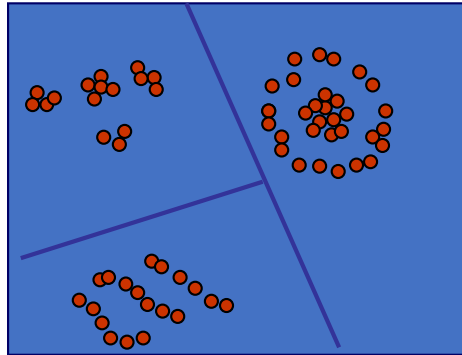


# Building Classification Models: Basics



# Classification Vs. Clustering

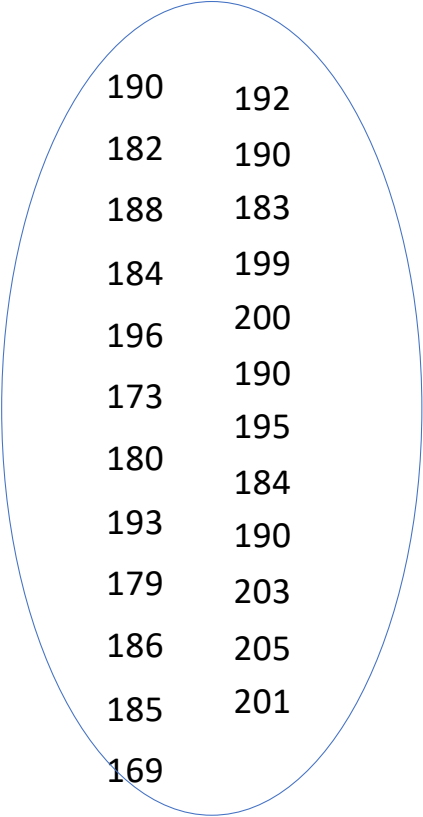
- Building a classifier requires marked or **labeled** training data. The number of classes, i.e. the number of distinct labels, is known beforehand. In a way, *the training data for classification comes with known answers*. Thus, the process is often termed **supervised learning**.
- In clustering, we seek a grouping in data without often knowing how many groups are there and how the examples in each group look like. Thus, it is often termed **unsupervised learning**.

# What is Classification?

190	F	192	M
182	F	190	M
188	F	183	M
184	F	199	M
196	F	200	M
173	F	190	M
180	F	195	M
193	F	184	M
179	F	190	M
186	F	203	M
185	F	205	M
169	F	201	M

Given the measurements for several male and female basketball players, can we predict the gender of the player whose height is 194 ?

# What is Clustering?



190  
182  
188  
184  
196  
173  
180  
193  
179  
186  
185  
169

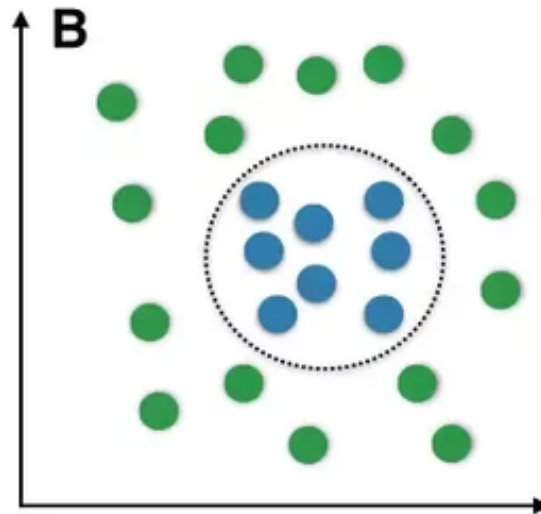
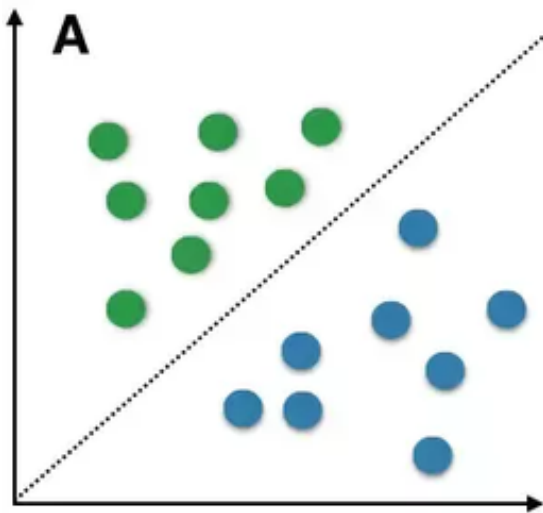
192  
190  
183  
199  
200  
190  
195  
184  
190  
203  
205  
201

