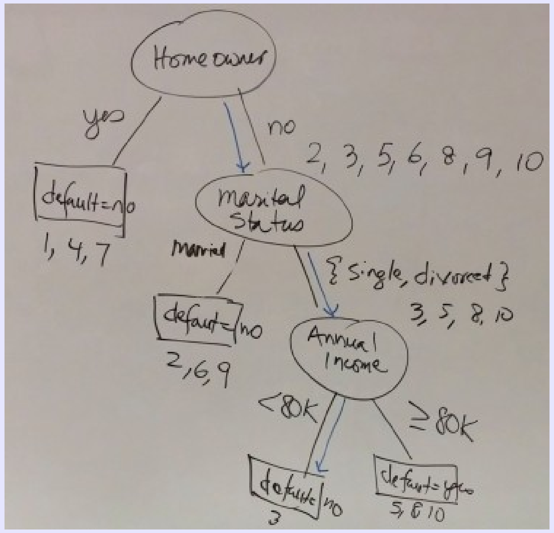
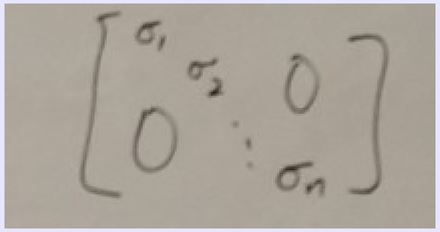
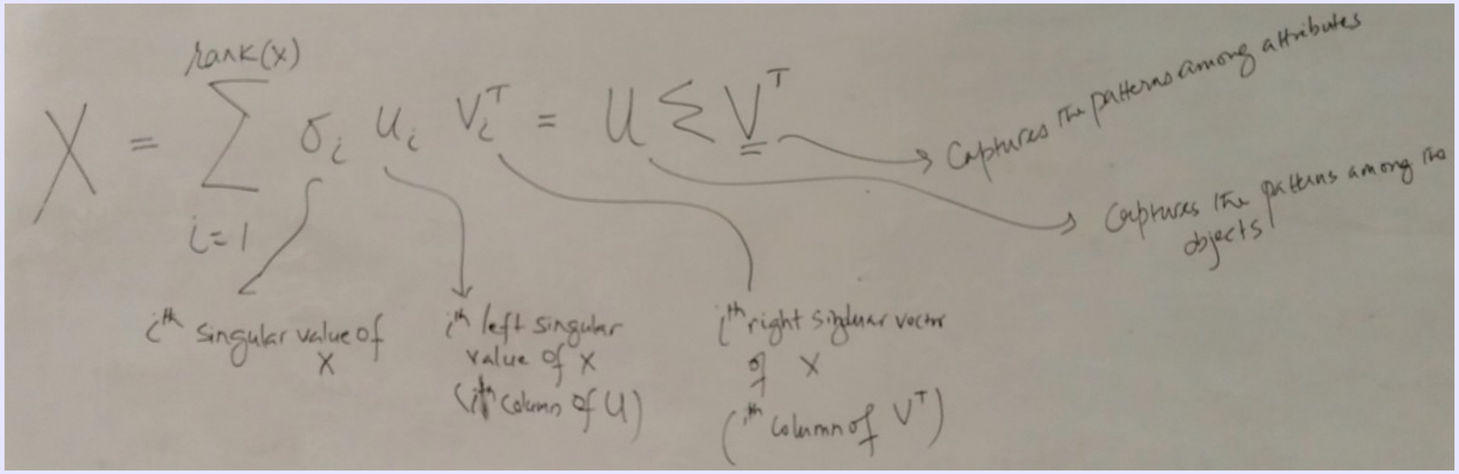
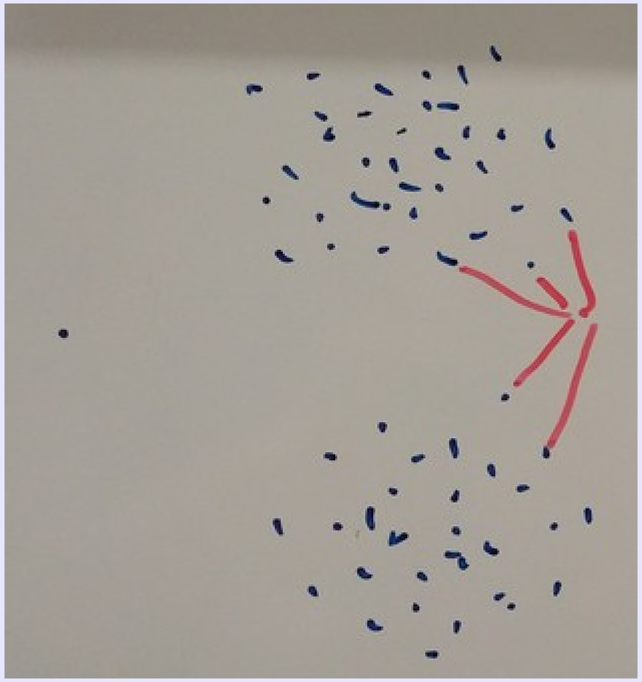



$\begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & 0 \\ 0 & & \ddots & \\ & & & \sigma_n \end{bmatrix}$


$X = \sum_{i=1}^{rank(X)} \sigma_i u_i v_i^T = U \Sigma V^T$
Captures the patterns among attributes
Captures the patterns among the objects
ith singular value of X
ith left singular value of X (ith column of U)
ith right singular vector of X (ith column of V^T)



CS 422-04: Data Mining

Vijay K. Gurbani, Ph.D., Illinois Institute of Technology

Lecture 5: **Decision Trees (continued)**
Interpretation and evaluation of Decision Trees
Advanced Decision Trees

Tree Induction

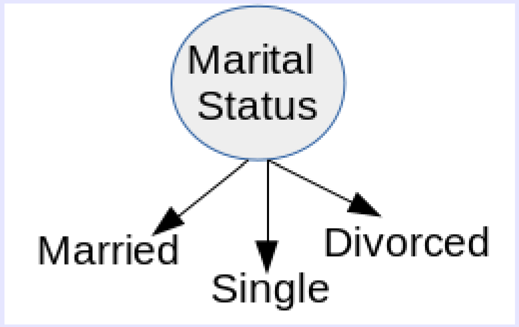
- Now that we know how to construct a Decision Tree … let’s see how to split the records at each level.
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Issues
  - How to split the records?
    - Specify attribute test condition
    - How to determine the best split?
  - When to stop splitting.

Tree Induction: Specify attribute test conditions

- Depends on attribute type
  - Binary (simple: 2-way split)
  - Nominal
  - Ordinal
  - Continuous
- Depends on number of ways to split
  - 2-way split
  - N-way split

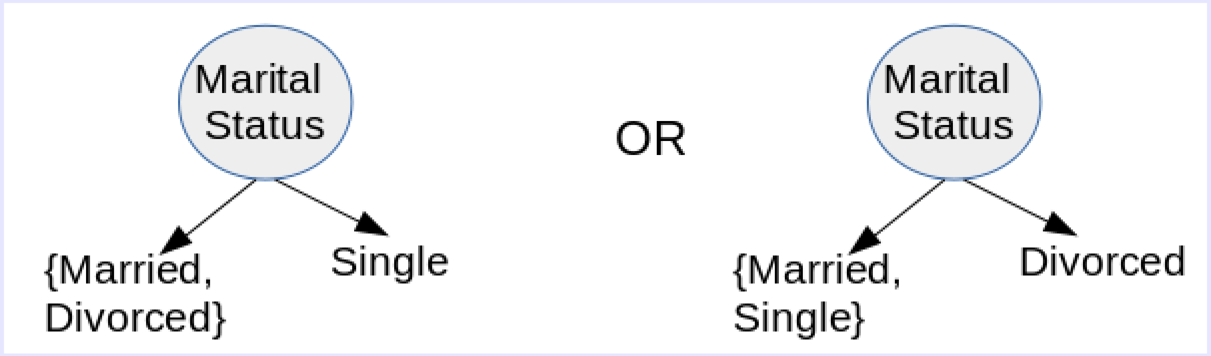
Tree Induction: Specify attribute test conditions

Nominal attributes



Multi-way split: Use as many partitions as there are distinct values.

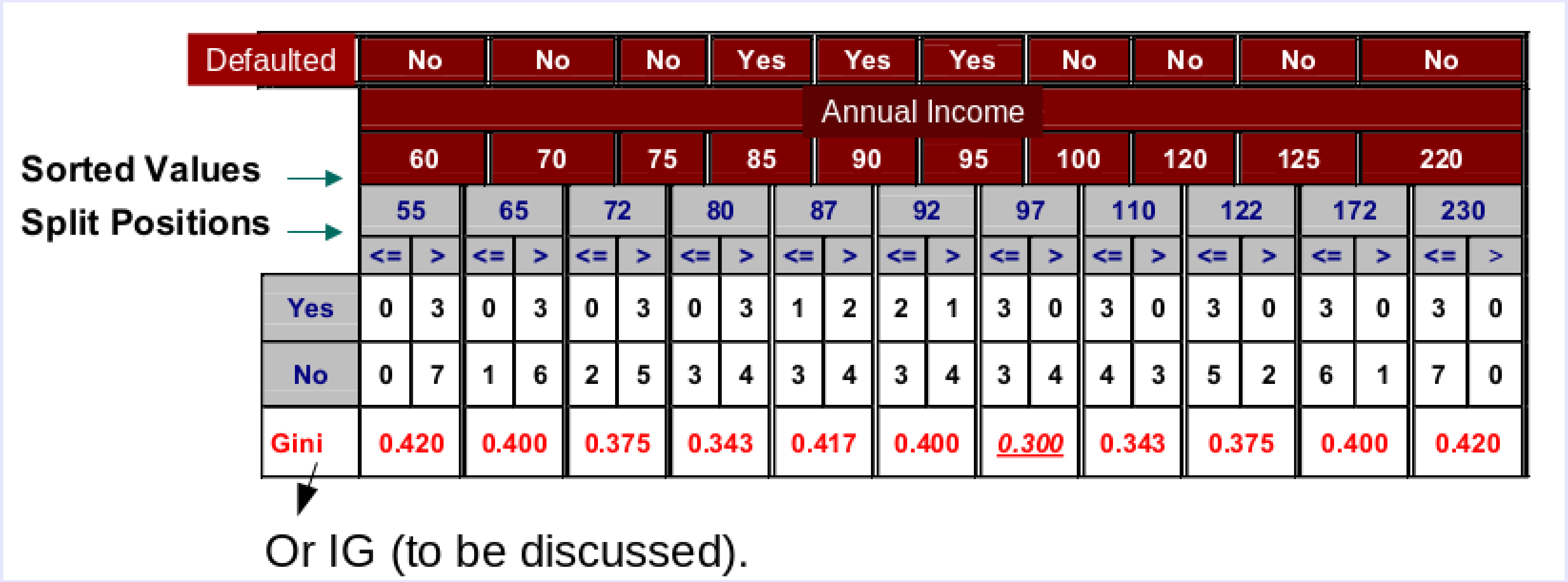
Binary split: Divides values in two subsets; need to find optimal partitioning.



