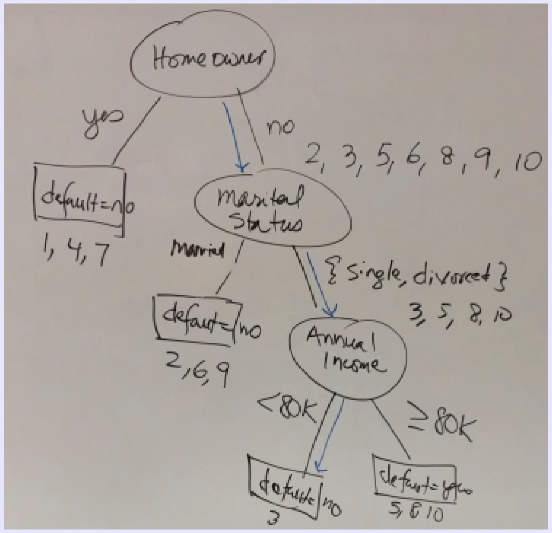
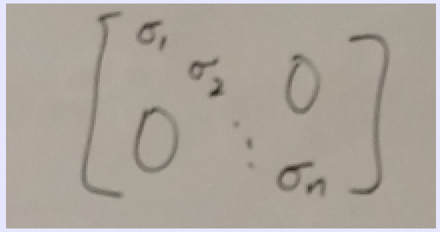
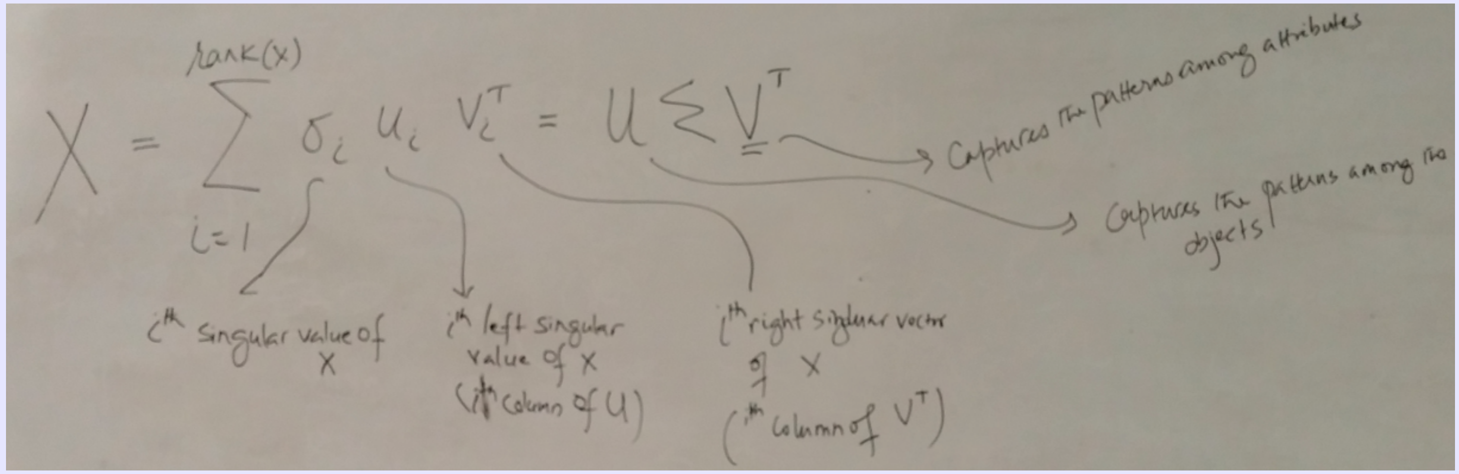
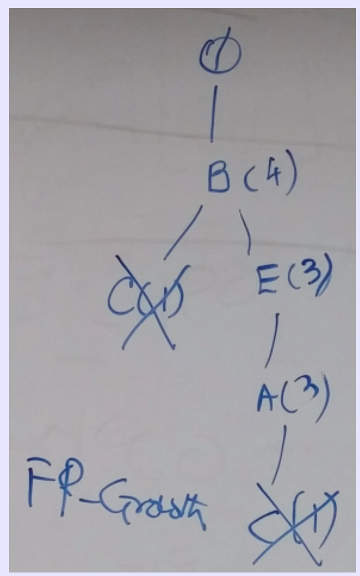
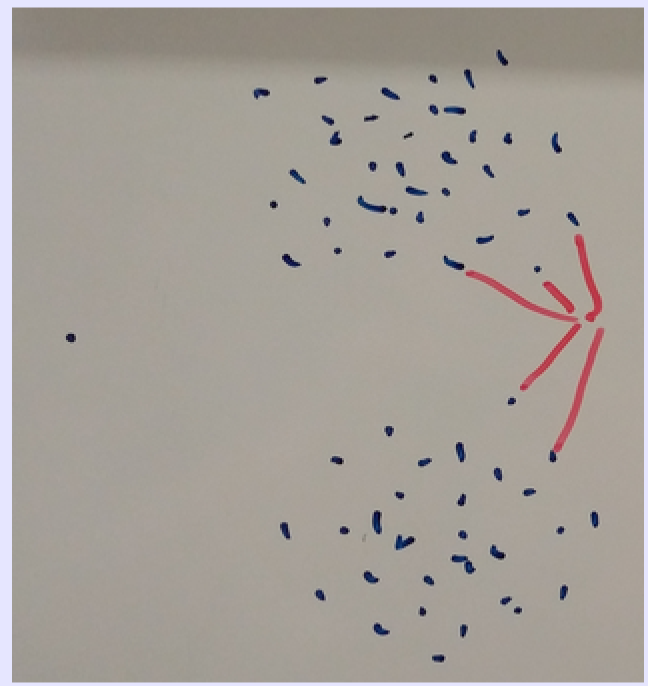