Given a collection of height measurements,  
form two groups

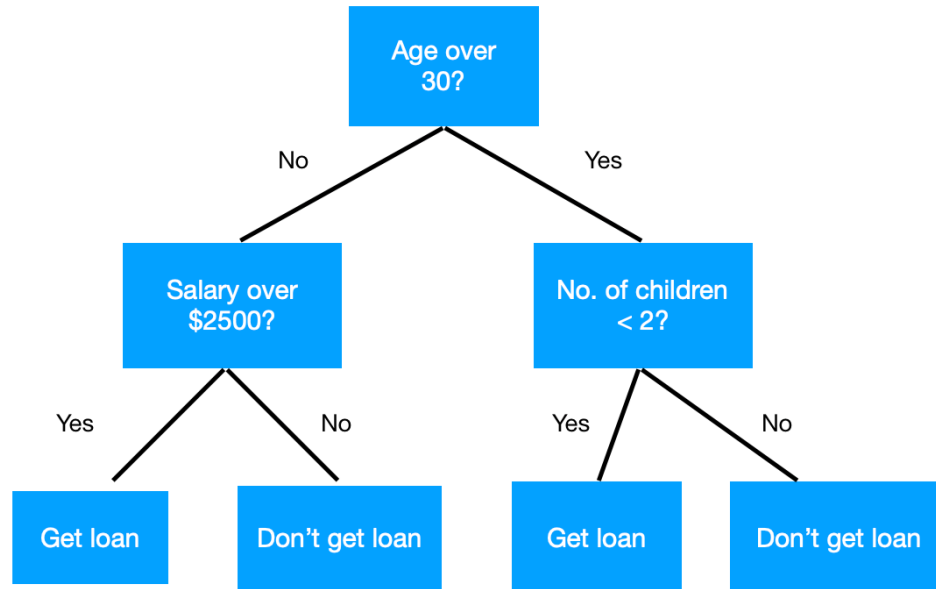
# Classifier Types

- Linear vs. nonlinear
  - A linear classifier separates examples from different classes by using linear decision functions
  - A nonlinear classifier uses nonlinear functions and is thus more powerful in achieving separation
- One shot vs. multi-stage classification
  - How the classification decision is arrived at?

# Linear & Nonlinear Classification



# Single Stage Vs. Multi-Stage Classification



# Hard Vs. Soft Classification

- Hard classification implies that the label(s) assigned to a document/object carry equal weight
- Soft classification implies ordering among the labels assigned to a document/object. Soft classifiers are often called **fuzzy classifiers**



# Binary Vs. Multi-way Classification

- Binary classification implies only two classes or categories. For example, spam vs. non-spam email
- Multi-way classification implies more than two categories. For example, Reuters collection has 135 topic categories.
  - A M-way multi-class problem can be converted to M binary classification problems

# Single Vs. Multi-label Classification

- A single label classifier assigns only one class/category label to an input pattern. The traditional classifiers are single label classifiers
- A multi-label classifier assigns more than one label to an input, for example a student classification system might label a student as *academic high achiever*, *athletic*, and *popular*. Many information retrieval applications need multiple labels to stored artifact

# Generative Vs. Discriminative Vs. Geometric Models

- In generative models, we assume a functional form for conditional probabilities and use data to estimate these probabilities and apply Bayes rule to estimate posteriori probabilities
- In discriminative models, we try to estimate directly the posteriori probabilities or learn to map input directly to class labels
- In geometric models, we try to learn a geometric function that can separate examples from different classes. The method doesn't use any probabilistic concepts

# Walk Through An Example: Flower Classification

- Build a classification model (smartphone app) to differentiate between two classes of flower