Tree Induction: Specify attribute test conditions

Continuous attributes

- Discretize: That is, convert from continuous to binary, or n-ary.
  - How: Step 1: Sort the data
    Step 2: Split them by specifying n-1 split points and bin them by frequency counting based on response variable.



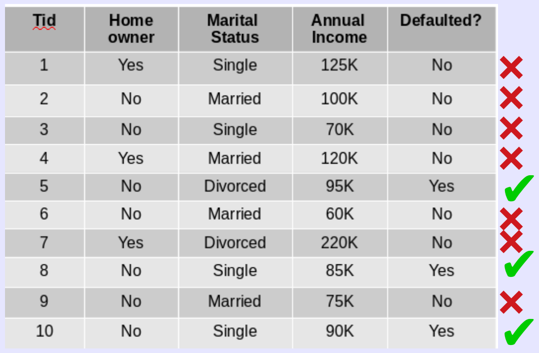
Defaulted	No	No	No	Yes	Yes	Yes	No	No	No	No
Annual Income										
Sorted Values	60	70	75	85	90	95	100	120	125	220

Split Positions	55		65		72		80		87		92		97		110		122		172		230	
	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>	<=	>
Yes	0	3	0	3	0	3	0	3	1	2	2	1	3	0	3	0	3	0	3	0	3	0
No	0	7	1	6	2	5	3	4	3	4	3	4	3	4	4	3	5	2	6	1	7	0
Gini	0.420		0.400		0.375		0.343		0.417		0.400		*0.300*		0.343		0.375		0.400		0.420	

Or IG (to be discussed).

Tree Induction: How to determine best split?

- Entropy: Amount of uncertainty involved in the value of a random variable, or the measure of disorder in a system.
- Entropy is defined as: $H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$ where y_i is the class label.
- Example:



Tid	Home owner	Marital Status	Annual Income	Defaulted?	
1	Yes	Single	125K	No	✗
2	No	Married	100K	No	✗
3	No	Single	70K	No	✗
4	Yes	Married	120K	No	✗
5	No	Divorced	95K	Yes	✓
6	No	Married	60K	No	✗
7	Yes	Divorced	220K	No	✗
8	No	Single	85K	Yes	✓
9	No	Married	75K	No	✗
10	No	Single	90K	Yes	✓

$$H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$$

At the root node of the tree, Entropy is calculated as follows:

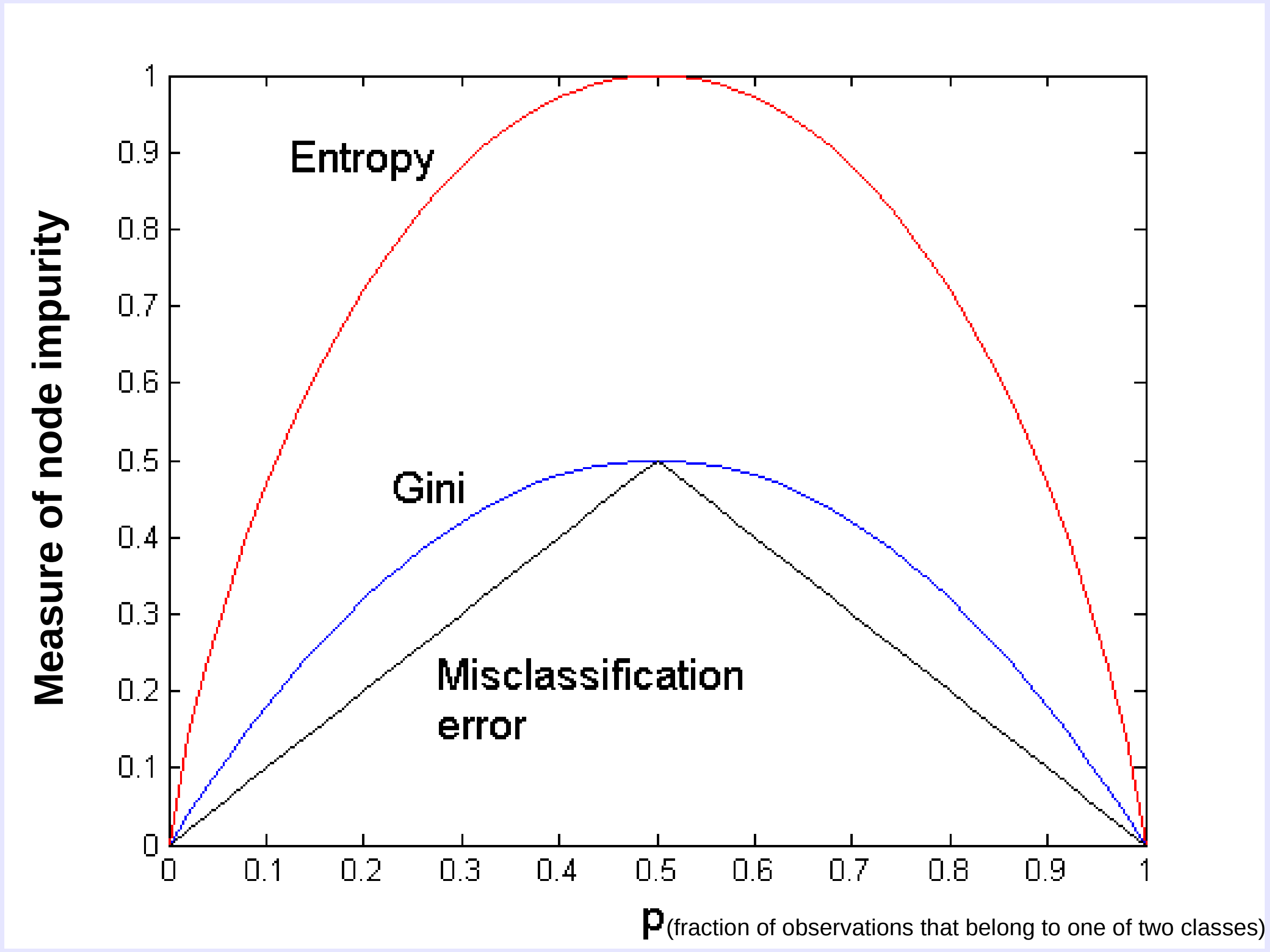
$$H(Y) = -[\frac{7}{10}\ log_2 \frac{7}{10} + \frac{3}{10}\ log_2 \frac{3}{10}] = 0.88$$

Tree Induction: How to determine best split?

- In decision tree algorithms, entropy measures *purity*.
  - *Purity* is defined as the fraction of observations belonging to a particular class in a node.
    - If all observations belong to the same class, we have a pure node → when we have a pure node, we minimize entropy.

Tree Induction: How to determine best split?

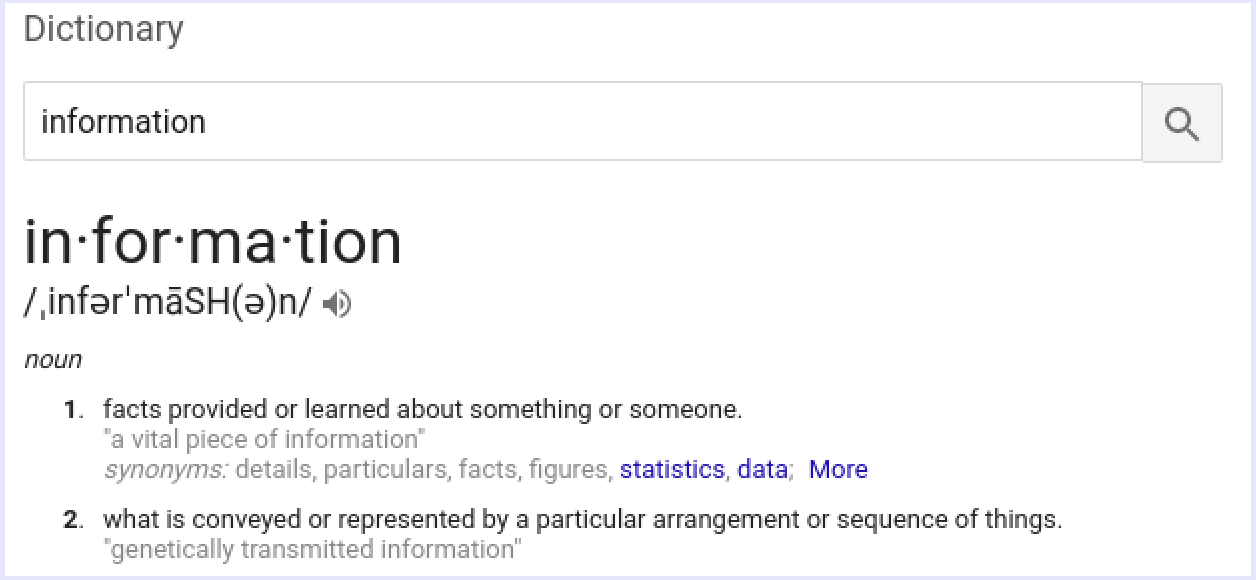
- Measures of node impurity
  - Gini index (used by CART, and *rpart*)
  - Entropy
  - Misclassification error



At p = 0.5, **maximum impurity**
when p = 0 or 1, distribution is **pure** (or **minimum impurity**)

Tree Induction: How to determine best split?