CS 422: Data Mining
Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

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

Tree Induction

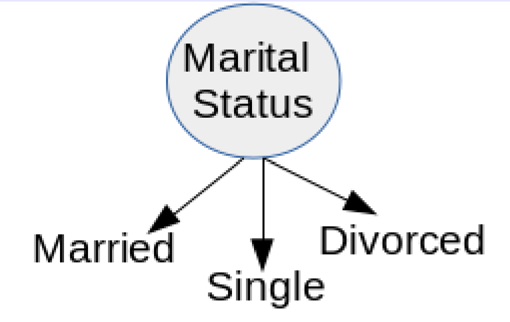
- Now that we know how to construct a Decision Tree … let's see how to split the records at each level.
- Greedy strategy.
  - Split the records based on an attribute test that optimizes certain criterion.
- Issues
  - How to split the records?
    - Specify attribute test condition
    - How to determine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

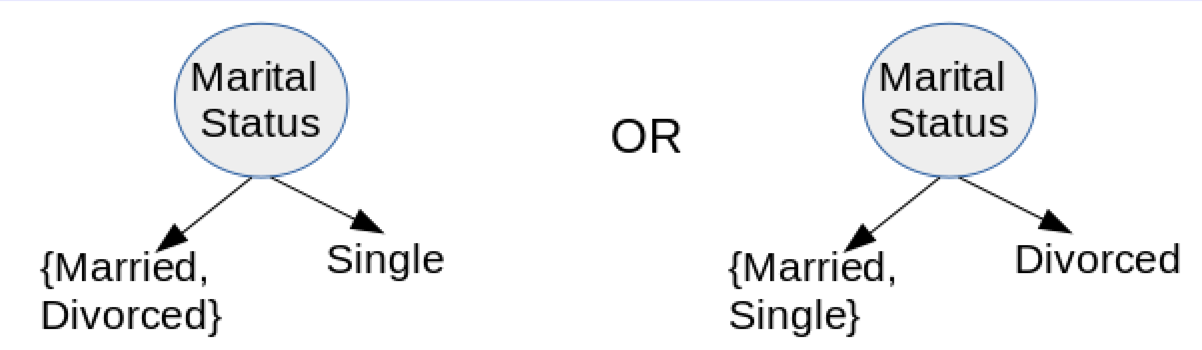
Tree Induction: Specify attribute test conditions

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 of 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).

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

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:

