Machine Learning Algorithms

Introduction

Motivating questions

- Why is dimensionality reduction important?
- How can we group records together without labels to inform predictions using unsupervised classification?
- What are some commonly used examples of supervised machine learning algorithms?
- How do we measure the performance of these algorithms?

Dimensionality reduction

Review of unsupervised learning

- Finding and extracting patterns from multidimensional data is known as unsupervised learning
  - Learning – inferring previously unknown patterns from the dataset
  - Unsupervised – no reference to the class or outcome labels because correct answer is unknown

Dimensionality reduction

- Dimensionality reduction is a term for reducing the number of features included in analysis

- Why?
  - Results need to be interpreted by humans, should be tractable
  - Increasing the number of features included in an analysis increases the required sample size exponentially
    - Height, weight, education level, income versus a study of height and weight
    - Height and weight are correlated and education level and income are correlated, how can this be used?

A word to the wise

- Trying to reduce dimensionality randomly or manually can lead to poor results

Source: Pexels

A word to the wise

- Features should not be included or discarded from analysis based on instinct, dimensionality reduction techniques should be employed, such as principal component analysis (PCA)

Source: Pexels

Summary statistics as a means of dimensionality reduction

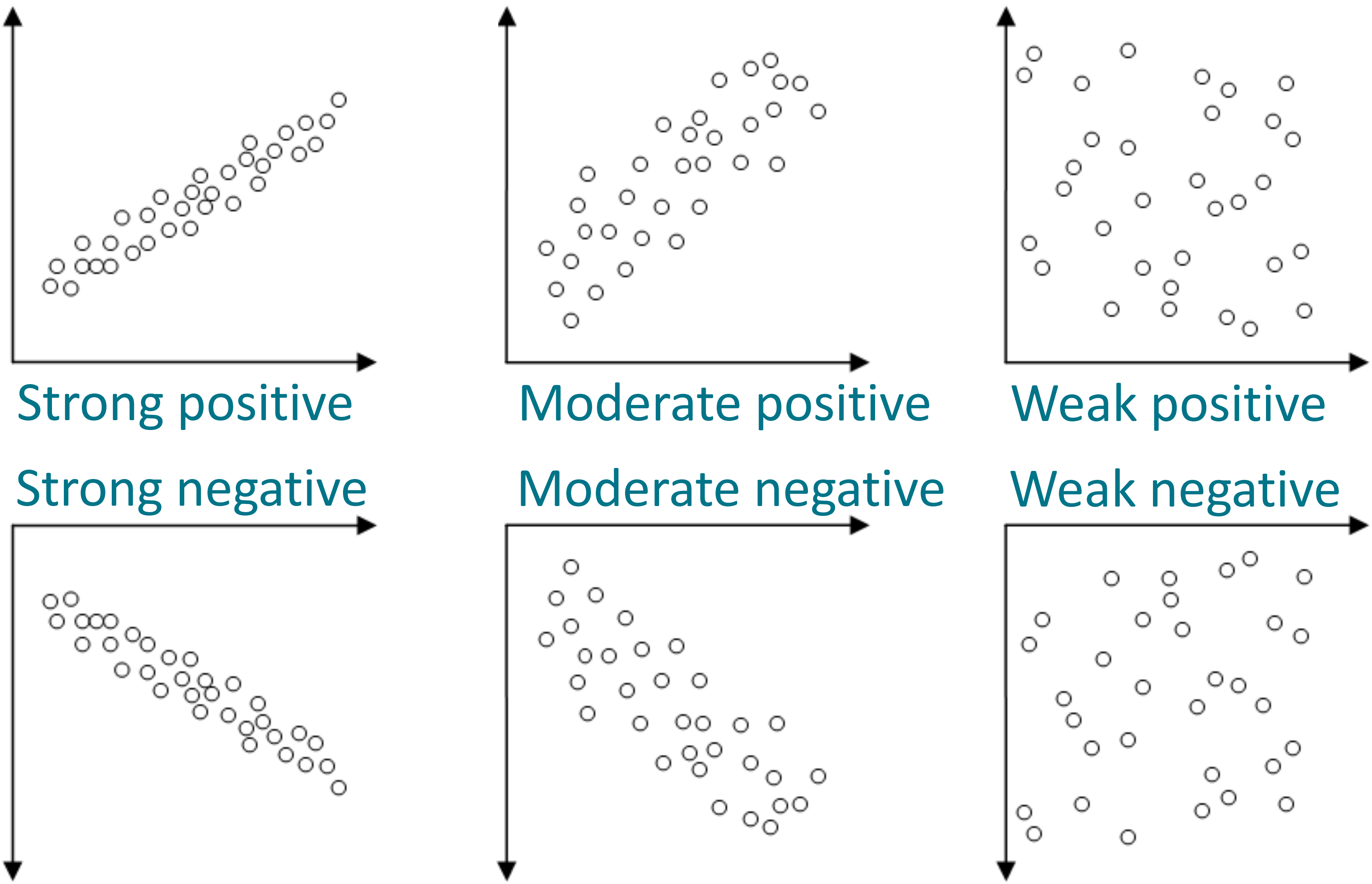
- Dimensionality can also be reduced with a summary statistic to describe a large number of data points with one or a few number(s)
  - Statistics can be misleading for all but the most common distributions (normal)
    - Average city in the US has ~8,000 people
    - Average human on Earth is female
    - Crime rates in small cities (highly variant)

