GENERAL ASSEMBLY

DATA SCIENCE
REVIEW

I. DATA SCIENCE WORKFLOW
II. SUPERVISED LEARNING
III. K-NEAREST NEIGHBORS
IV. LINEAR REGRESSION
V. LOGISTIC REGRESSION
VI. NAÏVE BAYES
VII. DECISION TREES
VIII. RANDOM FORESTS
IX. ADABOOST
X. NEURAL NETWORKS
XI. RECOMMENDERS
XII. CATEGORICAL VARIABLES
XIII. MODEL EVAL. PROCEDURES
XIV. MODEL EVAL. METRICS
XV. UNSUPERVISED LEARNING
XVI. CLUSTER ANALYSIS
XVII. DIMENSION REDUCTION

I. DATA SCIENCE WORKFLOW

I. Define the problem / question

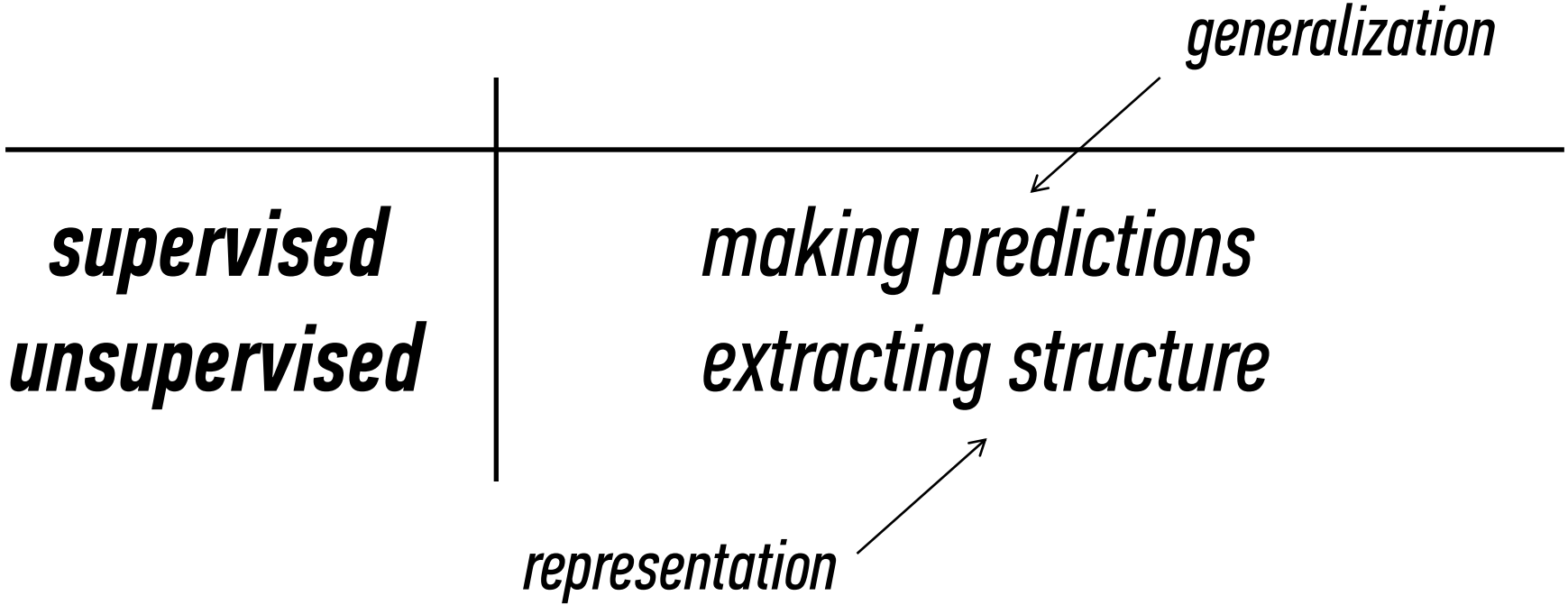
II. Identify and collect data

III. Explore and prepare data

IV. Build and evaluate model

V. Communicate results

II. SUPERVISED LEARNING


generalization
supervised
unsupervised
making predictions
extracting structure
representation

	continuous	***categorical***
supervised	*regression*	*classification*
unsupervised	*dimension reduction*	*clustering*

- Objectives of Supervised Learning:
  - Accurately predict unseen test cases
  - Understand which predictors affect the response, and how
  - Assess the quality of our predictions

- Vector of “Predictors” X
  - Also known as features, independent variables, inputs, regressors, covariates, attributes
- “Response” y
  - Also known as outcome, label, target, dependent variable
- Regression: y is continuous
  - e.g., price, blood pressure
- Classification: y is categorical (values in a finite, unordered set)
  - e.g., spam/ham, digit 0-9, cancer class of tissue sample
- Data is composed of “observations” (predictors and the associated response)
  - Also known as samples, examples, instances, records

150 observations
(n = 150)

Fisher's *Iris* Data

Sepal length	Sepal width	Petal length	Petal width	Species
5.1	3.5	1.4	0.2	*I. setosa*
4.9	3.0	1.4	0.2	*I. setosa*
4.7	3.2	1.3	0.2	*I. setosa*
4.6	3.1	1.5	0.2	*I. setosa*
5.0	3.6	1.4	0.2	*I. setosa*
5.4	3.9	1.7	0.4	*I. setosa*
4.6	3.4	1.4	0.3	*I. setosa*
5.0	3.4	1.5	0.2	*I. setosa*

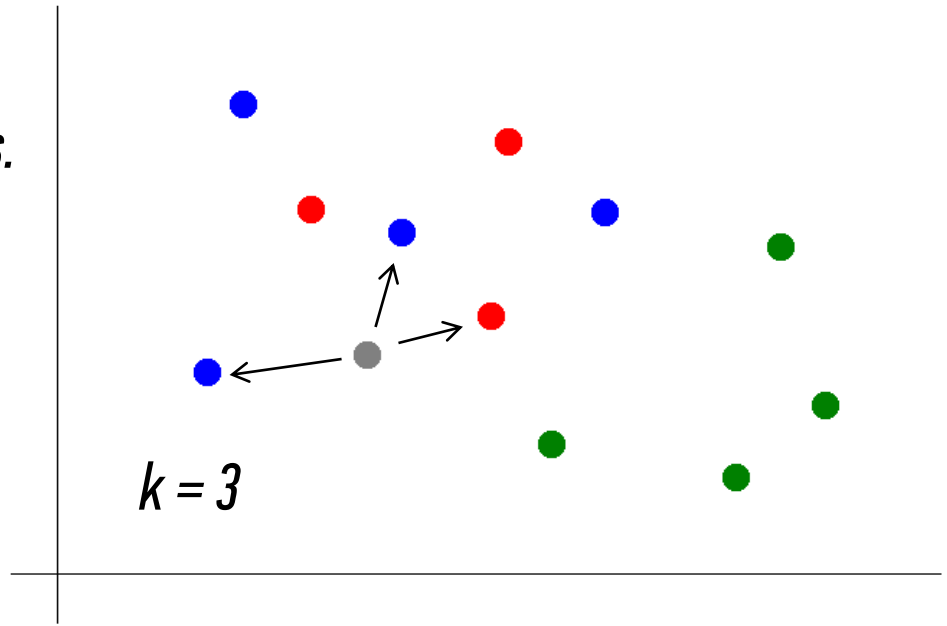
response

4 predictors (p = 4)

III. K-NEAREST NEIGHBORS

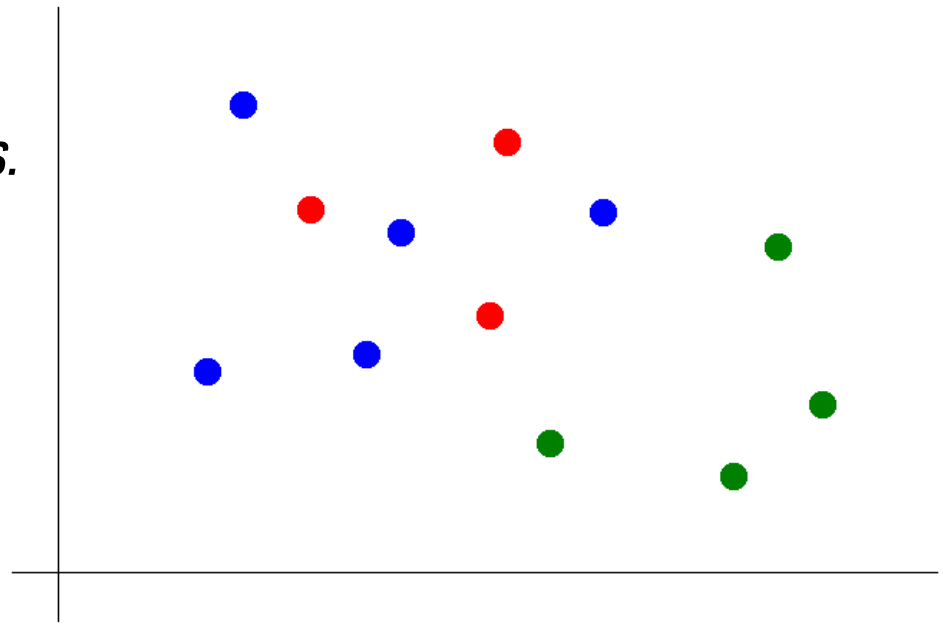
Suppose we want to predict the color of the gray dot.

