LECTURER: DR AKBAR GHOBAKHLOU
SCHOOL OF ENGINEERING, COMPUTER AND MATHEMATICAL SCIENCES

# Classification Basic Concept- Decision Tree (2)

## Weekly Learning Outcomes

1. Practical Issues of Classification
2. Metrics for Performance Evaluation
3. Methods for Model Comparison
4. Methods for Performance Evaluation

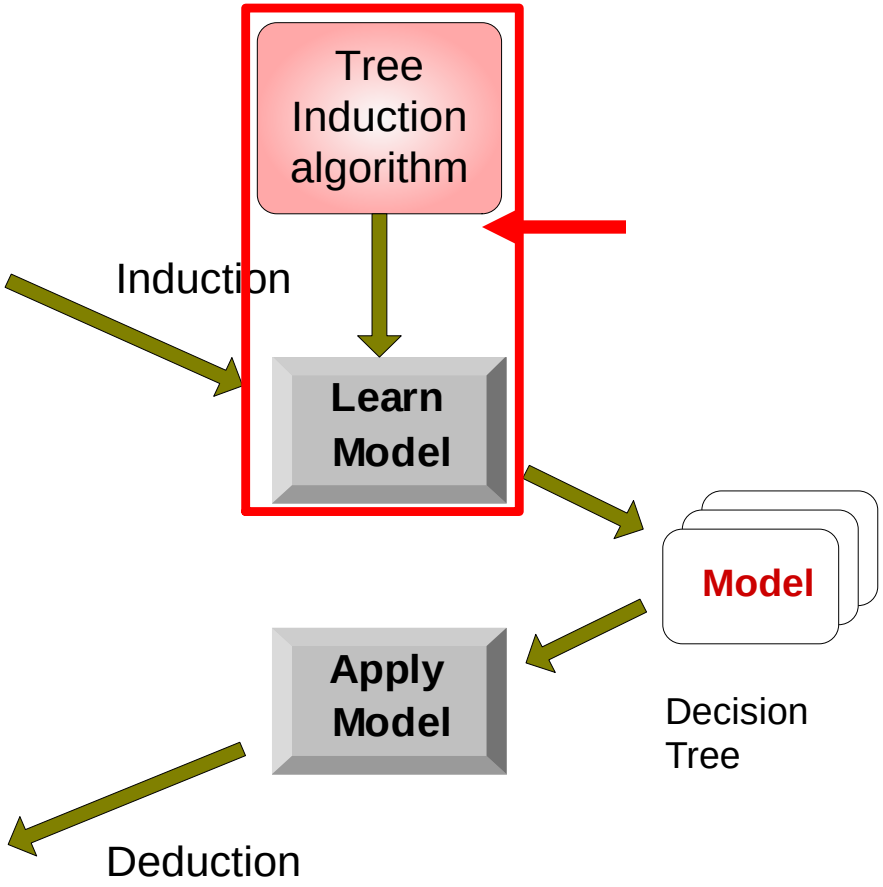
# Decision Tree Classification Task

| *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 | ? |

