Overfitting & Model Selection

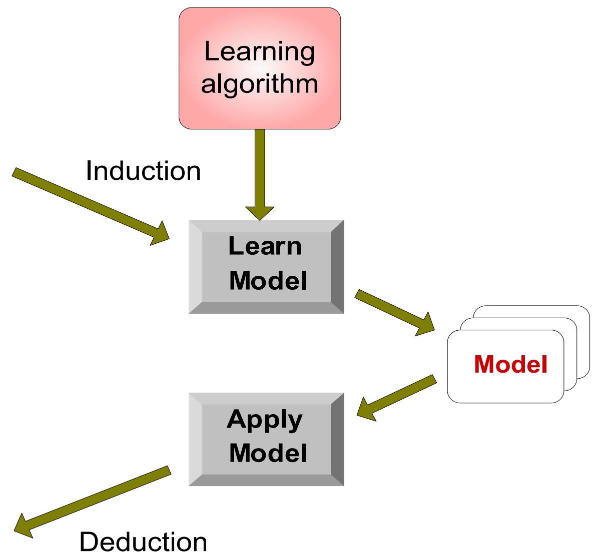
Data Science & Analytics
University of Missouri

Classification Errors

- **Training errors:** **Errors committed on the training set**
- **Test errors:** **Errors committed on the test set**
- **Generalization errors:** **Expected error of a model over random selection of records from same distribution**

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

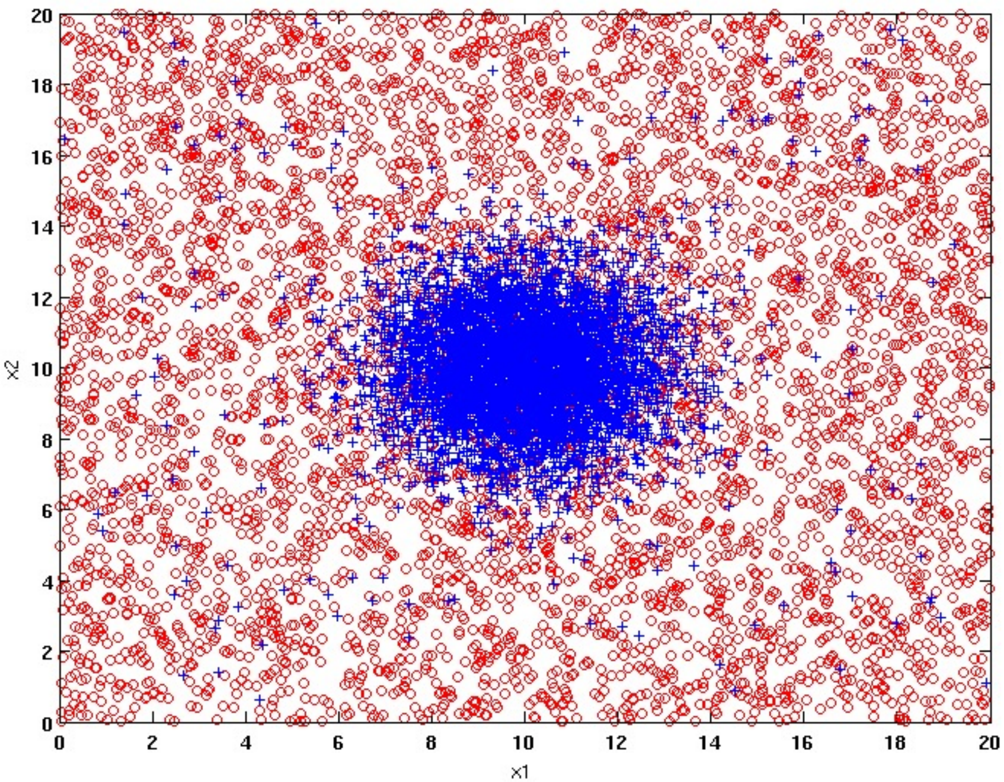


Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

source: Tan et. Introduction to Data Mining

Example Data Set



Two class problem with two features:

+ : 5400 instances

- **5000 instances generated from a Gaussian centered at (10,10)**
- **400 noisy instances added**