- Related to entropy is **Information**.
- Entropy: Amount of uncertainty involved in the value of a random variable, or the measure of disorder in a system.
- Information is →
- So, informally, information is the opposite of entropy.
  - We want to *maximize* Information, or **Information Gain (IG)** while *minimizing* entropy.



Dictionary

information

in·for·ma·tion

/ˌinfərˈmāSH(ə)n/

noun

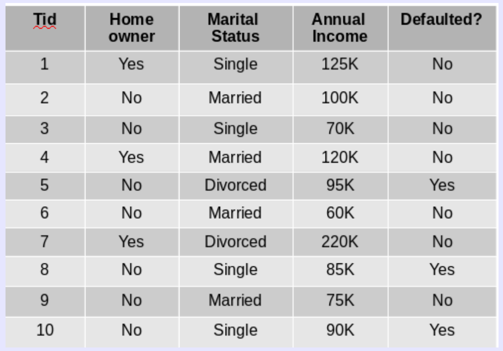
1. facts provided or learned about something or someone.
   "a vital piece of information"
   synonyms: details, particulars, facts, figures, statistics, data; More
2. what is conveyed or represented by a particular arrangement or sequence of things.
   "genetically transmitted information"

Tree Induction: How to determine best split?

- When we split, key question we want to answer is:
  - How much “information” does an attribute give us about the class?
    - Attributes that perfectly partition the observations should give us maximal information (pure partitions).
    - Unrelated attributes should give no (or very little) information.
- So we need to choose the split that maximizes **Information Gain (IG)** while minimizing entropy after the split.

Information Gain = Entropy before the split – Entropy after the split.

Tree Induction: How to determine best split?



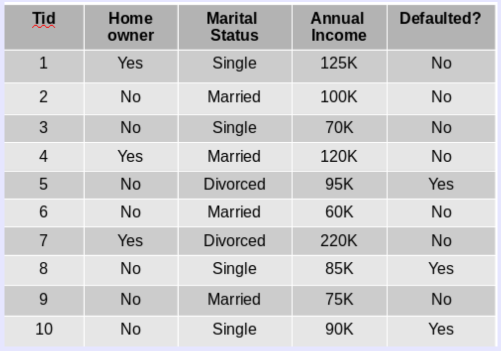
Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$$

At the root node of the tree, Entropy is calculated as follows:

$$H(Y) = -[\frac{7}{10}\ log_2\frac{7}{10} + \frac{3}{10}\ log_2\frac{3}{10}] = 0.88$$

Tree Induction: How to determine best split?



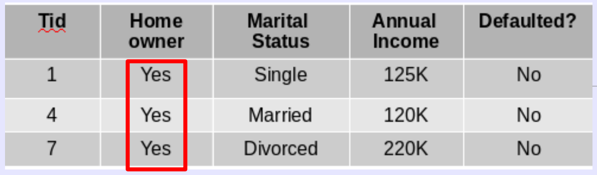
Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$$

At the root node of the tree, Entropy is calculated as follows:

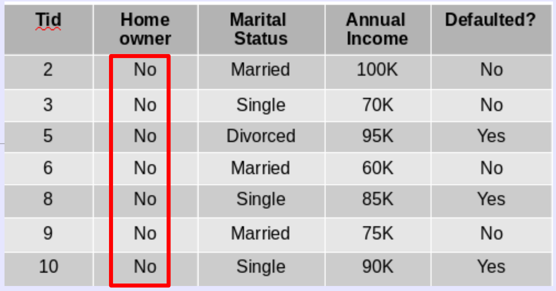
$$H(Y) = -[\frac{7}{10}\ log_2\frac{7}{10} + \frac{3}{10}\ log_2\frac{3}{10}] = 0.88$$

Let’s see what’s the IG if we split on Homeowner attribute.



Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

$$H(Y) = -[\frac{0}{3}\ log_2\frac{0}{3} + \frac{3}{3}\ log_2\frac{3}{3}] = 0.0$$



Tid	Home owner	Marital Status	Annual Income	Defaulted?
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -[\frac{4}{7}\ log_2\frac{4}{7} + \frac{3}{7}\ log_2\frac{3}{7}] = 0.99$$

Tree Induction: How to determine best split?

Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$$

At the root node of the tree, Entropy is calculated as follows:

$$H(Y) = -[\frac{7}{10}\ log_2\frac{7}{10} + \frac{3}{10}\ log_2\frac{3}{10}] = 0.88$$

Let's see what's the IG if we split on Homeowner attribute.

Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

$$H(Y) = -[\frac{0}{3}\ log_2\frac{0}{3} + \frac{3}{3}\ log_2\frac{3}{3}] = 0.0$$

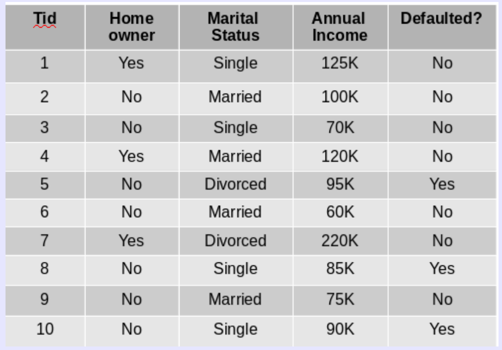
Tid	Home owner	Marital Status	Annual Income	Defaulted?
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -[\frac{4}{7}\ log_2\frac{4}{7} + \frac{3}{7}\ log_2\frac{3}{7}] = 0.99$$

Now calculate the conditional entropy H(Y|X), or the remaining entropy of Y given X:

$$H(Defaulted|Homeowner) = \frac{3}{10} * 0 + \frac{4}{10} * 0.99 = 0.69$$

Tree Induction: How to determine best split?



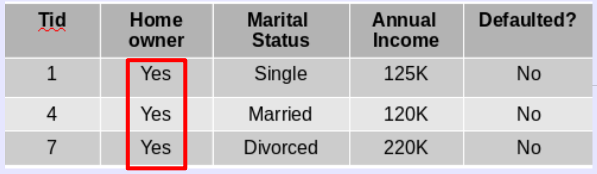
Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -\sum_{i=0}^{c-1} P(Y = y_i)\ log_2 P(Y = y_i)$$

At the root node of the tree, Entropy is calculated as follows:

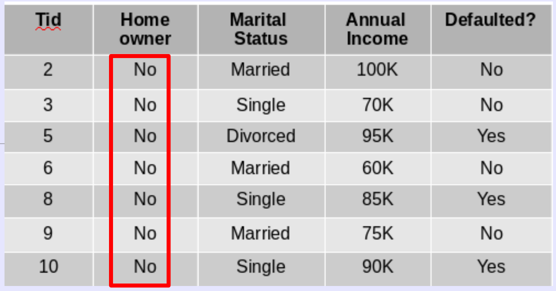
$$H(Y) = -[\frac{7}{10}\ log_2\frac{7}{10} + \frac{3}{10}\ log_2\frac{3}{10}] = 0.88$$

Let’s see what’s the IG if we split on Homeowner attribute.



Tid	Home owner	Marital Status	Annual Income	Defaulted?
1	Yes	Single	125K	No
4	Yes	Married	120K	No
7	Yes	Divorced	220K	No

$$H(Y) = -[\frac{0}{3}\ log_2\frac{0}{3} + \frac{3}{3}\ log_2\frac{3}{3}] = 0.0$$



Tid	Home owner	Marital Status	Annual Income	Defaulted?
2	No	Married	100K	No
3	No	Single	70K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

$$H(Y) = -[\frac{4}{7}\ log_2\frac{4}{7} + \frac{3}{7}\ log_2\frac{3}{7}] = 0.99$$

Now calculate the conditional entropy H(Y|X), or the remaining entropy of Y given X:

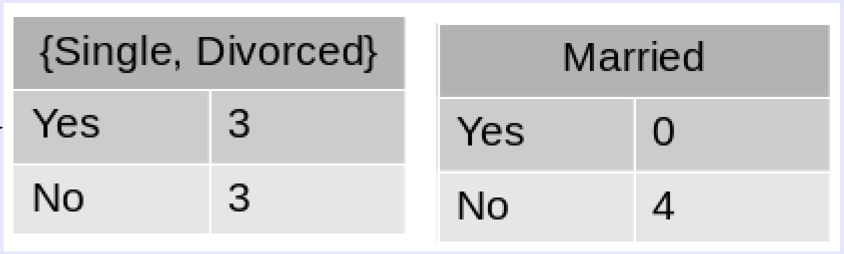
$$H(Defaulted|Homeowner) = \frac{3}{10} * 0 + \frac{4}{10} * 0.99 = 0.69$$

IG on splitting on Homeowner = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

Tree Induction: How to determine best split?

IG on splitting on Homeowner = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

IG on splitting on Marital Status = Entropy before – Entropy after
= 0.88 – 0.60 = 0.28



{Single, Divorced}	
Yes	3
No	3

Married	
Yes	0
No	4

Entropy: 1

Entropy: 0

$$H(Defaulted|Marital\,Status) = \frac{6}{10} * 1.00 + \frac{4}{10} * 0.00 = 0.60$$

Tree Induction: How to determine best split?

IG on splitting on Homeowner = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

IG on splitting on Marital Status = Entropy before – Entropy after
= 0.88 – 0.60 = 0.28

IG on splitting on Annual Income = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

{Single, Divorced}	
Yes	3
No	3

Entropy: 1

Married	
Yes	0
No	4

Entropy: 0

$$H(Defaulted|Marital\,Status) = \frac{6}{10} * 1.00 + \frac{4}{10} * 0.00 = 0.60$$

Income >= 80K	
Yes	3
No	4

Entropy: 0.99

Income < 80K	
Yes	0
No	3

Entropy: 0

$$H(Defaulted|Income) = \frac{7}{10} * 0.99 + \frac{3}{10} * 0.00 = 0.69$$

Tree Induction: How to determine best split?