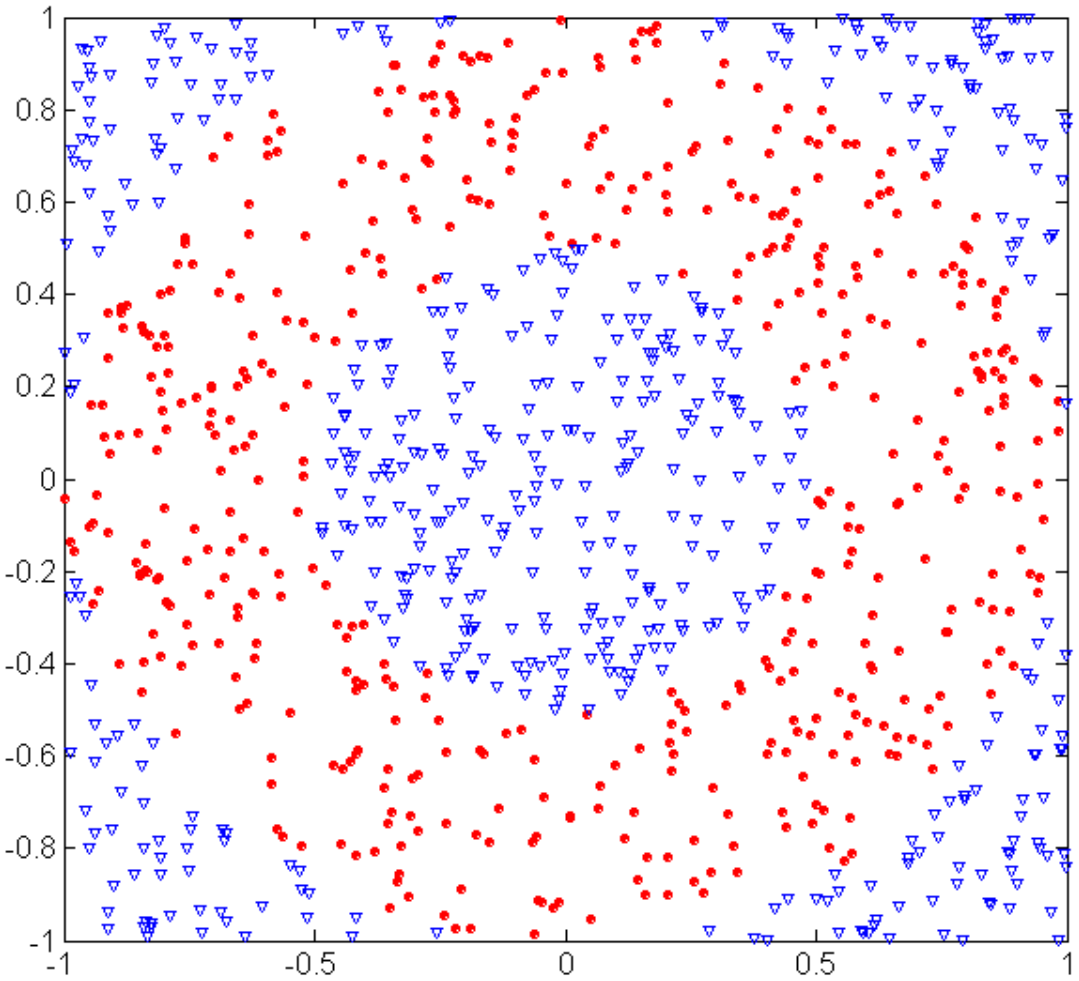
Test Set



# Practical Issues of Classification

- Underfitting and Overfitting
- Missing Values
- Costs of Classification

# Under-fitting and Over-fitting (Example)




500 circular and 500 triangular data points.

Circular points:

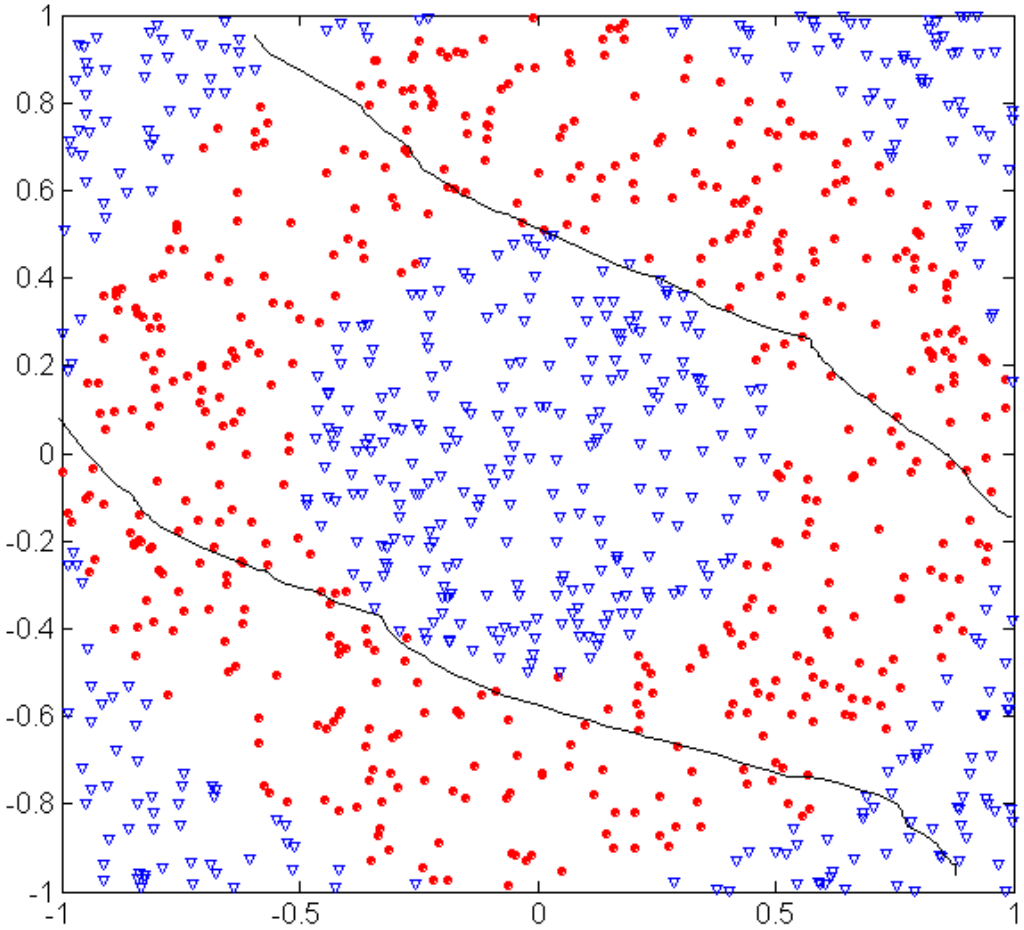
$0.5 \leq \text{sqrt}(x_1^2+x_2^2) \leq 1$

Triangular points:

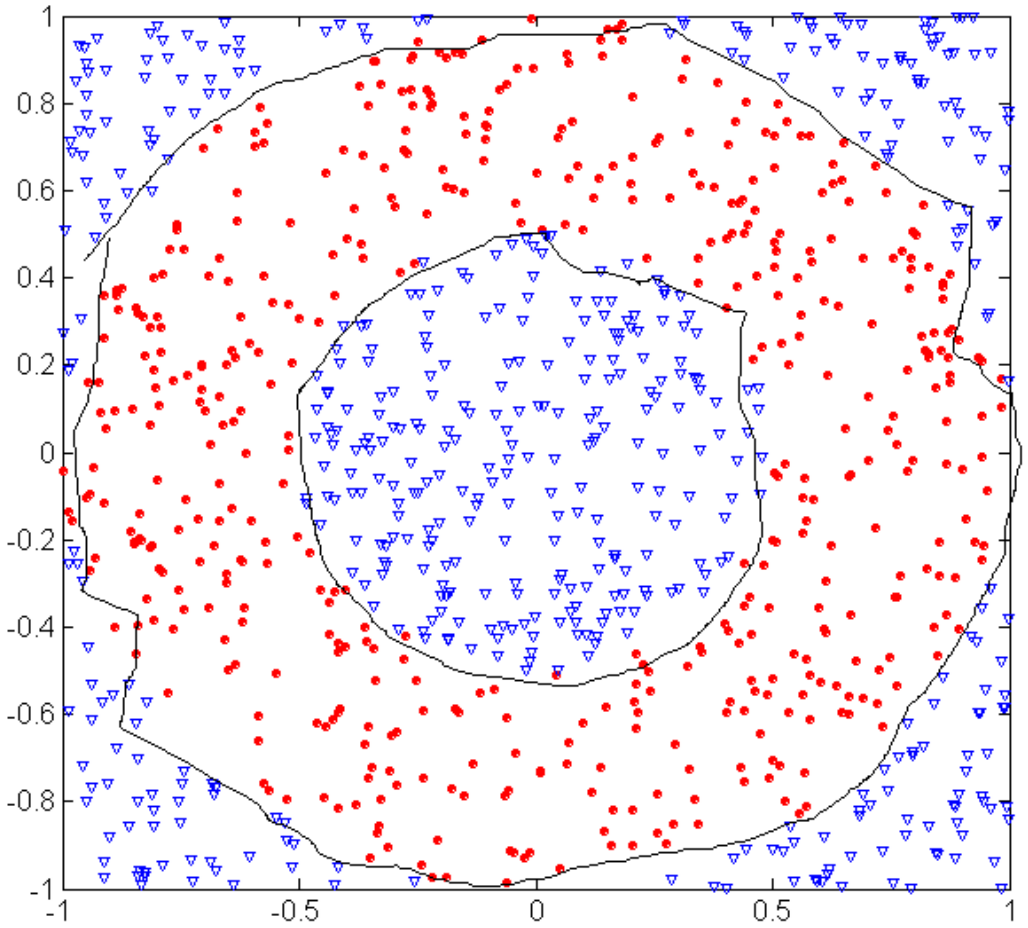
$\text{sqrt}(x_1^2+x_2^2) < 0.5$ or

$\text{sqrt}(x_1^2+x_2^2) > 1$

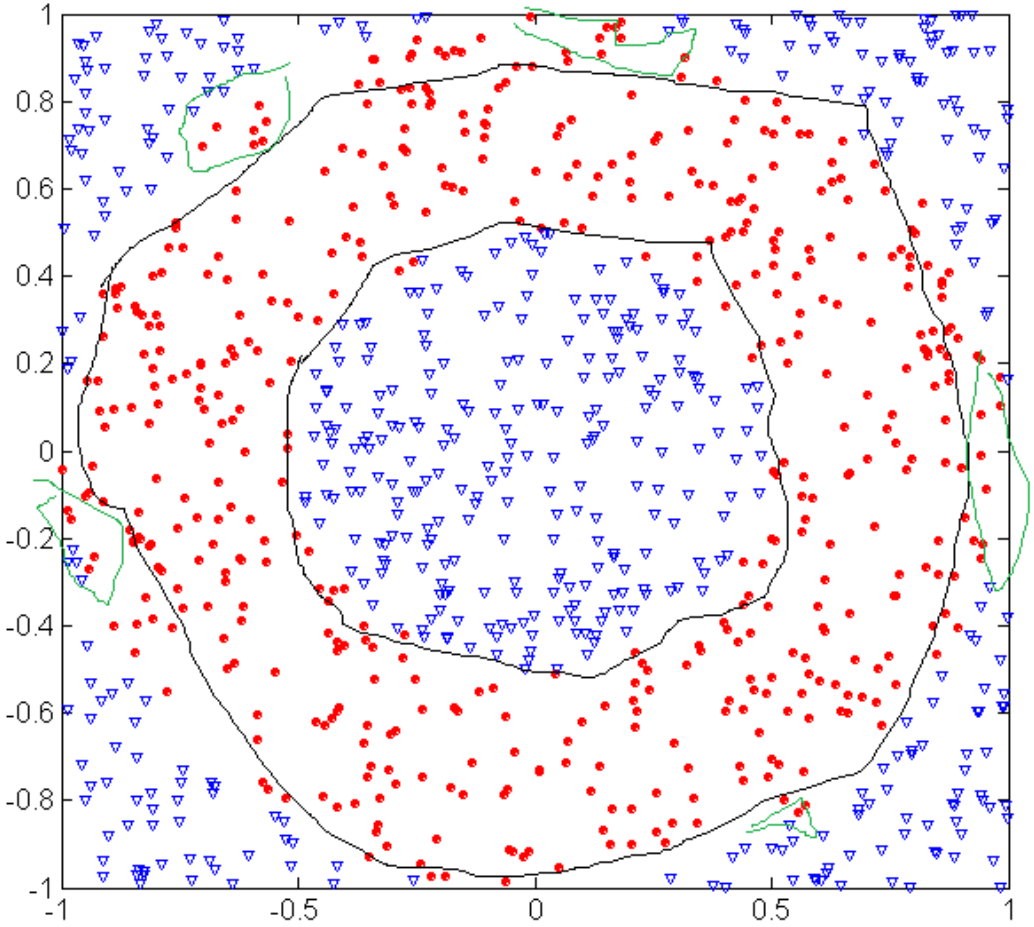
# Is this Model Good?