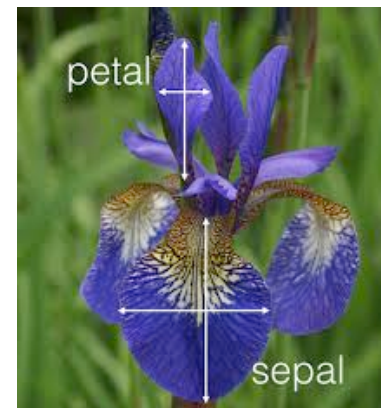
Iris Virginia



Iris Versicolor

# How Do We Go About It?

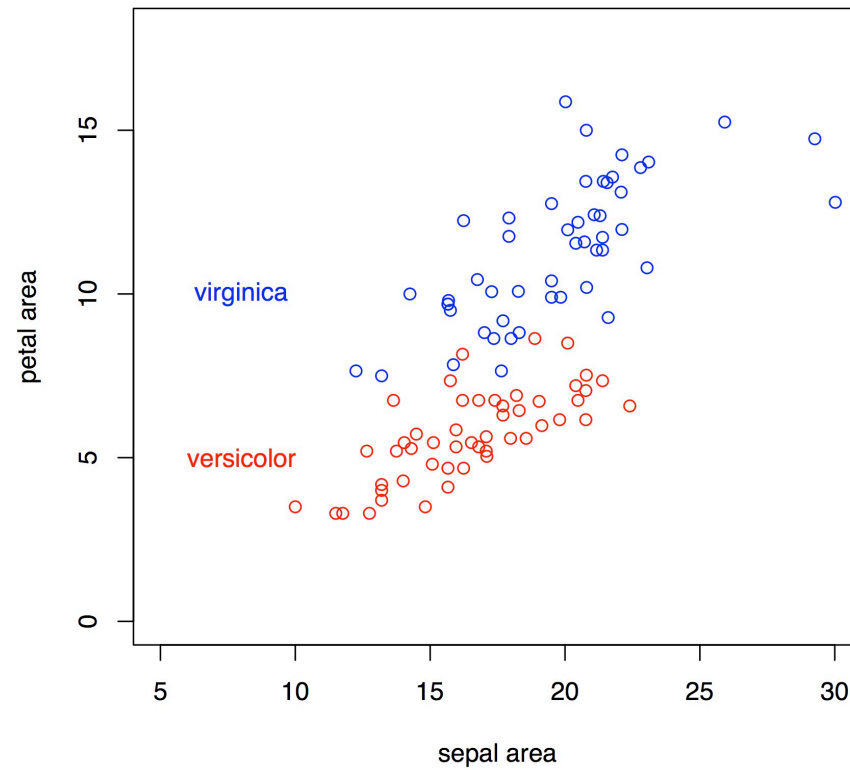
- Collect a large number of both types of flowers with the help of an expert
- Measure some attributes that can help differentiate between the two types of flowers. Let those attributes be petal area and sepal area.



