Machine Learning:

Classification

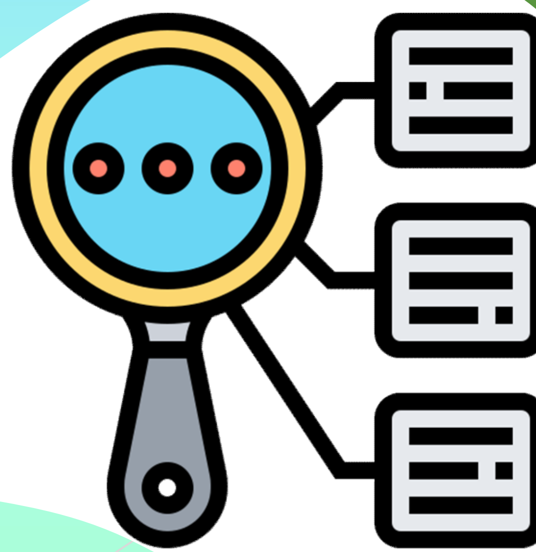
Dr. Wendy Bong Chin Wei

Learning Outcome

- By the end of this lesson, the student shall be able to:
  - Understand the concept of classification
  - Understand how decision tree works
  - Evaluate the model based on confusion matrix

01

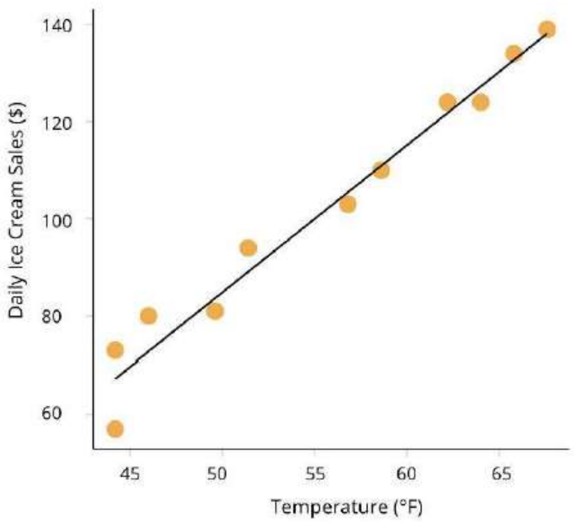
What is Classification?



Type of Supervised Learning

Regression

- Predicting a continuous output



Classification

- Predicting a categorical/discrete output

Definition

- Given a collection of records (training set)
  - Each record is by characterized by a tuple (x,y), where x is the attribute set and y is the class label
    - x: attribute, predictor, independent variable, input
    - y: class, response, dependent variable, output

A schematic illustration of a classification task.

Classification

- A **classification model** is an abstract representation of the relationship between the attribute set and the class label.
- As will be seen in the next chapters, the model can be represented in many ways, e.g., as a tree, a probability table, or simply, a vector of real-valued parameters.
- More formally, we can express it mathematically as a target function f that takes as input the attribute set and produces an output corresponding to the predicted class label.

Classification

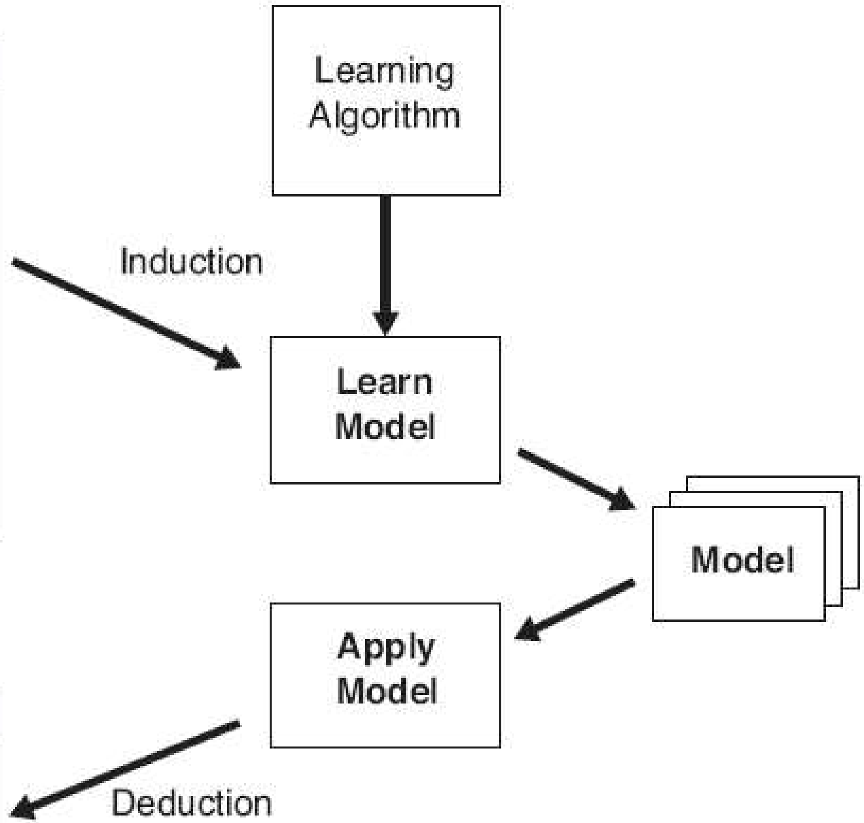
- serves two important roles in data mining.
- a **predictive model**
  - to ***classify*** previously ***unlabeled instances***. A good classification model must provide accurate predictions with a fast response time.
- a **descriptive model**
  - to ***identify the characteristics*** that distinguish instances from different classes.
- This is particularly useful for critical applications, such as medical diagnosis, where it is insufficient to have a model that makes a prediction without justifying how it reaches such a decision.

Classification Task

Task	Attribute set, ***x***	Class label, *y*
Categorizing email messages	Features extracted from email message header and content	spam or non-spam
Identifying tumor cells	Features extracted from x-rays or MRI scans	malignant or benign cells
Cataloging galaxies	Features extracted from telescope images	Elliptical, spiral, or irregular-shaped galaxies

Training Set

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



Test 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	?

