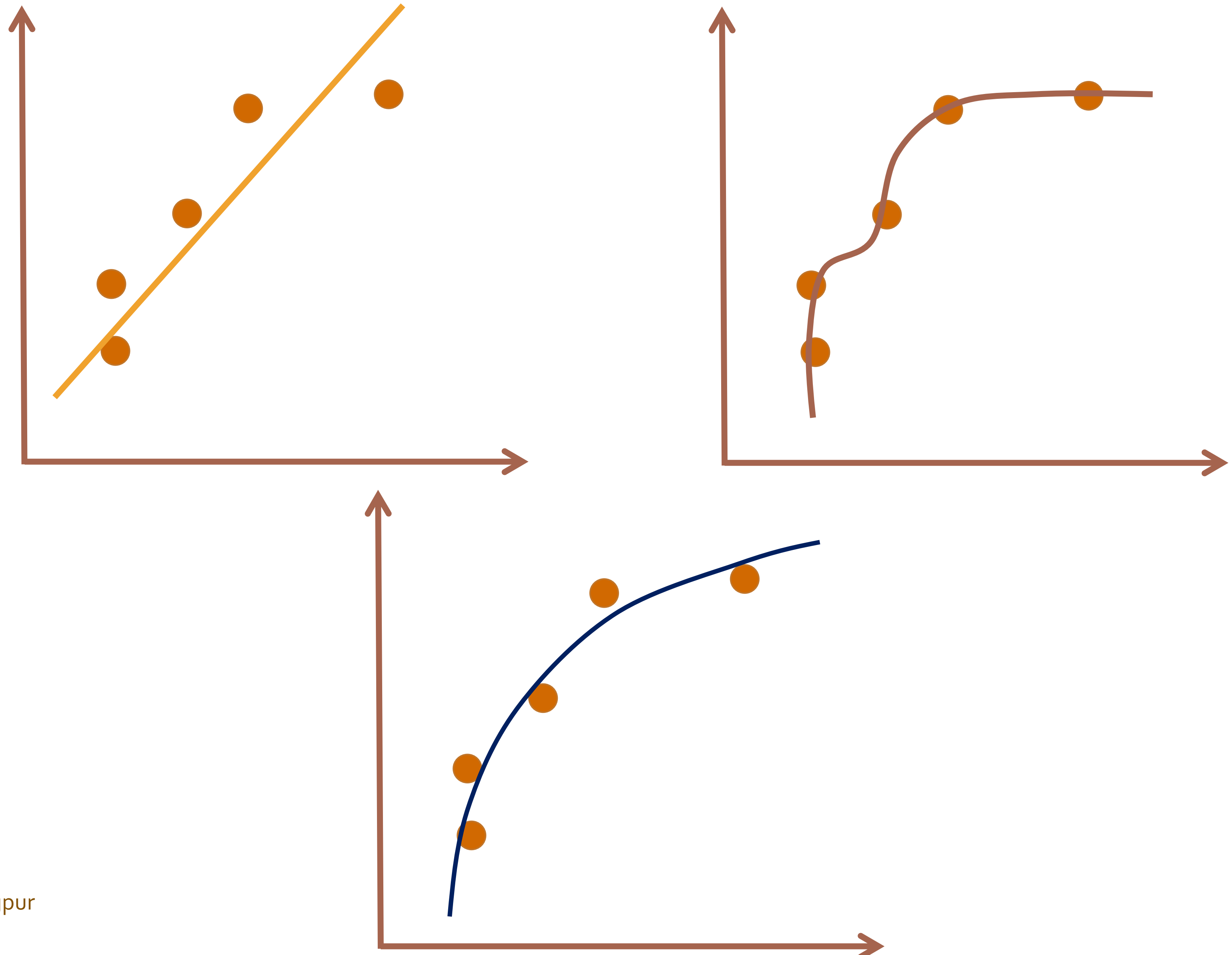


Overfitting results in decision trees that are more complex than necessary

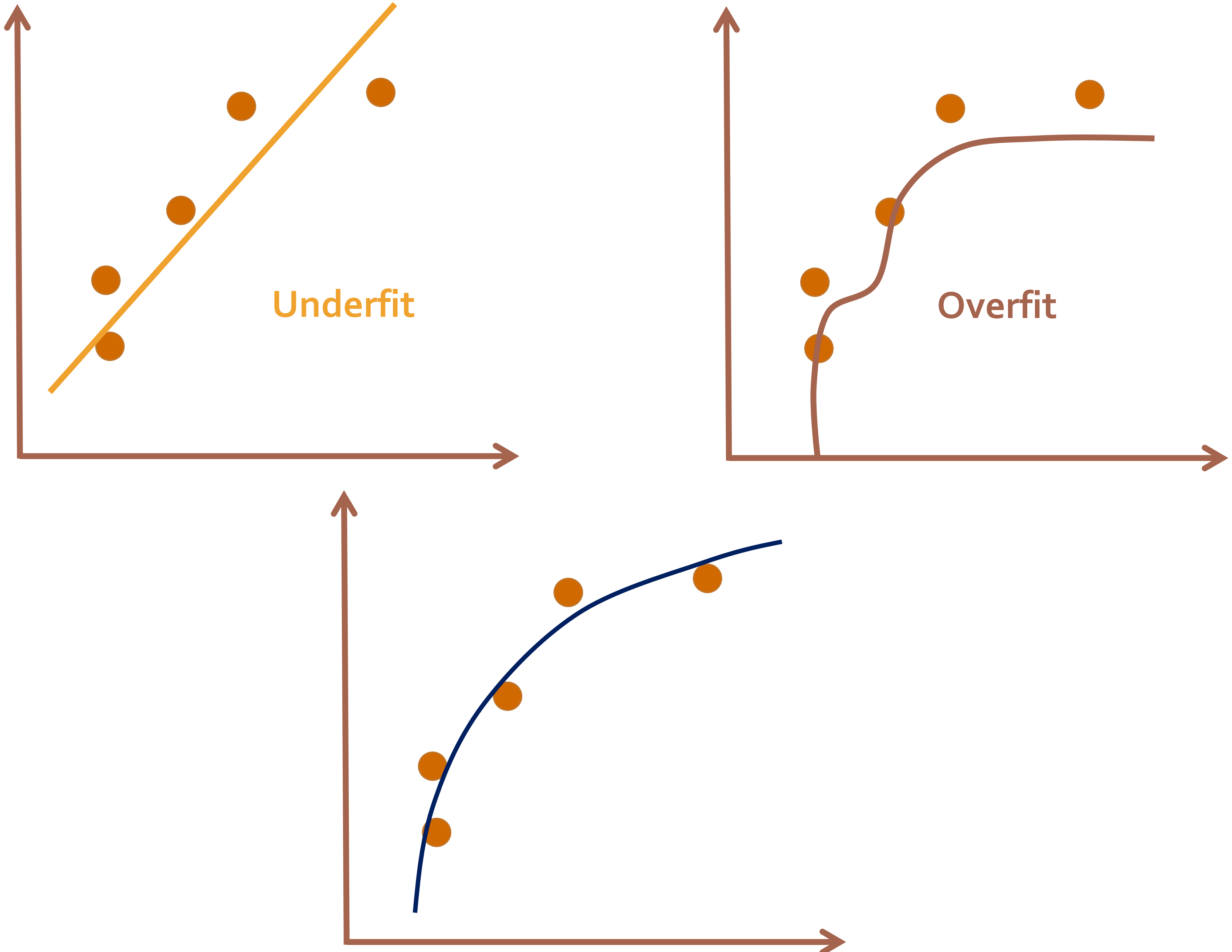
Regression



Regression



Regression



Regularization

- In a linear regression model overfitting is characterized by large weights
- Penalize large weights in Linear Regression
  - L2-Regularization or Ridge Regression
  - L1-Regularization