1) *Pick a value for k.*
2) *Find colors of k nearest neighbors.*



Suppose we want to predict the color of the gray dot.

1) *Pick a value for k.*
2) *Find colors of k nearest neighbors.*
3) *Assign the most common color to the gray dot.*



IV. LINEAR REGRESSION

Q: What is a **regression** *model?*
A: A functional relationship between input & response variables.

The **simple linear regression** *model captures a linear relationship between a single input variable x and a response variable y:*

$$y = \alpha + \beta x + \varepsilon$$

Q: What do the terms in this model mean?

$$y = \alpha + \beta x + \varepsilon$$

A: y = **response variable** *(the one we want to predict)*

x = **input variable** *(the one we use to train the model)*

α = **intercept** *(where the line crosses the y-axis)*

β = **regression coefficient** *(the model parameter)*

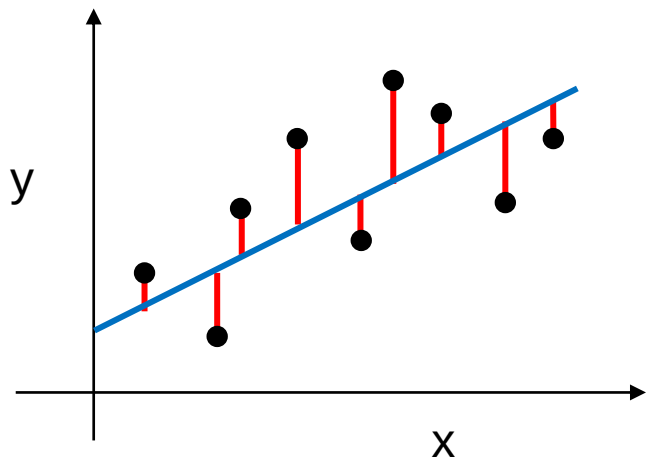
ε = **residual** *(the error)*

We can extend this model to several input variables, giving us the **multiple linear regression** *model:*

$$y = \alpha + \beta_1 x_1 + \ldots + \beta_n x_n + \varepsilon$$

*Q: How to **estimate** coefficients for a linear model?*

A: By finding the line that minimizes the sum of squared residuals.



Model Prediction

$$SS_{residuals} = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

Observed Result

V. LOGISTIC REGRESSION

Q: What is **logistic regression***?*

A: A generalization of the linear regression model to classification problems.

In linear regression, we used a set of input variables to predict the value of a continuous response variable.

In logistic regression, we use a set of input variables to predict probabilities of class membership.

These probabilities can then mapped to class labels, thus predicting the class for each observation.

When performing linear regression, we use the following function:

$$y = \beta_0 + \beta_1 x$$

When performing logistic regression, we use the following form:

$$\pi = \Pr(y = 1 \mid x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Probability of y = 1, given x

In linear regression, the parameter β_1 represents the change in the ***response variable*** *for a unit change in x.*

In logistic regression, β_1 represents the change in the ***log-odds*** *for a unit change in x.*

This means that e^{β_1} gives us the change in the ***odds*** *for a unit change in x.*

Example: *Suppose we are interested in mobile purchase behavior. Let y be a class label denoting purchase/no purchase, and let x denote whether phone was an iPhone.*

We perform a logistic regression, and we get $\beta_1 = 0.693$.

In this case the odds ratio is exp(0.693) = 2, meaning the likelihood of purchase is twice as high if the phone is an iPhone.

VI. NAÏVE BAYES

Suppose we have a dataset with features $x_1, \ldots, x_n$ and a class label C. What can we say about classification using Bayes' theorem?

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

Bayes' theorem can help us to determine the probability of a record belonging to a class, given the data we observe.

source: Data Analysis with Open Source Tools, by Philipp K. Janert. O'Reilly Media, 2011.

This term is the **posterior probability** *of c. It represents the probability of a record belonging to class c after the data is taken into account.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

The goal of any Bayesian computation is to find ("learn") the posterior distribution of a particular feature. This constitutes the training phase of the model.

The idea of Bayesian inference, then, is to **update** *our beliefs about the distribution of c using the data ("evidence") at our disposal.*

$$P(\text{class } C \mid \{x_i\}) = \frac{P(\{x_i\} \mid \text{class } C) \cdot P(\text{class } C)}{P(\{x_i\})}$$

Then we can use the posterior for prediction.

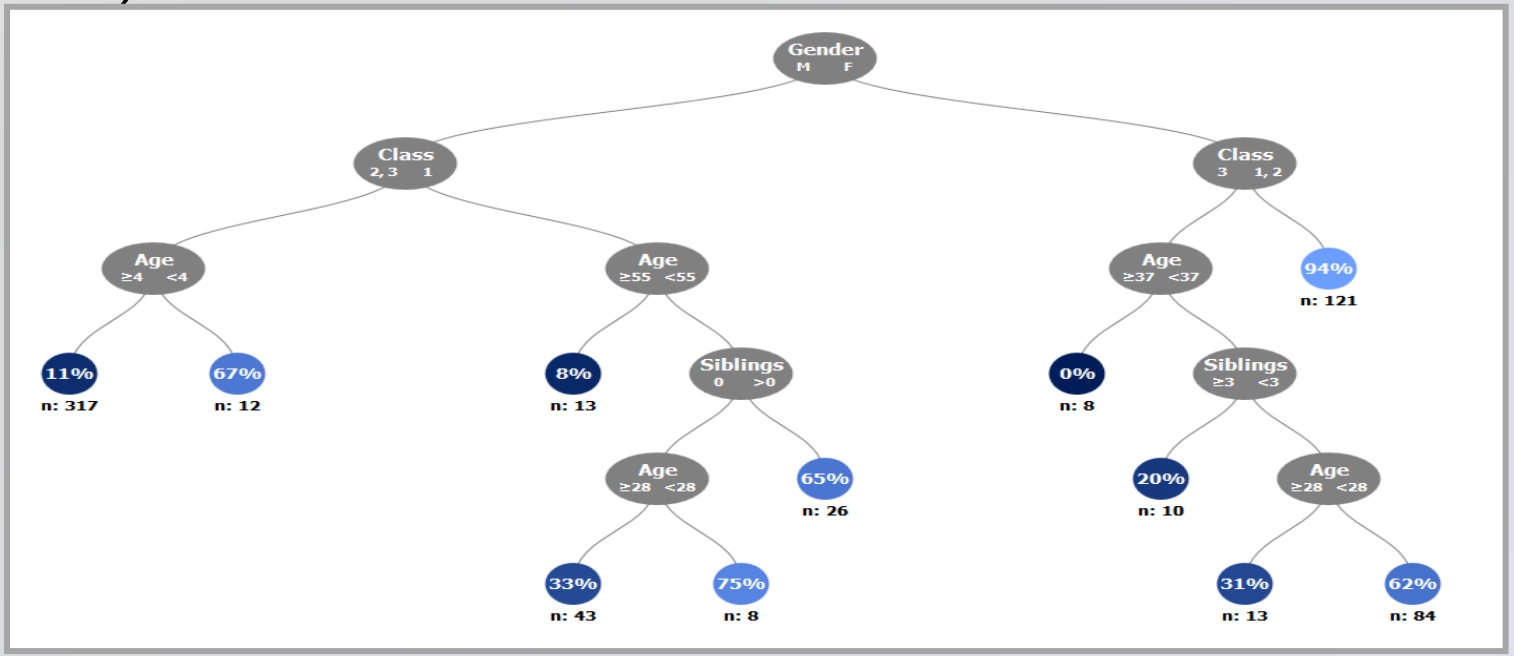
VII. DECISION TREES

What are decision trees?