Key points from lesson

- Dimensionality reduction is often needed for analysis with many features
- There is a need for the ability to confidently reduce the number of features included in an analysis without losing information
- Recall height, weight, education level, income example: correlations between attributes should be leveraged

Principal component analysis

Recall: correlation of features



Source: Wikimedia

Principal component analysis (PCA)

- PCA is a mathematical approach to reduce dimensionality for analysis or visualization

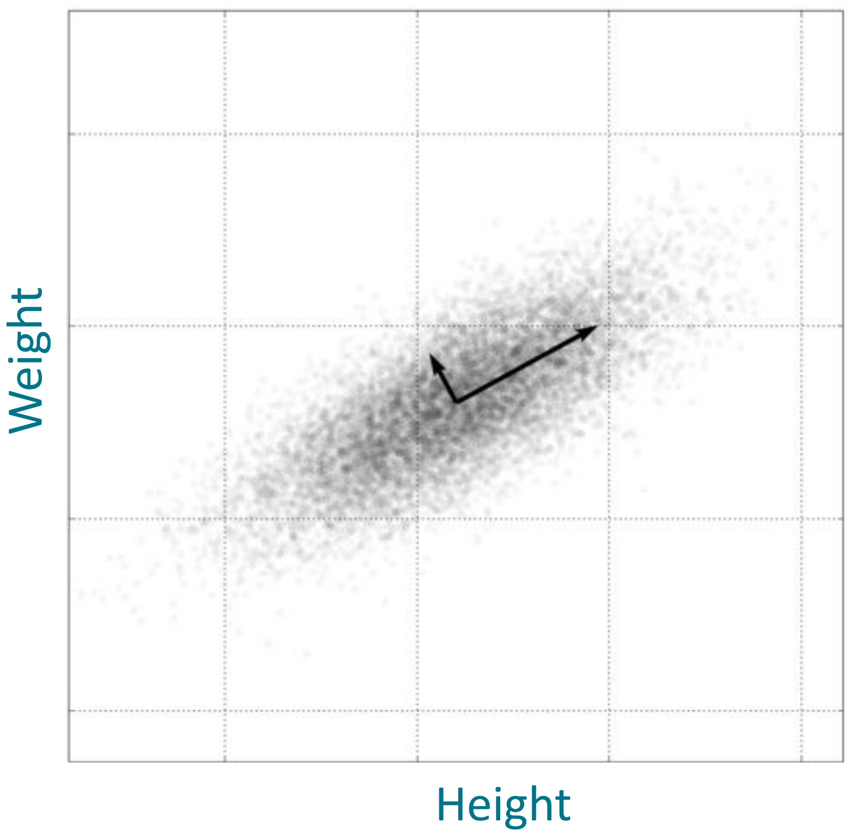
- PCA exploits correlations to transform the data such that the first few dimensions or features contain a majority of the information or variance in the dataset

- For example:
  - Heights and weights are highly correlated in most individuals
  - Income is much less strongly correlated with height of most individuals