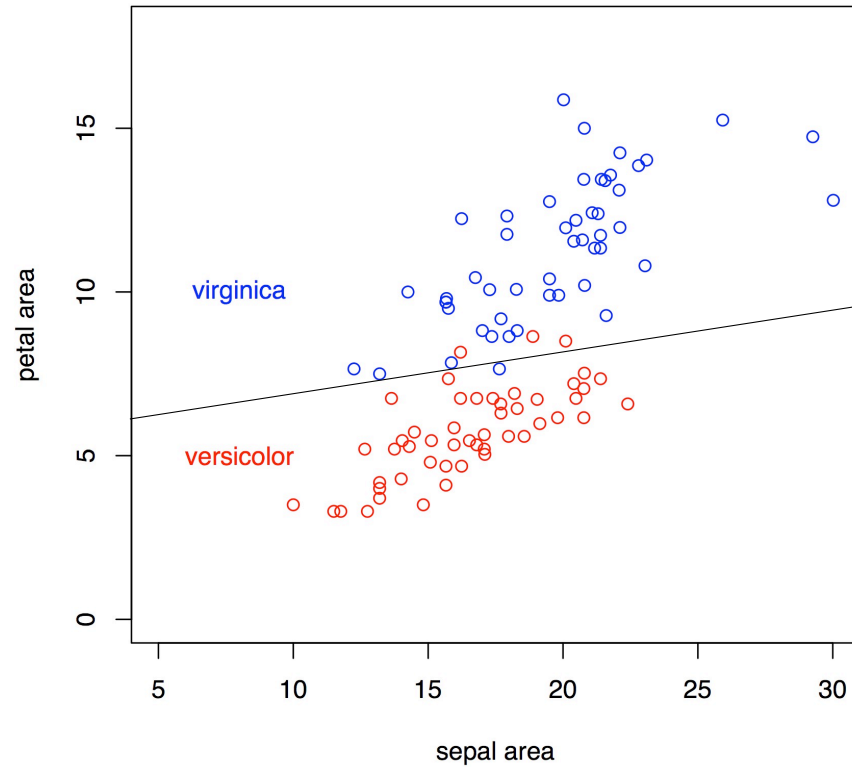
Scatter plot of 100 examples of flowers

**Anderson's Iris Data**

Let's plot the collected examples.

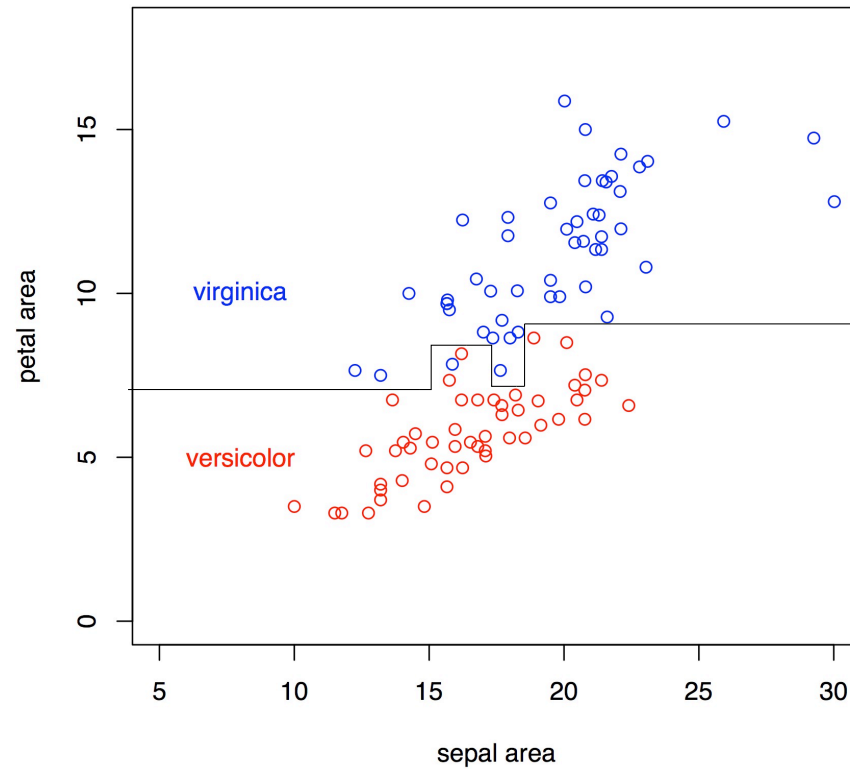


Anderson's Iris Data



We can separate the flower types using the linear boundary shown above. The parameters of the line represent the learned classification model.