General Framework

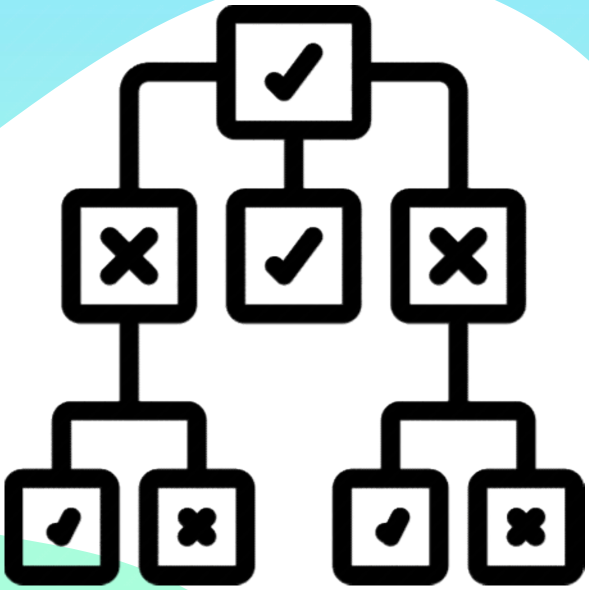
- Classification is the **task of assigning labels to unlabeled data instances** and a classifier is used to perform such a task.
- A **classifier** is typically described in terms of a **model**
- The model is created using a given a set of instances, known as the **training set**, which contains **attribute values** as well as **class labels** for each instance.
- The **systematic approach for learning a classification model** given a training set is known as a **learning algorithm**.

General Framework

- The process of **using a learning algorithm to build a classification model** from the training data is known as **induction**.
- This process is also often described as “learning a model” or “building a model.”
- This **process of applying a classification model on unseen test** instances to predict their class labels is known as **deduction**.
- Thus, the process of classification involves two steps:
  - applying a learning algorithm to training data to learn a model,
  - and then applying the model to assign labels to unlabeled instances..

Techniques

- Decision Tree based Methods
- Rule-based Methods
- Memory based reasoning
- Neural Networks / Deep Learning
- Naïve Bayes and Bayesian Belief Networks
- Support Vector Machines

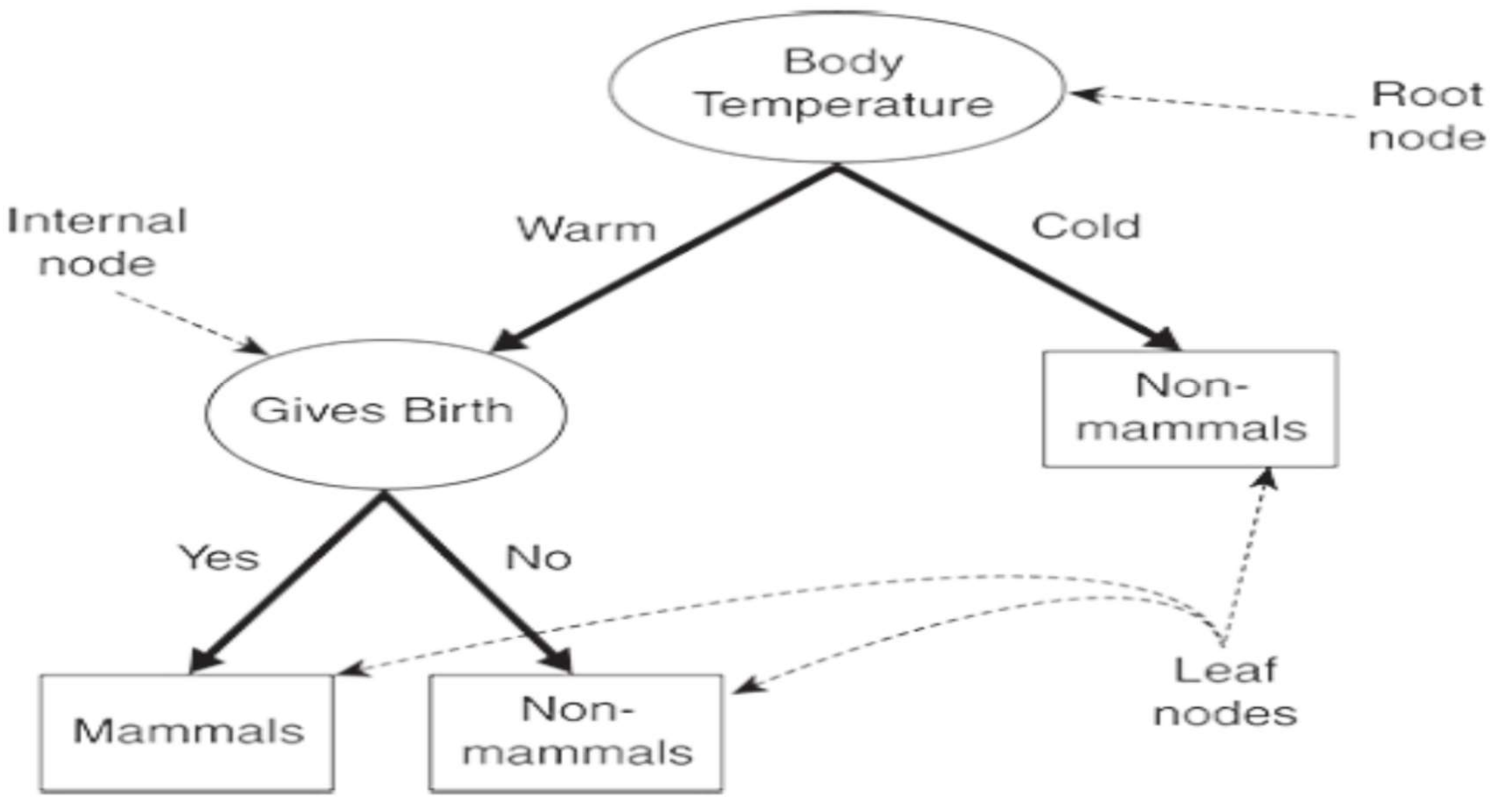


Decision Tree

Decision Tree Classifier

- **It has three types of nodes:**
  - A **root node**, with no incoming links and zero or more outgoing links.
  - **Internal nodes**, each of which has exactly one incoming link and two or more outgoing links.
  - **Leaf** or **terminal** nodes, each of which has exactly one incoming link and no outgoing links - Every leaf node in the decision tree is associated with a class label.
- The **non-terminal** nodes, which include the root and internal nodes, contain **attribute test conditions** that are typically defined using a single attribute.
- Each possible outcome of the attribute test condition is associated with exactly one child of this node link and no outgoing links.