# What about this?



# And This?



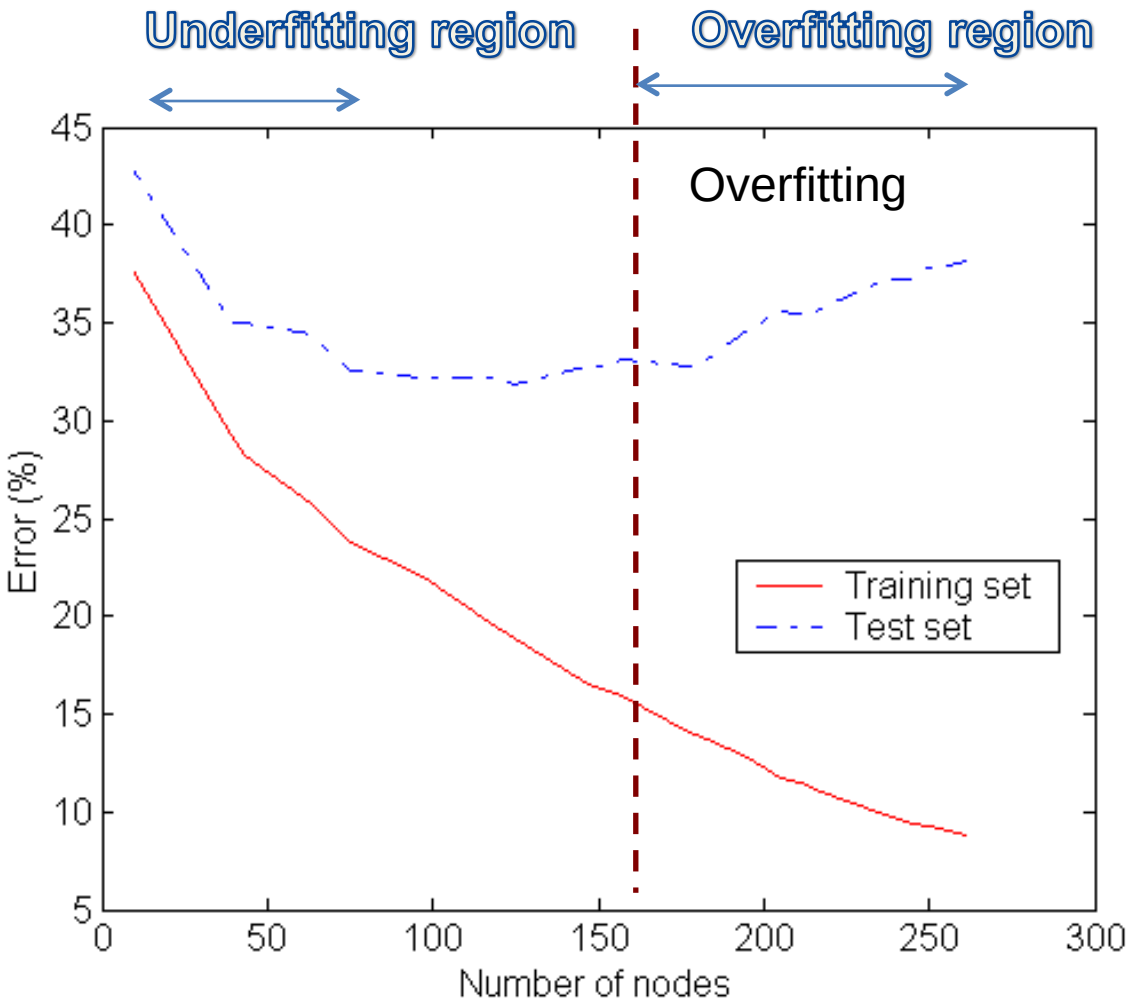
# Underfitting and Overfitting

- Two problems that can arise with models developed with Data Mining are: Overfitting and Underfitting
- Underfitting occurs when the model has not fully learned all the patterns in the data, resulting in poor prediction accuracy (test accuracy).
- Underfitting is generally caused by *the inability of the algorithm to find all patterns in the training dataset.*
- In the case of a Decision Tree method the tree developed is not of sufficient depth and size to learn all the patterns present in the data