IG on splitting on Homeowner = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

Best Split

IG on splitting on Marital Status = Entropy before – Entropy after
= 0.88 – 0.60 = 0.28

IG on splitting on Annual Income = Entropy before – Entropy after
= 0.88 – 0.69 = 0.19

{Single, Divorced}	
Yes	3
No	3

Entropy: 1

Married	
Yes	0
No	4

Entropy: 0

$$H(Defaulted|Marital\,Status) = \frac{6}{10} * 1.00 + \frac{4}{10} * 0.00 = 0.60$$

Income >= 80K	
Yes	3
No	4

Entropy: 0.99

Income < 80K	
Yes	0
No	3

Entropy: 0

$$H(Defaulted|Income) = \frac{7}{10} * 0.99 + \frac{3}{10} * 0.00 = 0.69$$

Tree Induction: When to stop splitting

- If we allow the tree to grow (become deep), we run the risk of overfitting.
  - Overfitting can be mitigated by **pruning** the tree: Grow the tree to its entirety, then trim nodes in a bottom-up fashion. If generalization error improves, replace sub-tree by a leaf node. Class label of the leaf node is determined by majority class of the instances in the sub-tree.
- If we stop early, we may underfit (error on training data may be low).
- Strategies on when to stop splitting:
  - When the best candidate split at a node reduces the impurity by less than a threshold.
    - How to set this threshold?
      - Stop when node has a certain number of observations.
  - When all observations in a node belong to the same class.
- Tradeoff between tree complexity vs. test set accuracy.

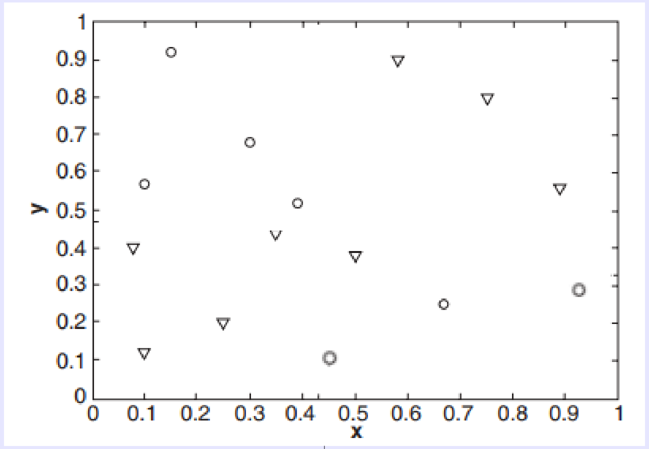
- Pruning: Two approaches:
  - Prepruning: Halt growth of tree based on some constraint (e.g., gain in impurity < threshold).
    - + : Shorter trees.
    - - : When to stop?
  - Post-pruning: Grow tree to maximum size, then trim (e.g., replace subtree with new leaf node whose class label is determined from majority class of records affiliated with the subtree.)
    - + : Gives better results than prepruning since we have benefit of the fully grown tree.
    - - : Wasted compute cycles in constructing the subtree if we have to eventually prune it.

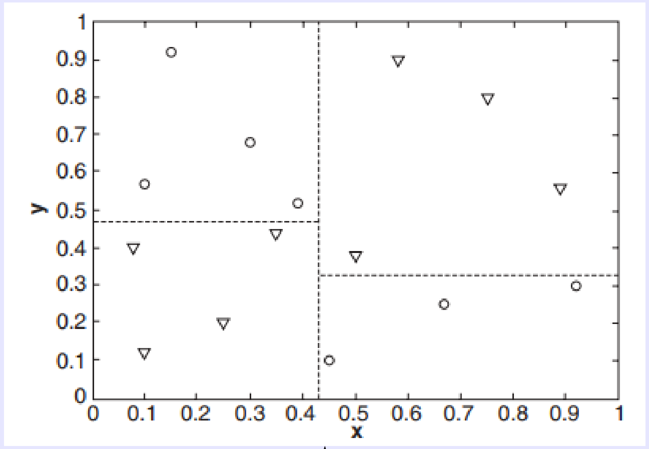
Decision Trees: Characteristics

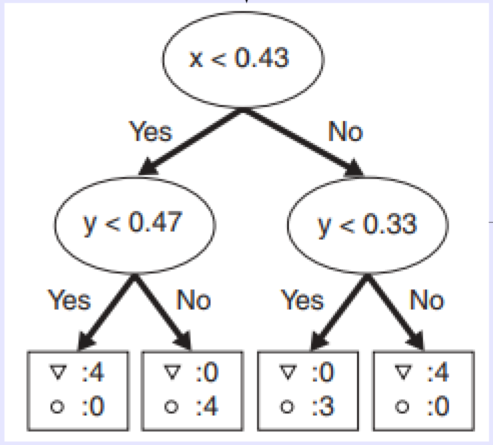
- Non-parametric approach for classification.
- Finding an optimal decision tree is NP-complete.
- Building a tree is computationally inexpensive; using it is *O(log n)*, where *n* is number of nodes in the tree.
- Multicollinearity does not affect accuracy (though it will affect the height).

Decision Trees: Characteristics

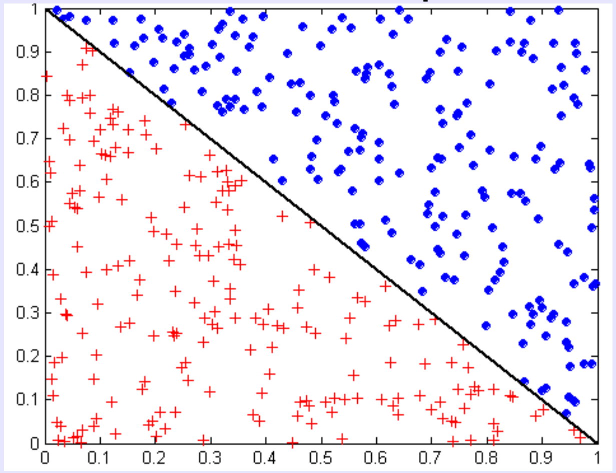
- Rectilinear decision boundaries.







Not possible



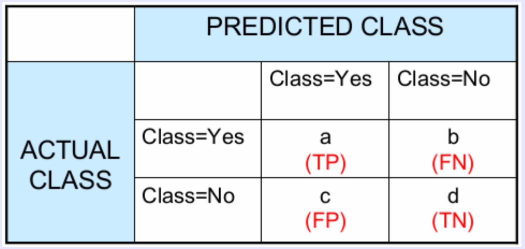
Model evaluation metrics

- Model = a particular classifier trained on a dataset, or informally, the *target function*.
- How to evaluate the performance of a model?
  - Confusion matrix: widely used measure.
    - Provides numerous metrics computed from the matrix (TPR, TNR, PPR)

      Positive Predicted Value
  - Receiver Operating Characteristics (ROC) curve.
    - Characterize the trade-off between positive hits and false alarms
- Other performance metrics exist as well, but these are the most commonly used when evaluating model performance.
- Remember: Focus on the predictive capability of the model, not how long it takes to train (mostly offline), how fast it takes to classify (it shouldn't be too slow, of course).

Model evaluation metrics

Confusion Matrix:



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

From book; different format but same information as one on right.

	Actual Class		
Predicted Class		Class = Yes	Class = No
	Class = Yes	TP	FP
	Class = No	FN	TN

Classification accuracy: $$\frac{(TP+TN)}{(TP+TN+FP+FN)}$$

Error rate: $$\frac{(FP+FN)}{(TP+TN+FP+FN)}$$

TPR (sensitivity, hit rate, recall): $$\frac{TP}{\underbrace{TP+FN}_{\text{All actual positive observations in the test set}}}$$

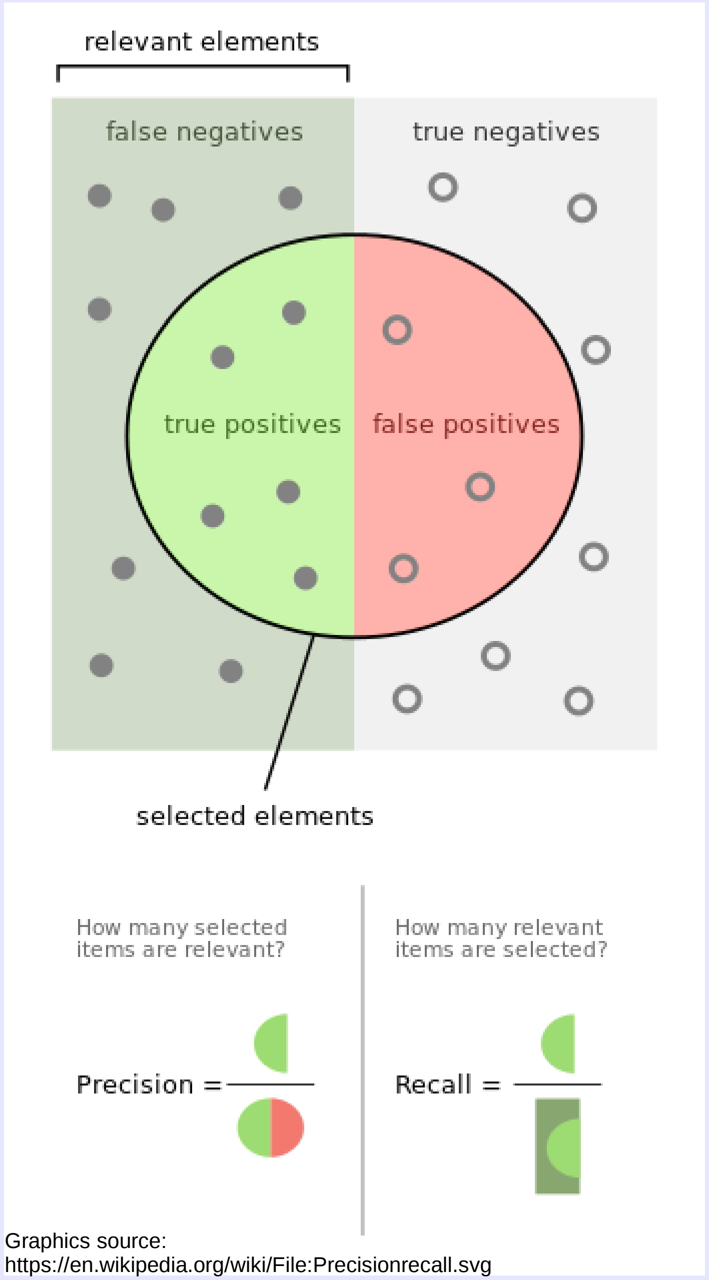
How many relevant items are selected?