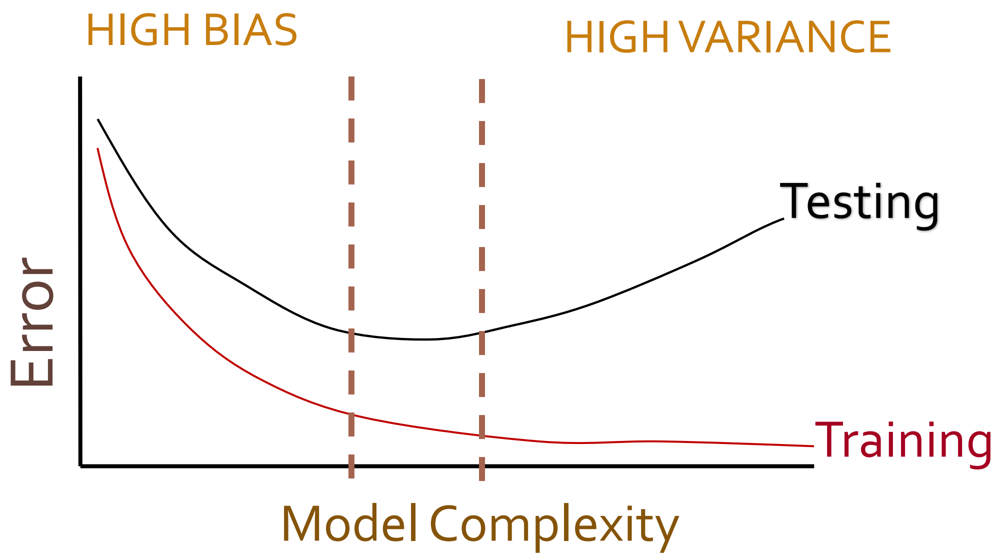
Overfitting vs Underfitting

Underfitting

- Not able to capture the concept
  - Features don't capture concept
  - Model is not powerful.

Overfitting

- Fitting the data too well



Bias

BIAS

- Error caused because the model can not represent the concept
- Bias is the expected difference between the model prediction and the true y's.
- Higher Bias:
    - Decision tree of lower depth
    - Linear functions
    - Important features missing

VARIANCE

- Error caused because the learned model reacts to small changes (noise) in the training data
- High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs
- Higher Variance
    - Decision tree with large no of nodes
    - High degree polynomials
    - Many features

Bias and Variance

BIAS

- if we train models $f_D(X)$ on many training sets D, bias is the expected difference between their predictions and the true y's.