- A supervised learning technique that can be used for discrete and continuous output.
- Visually engaging and easy to interpret.
- Excellent model for someone transitioning into the world of data science.
- Foundational to learning some very powerful techniques.
- Are prone towards high-variance.
- We will focus on the CART algorithm.

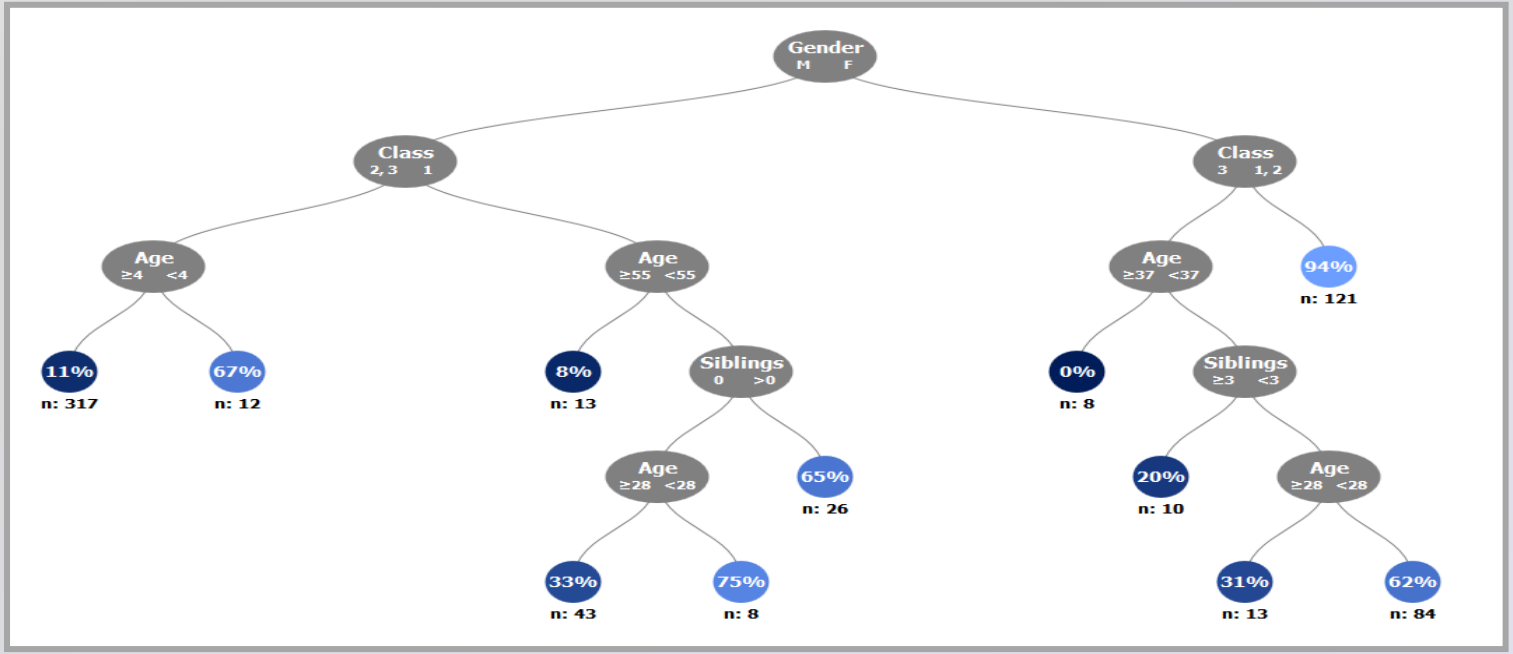
Description

Decision Trees are made up of interconnected nodes, which act as a series of questions / test conditions (e.g., is the passenger male or female?)



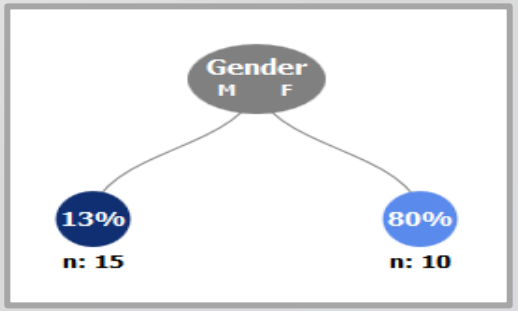
Description

Terminal nodes show the output metric, in this case the percentage of titanic survivors for a given combination of variables.

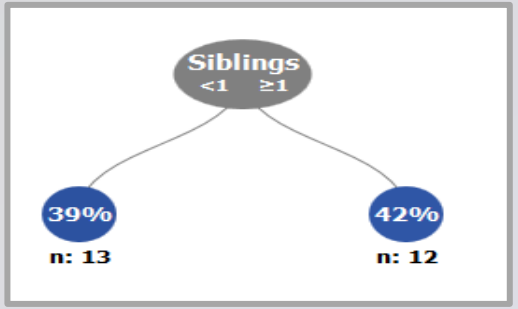


The Algorithm, Introduced

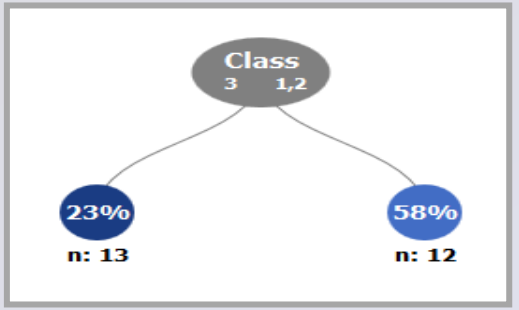
Different variables and split options are evaluated to determine which split will provide the greatest separation between classes.



Which split option would you select?



How can we determine the best split analytically?



The Algorithm – Overview

- Calculate the purity of the data
- Select a candidate split
- Calculate the purity of the data after the split
- Repeat for all variables
- Choose the variable with the greatest increase in purity
- Repeat for each split until some stopping criteria is met

Meta-Evaluation

Advantages:

- The nature of its output provides the decision tree algorithm with a degree of interpretability that other more complex algorithms don’t provide
- Understanding how the decision Tree works is foundational to understanding how more complex, and widely-used models work, such as random forests and boosted trees.

Disadvantages:

- The decision Tree tends to perform worse than more sophisticated modeling techniques due to their instability

VIII. RANDOM FORESTS

Definition of Random Forests:

"Each new training set is drawn, with replacement, from the original training set. Then a tree is grown on the new training set using random feature selection. The trees grown are not pruned."

Q: Why are trees not pruned?

A: To increase variance (i.e., model diversity). Variance is good because it means we have different trees. Remember how this was a disadvantage we discussed of CART? Well, now this is a key feature that makes enables the Random Forest to be so powerful.

Benefits of Random Forests:

- *Its accuracy is as good as AdaBoost and sometimes better.*
- *It's relatively robust to outliers and noise.*
- *It's faster than bagging or boosting.*
- *It gives useful internal estimates of error, strength, correlation and variable importance.*
- *It's simple and easily parallelized.*