TNR (specificity): $$\frac{TN}{\underbrace{TN+FP}_{\text{All actual negative observations in the test set}}}$$

PPV (precision): $$\frac{TP}{TP+FP}$$

How many selected items are relevant?

Model evaluation metrics



Graphics source: https://en.wikipedia.org/wiki/File:Precisionrecall.svg

TPR (sensitivity, hit rate, recall): $\underbrace{\frac{TP}{TP+FN}}_{\text{All actual positive observations in the test set}}$

How many relevant items are selected

TNR (specificity): $\underbrace{\frac{TN}{TN+FP}}_{\text{All actual negative observations in the test set}}$

PPV (precision): $\frac{TP}{TP+FP}$

How many selected items are relevant

		Actual Class	
		Class = Yes	Class = No
Predicted Class	Class = Yes	TP	FP
	Class = No	FN	TN

Model evaluation metrics

- Receiver Operation Characteristics (ROC) Curve
  - Developed in 1950s for signal detection theory to analyze noisy signals
    - Characterize the trade-off between positive hits and false alarms
- Performance of classifier represented as a point on the curve:
  - Curve can be changed by varying some parameter of classification.

Model evaluation metrics

(TPR=1,FPR=0): Ideal model

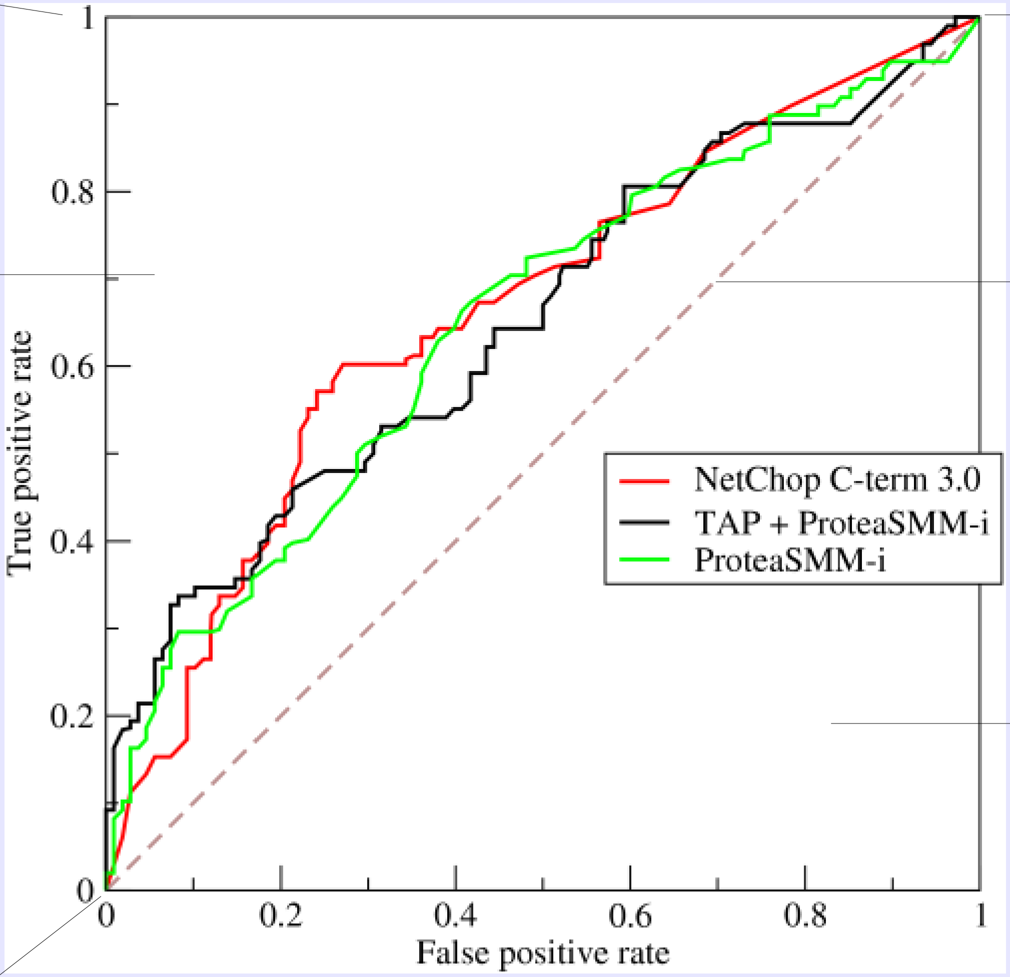
(TPR=1, FPR=1): model predicts every instance as positive class

Good area to be in for the model!

Random line (model should do better than this)

Bad area to be in for the model!

(TPR=0, FPR=0): model predicts every instance as negative class

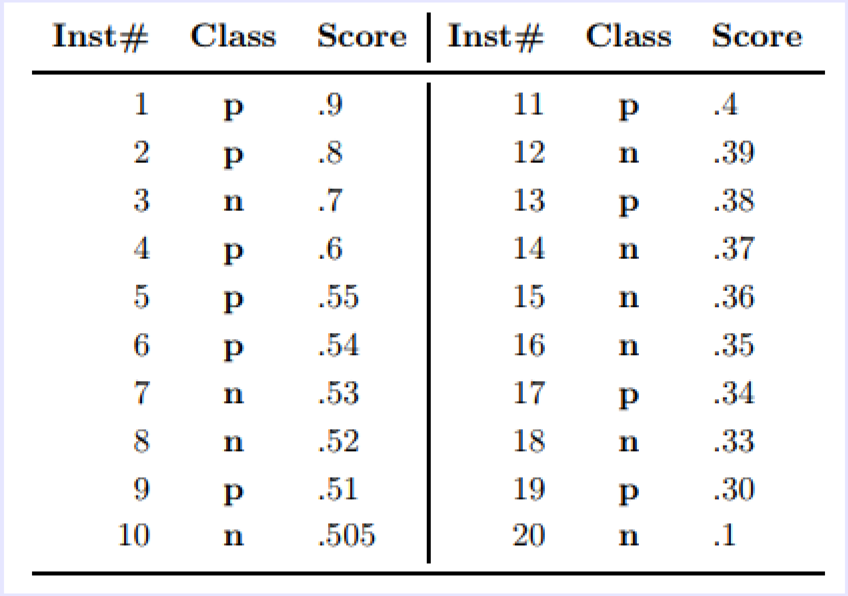


Graphic source: Wikipedia

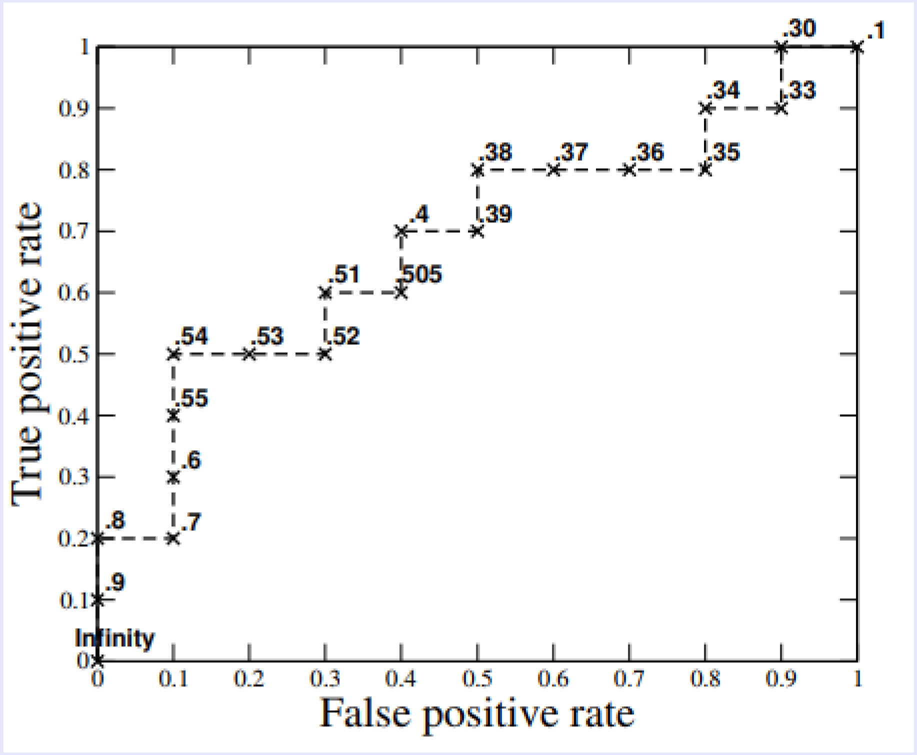
Which model is better? AUC.

Model evaluation metrics

How to calculate a ROC curve?