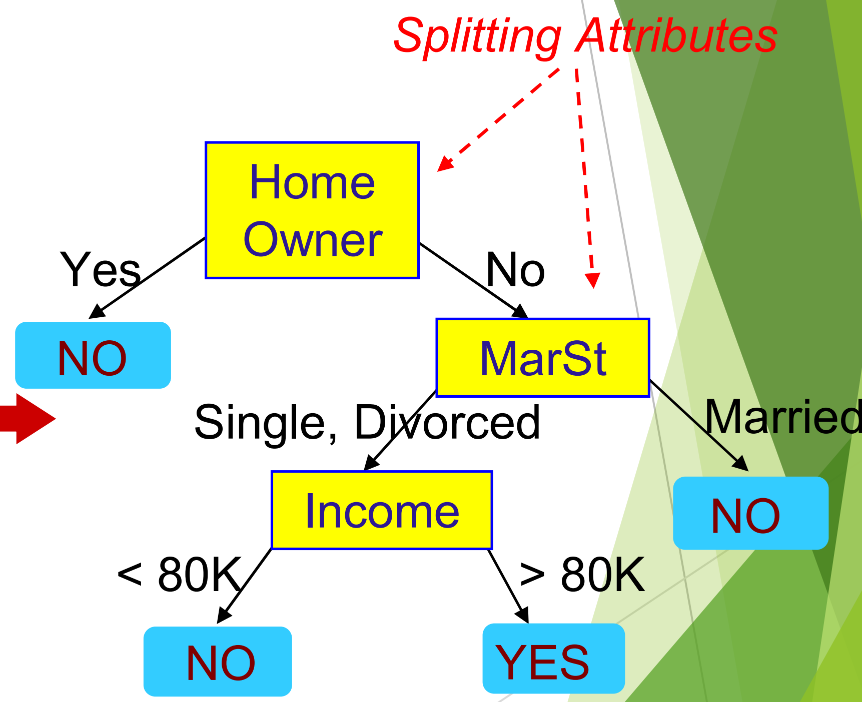
Example: Mammal



Example of a Decision Tree

	categorical	categorical	continuous	class
ID	**Home Owner**	**Marital Status**	**Annual Income**	**Defaulted Borrower**
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**

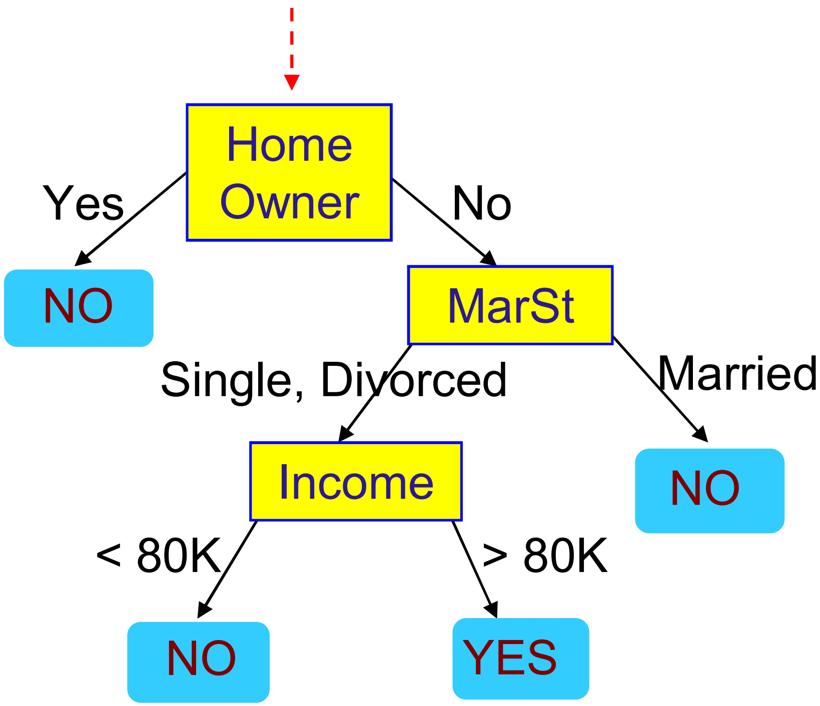
Training Data



Model: Decision Tree

Apply Model to Test Data

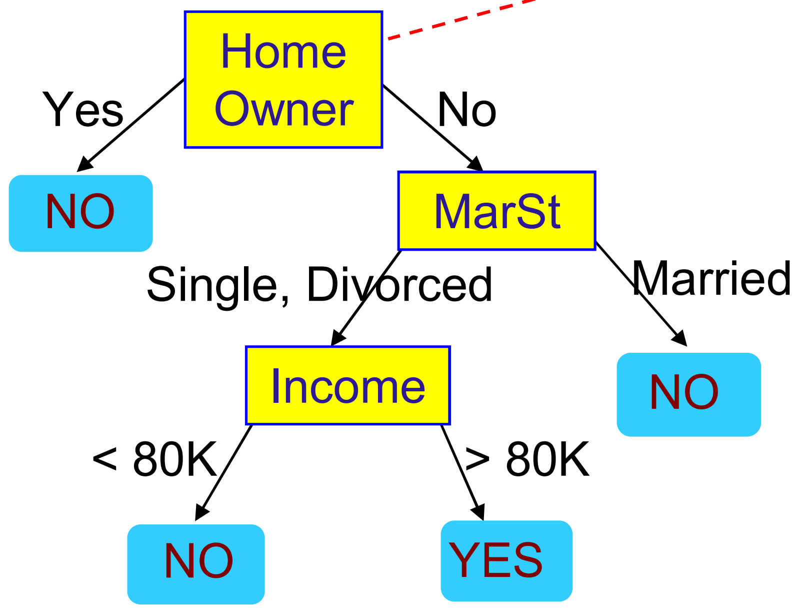
Start from the root of tree.