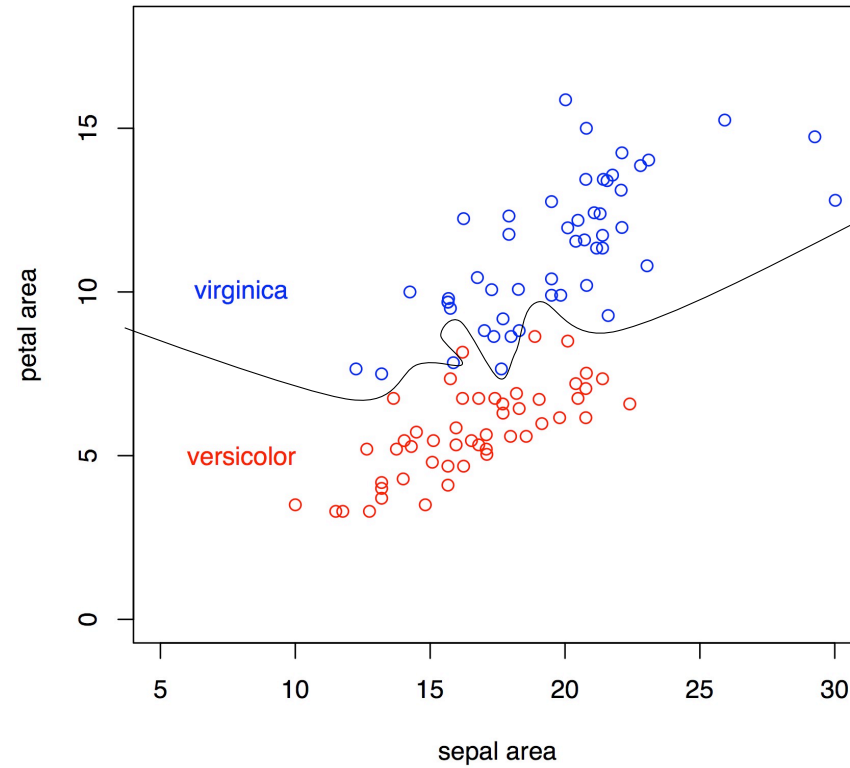
**Anderson's Iris Data**



Another possible boundary. This boundary cannot be expressed via an equation. However, a tree structure can be used to express this boundary. Note, this boundary does better prediction of the collected data



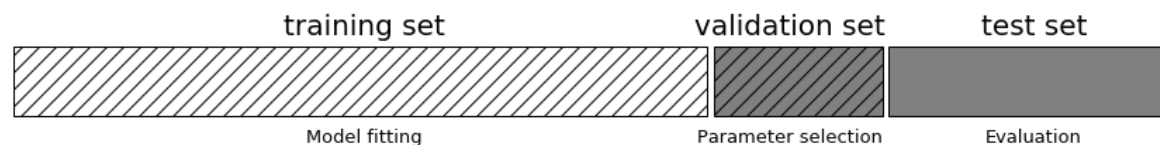
Anderson's Iris Data



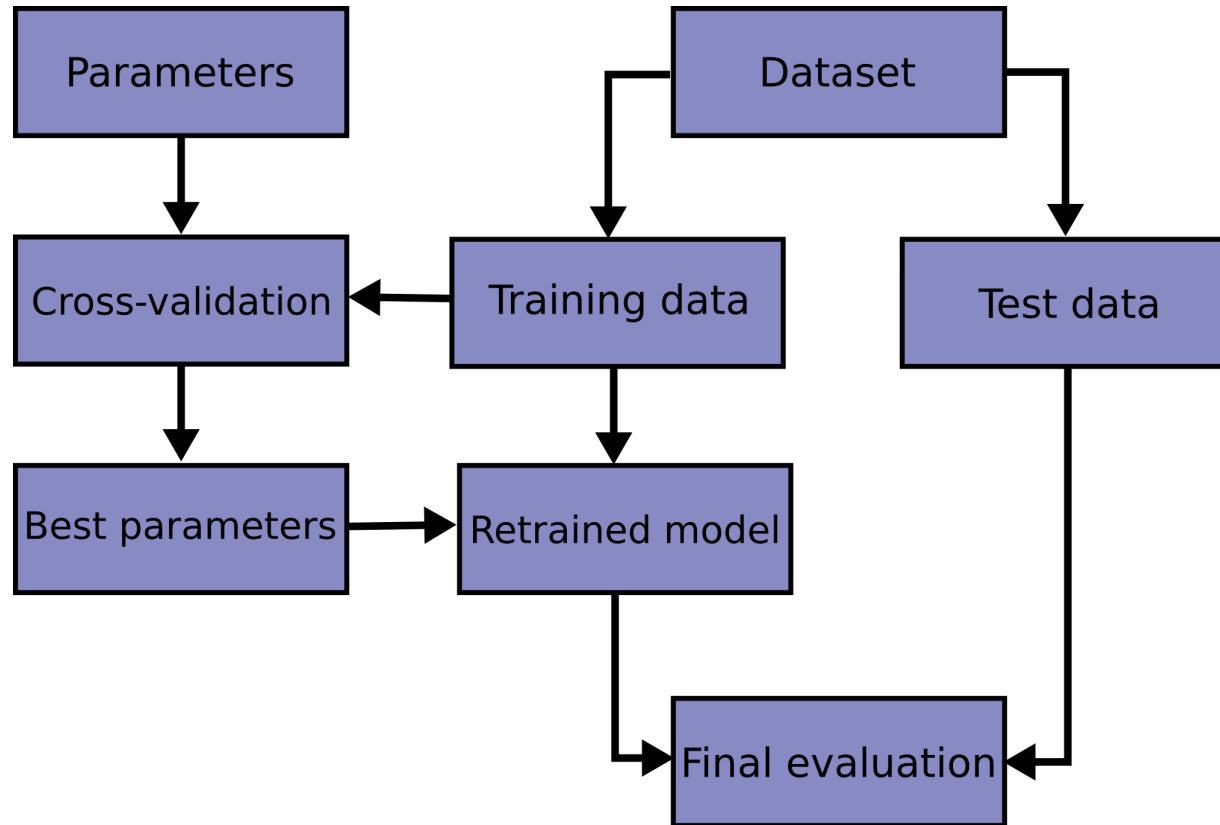
Yet another possible boundary. This boundary does prediction without any error. Is this a better boundary?