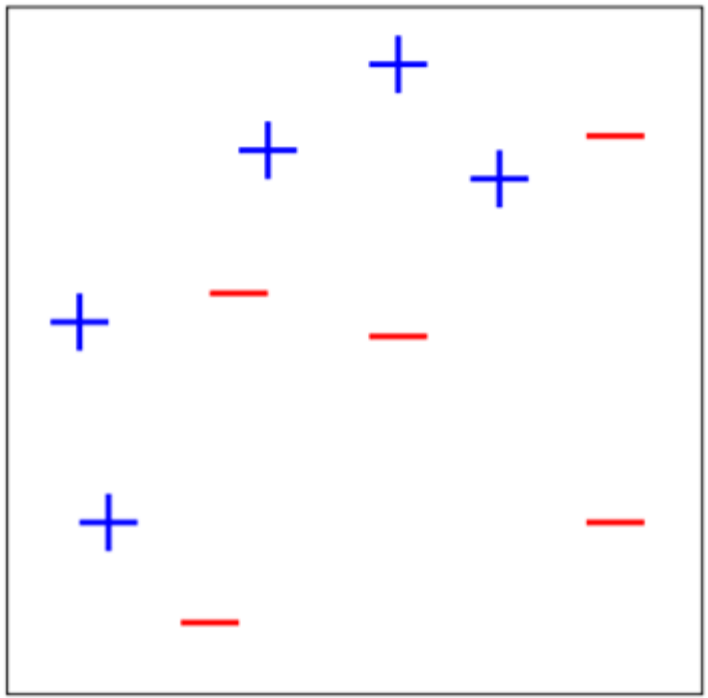
IX. ADABOOST

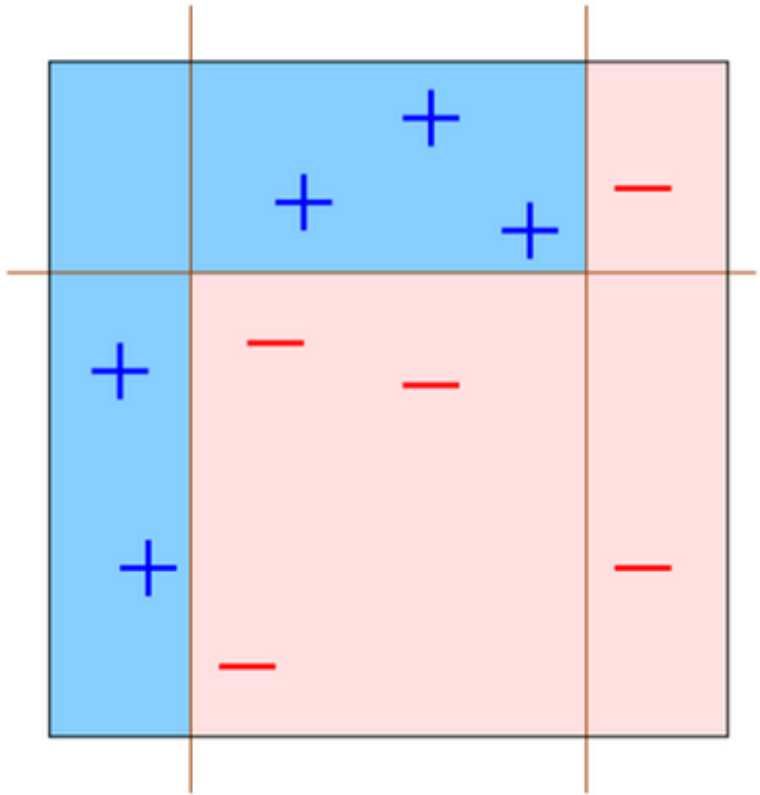
The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data.

The predictions from all of them are then combined through a weighted majority vote (or sum) to produce the final prediction.

The data modifications at each so-called boosting iteration consist of applying weights to each of the training samples. For each successive iteration, the sample weights are individually modified and the learning algorithm is reapplied to the reweighted data...

As iterations proceed, examples that are difficult to predict receive ever-increasing influence. Each subsequent weak learner is thereby forced to concentrate on the examples that are missed by the previous ones in the sequence.

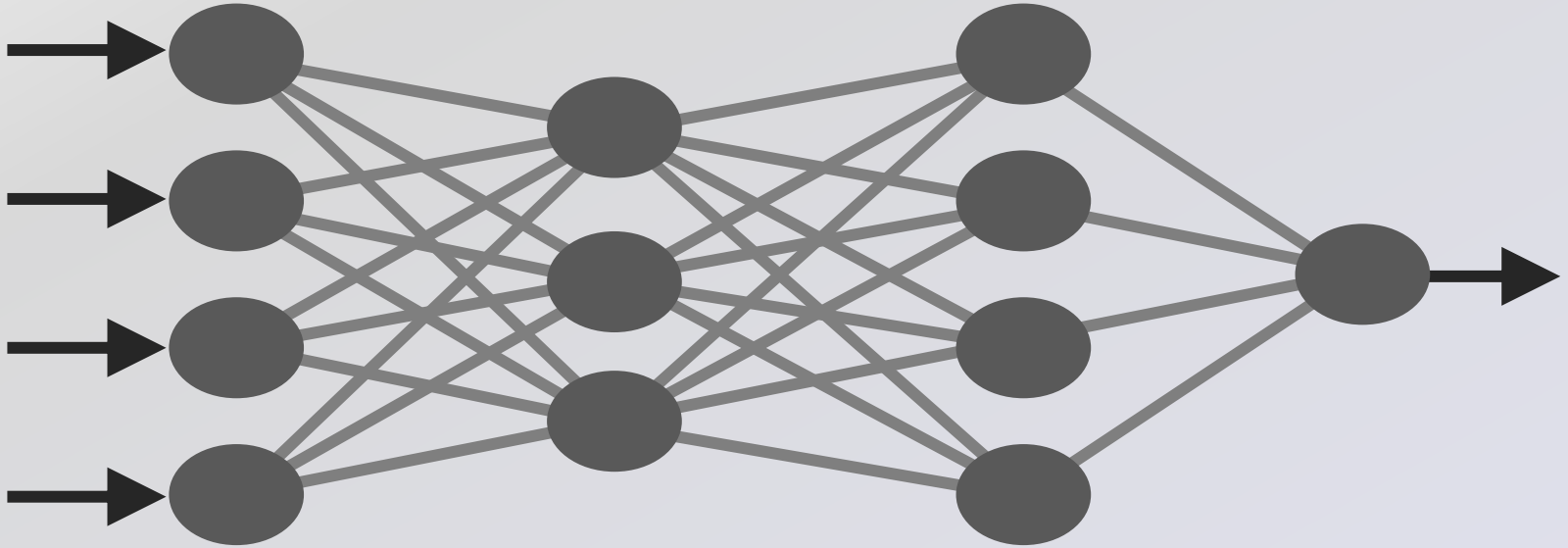




X. NEURAL NETWORKS

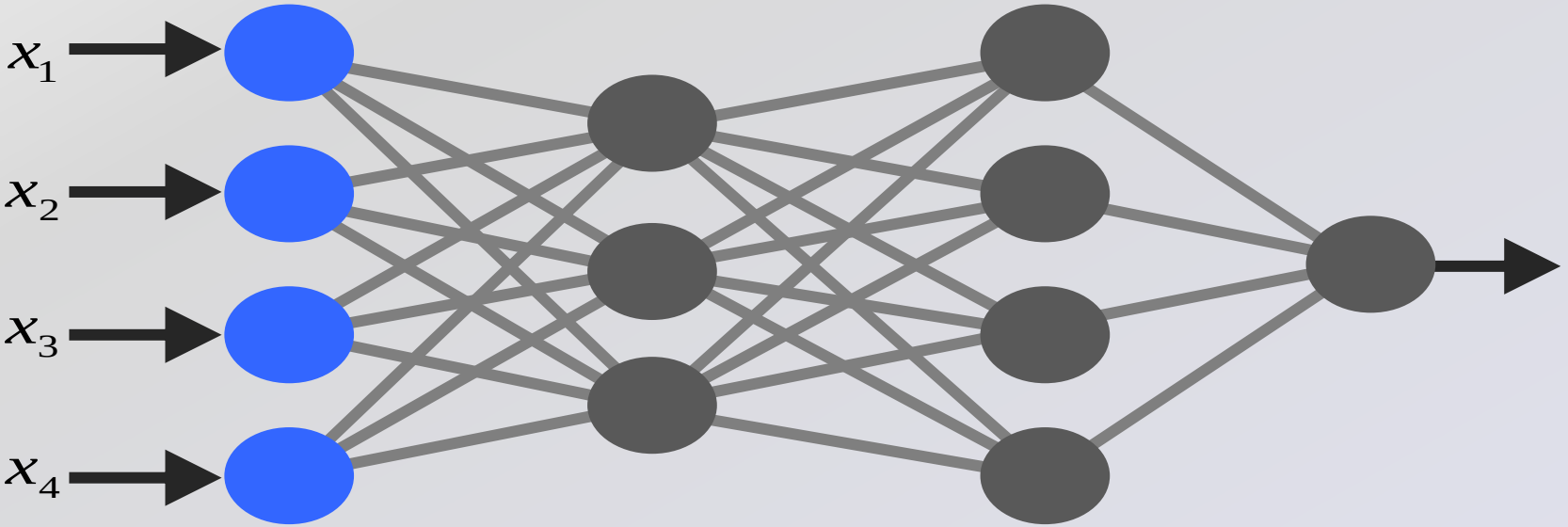
Representation

Artificial neural networks are represented as a system of interconnected neurons with multiple layers.