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	**?**

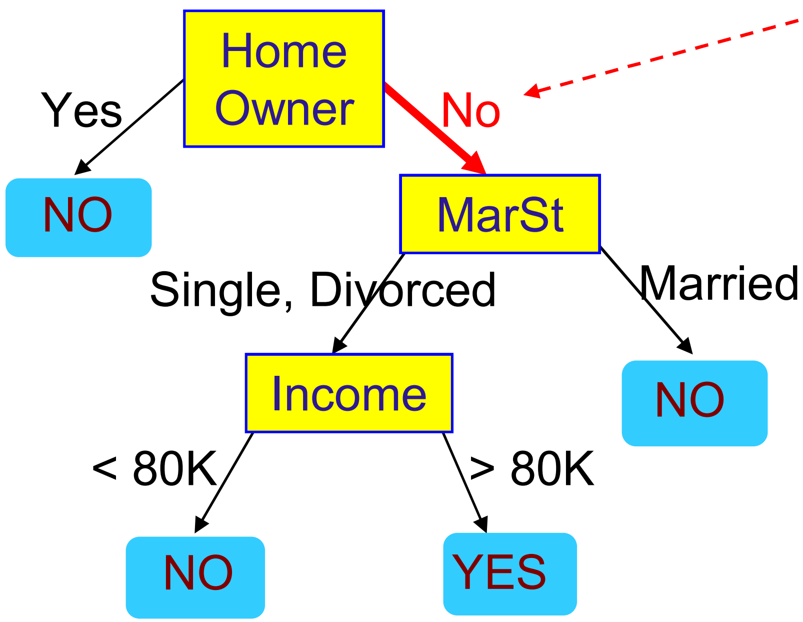
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

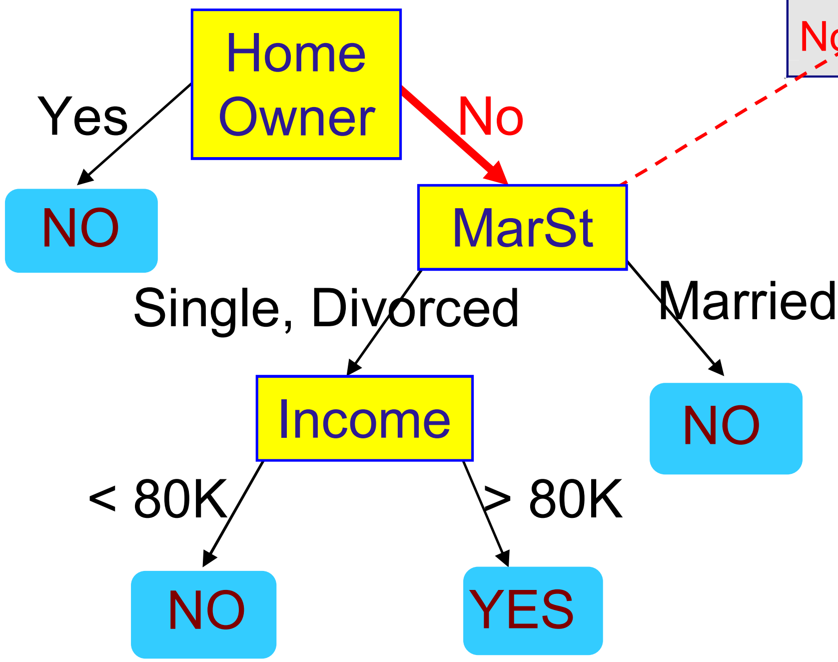
Apply Model to Test Data



Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data



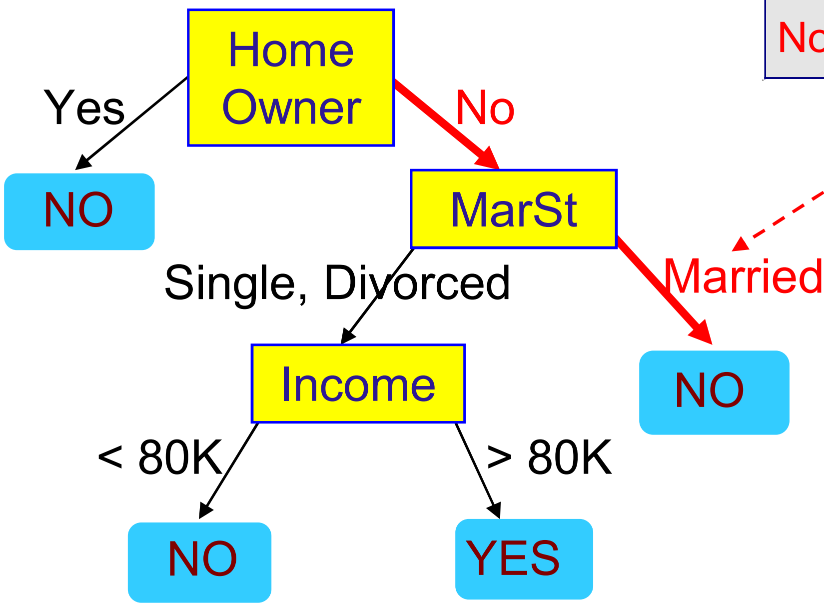
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?