Inst#	Class	Score	Inst#	Class	Score
1	p	.9	11	p	.4
2	p	.8	12	n	.39
3	n	.7	13	p	.38
4	p	.6	14	n	.37
5	p	.55	15	n	.36
6	p	.54	16	n	.35
7	n	.53	17	p	.34
8	n	.52	18	n	.33
9	p	.51	19	p	.30
10	n	.505	20	n	.1

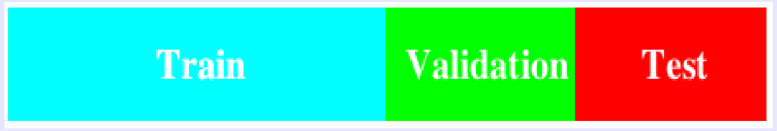


Source: Tom Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers (2004)"
Online at http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.9777

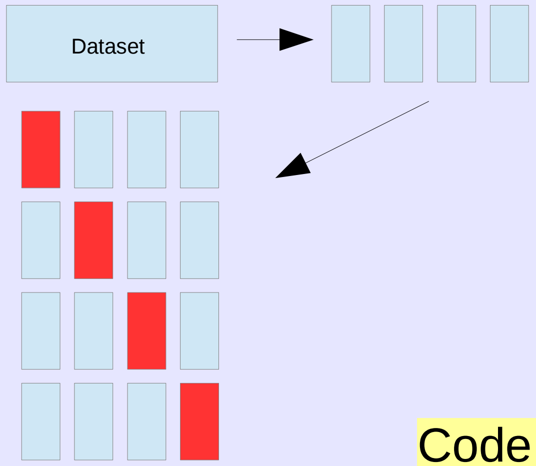
General approach to solving a classification problem

1) Get data.
2) Clean data, impute missing values, transform if needed, etc.
3) Do some exploratory analysis on the dataset (including visualization to gain familiarity for the dataset).
4) Choose a specific classifier (or a set of them).
5) Separate data into training and testing (sometimes validation). **NEVER MIX THESE!**
6) Create a model on training data (in-sample data).
7) Run model on test data (out of sample data).
8) Evaluate results.
    1) Good results? Relax, have a beer, go out to dinner, make a lot of money.
    2) No? Tune classifier or use different classifier, or get more data, or all of the above, and go to step 5.
9) Update model as and if needed. (Concept drift).

- How to split?
  - Holdout method.



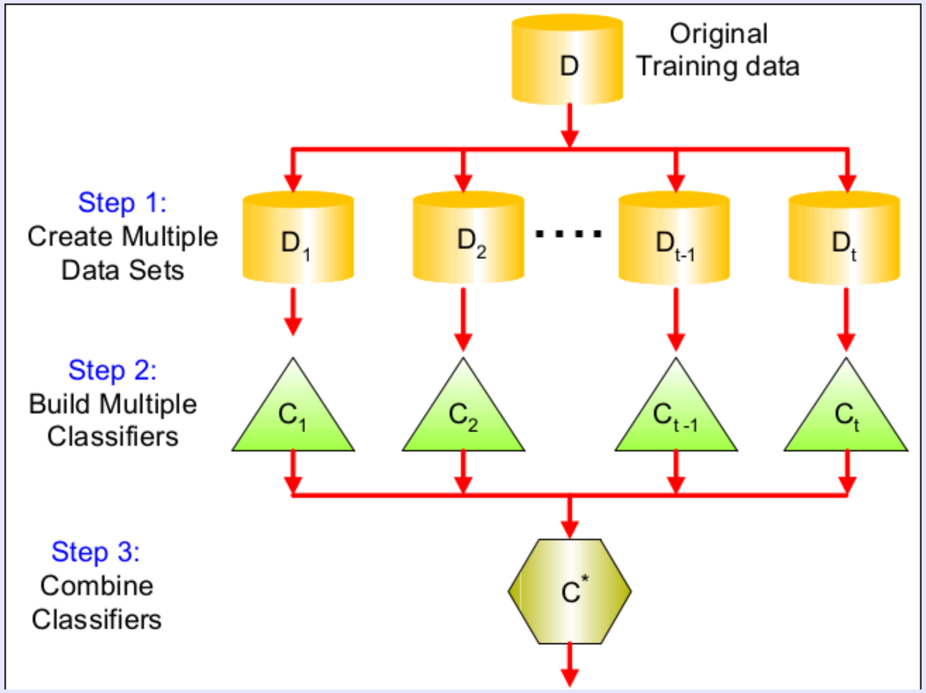
  - Random sub-sampling: Repeat holdout method several times and average accuracy over all runs.
  - Cross-validation



Code: diabetes.Rmd

Ensemble methods

- So far, we induce one classifier from training data.
- But, is *wisdom of the crowds* better?
- What if we created multiple classifiers and combined their prediction.
  - Will we get better results?



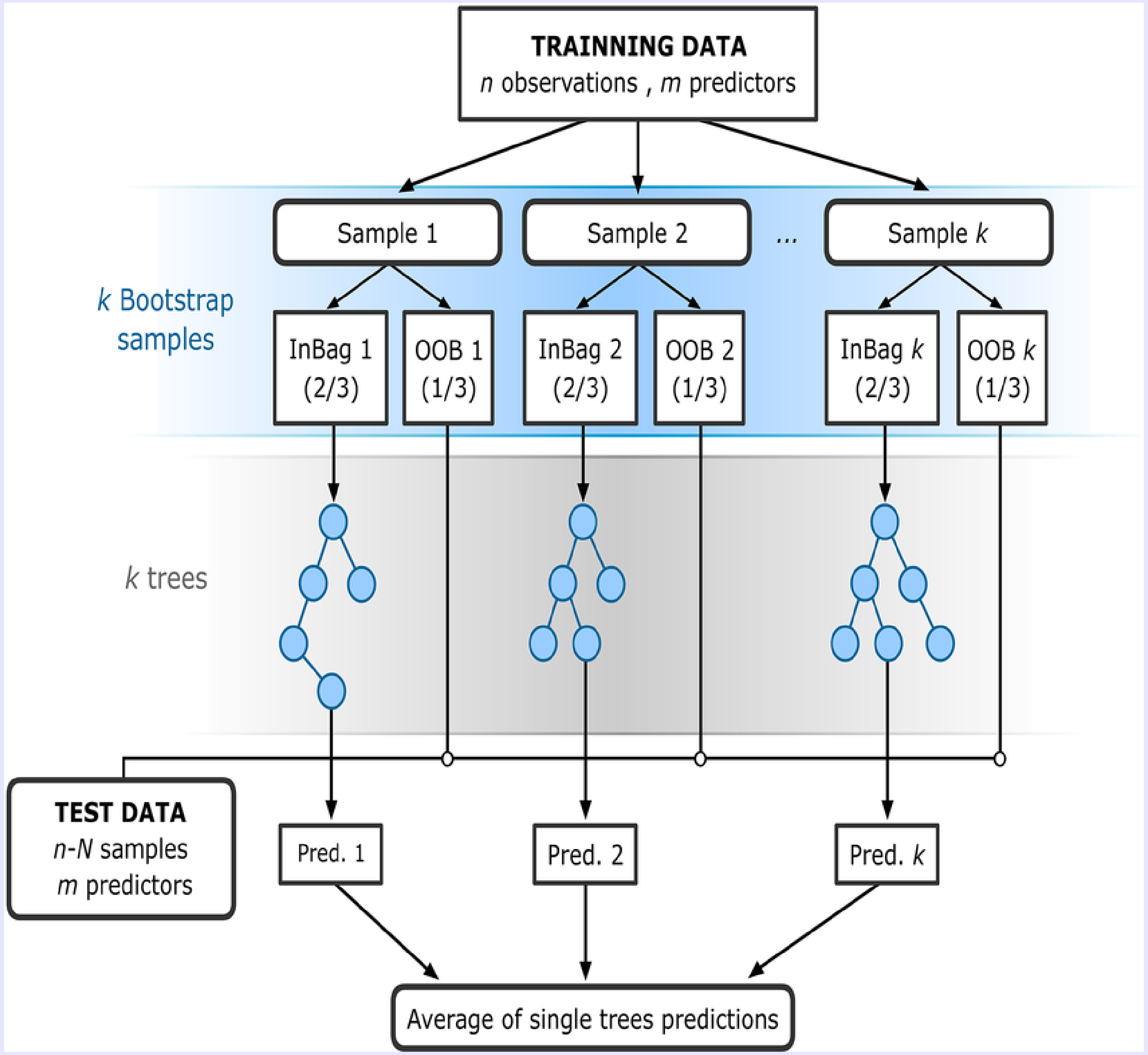
Original Training data
D
Step 1: Create Multiple Data Sets
D_1
D_2
....
D_{t-1}
D_t
Step 2: Build Multiple Classifiers
C_1
C_2
C_{t-1}
C_t
Step 3: Combine Classifiers
C^*

Ensemble methods

- The ensemble can be created in multiple ways.
  - Manipulate the training dataset.
  - Manipulate input features.
  - Manipulate class labels.
  - Manipulate learning algorithm.
- Can work effectively well (for some datasets).
  - Suppose we have 25 base classifiers, each of which has an error rate of ε = 0.35.
  - All classifiers are independent.
  - Ensemble makes the wrong prediction only if > ½ of base classifiers predict incorrectly:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

Ensemble methods: Random Forest



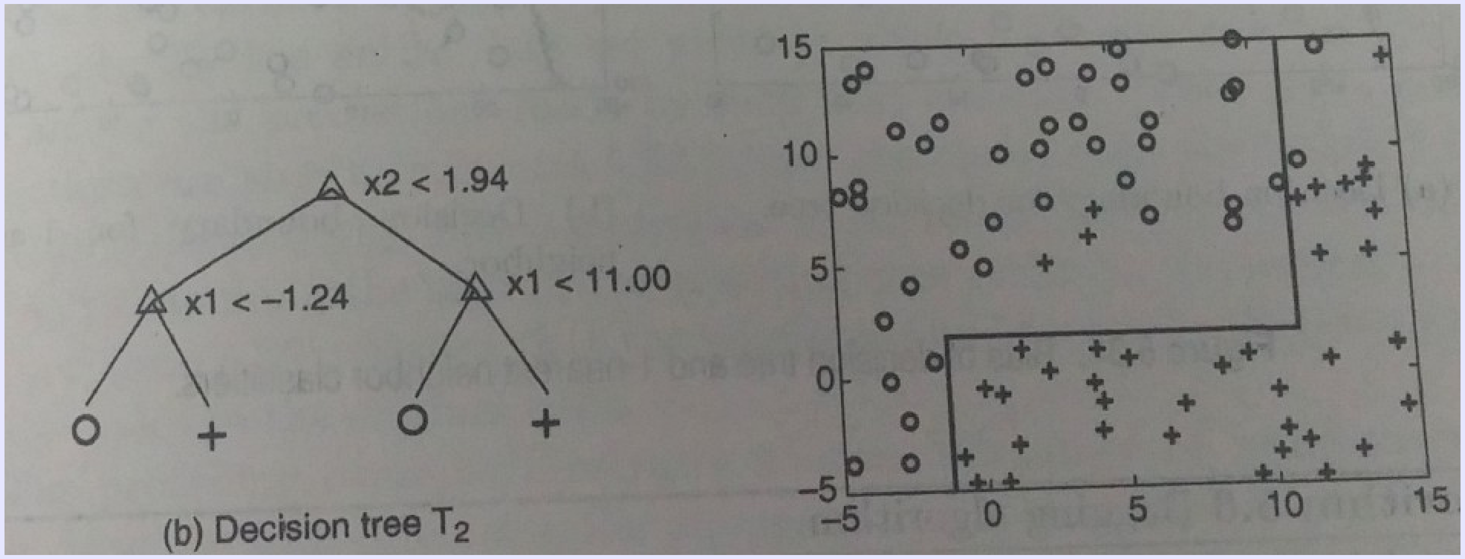
Modelling interannual variation in the spring and autumn land surface phenology of the European forest, Rodriguez-Galiano et al., 2016, Biogeosciences.