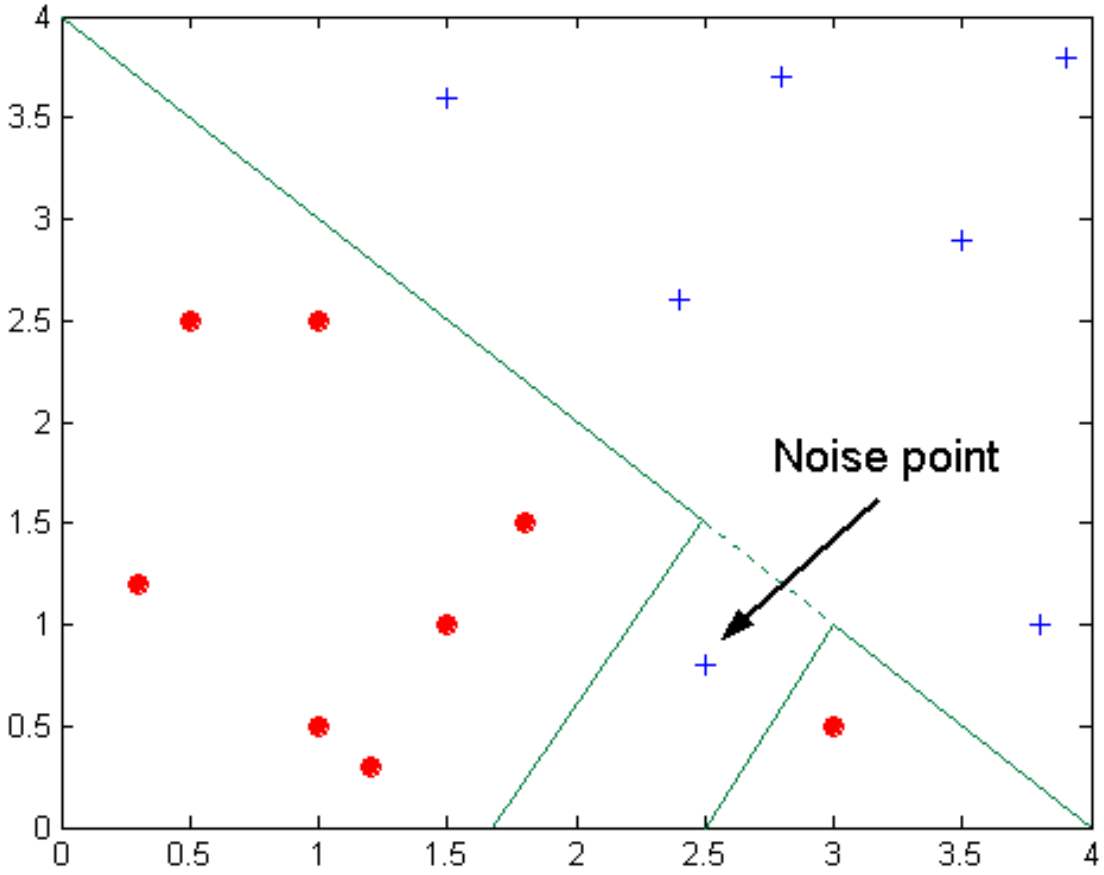
# Overfitting

- With Overfitting the model's learns the patterns in the training data very well but the model learnt cannot predict newly arriving data well
- In other words, accuracy on training dataset is high but accuracy drops drastically on newly arriving data – training set accuracy >> test set accuracy
- In the case of the Decision tree method the tree developed is too detailed (too large in size)
- Overfitting is generally caused by:
  1. *Noise (errors in assigning class labels) in the training dataset*
  2. *Lack of sufficient data to capture certain types of patterns*