Apply Model to Test Data

Test Data

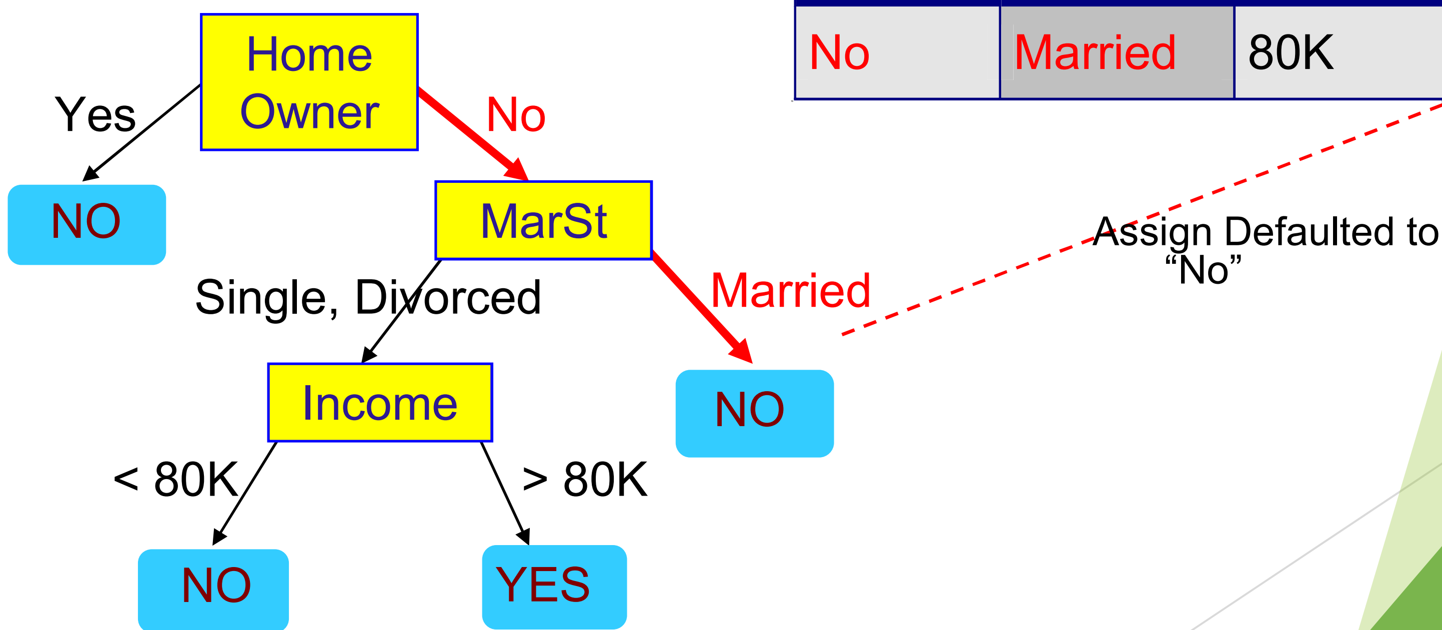
Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Apply Model to Test Data

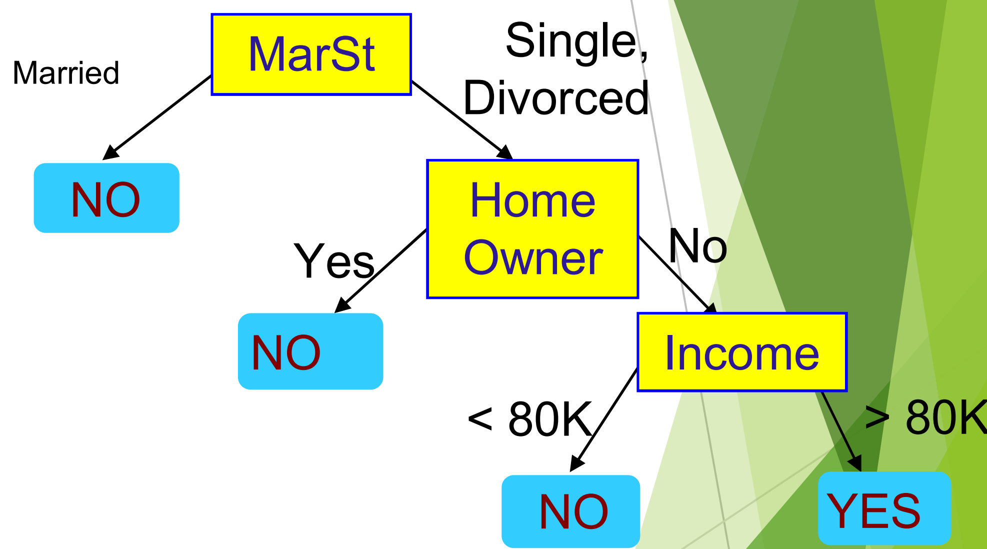
Test Data

Home Owner	Marital Status	Annual Income	Defaulted Borrower
No	Married	80K	?



Another Example of Decision Tree

	categorical	categorical	continuous	class
ID	**Home Owner**	**Marital Status**	**Annual Income**	**Defaulted Borrower**
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**

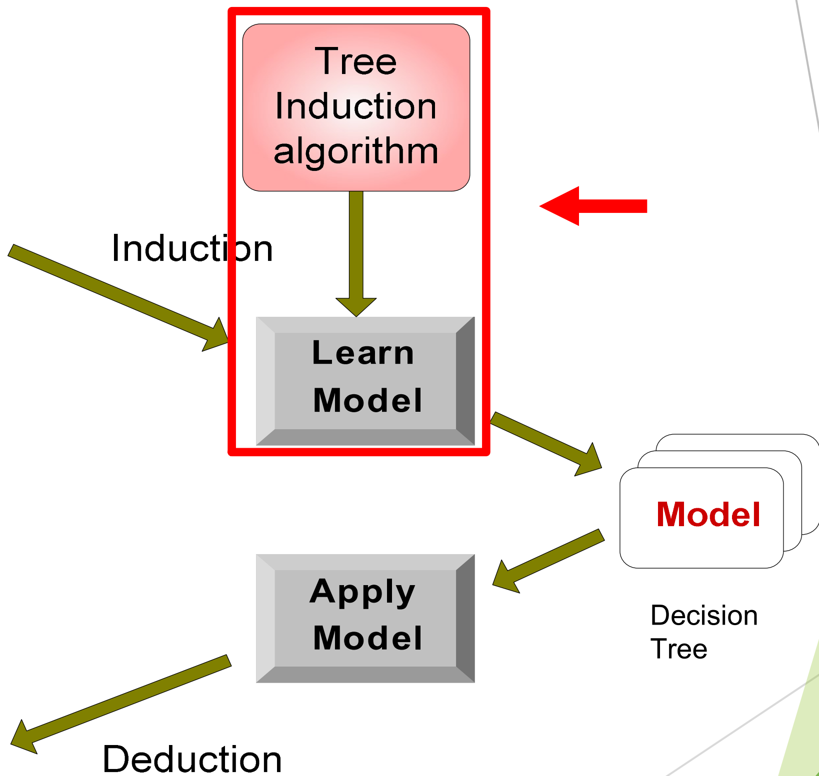


There could be more than one tree that fits the same data!

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

Design Issues of Decision Tree Induction

How should training records be split?	• Method for expressing test condition depending on attribute types • Measure for evaluating the goodness of a test condition
How should the splitting procedure stop?	• Stop splitting if all the records belong to the same class or have identical attribute values • Early termination

How to Specify Test Condition?