Input Layer

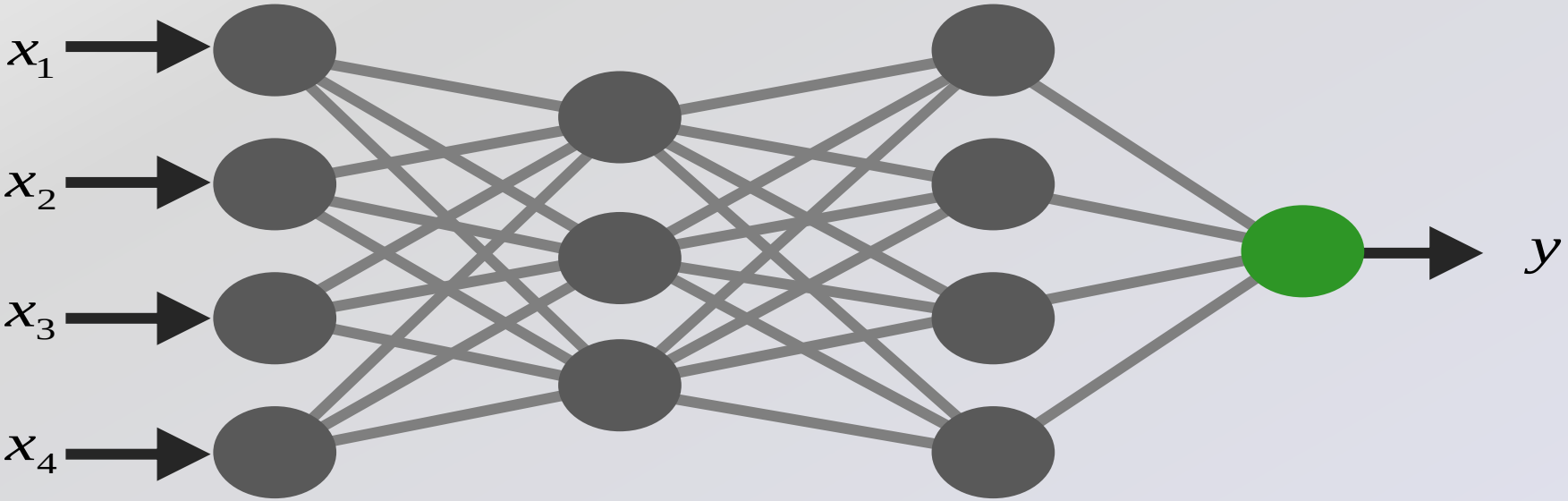
Input neurons represent the features of the dataset.



Input Neurons

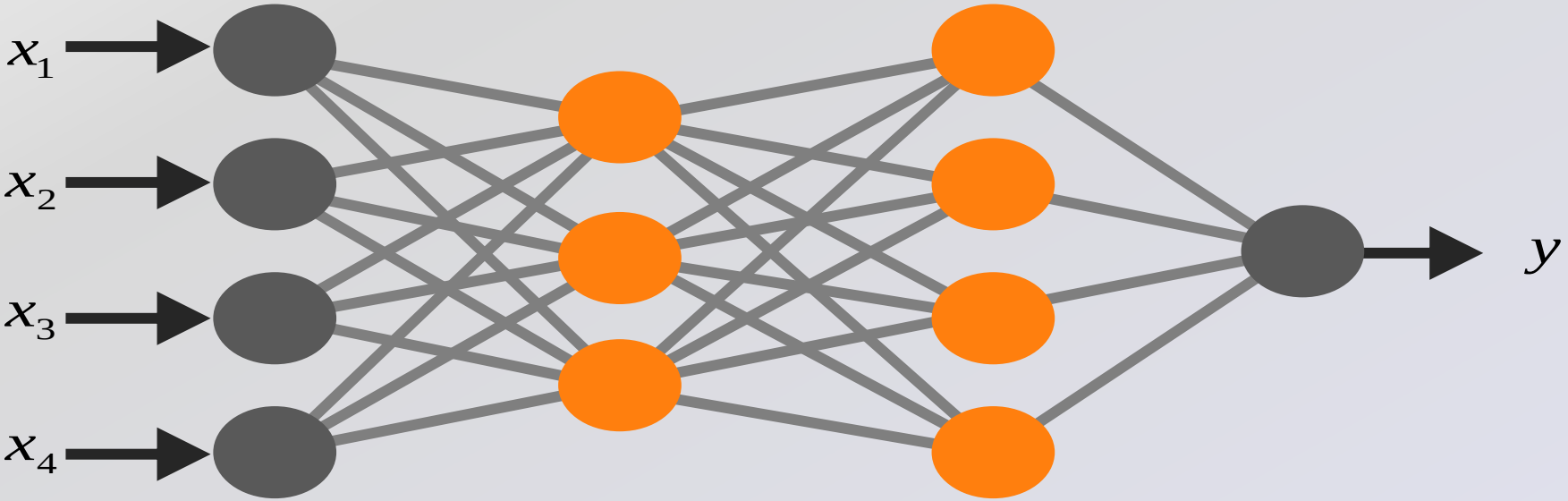
Output Layer

For supervised learning, the output neuron represents the response variable(s).



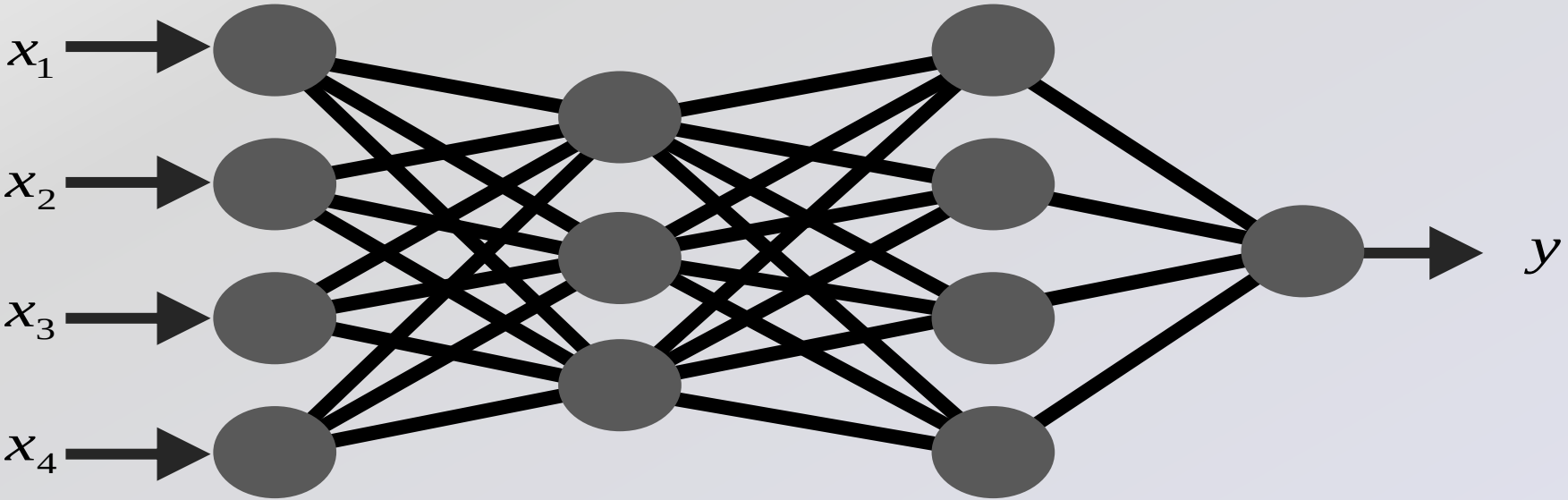
Hidden Layers

Hidden layer are what allow the network to fit highly non-linear patterns.