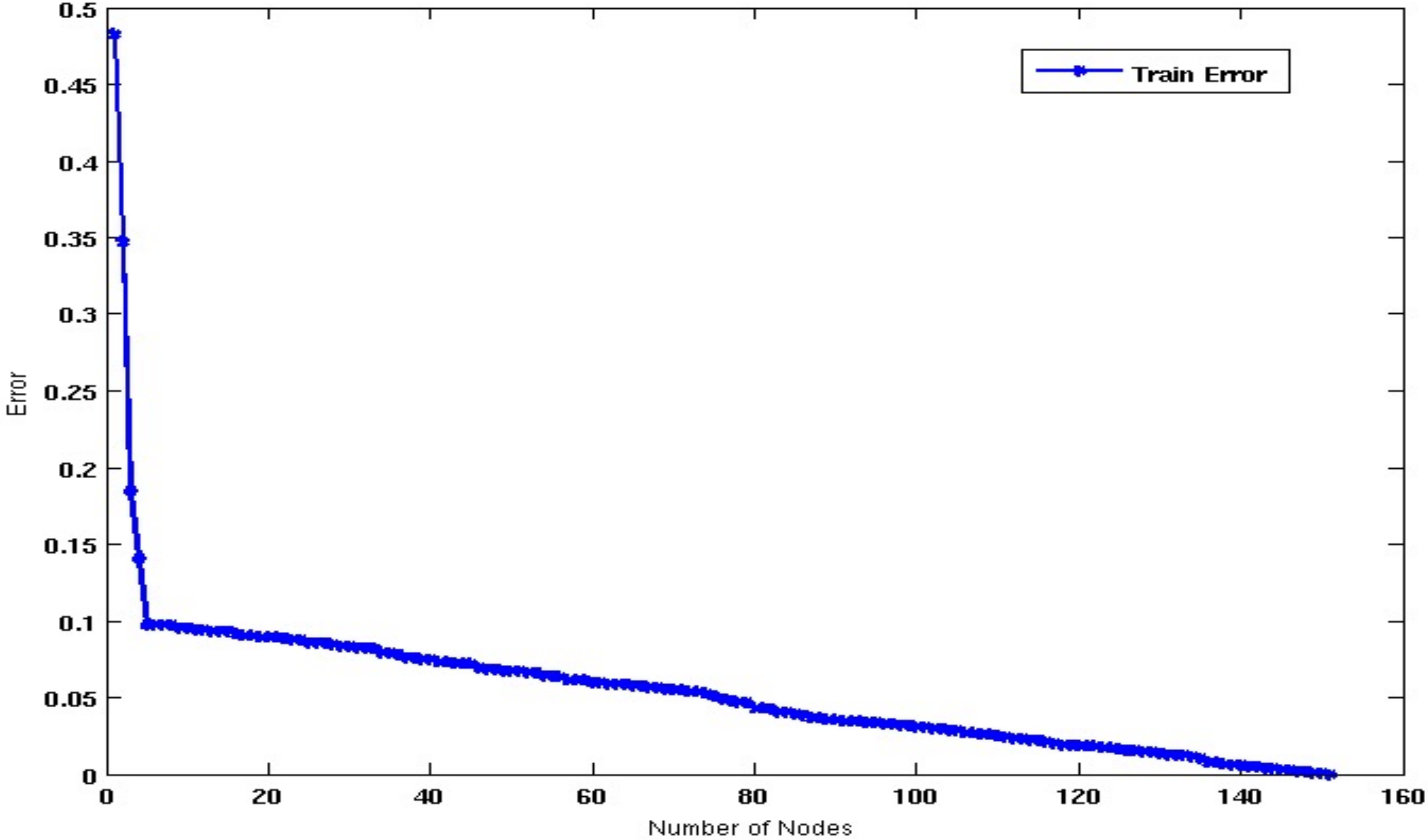
o : 5400 instances

- **Generated from a uniform distribution**

10 % of the data used for training and 90% of the data used for testing

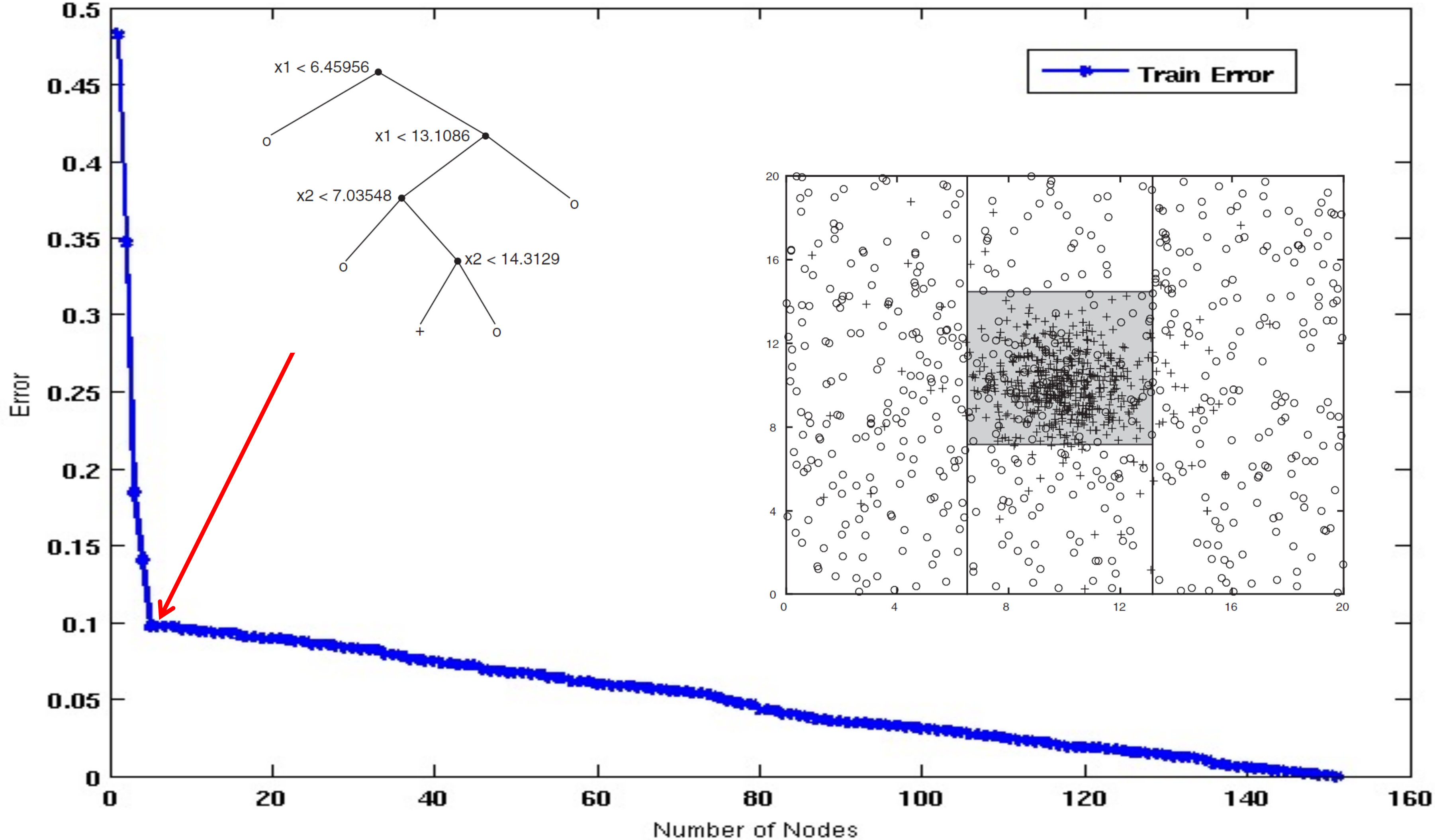
source: Tan et. Introduction to Data Mining