Depends on attribute types

Binary

Nominal

(name only, no ordering)
Direction: North, East, South, West

Ordinal

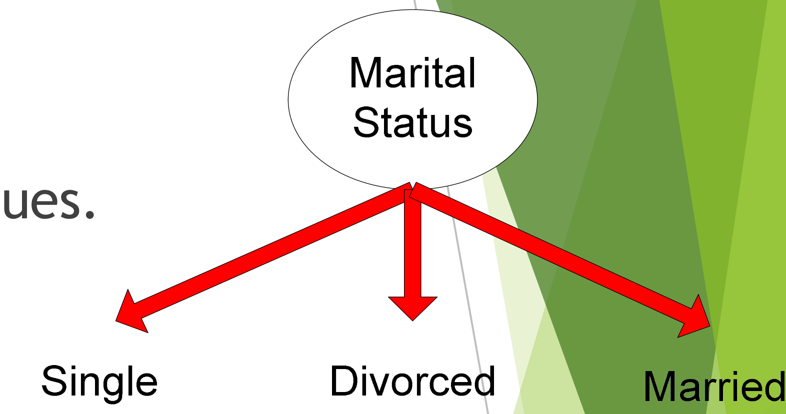
(ordered, not measurable)
First, second, third …
Hot, warm, cold

Continuous

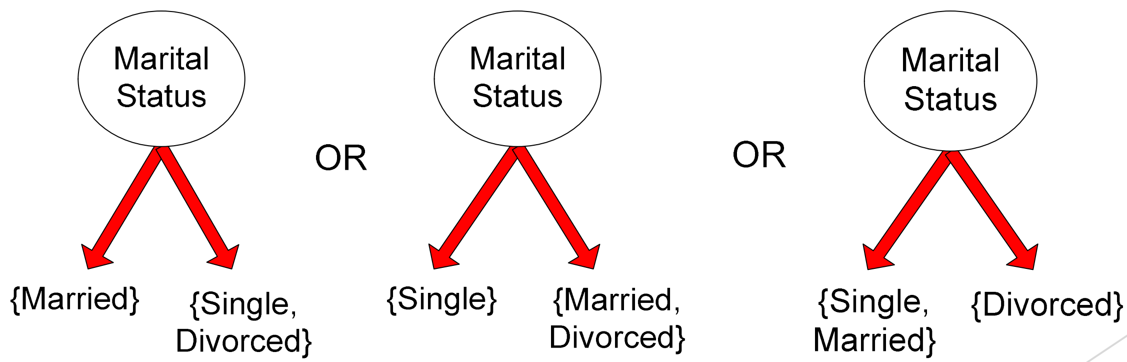
(allows arithmetic operations)
-123, 29.56, …