# Model Complexity

- There are tradeoffs between the complexity of models and their performance in the field. A good design (model choice) weighs these tradeoffs.
- A good design should avoid overfitting. How?
  - Divide the entire data into three sets
    - Training set (about 70% of the total data). Use this set to build the model
    - Test set (about 20% of the total data). Use this set to estimate the model accuracy after deployment
    - Validation set (remaining 10% of the total data). Use this set to determine the appropriate settings for free parameters of the model. May not be required in some cases.



## The overall stages of model building



# N-Fold Cross-validation

- Cross-validation refers to designing the classifier using the training data and testing the classifier's performance against the test data.
- In practice, the validation process should be repeated multiple times using different training and test sets. N-fold cross-validation means this process is repeated N times.
  - For example, let us say you have 100 labelled examples. For 5-fold cross-validation, we will divide these 100 examples into five groups of 20 examples each. We will pick the first group as the test set and use the remaining four sets as the training set. Next, we will use the second group as the test set and the remaining four groups as the training set. We will repeat this three more times with the third, fourth and fifth groups becoming training sets respectively.
- When N equals the number of training examples, the cross-validation process is known as *leave-one-out* method.

# Measuring Classifier Performance

	Correct category is $c_1$	Correct category is $c_2$
Assigned category is $c_1$	a	b
Assigned category is $c_2$	c	d

- Accuracy =  $(a + d)/(a + b + c + d)$
- Precision( $c_1$ ) =  $a/(a + b)$ ; Precision( $c_2$ ) =  $d/(c + d)$
- Recall( $c_1$ ) =  $a/(a + c)$ ; Recall( $c_2$ ) =  $d/(b + d)$

# Measuring Performance

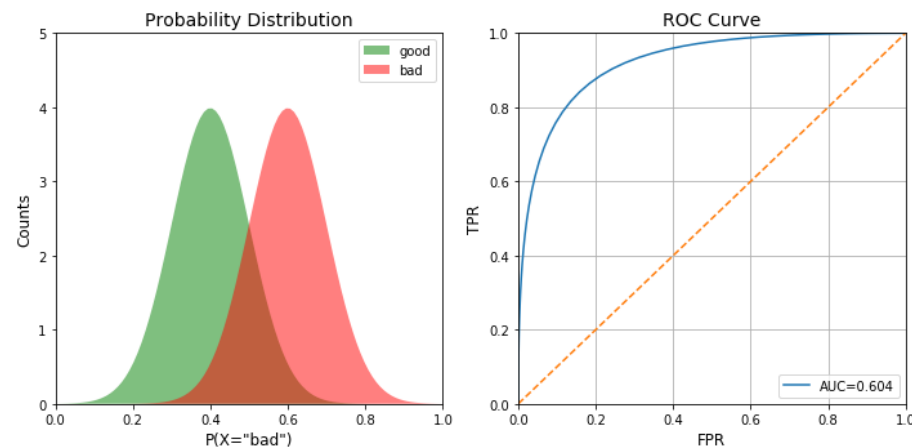
- True Positive: Correctly identified as relevant (“a” from prev. slide)
- True Negative: Correctly identified as not relevant (“d”)
- False Positive: Incorrectly labeled as relevant (“b”)
- False Negative: Incorrectly labeled as not relevant (“c”)