Increasing number of nodes in Decision Trees



source: Tan et. Introduction to Data Mining

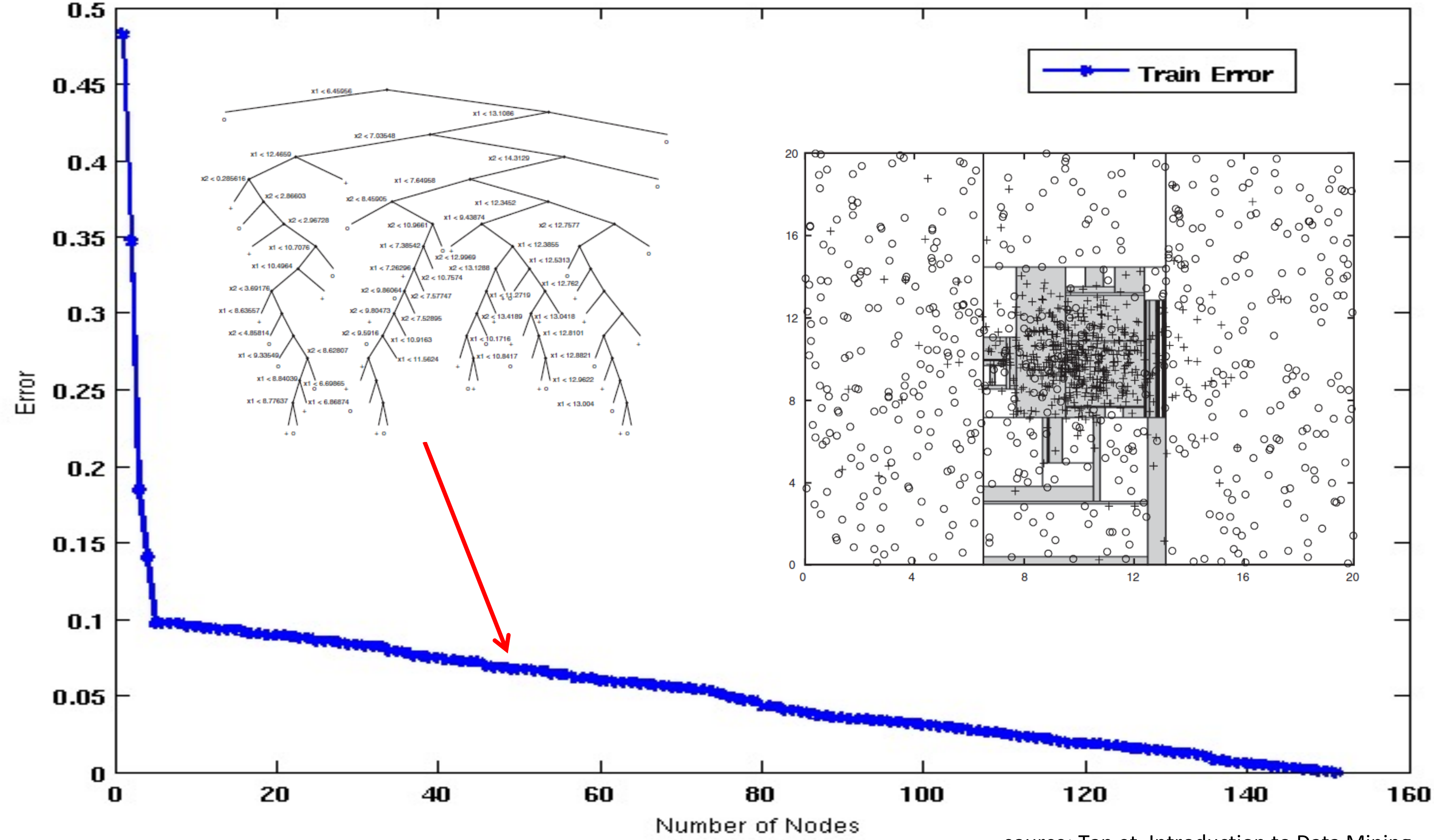