Ensemble methods: Random Forest

- Final word of wisdom: Ensemble methods are not universally better than their normal counterparts. For the ensemble method to be better, the normal classifier should demonstrate some *instability*.
  - Define *instability* as inappropriate sensitivity to input

Code: german-credit.Rmd

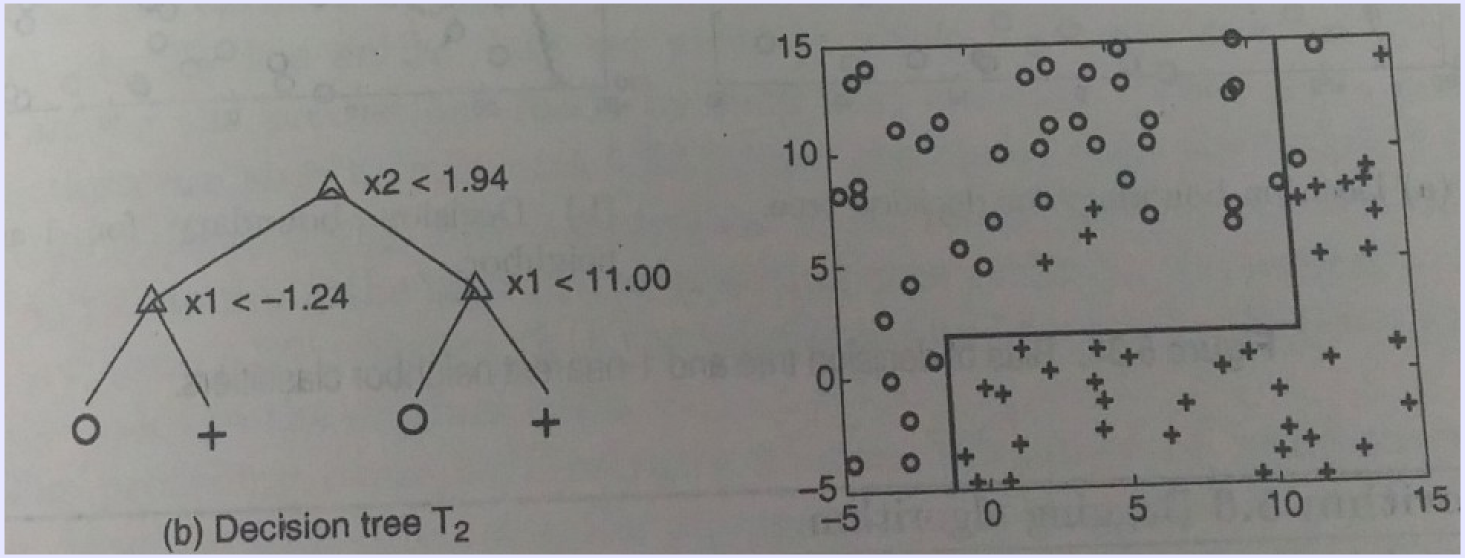
Bias, Variance, Under/Over-fitting



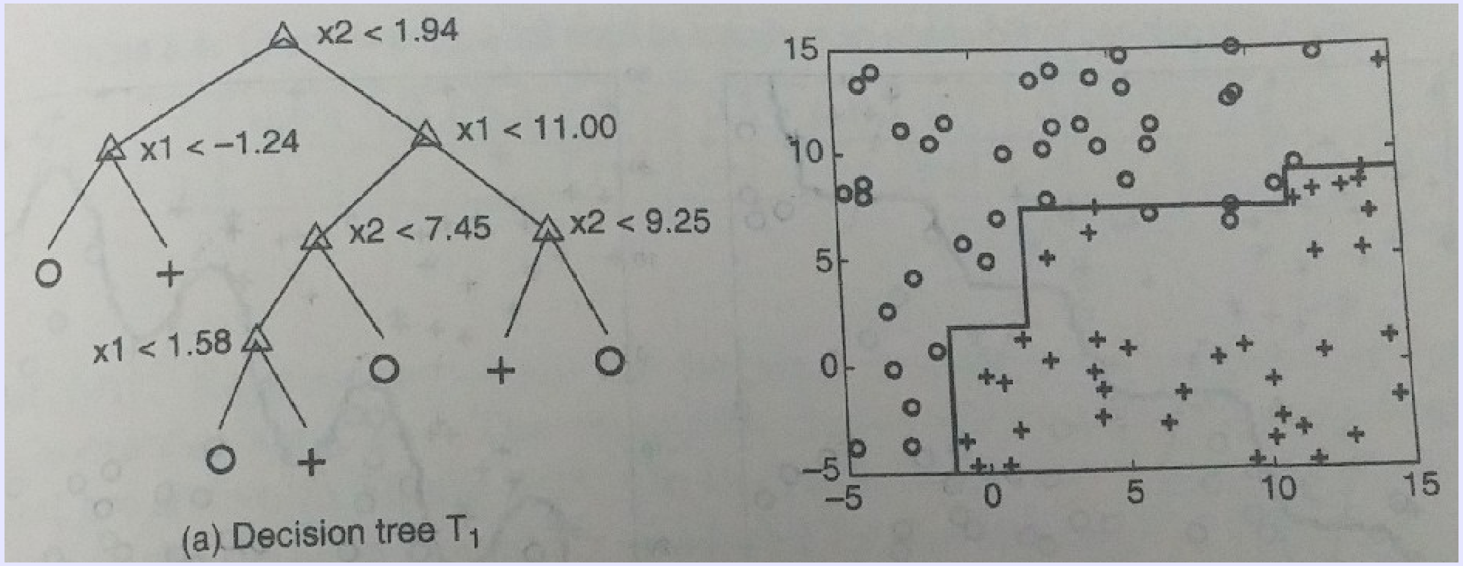
(b) Decision tree T_2

Bias, Variance, Under/Over-fitting

High Bias

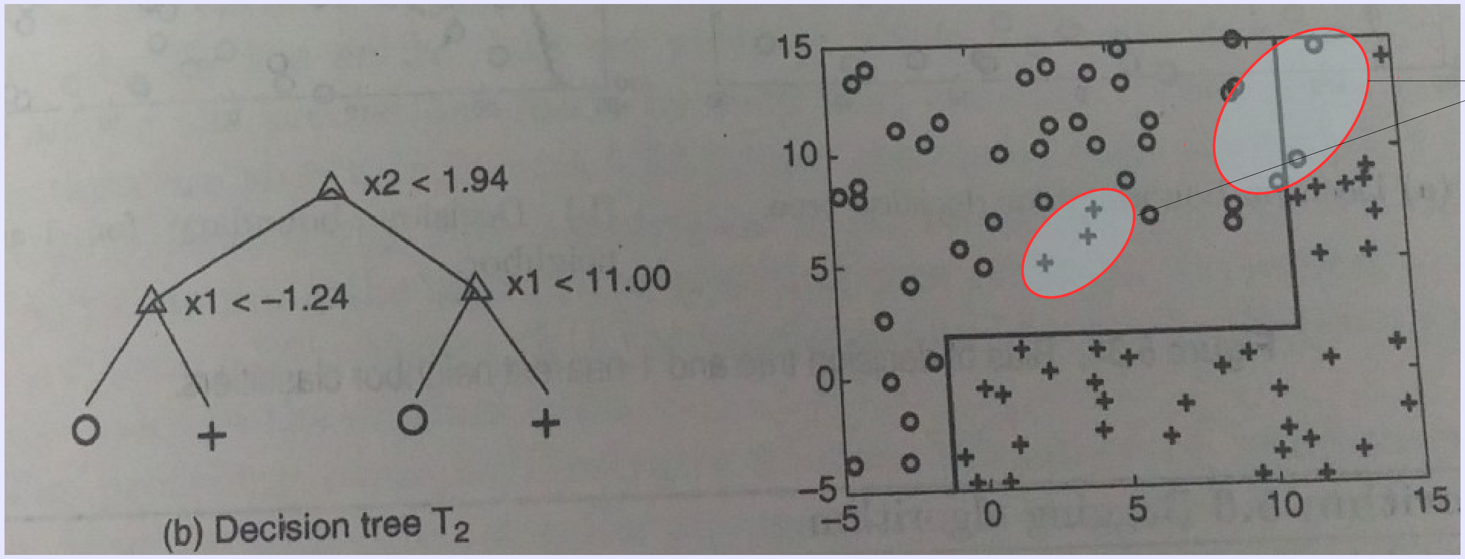


Low Bias

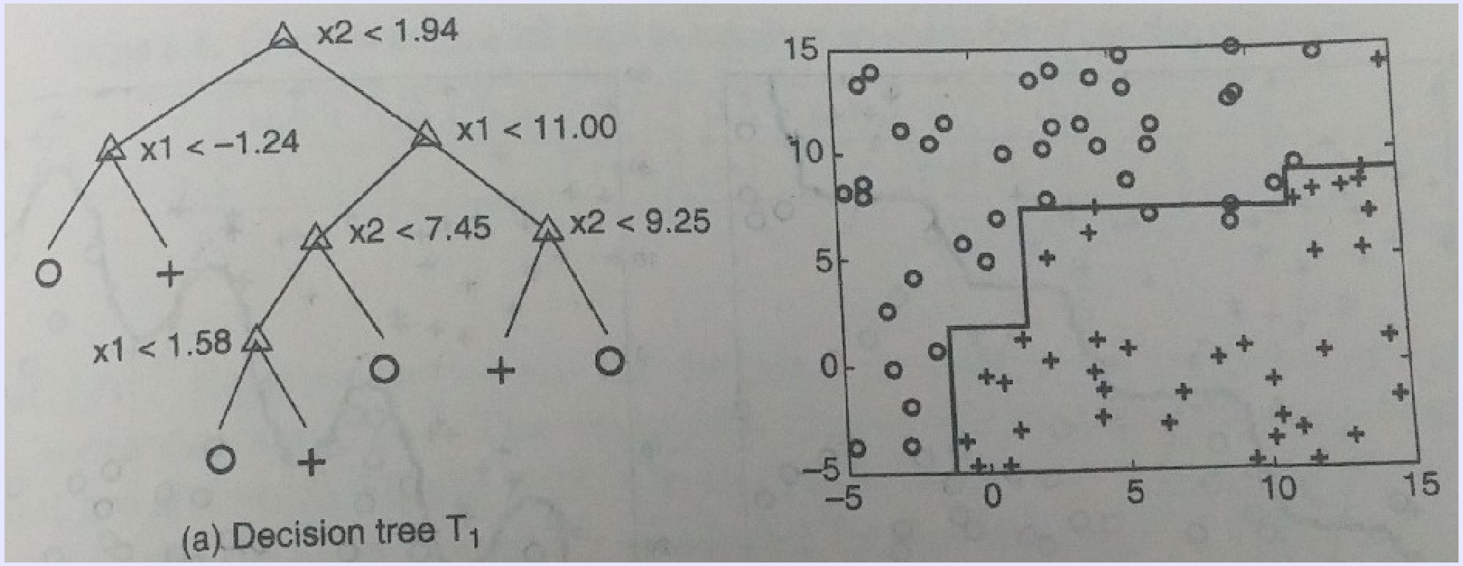


Bias, Variance, Under/Over-fitting

High Bias



Low Bias

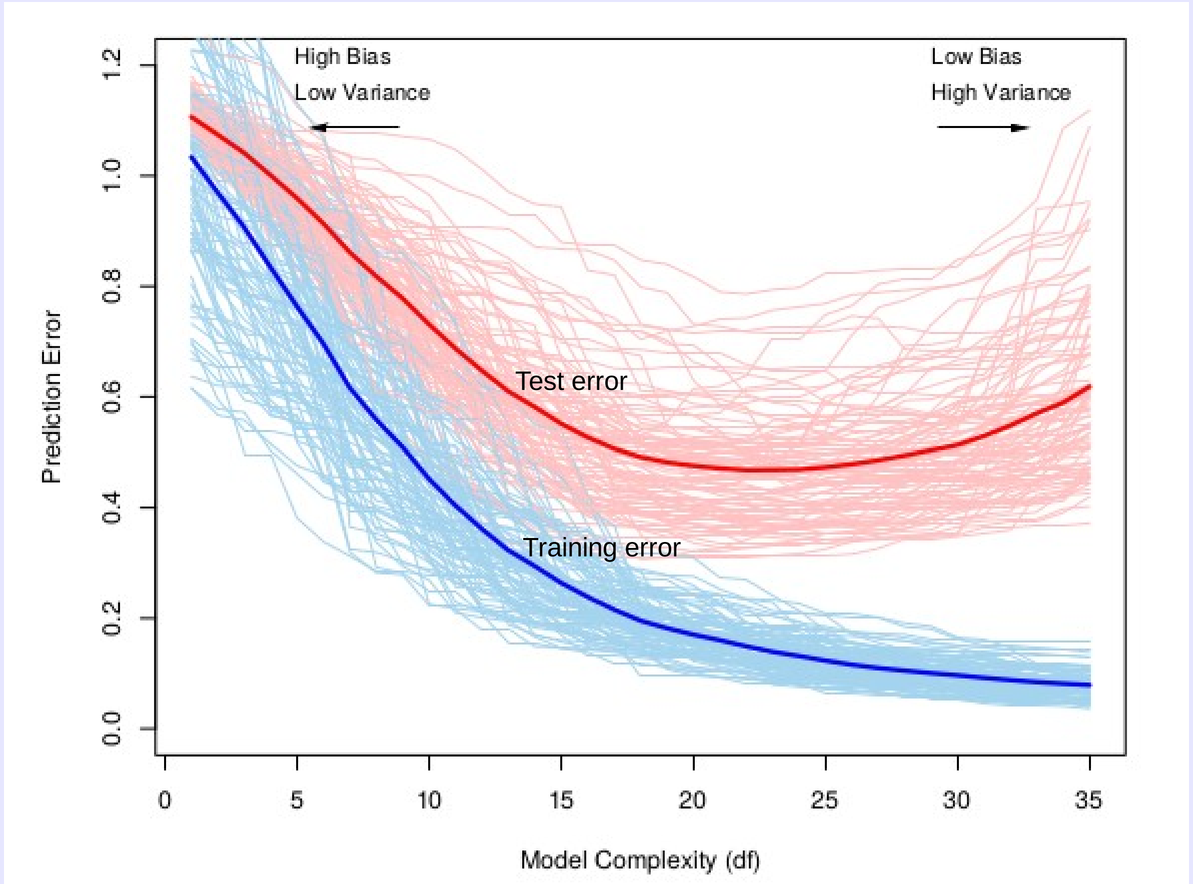


Overfit?

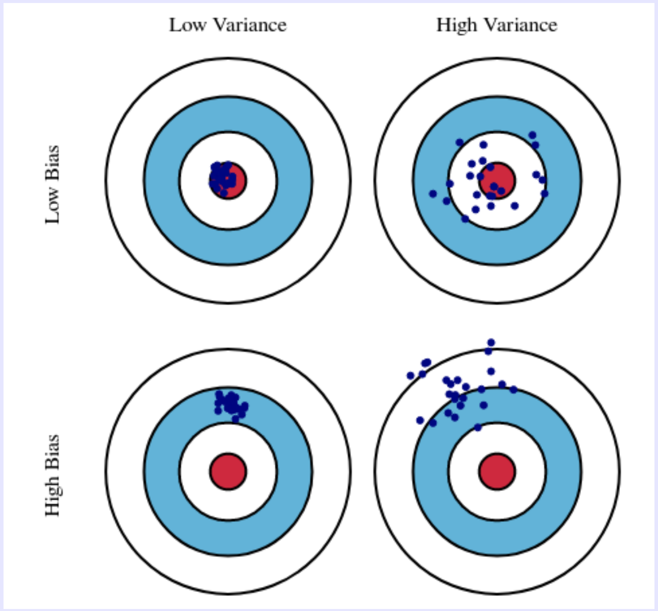
Bias, Variance, Under/Over-fitting

- Variance, on the other hand, is how your model reacts to changes in the training data.
- As the model becomes more complex, it picks up patterns in the training data, thus making it less generalizable.
  - A different training dataset (drawn from the same distribution) may induce results that vary from before.

Bias, Variance, Under/Over-fitting



Source: The Elements of Statistical Learning, 2e



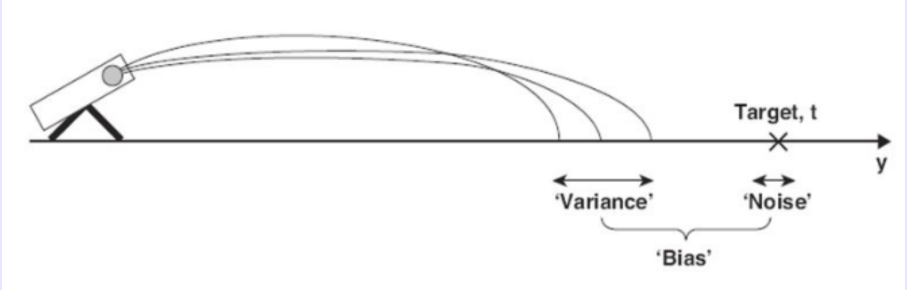


Figure 5.32. Bias-variance decomposition.

A model with many predictive attributes will exhibit **low bias, high variance**.
A model with too few predictive attributes will exhibit **low variance, but may be quite biased**.

Informally: **bias** is how far a model's predictions are from target (underfitting), and **variance** is the degree to which these predictions vary between model iterations (overfitting).

Bias, Variance, Under/Over-fitting

A model with many predictive attributes will exhibit **low bias, high variance**. A model with too few predictive attributes will exhibit **low variance, but may be quite biased**.