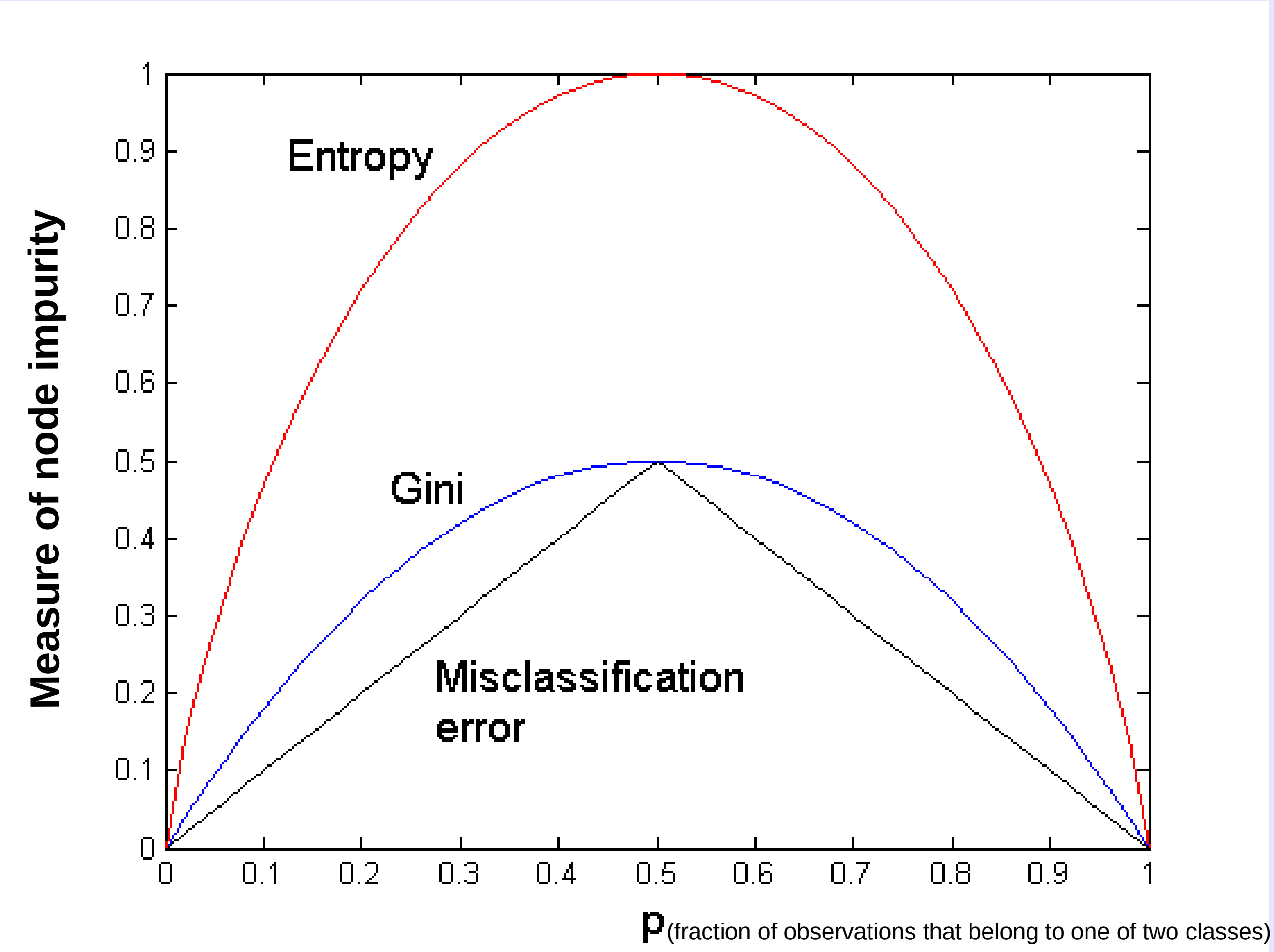
$$H(Y) = -[\frac{7}{10}\ log_2 \frac{7}{10} + \frac{3}{10}\ log_2 \frac{3}{10}] = 0.88$$

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?

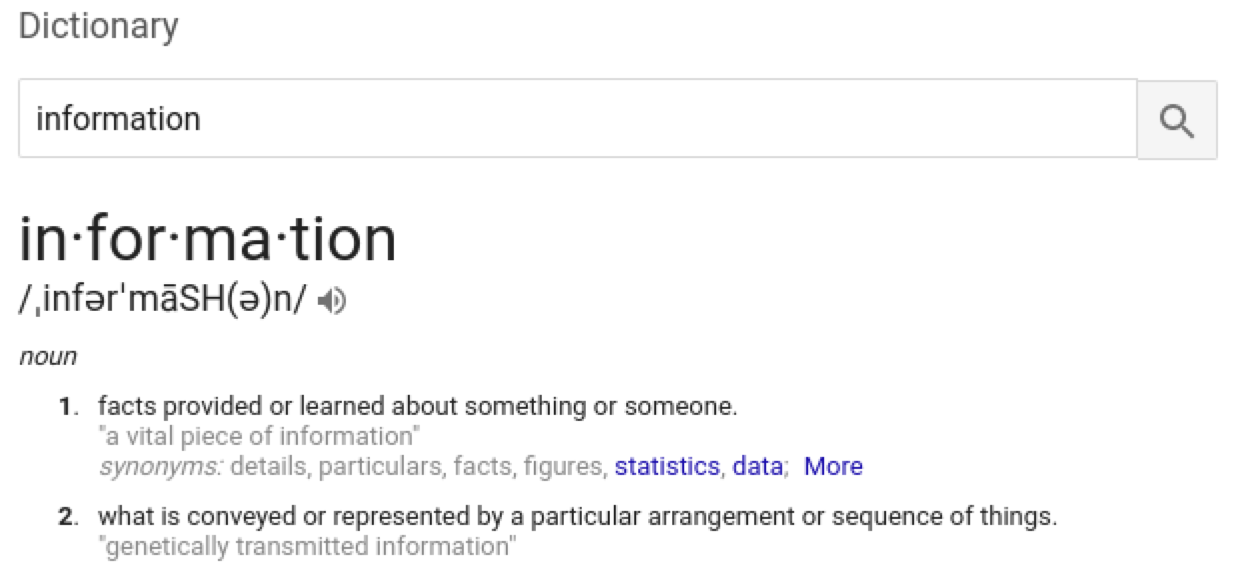
- Measures of node impurity
  - Gini index (used by CART, and *rpart*)
  - Entropy
  - Misclassification error



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.



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:

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

Question is: What do we split on now?
- Homeowner?
- Marital Status?
- Annual Income?