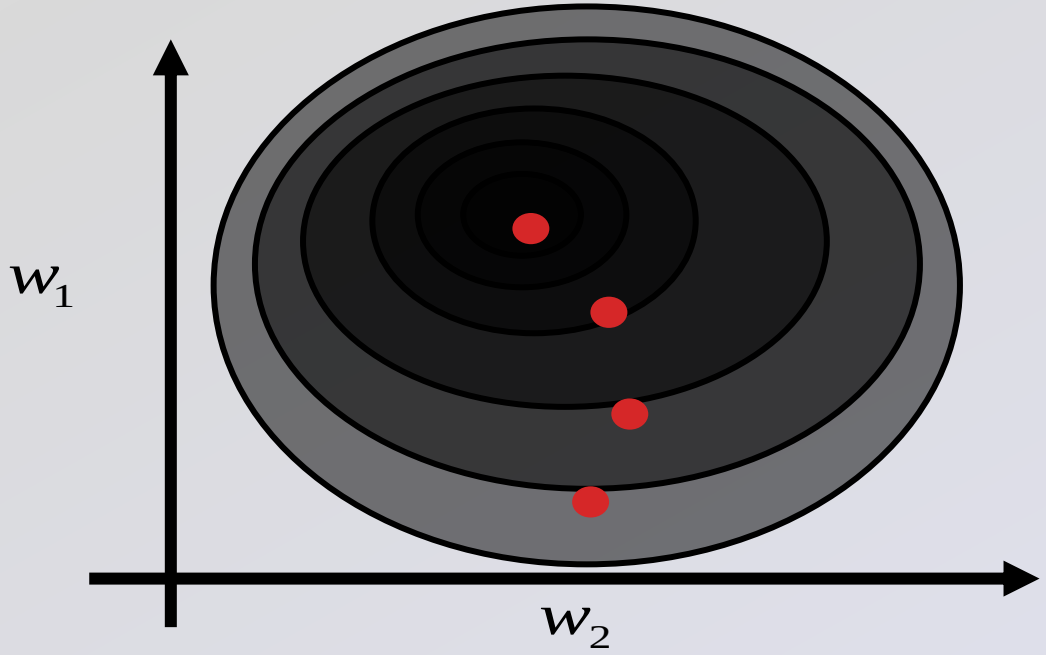
Weights

Connections between nodes are defined by a weight value which determines the influence of one neuron on another.



Gradient Descent - Intuition

Weights are updating according to an algorithm called the gradient descent, an iterative procedure for finding the optimal parameters for a given model.



Meta-Evaluation

Advantages:

- Neural networks give amazing flexibility to fit complex and non-linear functions
- For this reason, Neural networks can produce impressive predictive results that exceed any of the algorithms we have previously learned

Disadvantages:

- Neural networks can be computationally expensive, requiring a lot of time / processing power
- Neural networks function as a “black box” with low interpretive value

XI. RECOMMENDERS

A **recommendation system** *will predict a rating that a user will give an item that they have not yet rated.*

This rating is produced by analyzing other user/item ratings (and sometimes item characteristics) to provide personalized recommendations to users.

There are two general approaches to recsys design:

In **content-based filtering**, *items are mapped into a feature space, and recommendations depend on item characteristics.*

In contrast, the only data under consideration in **collaborative filtering** *are user-item ratings, and recommendations depend on user preferences.*

Content-based filtering has some difficulties:

- need to map each item into a feature space (usually by hand!)
- recommendations are limited in scope (items must be similar to each other)
- hard to create cross-content recommendations (eg books/music films...this would require comparing elements from different feature spaces!)

Collaborative filtering *refers to a family of methods for predicting ratings where instead of thinking about users and items in terms of a feature space, we are only interested in the existing user-item ratings themselves.*

The idea is that users get value from recommendations based on other users with similar tastes.

XII. CATEGORICAL VARIABLES

Q: How do we deal with categorical variables? (i.e., with k levels)

A: Create a k-1 binary ("dummy") variables.

Major (k=4)
Computer Science
Engineering
Business
Literature
Business
Engineering

Engineering	Business	Literature
0	0	0
1	0	0
0	1	0
0	0	1
0	1	0
1	0	0

Computer Science is the reference

Q: Why k-1 and not k?
A: Because k-1 captures all possible outputs, and to avoid multicollinearity.

Q: Does it matter which factor level I leave out?
A: Yes, this is the reference point for all other factor levels.

Q: Is this a limitation?
A: Not really, a comparison must have a baseline.

XII. MODEL EVALUATION PROCEDURES

Q: What's wrong with training error?

Thought experiment:
Suppose we train our model using the entire dataset.
Q: How low can we push the training error?

- *We can make the model arbitrarily complex (effectively "memorizing" the entire training set).*

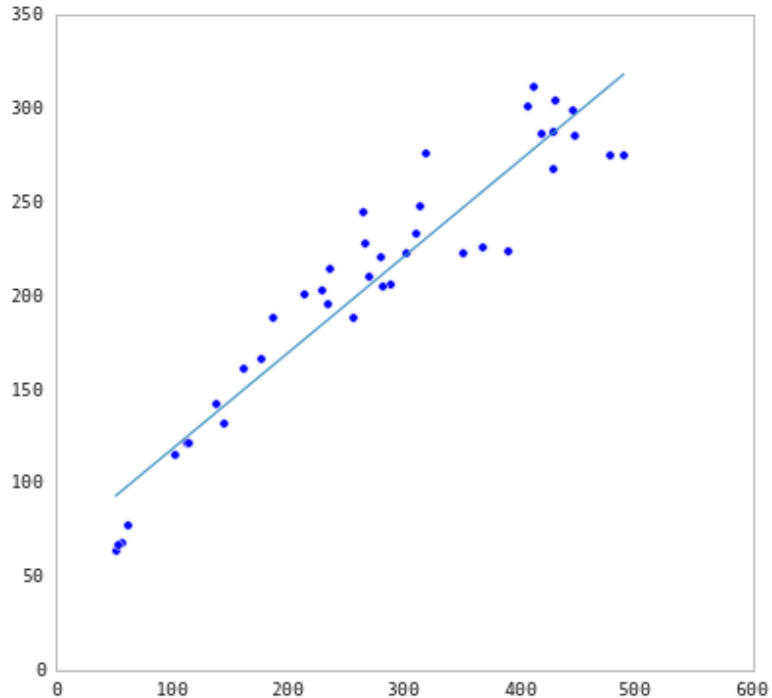
A: Down to zero!

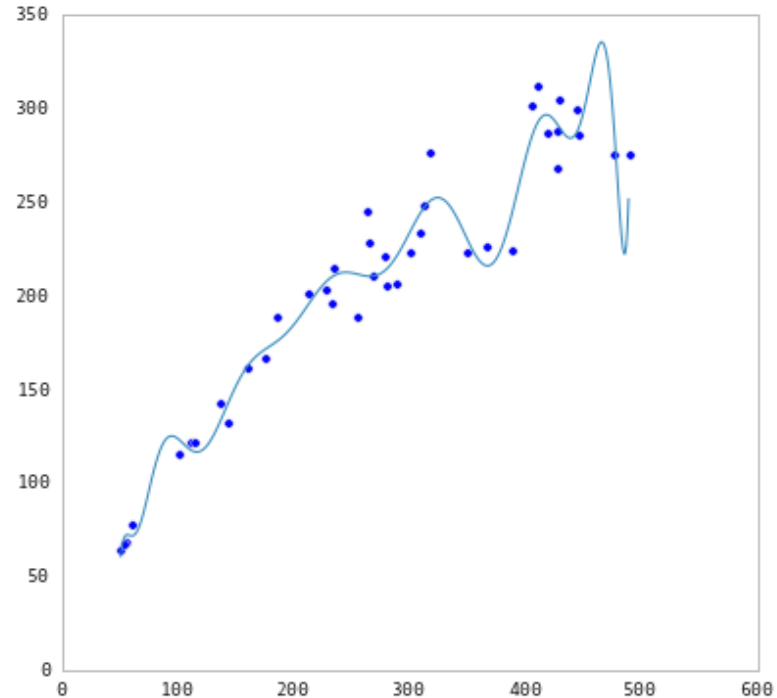
NOTE

This phenomenon is called *overfitting*.

A: Training error is not a good estimate of accuracy beyond training data.