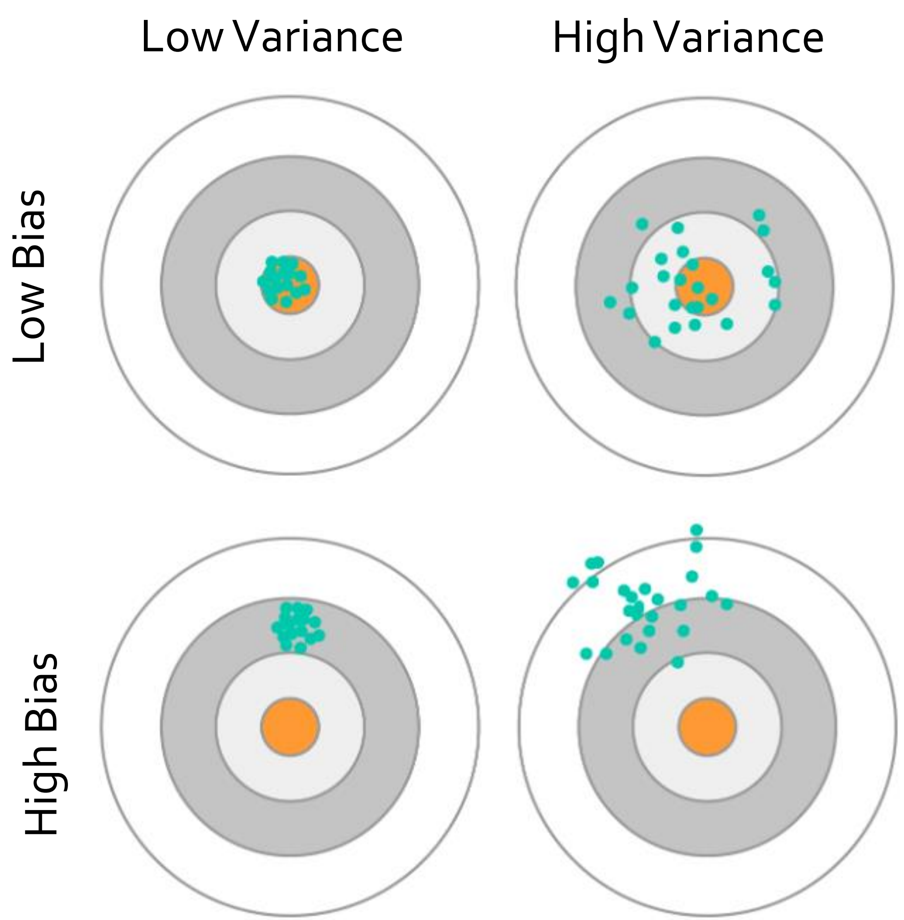
$$Bias = \mathrm{E}[f_D(X) - y]$$

VARIANCE

- if we train models $f_D(X)$ on many training sets D, variance is the variance of the estimates:

$$Variance = \mathrm{E}\left[\left(f_D(X) - \bar{f}(X)\right)^2\right]$$

Bias and Variance

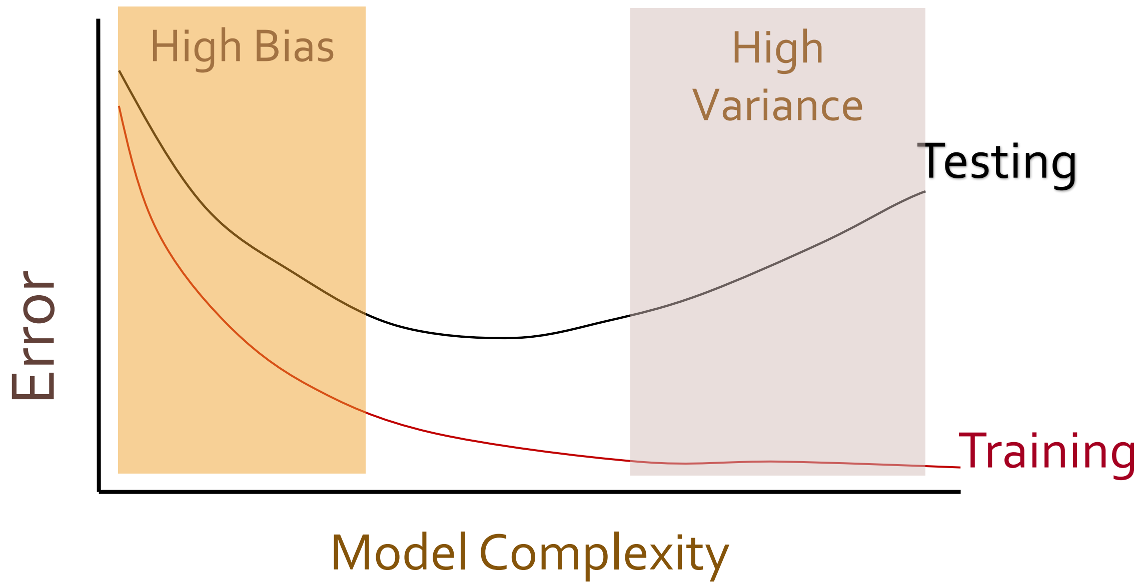


Bias and Variance Tradeoff

There is usually a bias-variance tradeoff caused by model complexity.

Complex models often have lower bias, but higher variance.

Simple models often have higher bias, but lower variance.