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:

$$H(Defaulted|Homeowner) = \frac{3}{10} * 0 + \frac{7}{10} * 0.99 = 0.69$$

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:

$$H(Defaulted|Homeowner) = \frac{3}{10} * 0 + \frac{7}{10} * 0.99 = 0.69$$

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

Entropy: 1

Married	
Yes	0
No	4

Entropy: 0

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

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

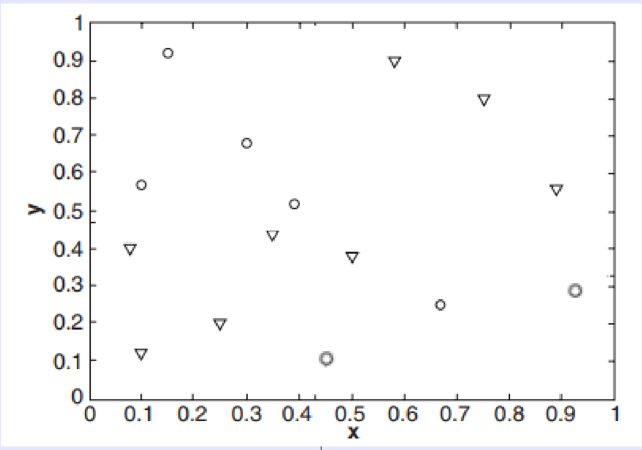
$$H(Defaulted|Income) = \frac{7}{10} * 0.99 + \frac{3}{10} * 0.00 = 0.69$$

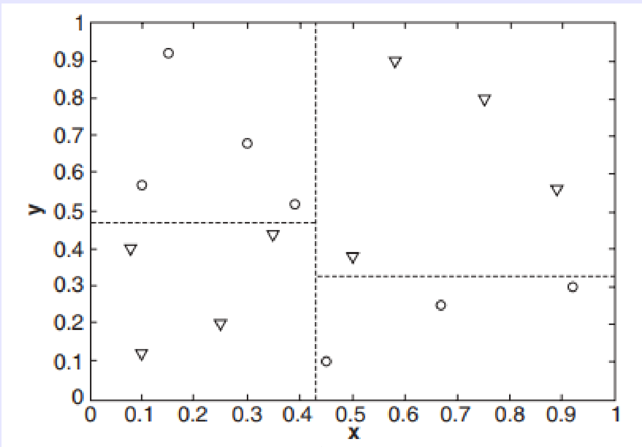
Decision Trees: Characteristics

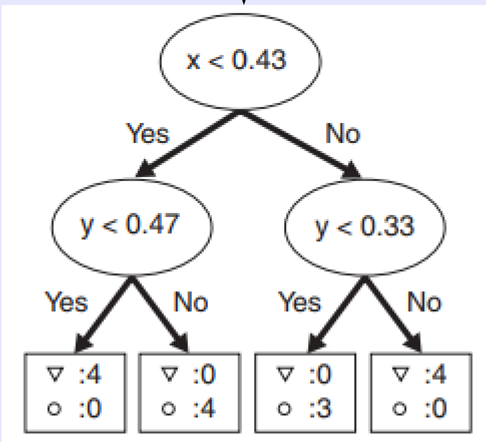
- Non-parametric approach for classification.
- Finding an optimal decision tree is NP-complete.
- Building a tree is computationally inexpensive; using it is *O(log n)*, where *n* is number of nodes in the tree.
- Multicollinearity does not affect accuracy (though it will affect the height).

Decision Trees: Characteristics

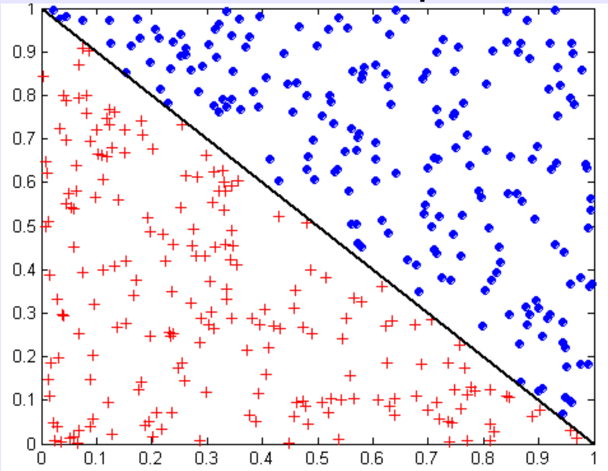
- Rectilinear decision boundaries.







Not possible



Model selection

- Remember: model = algorithm + hypothesis
- When we select a model, we evaluate all steps in the procedure:
  - Preparing the training data;
  - Choose a hypothesis set and an algorithm;
  - Tune the algorithm
  - Train the model, fit the model to out of sample data (test set) and evaluate results.