UNDERFITTING AND OVERFITTING

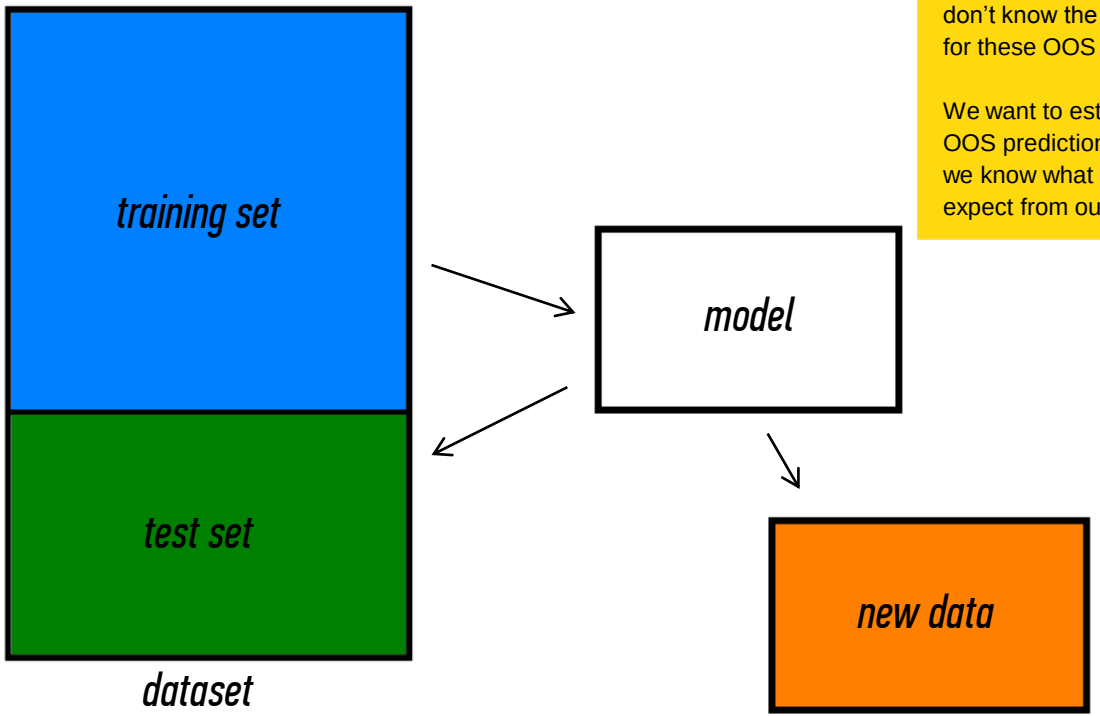




Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

Q: How can we make a model that generalizes well?

1) *split dataset*
2) *train model*
3) *test model*
4) *parameter tuning*
5) *choose final model*
6) *train on all data*
7) *make predictions*



NOTE

This new data is called *out of sample* data. We don't know the labels for these OOS records!

We want to estimate OOS prediction error so we know what to expect from our model.

Suppose we do the train/test split.
Q: How well does test set error predict OOS accuracy?

Thought experiment:
Suppose we had done a different train/test split.
Q: Would the test set error remain the same?
A: Of course not!

A: On its own, not very well.

NOTE

The test set error gives a *high-variance estimate* of OOS accuracy.

Something is still missing!

Q: How can we do better?

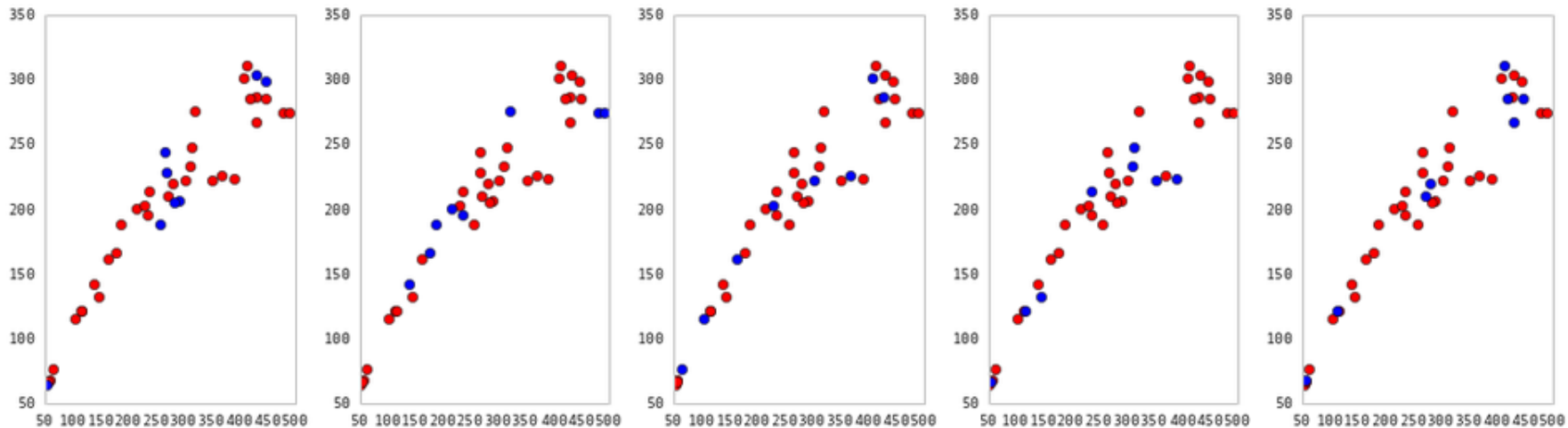
Thought experiment:
Different train/test splits will give us different test set errors.
Q: What if we did a bunch of these and took the average?
A: Now you're talking!

A: Cross-validation.

Steps for K-fold cross-validation:

1) *Randomly split the dataset into K equal partitions.*
2) *Use partition 1 as test set & union of other partitions as training set.*
3) *Calculate test set error.*
4) *Repeat steps 2-3 using a different partition as the test set at each iteration.*
5) *Take the average test set error as the estimate of OOS accuracy.*

CROSS-VALIDATION



5-fold cross-validation: red = training folds, blue = test fold

Source: http://nbviewer.ipython.org/github/fonnesbeck/Bios366/blob/master/notebooks/Section6_3-Model-Selection-and-Validation.ipynb

Features of K-fold cross-validation:

1) *More accurate estimate of OOS prediction error.*
2) *More efficient use of data than single train/test split.*
   - *Each record in our dataset is used for both training and testing.*
3) *Presents tradeoff between efficiency and computational expense.*
   - *10-fold CV is 10x more expensive than a single train/test split*
4) *Can be used for parameter tuning and model selection.*

XIV. MODEL EVALUATION METRICS

Classification:

- *Confusion Matrix*
- *ROC Curve (and AUC)*

Regression:

- *Root Mean Squared Error*

Confusion Matrix: table to describe the performance of a classifier

n=165	Actual: YES	Actual: NO
Predicted: YES	100	10
Predicted: NO	5	50

Example: Test for presence of disease

YES = positive test = True = 1

NO = negative test = False = 0

- *How many classes are there?*
- *How many patients?*
- *How many predictions of disease?*
- *How many patients actually have the disease?*

Email Number	Score	True Label
5	0.93	Spam
8	0.91	Spam
2	0.84	Spam
1	0.6	Ham
7	0.54	Spam
3	0.22	Ham
4	0.10	Ham
6	0.02	Ham

Every email gets a spamminess score.