Cat vs. No Cat



# ROC (Receiver Operating Characteristics) Curve AUC (Area under the ROC Curve)

- True Positive Rate (TPR) =  $TP/(TP+FN) = a/(a+c)$
- False Positive Rate (FPR) =  $FP/(FP+TN) = b/(b+d)$
- An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.



# Confusion Matrix for M Classes

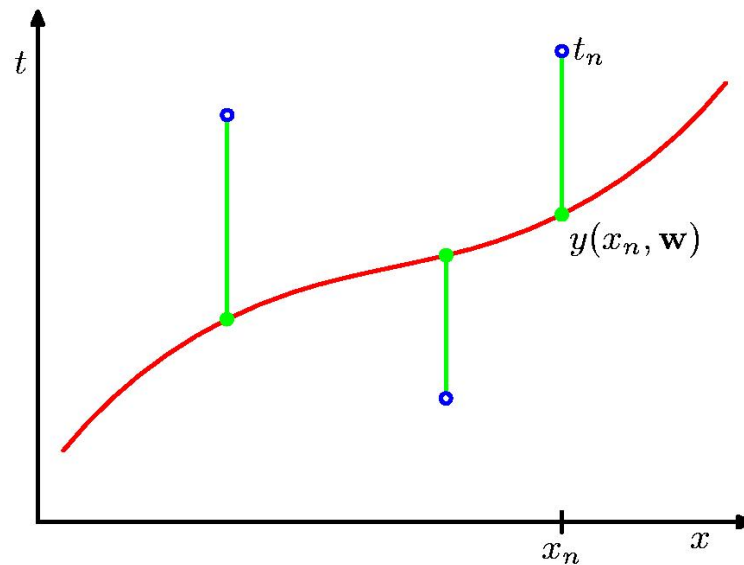
		Predicted class						
		Sit	Stand	Walk Jog	Ascend	Descend	Cycle	Class-specific recall
Actual class	Sit	3202	2	0	0	0	14	0.99
	Stand	7	3191	2	7	0	0	0.99
	Walk	0	0	10647	74	0	0	0.99
	Ascend	0	0	34	500	15	1	0.90
	Descend	0	0	41	60	405	0	0.80
	Cycle	146	3	0	0	0	2539	0.94
	Class-specific precision	0.95	1.00	0.99	0.78	0.96	0.99	0.98



# Accuracy?

- The previous definition of accuracy can be misleading when you have uneven representation from two classes. For example, consider a 2-class problem with 90 examples from class 1 and 10 from class 2.
  - Class predictive model 1 classifies 75 of class 1 examples and 3 of class 2 examples correctly
  - Another model classifies 60 of class 1 examples and 7 of class 2 examples correctly
  - Which model is a better predictor?
- What happens when different mistakes do not cost the same?

# Sum-of-Squares Error for Regression Models



For regression model, the error is measured by taking the square of the difference between the predicted output value and the target value for each training (test) example and adding this number over all examples as shown

# Bias and Variance

- Bias: expected difference between model's prediction and truth
- Variance: how much the model differs among training sets
- Model Scenarios
  - High Bias: Model makes inaccurate predictions on training data
  - High Variance: Model does not generalize to new datasets
  - Low Bias: Model makes accurate predictions on training data
  - Low Variance: Model generalizes to new datasets

# The Guiding Principle for Model Selection: Occam's Razor