# Underfitting and Overfitting

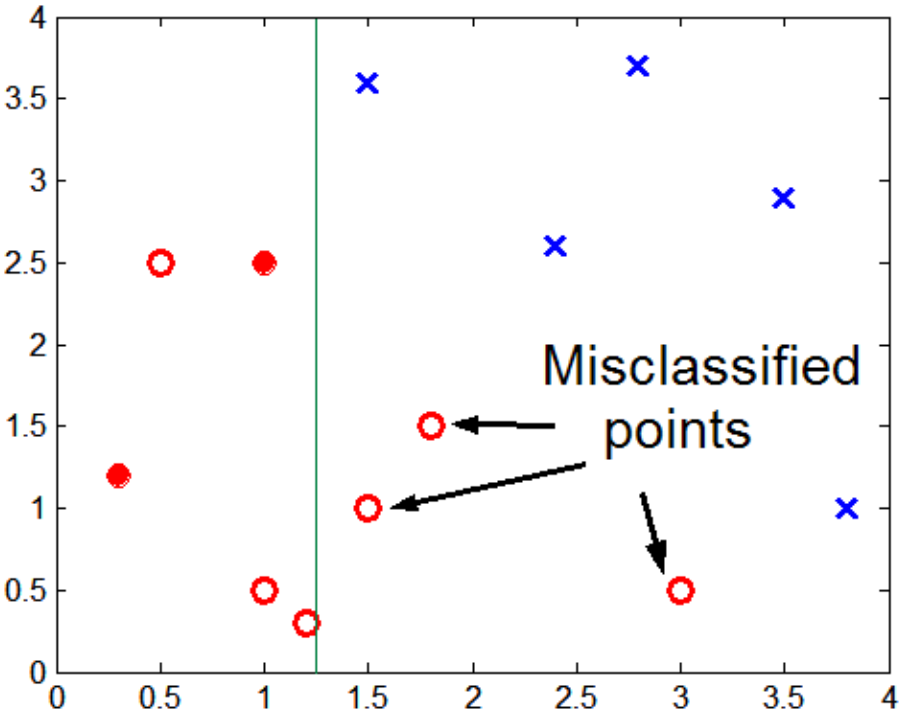


# Overfitting due to Noise




Decision boundary is distorted by noise point

# Overfitting due to Insufficient Examples




Lack of data points in the lower half of the diagram makes it difficult to predict correctly the class labels of that region

- Insufficient number of training records in the region causes the decision tree to predict the test examples using other training records that are irrelevant to the classification task

# Notes on Overfitting

- Overfitting results in decision trees that are more complex than necessary
- Training error no longer provides a good estimate of how well the tree will perform on previously unseen records
- Need new ways for estimating errors

# Methods for estimating the error

- Re-substitution errors: error on training ($\Sigma$ **e(t)** )
- Generalization errors: error on testing ($\Sigma$ **e'(t)**)
- Methods for estimating generalization errors:
  - Optimistic approach: **e'(t) = e(t)**
  - Pessimistic approach:
    - For each leaf node: **e'(t) = (e(t)+0.5)**
    - Total errors: **e'(T) = e(T) + N × 0.5** (N: number of leaf nodes)
    - For a Tree with 30 leaf nodes and 10 errors on training (out of 1000 instances):

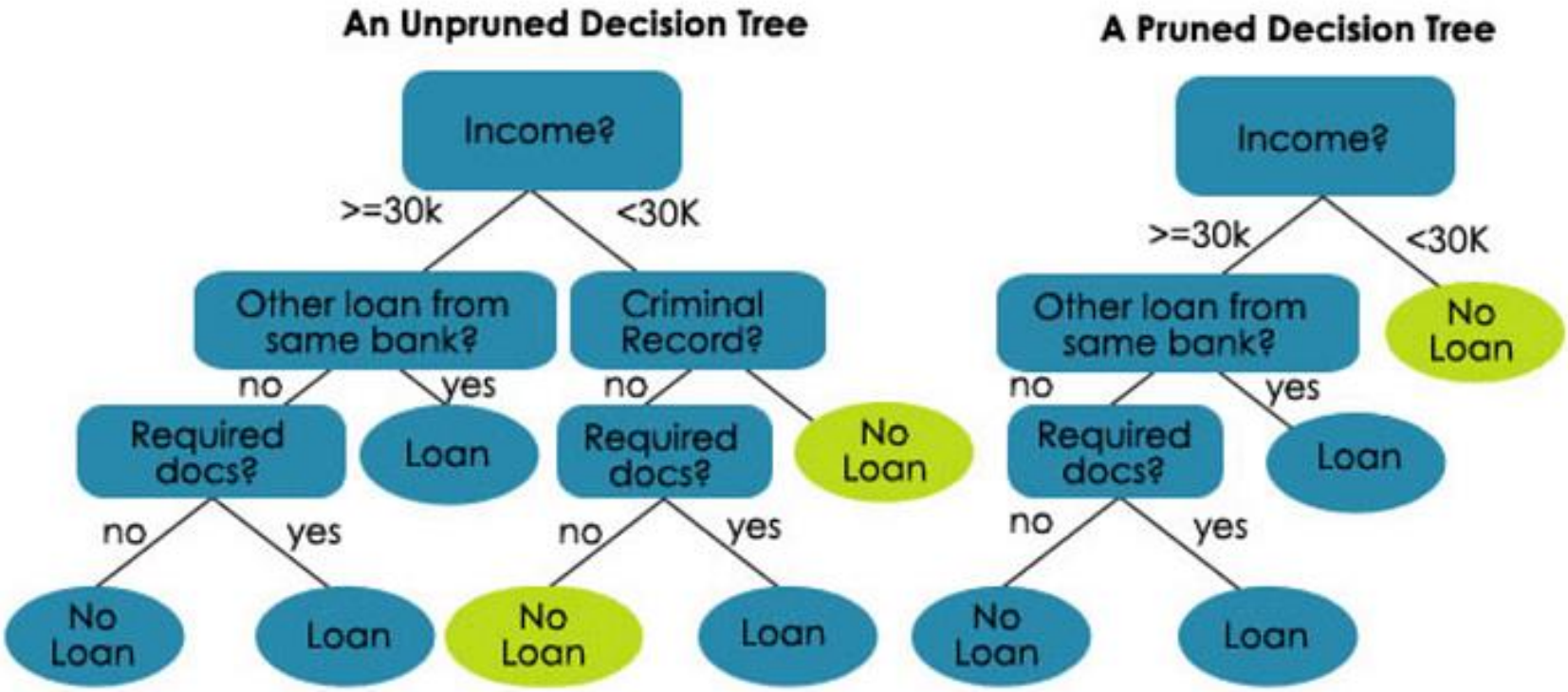
    Training error = 10/1000 = 1%

    Generalization error = (10 + 30×0.5)/1000 = 2.5%
  - Reduced error pruning (REP):
    - uses ***validation data set*** to estimate generalization error