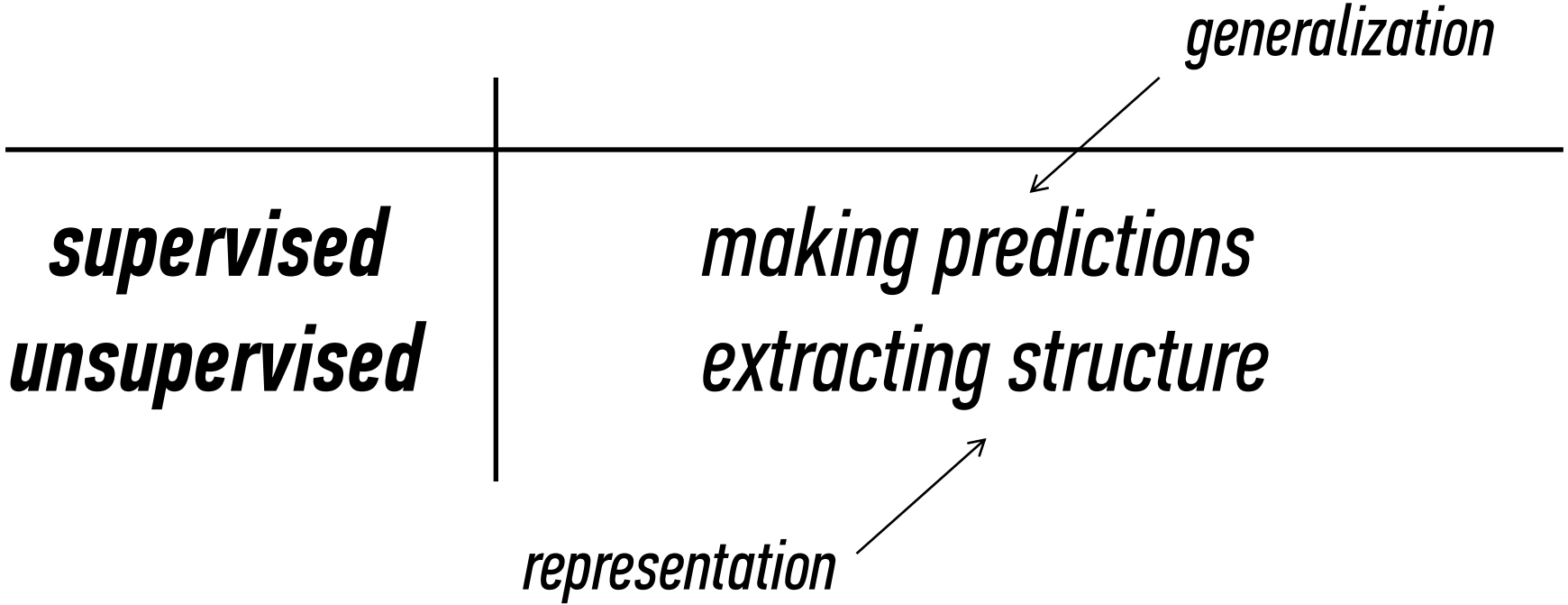
Choosing a cut-off, this becomes a classification.

How do we choose a cut-off?
How do we evaluate the ranking without choosing a cut-off?

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- *Used for regression problems*
- *Square root of the mean of the squared errors*
- *Easily interpretable (in the "y" units)*
- *"Punishes" larger errors*

XV. UNSUPERVISED LEARNING


generalization
supervised
unsupervised
making predictions
extracting structure
representation

	continuous	***categorical***
supervised	*regression*	*classification*
unsupervised	*dimension reduction*	*clustering*

- No response variable y, just a set of predictors X
- Objective is more fuzzy:
  - Find groups of observations that behave similarly
  - Find predictors that behave similarly
  - Find linear combinations of features that explain most of the variation in the data
- Difficult to evaluate how well you are doing
- Data is easier to obtain for unsupervised learning since it can be "unlabeled" (i.e., it hasn't been labeled with a response)
- Sometimes useful as a preprocessing step for supervised learning
- Common techniques: clustering, principal components analysis

XVI. CLUSTER ANALYSIS

Q: What is the purpose of cluster analysis?

A: To enhance our understanding of a dataset by dividing the data into groups.

Clustering provides a layer of abstraction from individual data points.

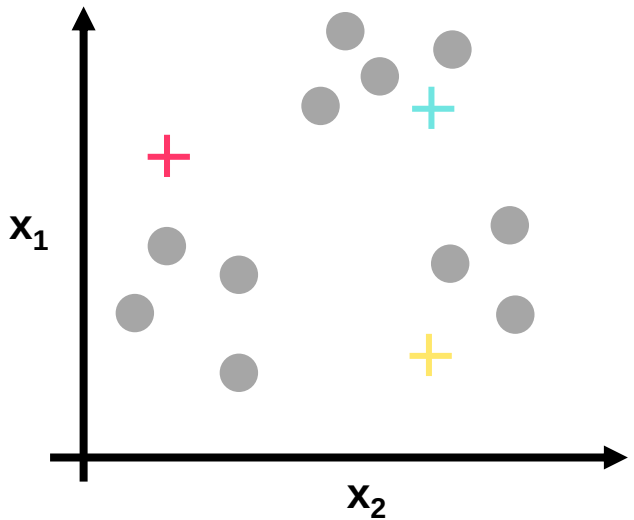
The goal is to extract and enhance the natural structure of the data

1) *choose k initial centroids (note that k is an input)*

2) *for each point:*
   - *find distance to each centroid*
   - *assign point to nearest centroid*

3) *recalculate centroid positions*

4) *repeat steps 2-3 until stopping criteria met*



In general, k-means will converge to a solution and return a partition of k clusters, even if no natural clusters exist in the data.

We will look at two validation metrics useful for partitional clustering, **cohesion** *and* **separation**.

Strengths:

K-means is a popular algorithm because of its computational efficiency and simple and intuitive nature.

Weaknesses:

However, K-means is highly scale dependent, and is not suitable for data with widely varying shapes and densities.

XVII. DIMENSION REDUCTION

Q: What is dimensionality reduction?

A: A set of techniques for reducing the size (in terms of features, records, and/or bytes) of the dataset under examination.

In general, the idea is to regard the dataset as a matrix and to decompose the matrix into simpler, meaningful pieces.

Q: What are the motivations for dimensionality reduction?

A: The number of features in our dataset can be difficult to manage, or even misleading (eg, if the relationships are actually simpler than they appear).

Q: How is dimensionality reduction performed?

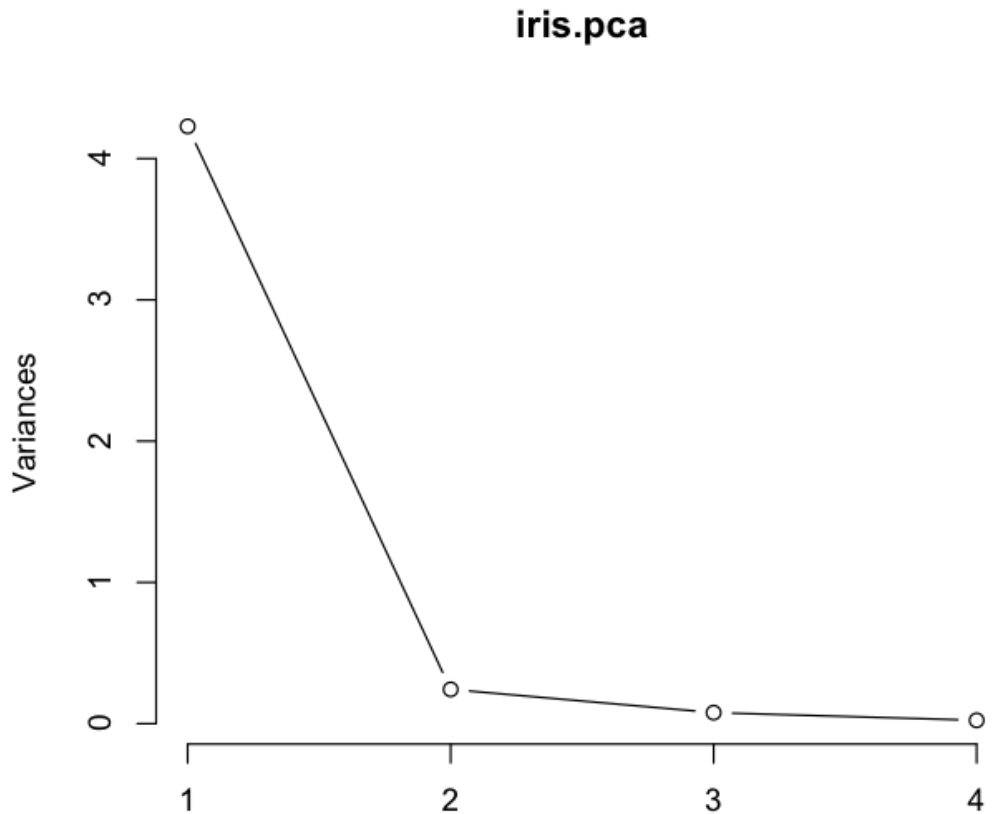
A: There are two approaches: feature selection and feature extraction.

feature selection – *selecting a subset of features using an external criterion*

feature extraction – *mapping the features to a lower dimensional space*

Feature selection is important, but typically when people say dimensionality reduction, they are referring to feature extraction.

The goal of feature extraction is to create a new set of coordinates that simplify the representation of the data.



NOTE

Looking at this plot also gives you an idea of how many principal components to keep.

Apply the *elbow test*: keep only those pc's that appear to the left of the elbow in the graph.