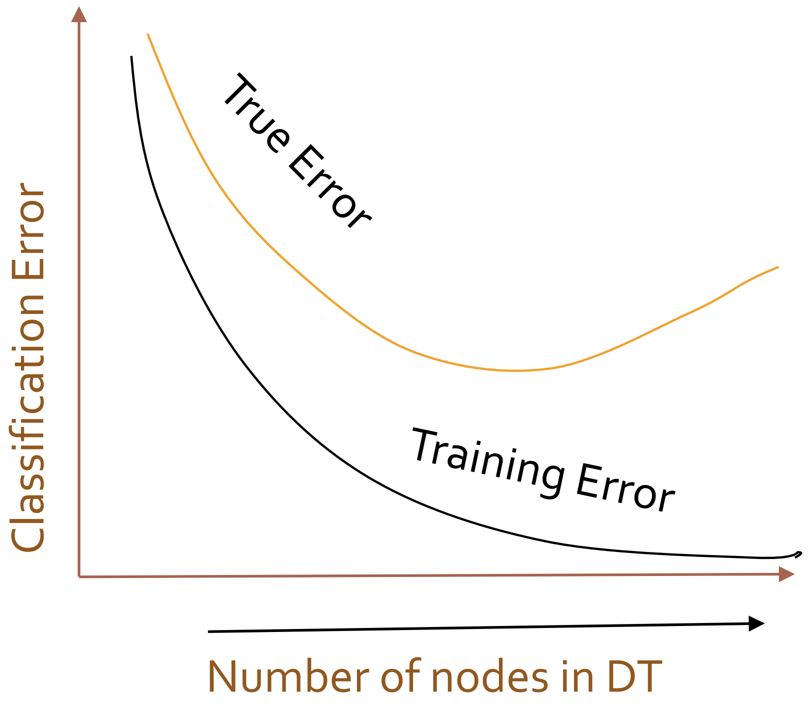


Avoid Overfitting

- How can we avoid overfitting a decision tree?
  - Prepruning: Stop growing when data split not statistically significant
  - Postpruning: Grow full tree then remove nodes

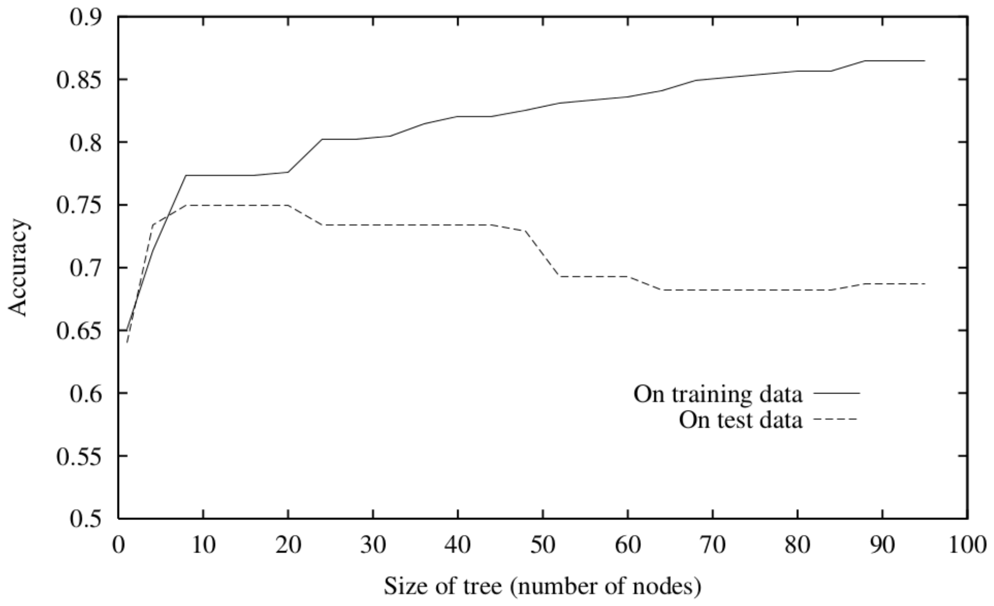
Pre-Pruning (Early Stopping)

- Typical stopping conditions for a node:
  - All instances belong to the same class
  - All the attribute values are the same
- Early Stopping:
  - Stop the learning algorithm before tree becomes too complex



Pre-Pruning (Early Stopping)

- Typical stopping conditions for a node:
  - All instances belong to the same class
  - All the attribute values are the same
- Early Stopping:
  - Stop the learning algorithm before tree becomes too complex



Pre-Pruning (Early Stopping)

- Stopping conditions:
  - Do not split a node which contains too few instances
  - Stop if expanding the current node does not improve impurity measures significantly (e.g., Gini or information gain)
  - Limit tree depth

Reduced-error Pruning

Partition data into train set and validation set

- Build a tree using the train set.
- Until accuracy on validation set decreases, do:

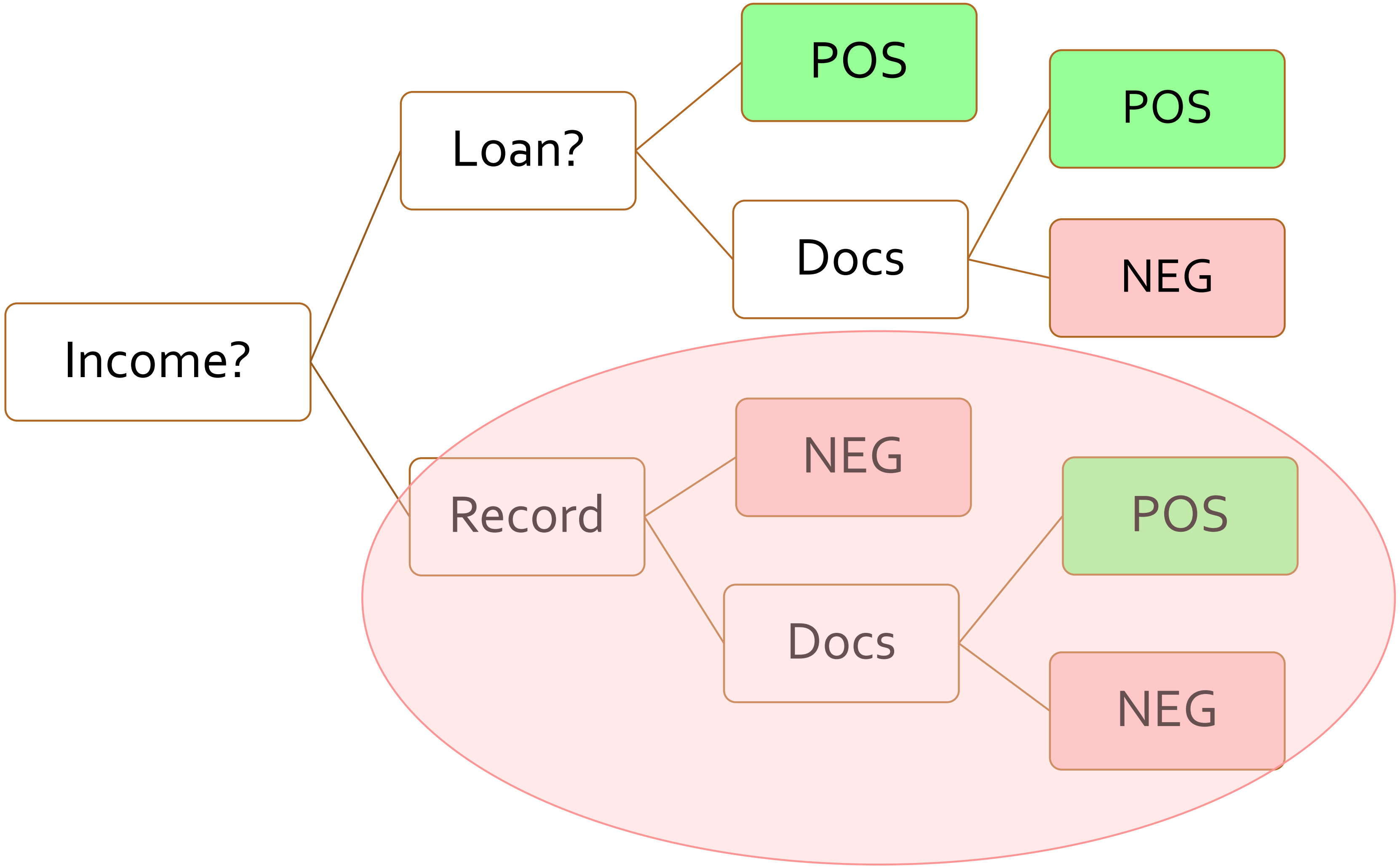
  For each non-leaf node in the tree

  Temporarily prune the tree below; replace it by majority vote

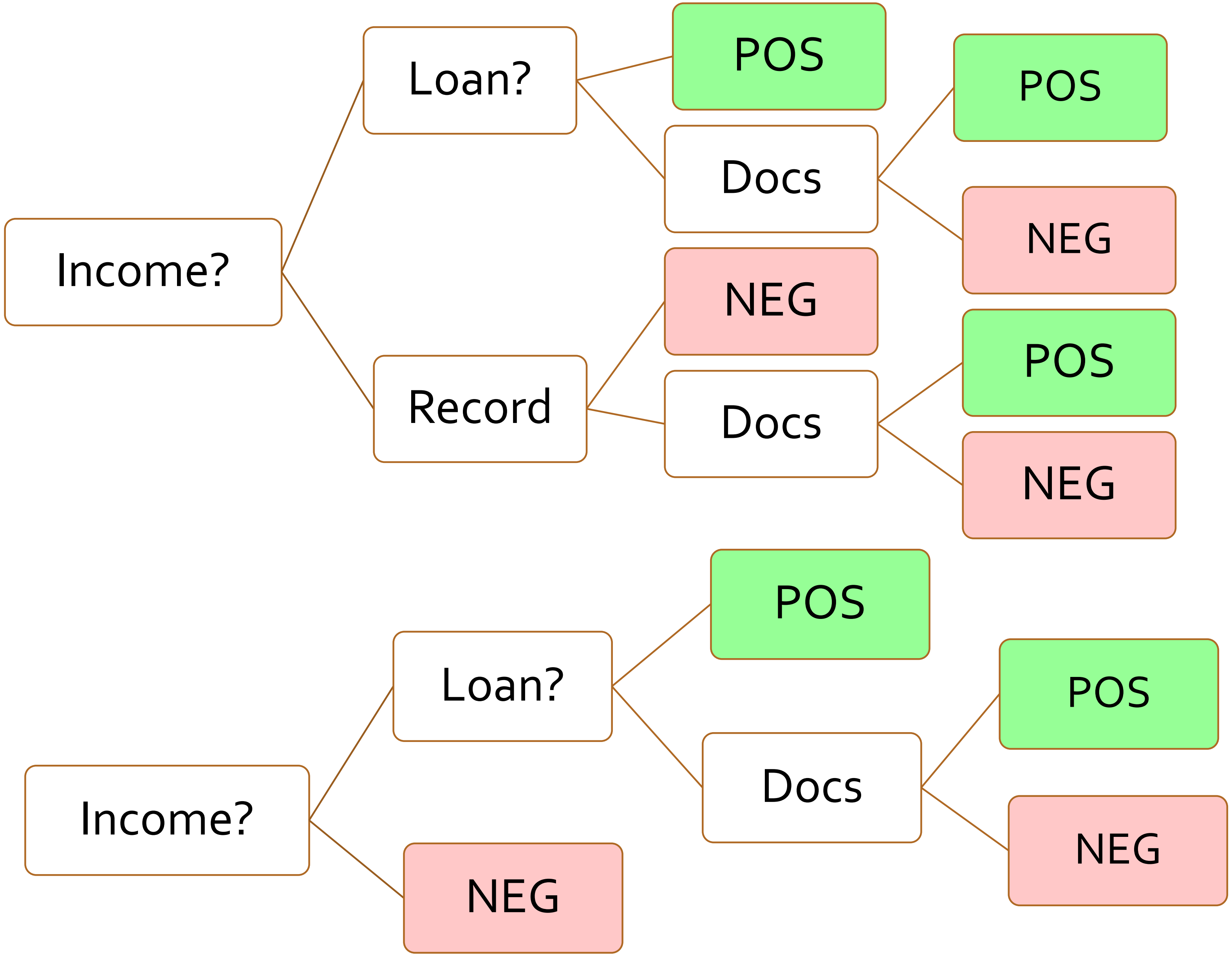
  Test the accuracy of the hypothesis on the validation set

  Permanently prune the node with the greatest increase in accuracy on the validation test.