Decision Tree with 4 nodes



5

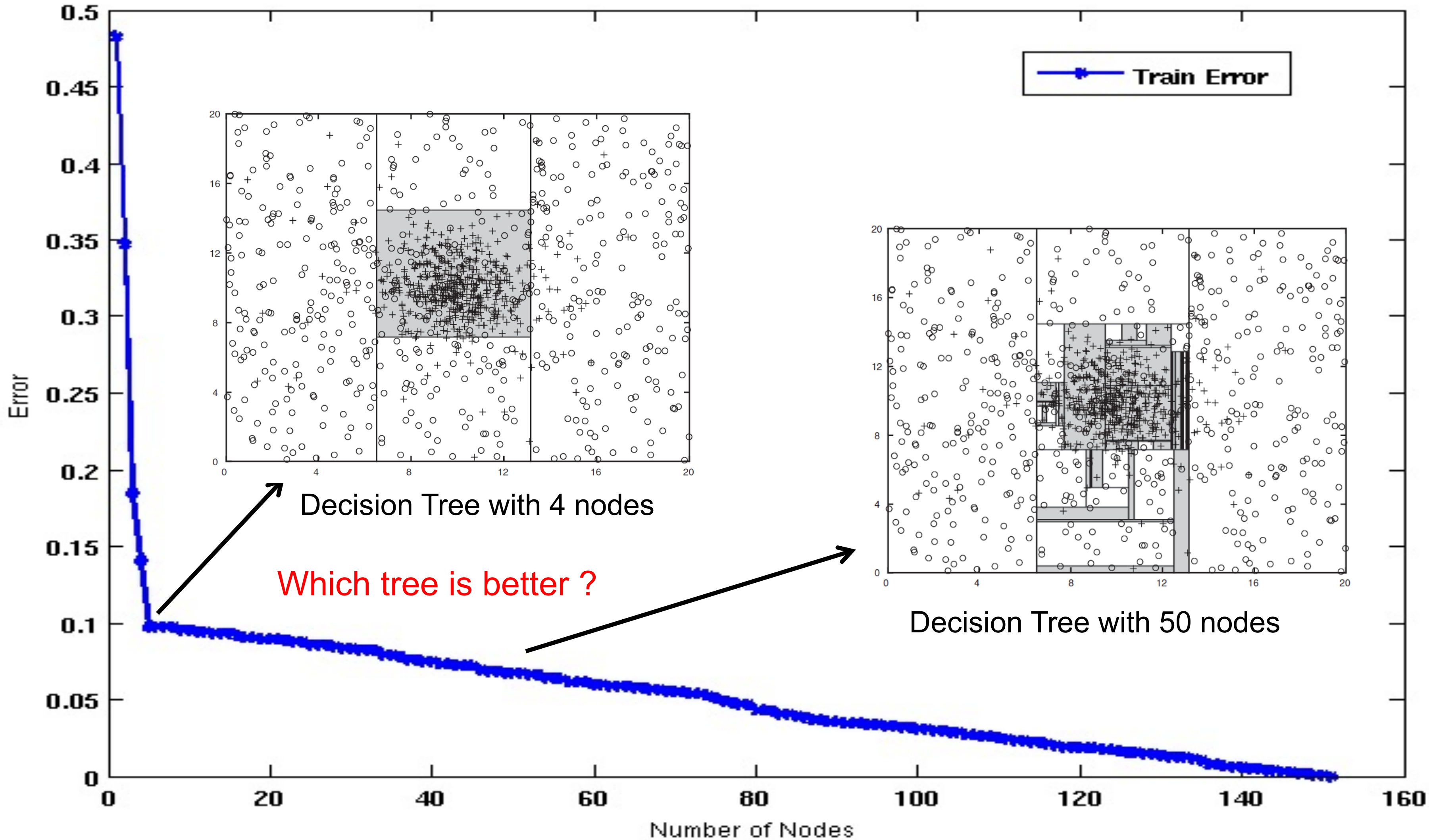
source: Tan et. Introduction to Data Mining