Model selection

- Assume you have two models (regression and neural networks). Important question: Which model performs better on your dataset?
- Such evaluation metrics are motivated by two fundamental problems:
  - Model checking
    - All algorithms have hyper-parameters
      - *k* in kNN
      - Weights, network size, …
    - How do we select the optimal parameters for the model?
  - Performance estimation
    - How should we evaluate a model's *goodness of fit*?
- Goal of model selection: Select the **best** model from training phase.
- Problem: How do we define *best*? (The best model is also called the *final* model.)

Model selection

- Assume you have two models (regression and neural networks). Important question: Which model performs better on your dataset?
- Such evaluation metrics are motivated by two fundamental problems:
  - Model checking
    - All algorithms have hyper-parameters
      - *k* in kNN
      - Weights, network size, …
    - How do we select the optimal parameters for the model?
  - Performance estimation
    - How should we evaluate a model's *goodness of fit*?
- Goal of model selection: Select the **best** model from training phase.
- Problem: How do we define *best*? (The best model is also called the *final* model.)
- **The best model is the one that gives you the smallest prediction error (or minimizes the loss function) on the training set and generalizes well on the testing set.**

Model checking

- Given a dataset, you divide it into training and testing dataset.
  - Model is trained on the training dataset and evaluated on the test dataset.
- Problem with this?

Model checking

- Given a dataset, you divide it into training and testing dataset.
  - Model is trained on the training dataset and evaluated on the test dataset.
- Problem with this?
  - What happens if you randomly select test points that are not representative of the population in general?
- Solution: Cross validation.

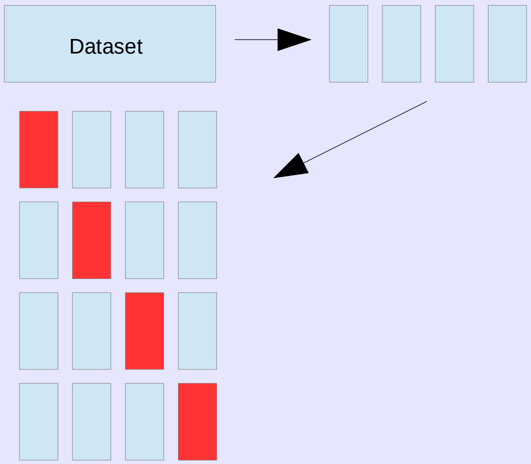
Model checking

- Solution: Cross validation --- an approach to systematically create and evaluate multiple models on multiple subsets of the dataset.
- Cross validation approaches:
  - Holdout method.
  - *k*-fold cross validation.
  - Leave one out cross-validation (LOOC).

Model checking

- *k*-fold cross validation.

  Split data into *k* chunks, train on *k-1* chunks and test on the k^{th} chunk. Do this *k* times and calculate average error.



- LOOC: extreme version of *k*-fold, where *k* = 1. (1 observation, not chunk!)

  for all *i* in *{1..n}*
     train model on every point except *i*
     compute the test error at the held out point *i*
  average the test errors

Model checking

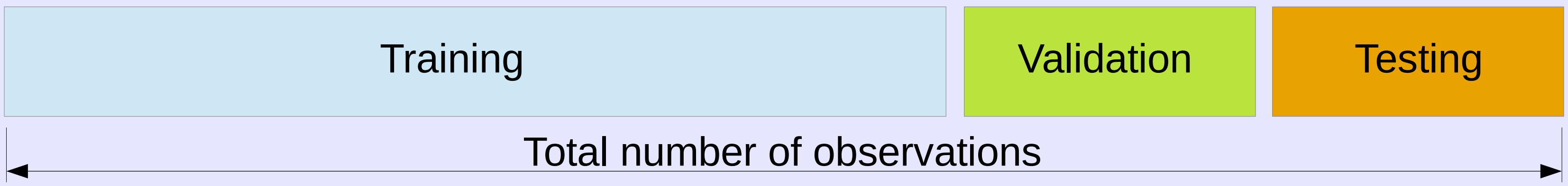
Example:

Training Set		
Train	Train	Test
Train	Test	Train
Test	Train	Train

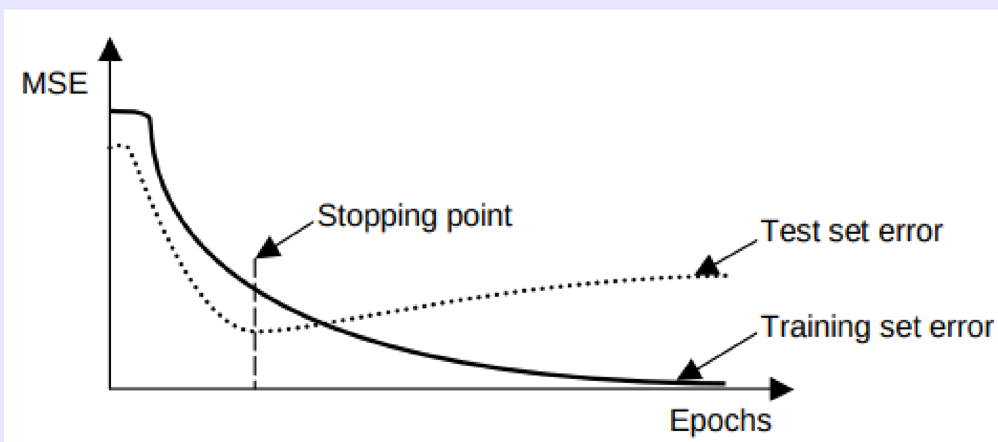
Model 1	Model 2	Model 3
Err = 0.1	Err = 0.4	Err = 0.4
Err = 0.4	Err = 0.5	Err = 1.2
Err = 0.3	Err = 0.8	Err = 0.6
Avg. Err 0.27	Avg. Err 0.57	Avg. Err 0.73