# How to Address Overfitting…

- Post-pruning
  - *Grow decision tree to its entirety*
  - *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*

# Decision Trees Pruned vs Unpruned



# Occam's Razor

- Given two models of similar generalization errors, one should prefer the simpler model over the more complex model

- For complex models, there is a greater chance that it was fitted accidentally by errors in data

- Therefore, one should include model complexity when evaluating a model

# Model Evaluation

- Metrics for Performance Evaluation
  - *How to evaluate the performance of a model?*
- Methods for Performance Evaluation
  - *How to obtain reliable estimates?*
- Methods for Model Comparison
  - *How to compare the relative performance among competing models?*

# Model Evaluation

- Metrics for Performance Evaluation
  - *How to evaluate the performance of a model?*
- Methods for Performance Evaluation
  - *How to obtain reliable estimates?*
- Methods for Model Comparison
  - *How to compare the relative performance among competing models?*

# Why Evaluate?

- Multiple methods are available to classify or predict
- For each method, multiple choices are available for settings
- To choose best model, need to assess each model's performance

# Misclassification error

- Error = classifying a record as belonging to one class when it belongs to another class.

- Error rate = percent of misclassified records out of the total records in the validation data

# Metrics for Performance Evaluation

- Focus on the predictive capability of a model
  - *Rather than how fast it takes to classify or build models, scalability, etc.*
- Confusion Matrix:

| | | PREDICTED CLASS | |
|---|---|---|---|
| | | Class=Yes | Class=No |
| ACTUAL CLASS | Class=Yes | a | b |
| | Class=No | c | d |

a: TP (true positive)

b: FN (false negative) Type 2 error

c: FP (false positive) Type 1 error

d: TN (true negative)

# Confusion Matrix

| Classification Confusion Matrix | | |
|---|---|---|
| | Predicted Class | |
| Actual Class | 1 | 0 |
| 1 | 201 | 85 |
| 0 | 25 | 2689 |

# Confusion Matrix

201 1's correctly classified as "1"

85 1's incorrectly classified as "0"

25 0's incorrectly classified as "1"

2689 0's correctly classified as "0"

| Classification Confusion Matrix | | |
|---|---|---|
| | Predicted Class | |
| Actual Class | 1 | 0 |
| 1 | 201 | 85 |
| 0 | 25 | 2689 |

# Metrics for Performance Evaluation…

| | | PREDICTED CLASS | |
|---|---|---|---|
| | | Class=Yes | Class=No |
| ACTUAL CLASS | Class=Yes | a (TP) | b (FN) |
| | Class=No | c (FP) | d (TN) |

- Most widely-used metric:

$$\text{Accuracy} = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{TP+TN+FP+FN}$$

# Error Rate

**Overall error rate** = (25+85)/3000 = 3.67%

**Accuracy** = 1 – err = (201+2689) = 96.33%

If multiple classes, error rate is:

(sum of misclassified records)/(total records)

| Classification Confusion Matrix | | |
|---|---|---|
| | Predicted Class | |
| Actual Class | 1 | 0 |
| 1 | 201 | 85 |
| 0 | 25 | 2689 |

# Limitation of Accuracy

- Consider a 2-class problem
  - *Number of Class 0 examples = 9990*
  - *Number of Class 1 examples = 10*

- If model predicts everything to be class 0, accuracy is 9990/10000 = 99.9 %
  - *Accuracy is misleading because model does not detect any class 1 example*

# Main Metrics

**Accuracy**: the ratio of correctly classified (TP+TN) to the total number samples

$$\text{Accuracy} = \frac{\text{TN+TP}}{\text{TN+FP+FN+TP}}$$

**Precision:** the ratio of correctly classified (*TP*) to the total samples **predicted** as positive samples

$$Precision = \frac{TP}{TP+FP}$$

**Recall :** the ratio of correctly classified (*TP*) divided by total number of **actual** positive samples

$$\text{Re}\,call = \frac{TP}{TP+FN}$$

**F1 score** is also known as the **F Measure**. The F1 score states the equilibrium between the precision and the recall.

$$F1Score = \frac{2*precision*recall}{precision+recall}$$

# Model Evaluation

- Metrics for Performance Evaluation
  - *How to evaluate the performance of a model?*
- Methods for Performance Evaluation
  - *How to obtain reliable estimates?*
- Methods for Model Comparison
  - *How to compare the relative performance among competing models?*