Decision Tree with 50 nodes



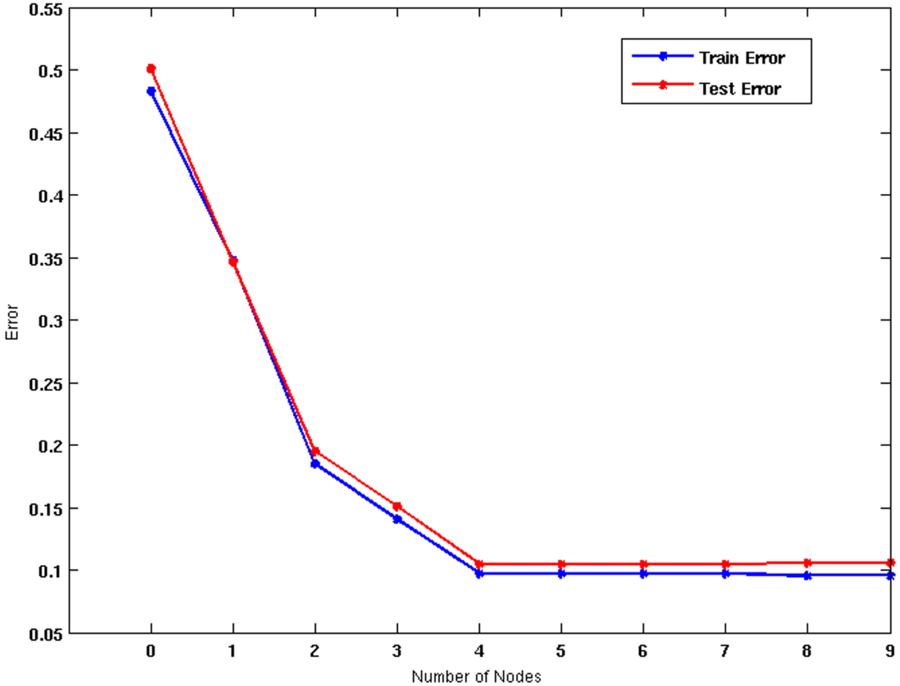
source: Tan et. Introduction to Data Mining

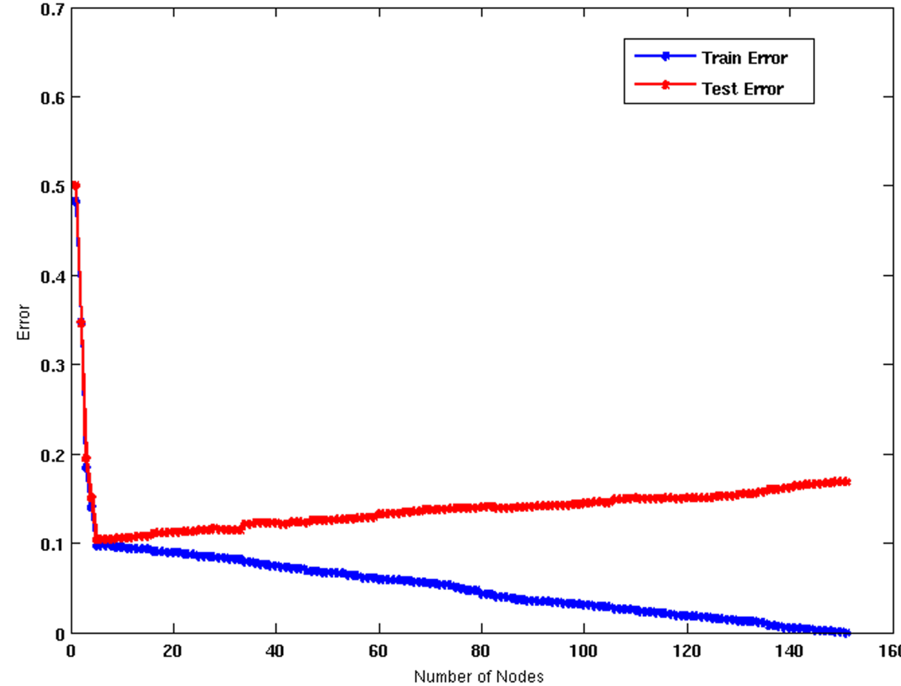
Which tree is better?