Model checking

- Use of *k*-fold or LOOC resampling methods more robust if the dataset is split into three parts:



- Typical application of these holdout methods is determining a stopping point with respect to error



Graphic source: Ricardo Gutierrez-Osuna, Wright State University

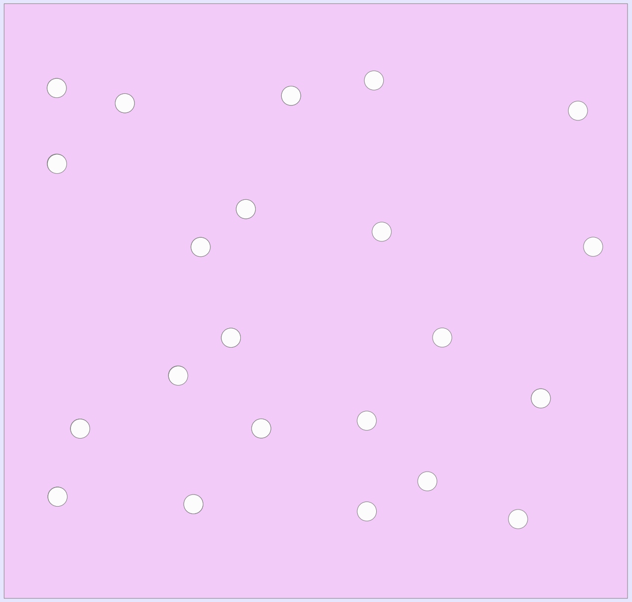
Performance estimation

- 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).