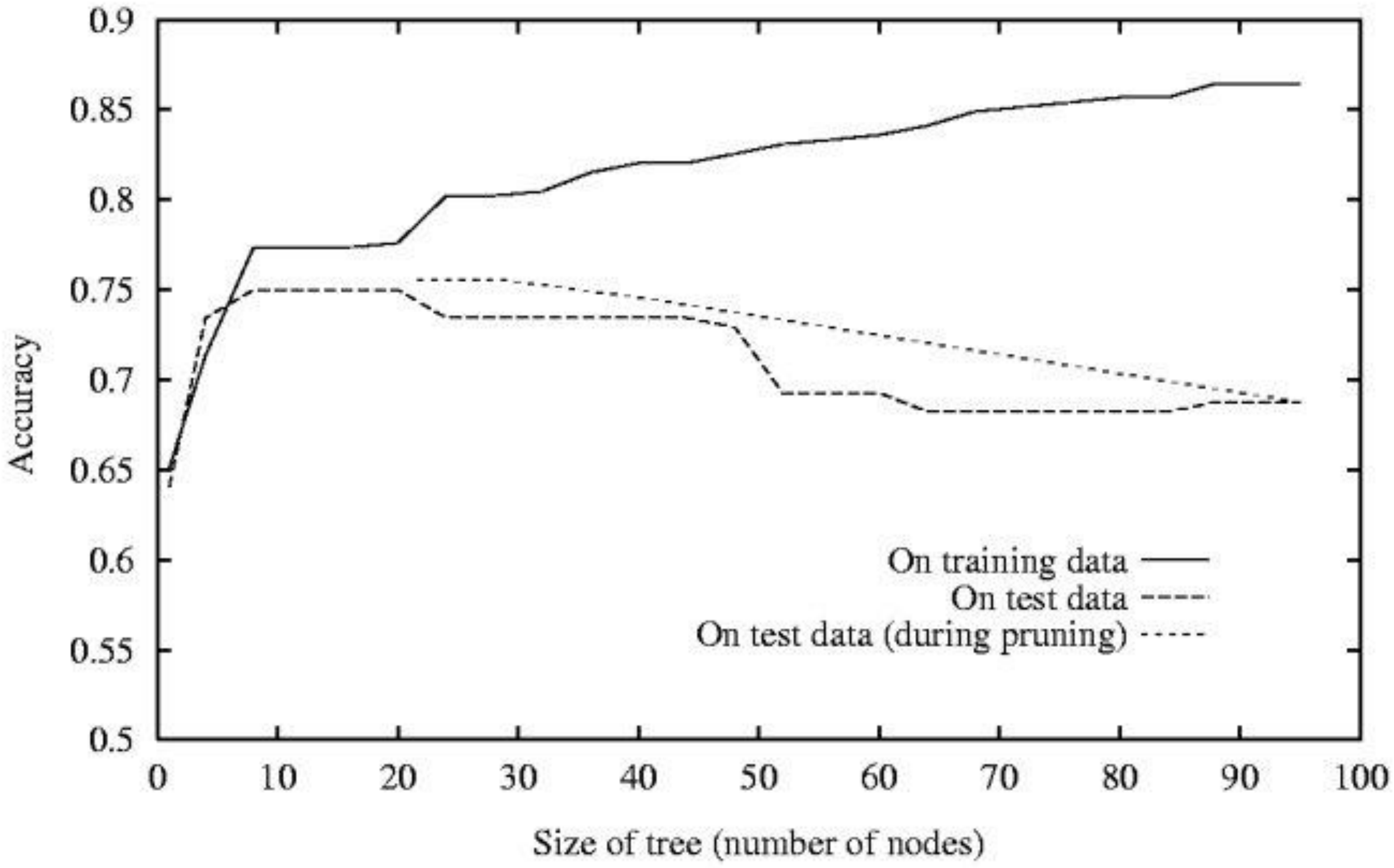
Tree Pruning



Tree Pruning



Reduced Error Pruning



Pruning

- Methods for evaluating subtrees to prune:

1. Cross-validation
2. Minimum description length (MDL):

Minimize: size(tree) + size(misclassifications(tree))

Trade-Offs

- There is a trade-off between these factors:
  - Complexity of Model $c(H)$
  - Training set size, *m*,
  - Generalization error, *E* on new data

1. As *m increases*, *E* decreases
2. As *c* (H) *increases*, first *E decreases* and then *E increases*
3. As *c* (H) *increases*, the training error *decreases* for some time and then stays constant (frequently at 0)

As m increases, E decreases