source: Tan et. Introduction to Data Mining

Model Underfitting and Overfitting





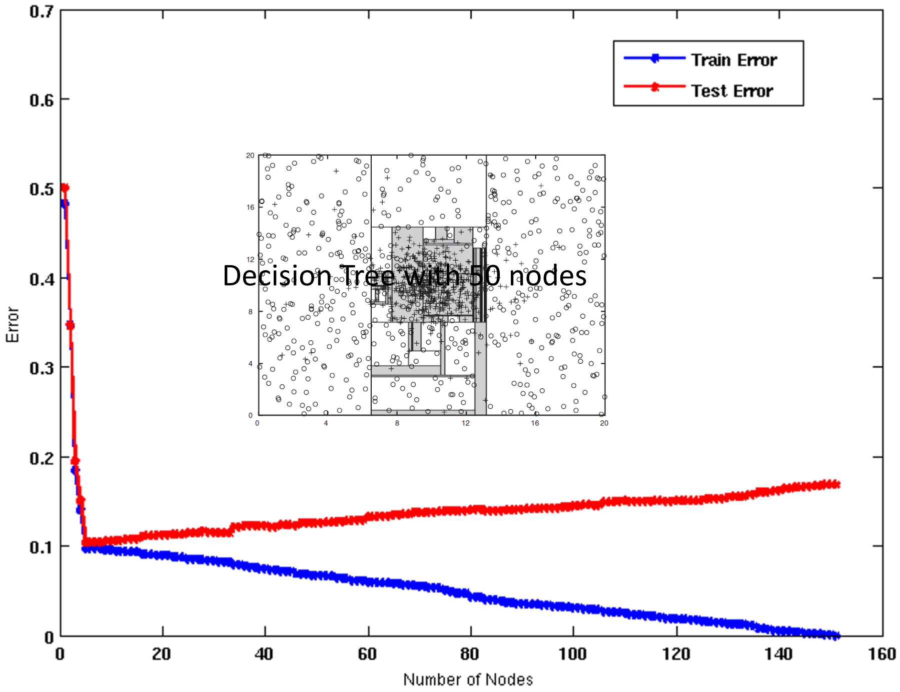
- As the model becomes more and more complex, test errors can start increasing even though training error may be decreasing

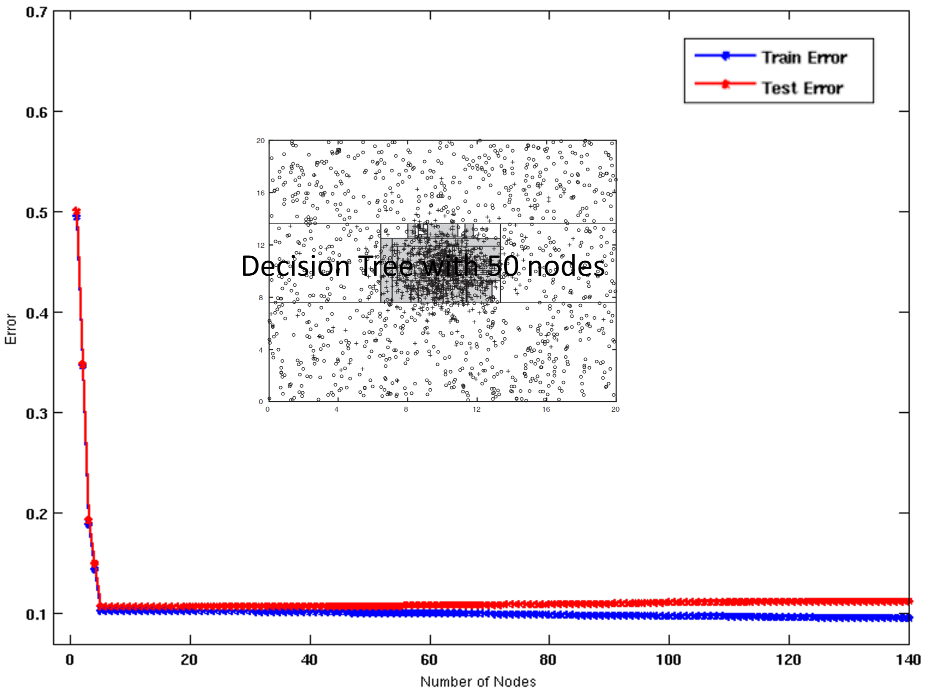
Underfitting: when model is too simple, both training and test errors are large

Overfitting: when model is too complex, training error is small but test error is large

source: Tan et. Introduction to Data Mining

Model Overfitting – Impact of Training Data Size





Using twice the number of data instances

- Increasing the size of training data reduces the difference between training and testing errors at a given size of model

source: Tan et. Introduction to Data Mining

Reasons for Model Overfitting

- **Not enough training data**

- **High model complexity**
  - Multiple Comparison Procedure

Notes on Overfitting

- Overfitting results in decision trees that are <u>more complex</u> than necessary

- Training error does not provide a good estimate of how well the tree will perform on previously unseen records