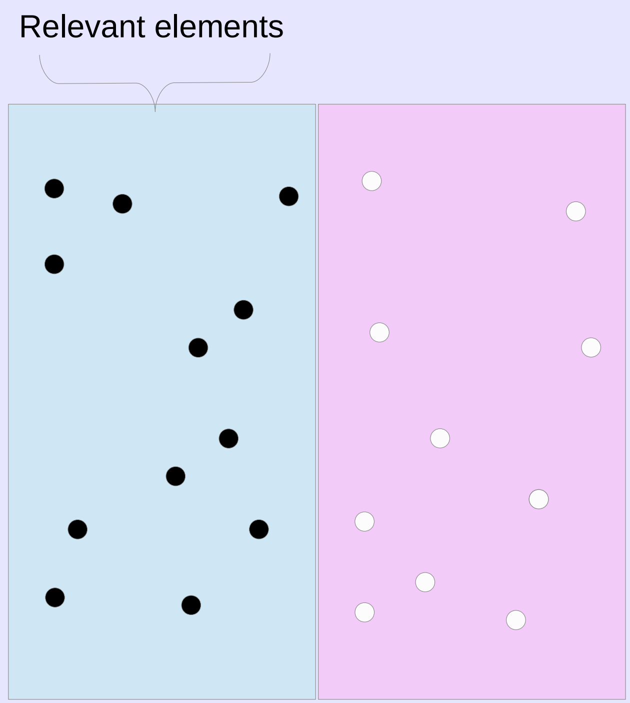
Performance estimation

- Population



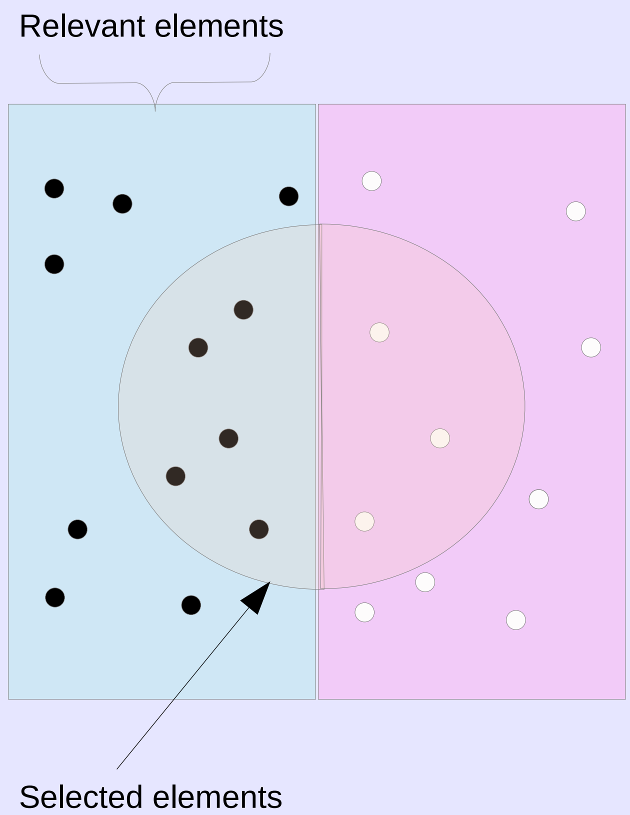
Performance estimation

- Population
- Samples in black are relevant elements



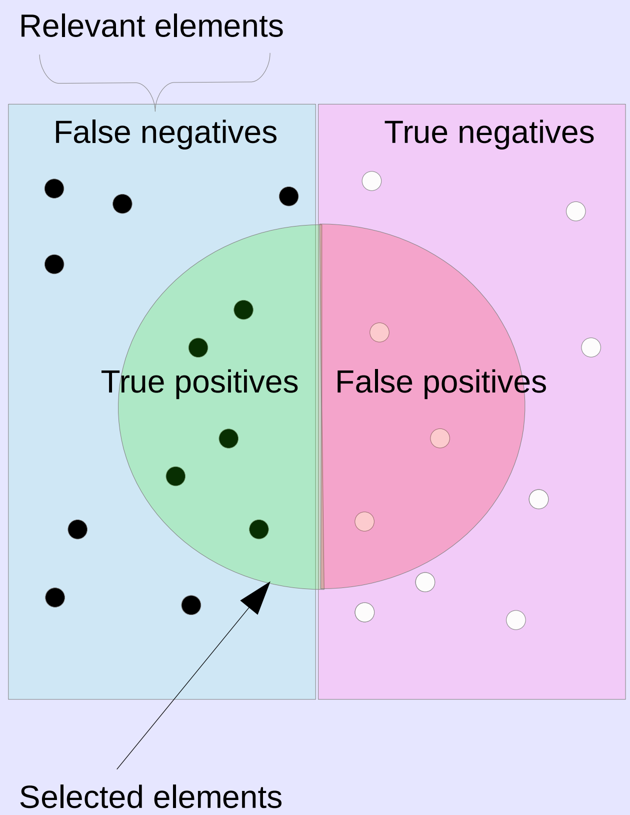
Performance estimation

- Population
- Samples in black are relevant elements
- You do a search and select elements



Performance estimation

- Population
- Samples in black are relevant elements
- You do a search and select elements



Performance estimation

Confusion Matrix:

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

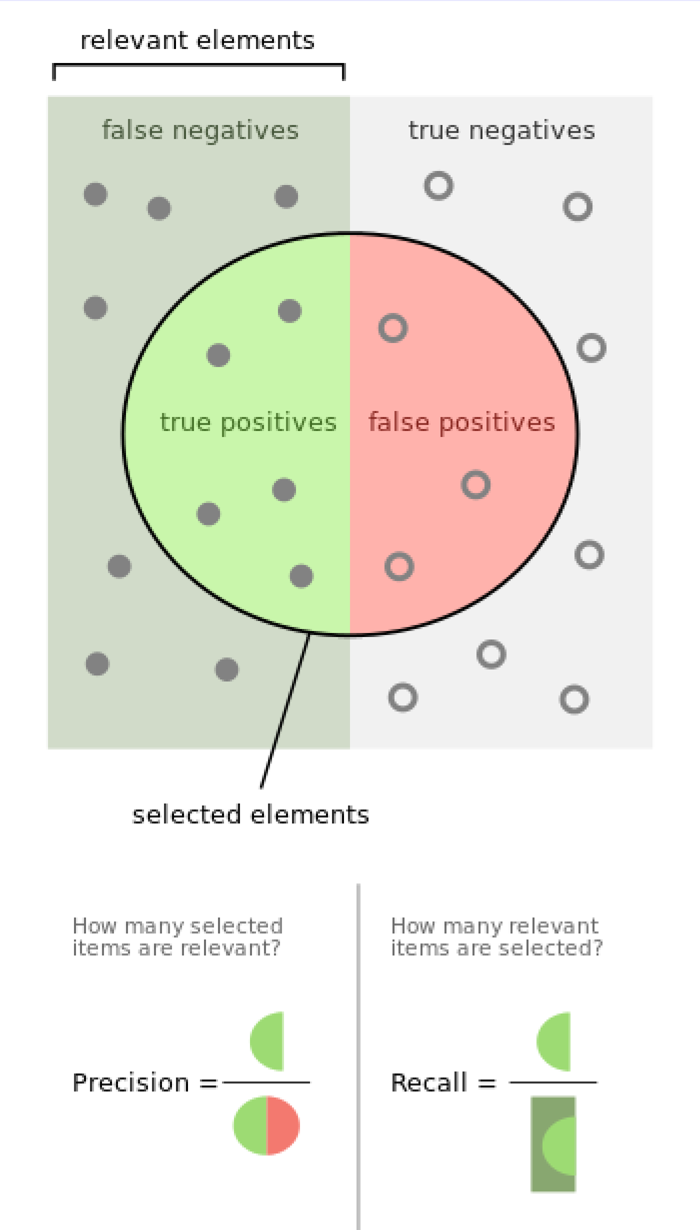
	PREDICTED CLASS		
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