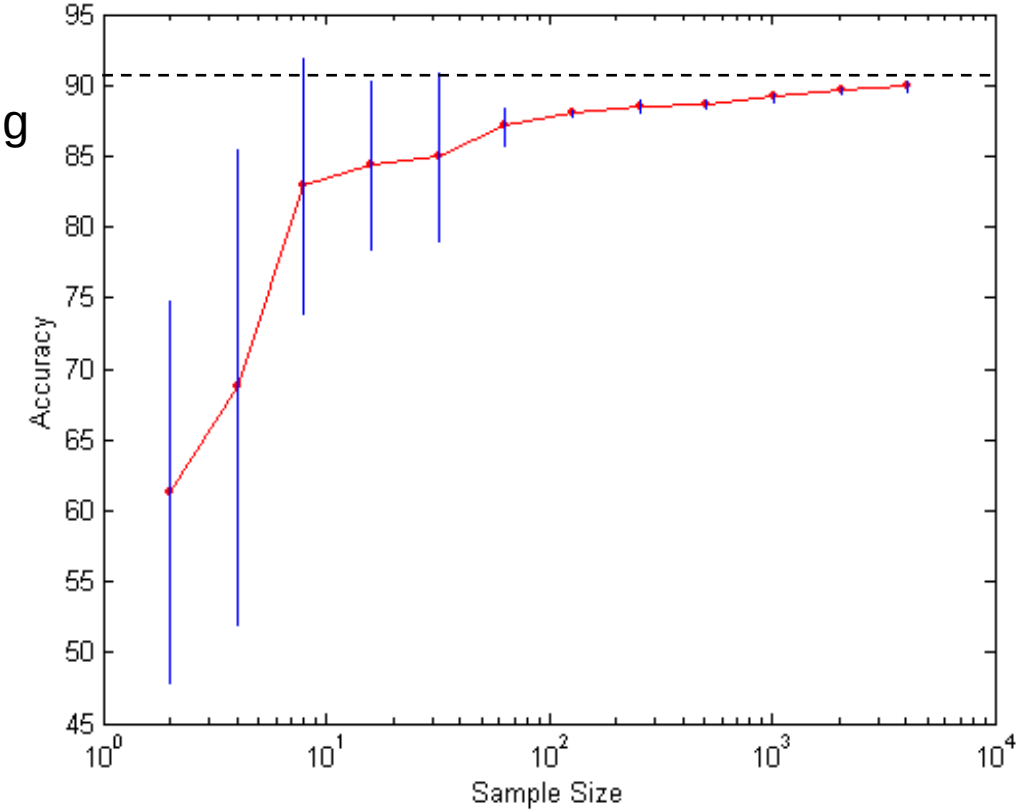
# Methods for Performance Evaluation

- How to obtain a reliable estimate of performance?

- Performance of a model may depend on other factors besides the learning algorithm:
  - *Class distribution*
  - *Cost of misclassification*
  - *Size of training and test sets*

# Learning Curve

- Learning curve shows how accuracy changes with varying sample size
- Effect of small sample size:
  - ✓ Bias in the estimate
  - ✓ Variance of estimate



# Methods of Estimation

- Holdout
  - *Reserve 2/3 for training and 1/3 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*
- Stratified sampling
  - *oversampling vs undersampling*
- Bootstrap
  - *Sampling with replacement*

# Model Evaluation

- Metrics for Performance Evaluation
  - *How to evaluate the performance of a model?*
- Methods for Performance Evaluation
  - *How to obtain reliable estimates?*
- Methods for Model Comparison
  - *How to compare the relative performance among competing models?*

# Test of Significance

- Given two models:
  - *Model M1: accuracy = 85%, tested on 30 instances*
  - *Model M2: accuracy = 75%, tested on 5000 instances*

- Can we say M1 is better than M2?
  - *How much confidence can we place on accuracy of M1 and M2?*
  - *Can the difference in performance measure be explained as a result of random fluctuations in the test set?*

# Comparing Performance of 2 Models

- Given two models, say M1 and M2, which is better?
  - *M1 is tested on D1 (size=n1), found error rate = $e_1$*
  - *M2 is tested on D2 (size=n2), found error rate = $e_2$*
  - *Assume D1 and D2 are independent*
  - *If n1 and n2 are sufficiently large, then*

$$e_1 \sim N(\mu_1, \sigma_1)$$

$$e_2 \sim N(\mu_2, \sigma_2)$$

  - *Approximate:*

$$\hat{\sigma}_i = \frac{e_i(1-e_i)}{n_i}$$

# Comparing Performance of 2 Models

- To test if performance difference is statistically significant: d = e1 – e2
  - *$d \sim N(d_t, \sigma_t)$ where $d_t$ is the true difference*
  - *Since D1 and D2 are independent, their variance adds up:*

$$\sigma_t^2 = \sigma_1^2 + \sigma_2^2 \cong \hat{\sigma}_1^2 + \hat{\sigma}_2^2$$

$$= \frac{e1(1-e1)}{n1} + \frac{e2(1-e2)}{n2}$$

  - *At (1-$\alpha$) confidence level,* $d_t = d \pm Z_{\alpha/2}\hat{\sigma}_t$

*Link to read more on* *Model evaluation, model selection*

# References

1. *Data Mining: Practical Machine Learning Tools and Techniques (3rd edition) /* Ian Witten, Eibe Frank; Elsevier, 2011, Chapter 4