- Need ways for estimating generalization errors

Model Selection

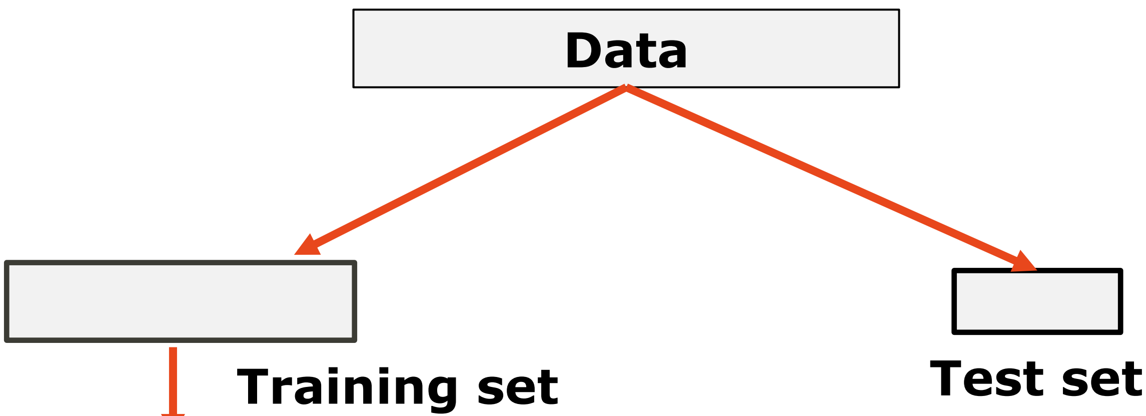
- **Performed during model building**
- **Purpose is to ensure that model is not overly complex (to avoid overfitting)**
- **Need to estimate generalization error**
  - Using Validation Set
  - Incorporating Model Complexity

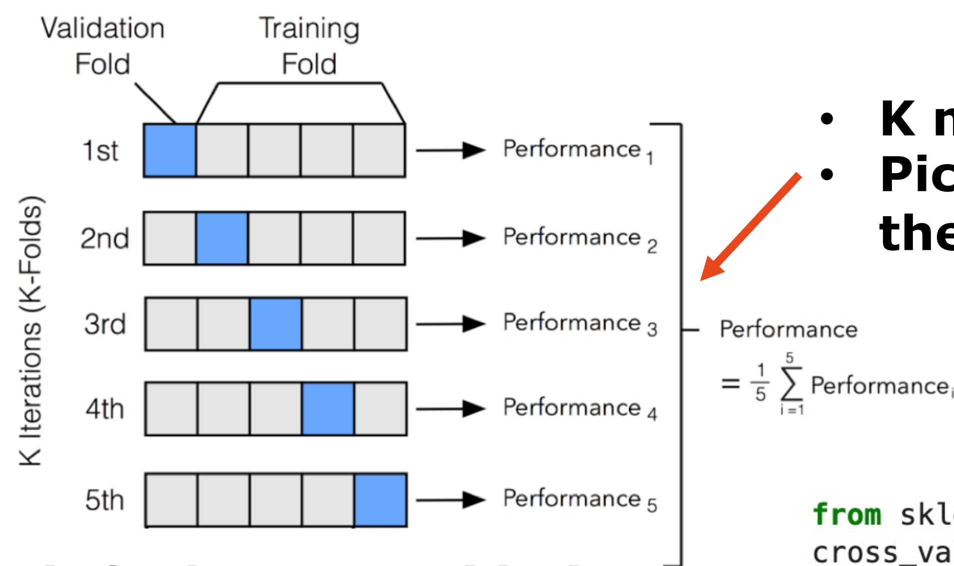
Model Selection Using Validation Set

- **Divide <u>training</u> data into two parts:**
  - Training set:
    - use for model building
  - Validation set:
    - use for estimating generalization error
    - Note: validation set is not the same as test set

- **Drawback:**
  - Less data available for training

k-Fold Cross Validation





k-fold cross-validation

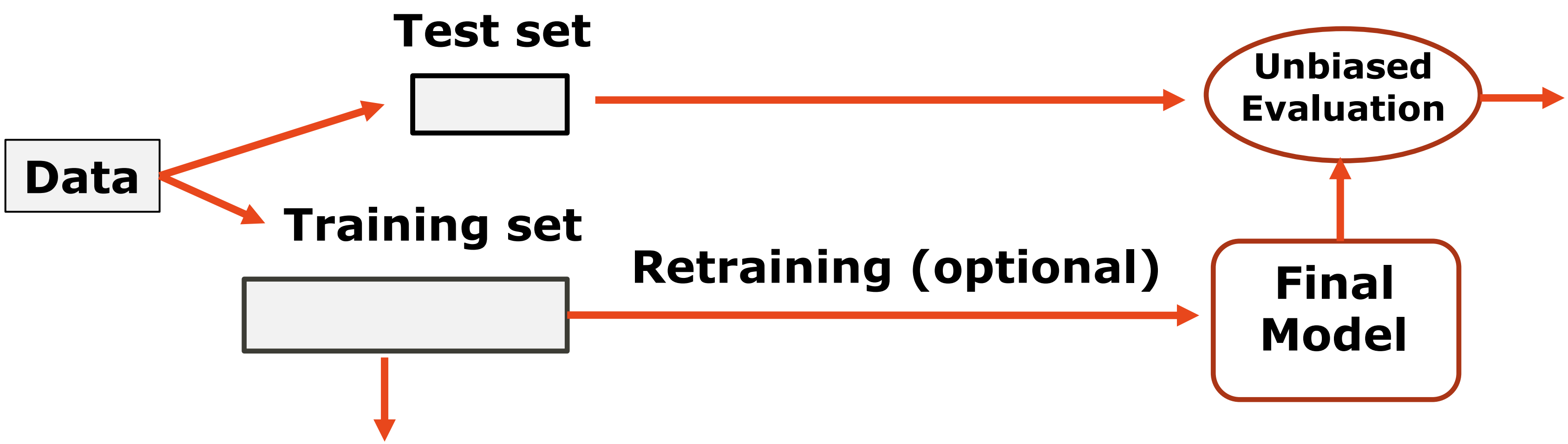
- **K models are created**
- **Pick the model with the best performance**