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

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

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

Performance estimation



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

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

How many relevant items are selected

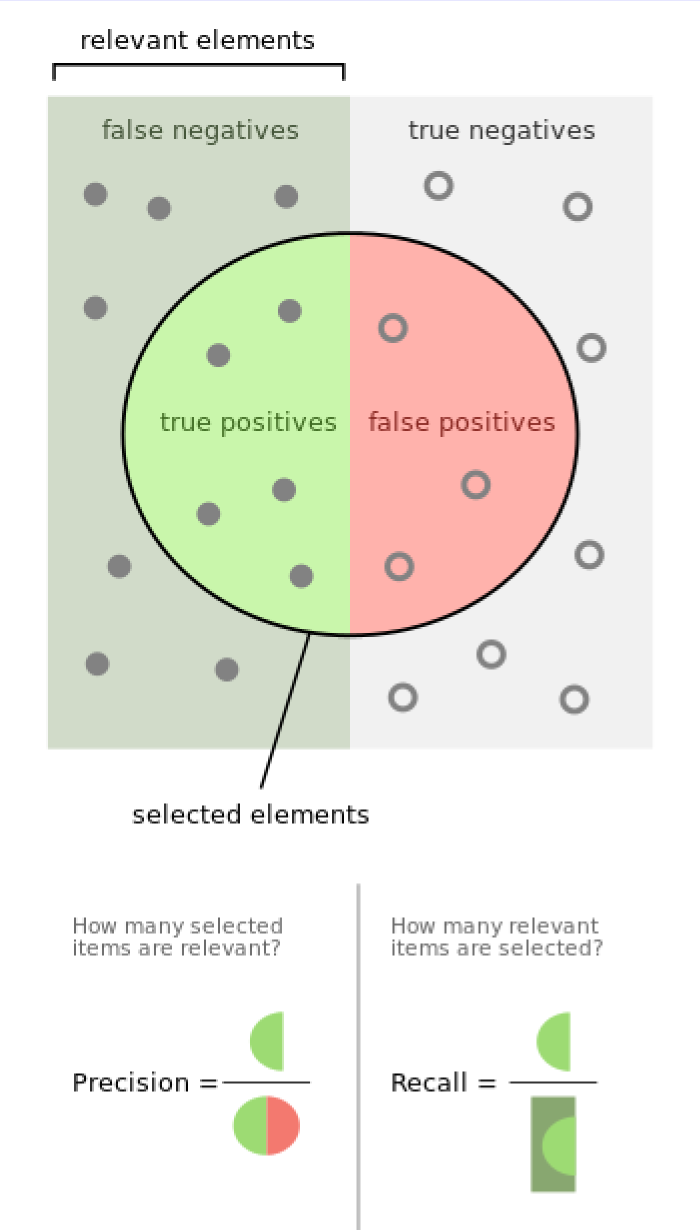
TNR (specificity): $\underbrace{\dfrac{TN}{TN+FP}}_{\text{All actual negative observations in the test set}}$

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

How many selected items are relevant

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

Performance estimation



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

$$\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{P}} = \frac{\mathrm{FN}}{\mathrm{FN}+\mathrm{TP}} = 1 - \mathrm{TPR}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = 1 - \mathrm{TNR}$$

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

In people with confirmed COVID-19, antigen tests correctly identified COVID-19 infection in an average of 72% of people with symptoms. In people who did not have COVID-19, antigen tests correctly ruled out infection in 99.5% of people with symptoms.
(Source: https://www.cochrane.org/CD013705/INFECTN_how-accurate-are-rapid-tests-diagnosing-covid-19).

Q. Is this a good test?

TPR = 0.72, **FNR** = 1 – TPR = 1 – 0.72 = 0.280
TNR = 0.995, **FPR** = 1 – TNR = 1 – 0.995 = 0.005

Performance estimation

Why accuracy alone is not a good measure.

Consider a spam detector model.

	Actual Class			Totals
Predicted Class		Class = Yes	Class = No	
	Class = Yes	TP 0	FP 0	0
	Class = No	FN 25	TN 125	150
Totals:		25	125	150

Accuracy = ???

Performance estimation

Why accuracy alone is not a good measure.

Consider a spam detector model.

	Actual Class			Totals
		Class = Yes	Class = No	
Predicted Class	Class = Yes	TP 0	FP 0	0
	Class = No	FN 25	TN 125	150
Totals:		25	125	150

Accuracy = 125/150 = 83.3%!!
But sensitivity and precision are 0!