Test Condition for Nominal Attributes

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

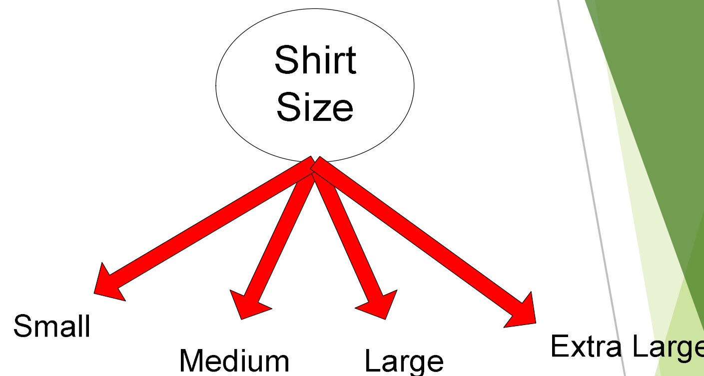


- Binary split:
  - Divides values into two subsets

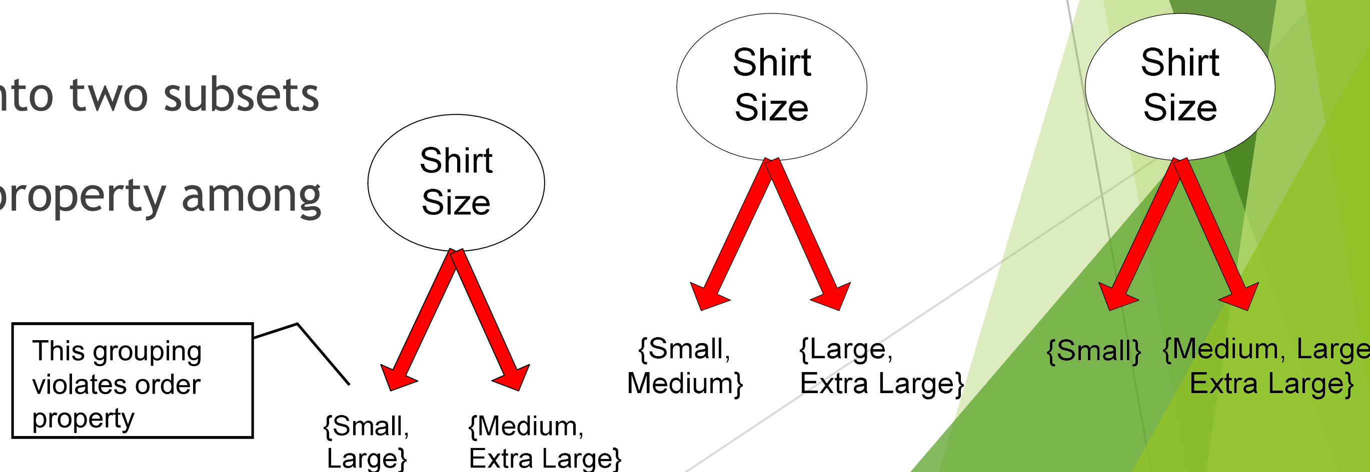


Test Condition for Ordinal Attributes

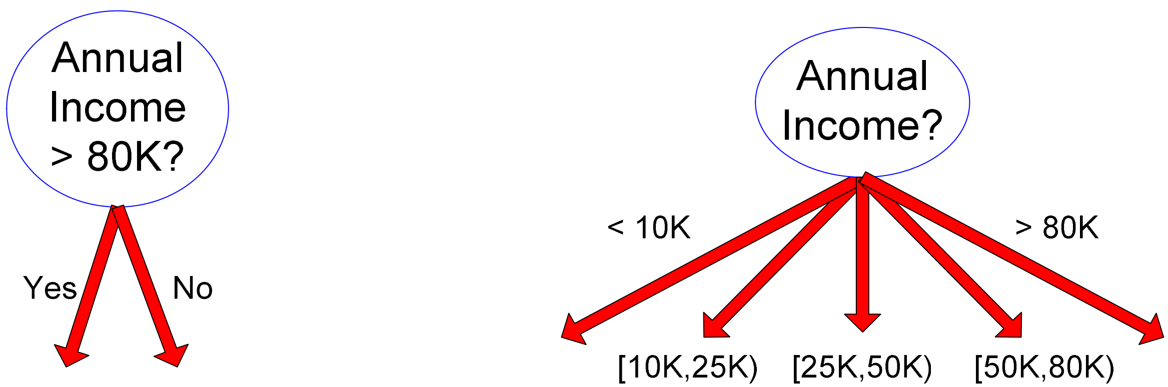
- Multi-way split:
  - Use as many partitions as distinct values



- Binary split:
  - Divides values into two subsets
  - Preserve order property among attribute values



Test Condition for Continuous Attributes



(i) Binary split

(ii) Multi-way split

Advantages of Decision Tree Based Classification

Inexpensive To Construct

Extremely Fast At Classifying Unknown Records

Easy To Interpret For Small-sized Trees

Accuracy Is Comparable To Other Classification Techniques For Many Simple Data Sets

Model Evaluation and Comparison

Metrics for Performance Evaluation: Confusion Matrix

- Focus on the predictive capability of a model (not speed, scalability, etc.)
- Here we will focus on binary classification problems!

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

a: TP (true positive)
b: FN (false negative)
c: FP (false positive)
d: TN (true negative)

Metrics for Performance Evaluation: Statistical Test

From Statistics: Null Hypotheses H0 is that the actual class is yes

		PREDICTED CLASS	
ACTUAL CLASS		Class=Yes	Class=No
	Class=Yes		Type I error
	Class=No	Type II error	

Type I error: $P(NO \mid H0 \text{ } is \text{ } true)$
Type II error: $P(Yes \mid H0 \text{ } is \text{ } false)$

Metrics for Performance Evaluation: Accuracy

Most widely-used metric: How many do we predict correct (in percent)?

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

$$Accuracy = \frac{a+d}{a+b+c+d} = \frac{TP+TN}{N}$$

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 the model does not detect any class 1 example

→ **Class imbalance problem!**

Cost Matrix

Different types of error can have different cost!

	PREDICTED CLASS		
ACTUAL CLASS	***C(i|j)***	**Class=Yes**	**Class=No**
	Class=Yes	$C(Yes\|Yes)$	$C(No\|Yes)$
	Class=No	$C(Yes\|No)$	$C(No\|No)$

$C(i|j)$: Cost of misclassifying class j example as class i

Computing Cost of Classification

Cost Matrix	PREDICTED CLASS		
ACTUAL CLASS	*C(i|j)*	+	-
	+	-1	100
	-	1	0

Missing a + case is really bad!

Model M_1	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	150	40
	-	60	250

Accuracy = 80%

Cost = -1*150+100*40+
1*60+0*250 = 3910

Model M_2	PREDICTED CLASS		
ACTUAL CLASS		+	-
	+	250	45
	-	5	200

Accuracy = 90%

Cost = 4255

Cost-Biased Measures

$$Precision\ (p) = \frac{a}{a+c}$$

$$Recall\ (r) = \frac{a}{a+b}$$

$$F-measure\ (F) = \frac{2rp}{r+p} = \frac{2a}{2a+b+c}$$

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

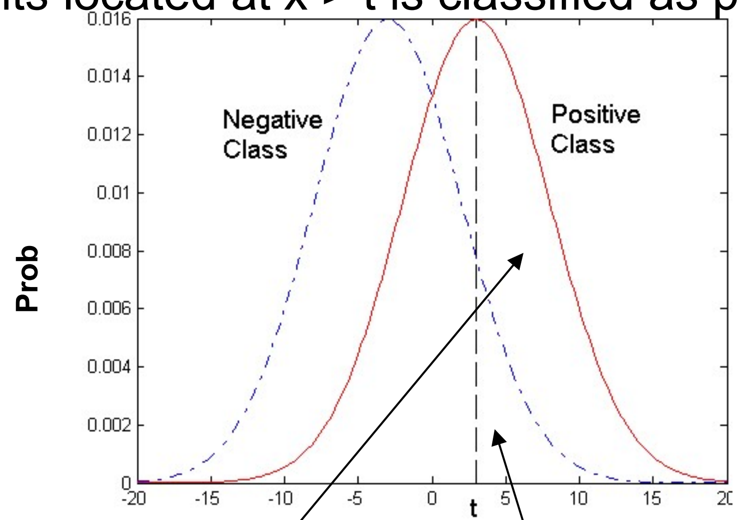
- Precision is biased towards *C(Yes|Yes)* & *C(Yes|No)*
- Recall is biased towards *C(Yes|Yes)* & *C(No|Yes)*
- F-measure is biased towards all except *C(No|No)*

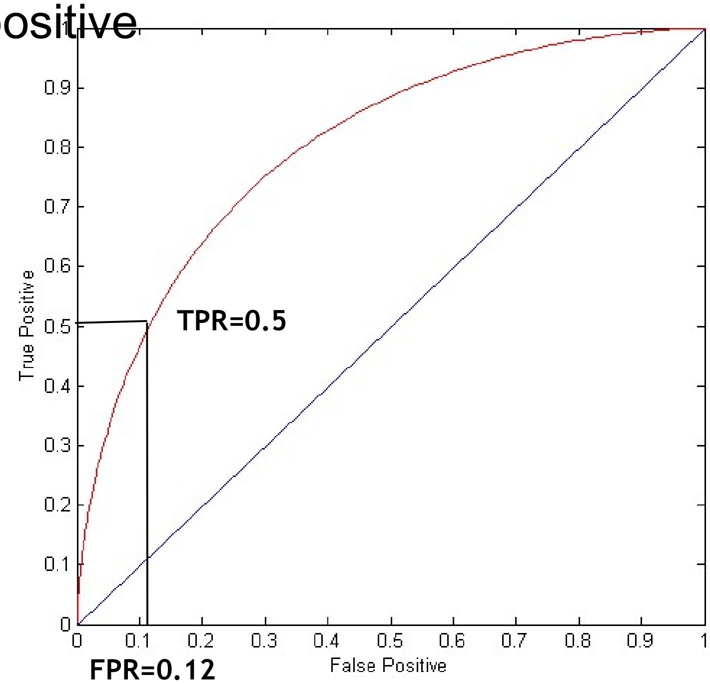
ROC (Receiver Operating Characteristic)