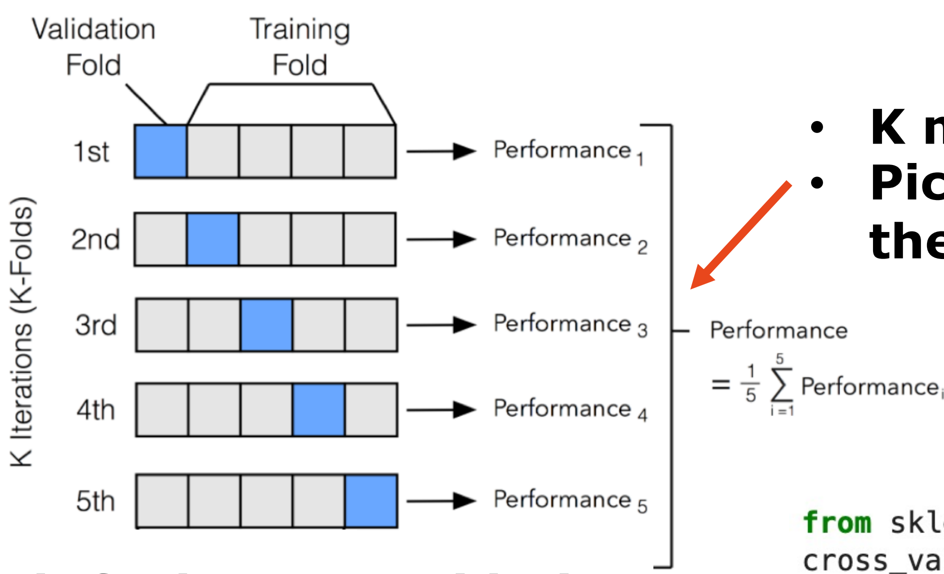
```python
from sklearn.model_selection import cross_val_score
cross_val_score(model, X_train, y_train, cv=5)
```

Model Evaluation

- **Purpose:**
  - To estimate performance of classifier on previously unseen data (test set)
- **Holdout/Test Set**
  - Reserve k% for training and (100-k)% for testing
  - Random subsampling: repeated holdout
- **Cross validation**
  - Partition data into k disjoint subsets
  - **k-fold**: train on k-1 partitions, test on the remaining one
  - **Leave-one-out**: k=n

Overall Workflow





k-fold cross-validation

- **K models are created**
- **Pick the model with the best performance**

```
from sklearn.model_selection import cross_val_score
cross_val_score(model, X_train, y_train, cv=5)
```

Model selection: M1 vs M2

- **In Module 1: Train/Test**
  - Some issues with this approach:
    - The test set is assumed to be unknown/will be encountered in future
    - Using entire train set for model fitting doesn't say much about the model's performance
- **In Module 2: Train/Validation/Test**
  - **Training set**: used for learning model
  - **Validation set**: used for tuning the parameters
    - Gives an early estimation of the model performance
  - **Test set**: used for assessing the performance of the final model

Model Selection for Decision Trees

- **Pre-Pruning (Early Stopping Rule)**
  - Stop the algorithm before it becomes a fully-grown tree
  - Typical stopping conditions for a node:
    - Stop if all instances belong to the same class
    - Stop if all the attribute values are the same
  - More restrictive conditions:
    - Stop if number of instances is less than some user-specified threshold
    - Stop if class distribution of instances are independent of the available features (e.g., using χ^2 test)
    - Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).
    - Stop if estimated generalization error falls below certain threshold

Model Selection for Decision Trees

- **Post-pruning**
  - Grow decision tree to its entirety
  - Subtree replacement
    - Trim the nodes of the decision tree in a bottom-up fashion
    - If generalization error improves after trimming, replace sub-tree by a leaf node
    - Class label of leaf node is determined from majority class of instances in the sub-tree