- Mitigation approaches
  - Dimensionality reduction and feature selection can reduce variance by simplifying models.
  - If you cannot reduce dimensions or engage in feature selection, a larger training set may decrease variance.
  - Adding features decreases bias, at the expense of additional variance.

Class Imbalance

- Datasets with **balanced** class distribution are the exception as most (all) datasets have a imbalanced class distribution.
  - Medical
  - Manufacturing
- Correct classification of the rare (minority) class has greater value than correct classification of the majority class.
- Imbalanced classes present a number of problems to classification algorithms:
  - Accuracy is no longer a reasonable measurement.
  - Balanced accuracy is a better measure when the test (or training) datasets exhibit class imbalance.

Class Imbalance

- Other mitigation techniques:
  - Cost sensitive learning penalizes the model when it commits a false negative error.
  - Sampling techniques modify the class distribution such that the rare class is well represented in the training set.
    - Undersampling gathers **less** of the majority class observations for training.
      - Disadvantage: useful observations may not be part of the sample. (Can be overcome by sampling multiple times and using an ensemble method).
    - Oversampling gathers **more** of the minority class observations for training.
      - Disadvantage: If training data is noisy, oversampling may amplify the noise.
    - Hybrid approach uses both of the above techniques to arrive at a equivalent dataset.
  - Synthetic data may be generated, if possible. If so, the generation could ensure that the class distribution is equivalent.

Multi-class decision trees

- Classification extends to differentiating between multiple classes as well.
  - Code: multiclass.Rmd