- Developed in 1950s for signal detection theory to analyze noisy signals to characterize the **trade-off between positive hits and false alarms.**
- Works only for **binary classification (two-class problems).** The classes are called the positive and the other is the negative class.
- **ROC curve plots TPR (true positive rate) on the y-axis against FPR (false positive rate) on the x-axis.**
- Performance of each classifier represented as a point. Changing the threshold of the algorithm, sample distribution or cost matrix changes the location of the point and forms a curve.

ROC Curve

- Example with 1-dimensional data set containing 2 classes (positive and negative)
- Any points located at x > t is classified as positive





At threshold t:

TPR=0.5, FNR=0.5, **FPR=0.12,** FNR=0.88

- Move t to get the other points on the ROC curve.

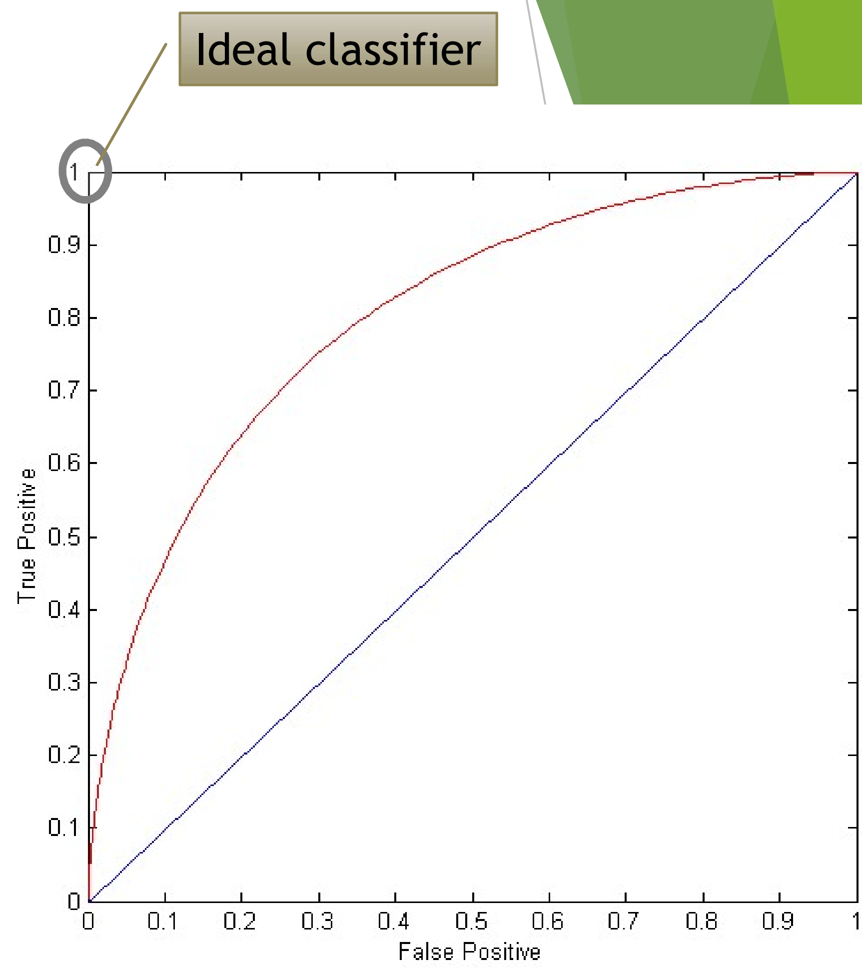
ROC Curve

(TPR,FPR):

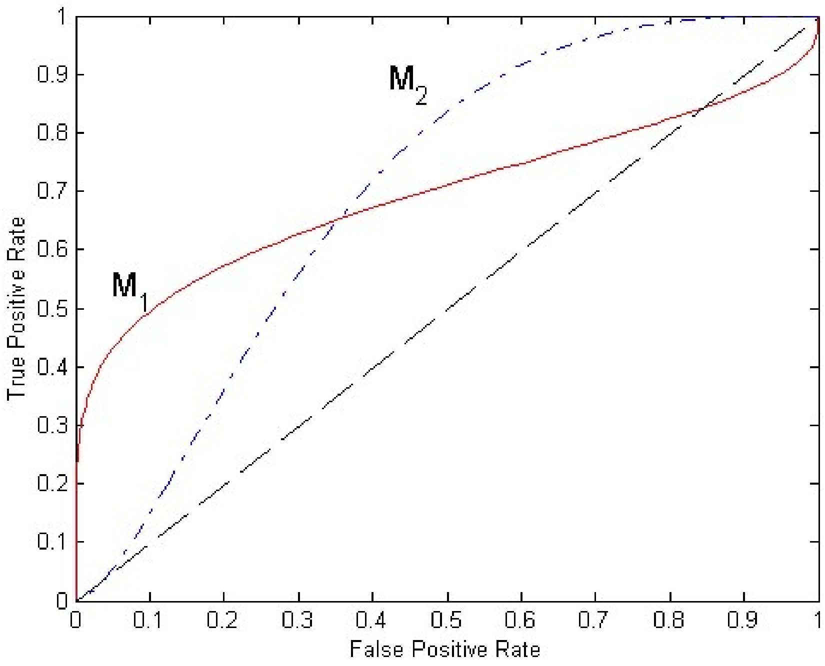
- (0,0): declare everything to be negative class
- (1,1): declare everything to be positive class
- (1,0): ideal

Diagonal line:

- Random guessing
- Below diagonal line: prediction is opposite of the true class



Using ROC for Model Comparison



No model consistently outperform the other

- M1 is better for small FPR
- M2 is better for large FPR

Area Under the ROC curve (AUC)

- Ideal:
  - AUC = 1
- Random guess:
  - AUC = 0.5


THANK
YOU