Combining height and weight

- Height and weight are correlated, and the linear regression solution describes much of the information about the dataset, shown with the long arrow – this would be the first principal component

- The smaller arrow is still informative in terms of differentiating individual points, so it forms the second principal component



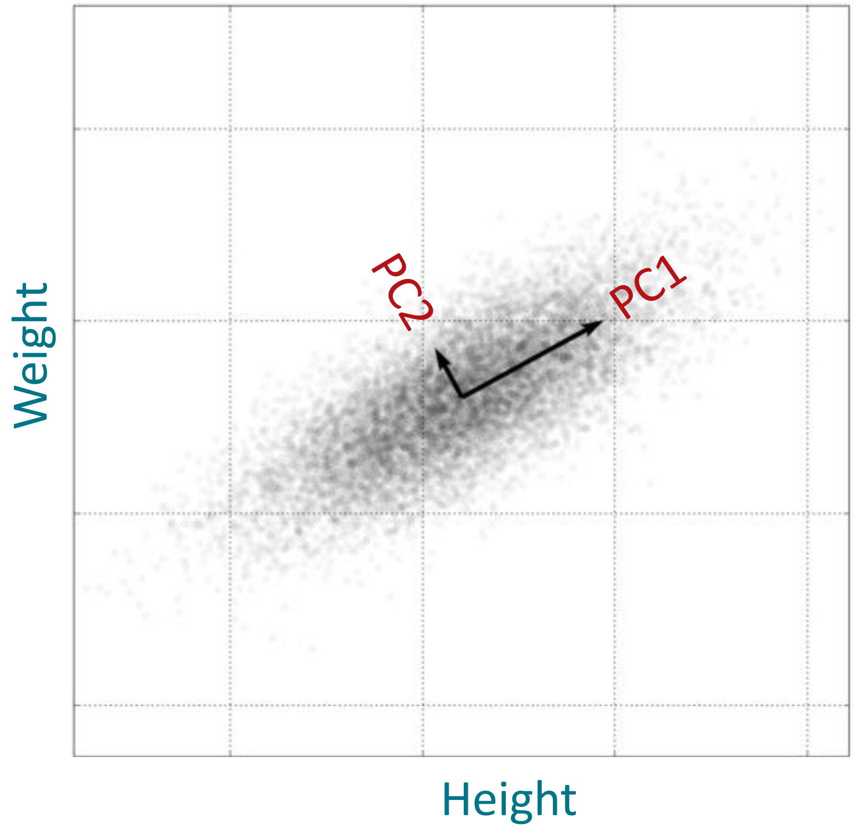
Source: Wikimedia

PCA linear transformations

- PCA determines which variables are most informative based on the distribution of data
- PCA calculates the most informative combinations of existing variables within the dataset:

PC1 = (*a*var1 + b*var2 + c*var3 + …*)
PC2 = (*d*var1 + e*var2 + f*var3 + …*)
PC3 = (*g*var1 + h*var2 + i*var3 + …*)

- No information is lost, first few components hold much of the information



Source: Wikimedia

How PCA works

- Same premise as linear regression, except without a dependent variable
  - Linear regression solution is the first principal component
  - Disregarding the information described by the first principal component, PCA calculates the second most informative component, then the third, and so on

- These linear combinations form a new set of variables which can be used to view the data – new axes

- Components are ranked by importance, so all but the first few can discarded, leaving only most important information with very few components

Example

- http://setosa.io/ev/principal-component-analysis/

Interpreting the transformation

- The coefficients shown in the table give the proportion of each of the original variables that went into each component
- Relative signs +/- indicate that two variables are positively or negatively correlated in that particular component
- The components are difficult to interpret using only the coefficient values, plotting often improves understanding

$$PC1 = (a*var1 + b*var2 + c*var3 + \ldots)$$
$$PC2 = (d*var1 + e*var2 + f*var3 + \ldots)$$
$$PC3 = (g*var1 + h*var2 + i*var3 + \ldots)$$

Component	var1	var2	var3
PC1	-0.522	0.263	-0.581
PC2	-0.372	-0.926	-0.021
PC3	-0.002	-0.010	0.011

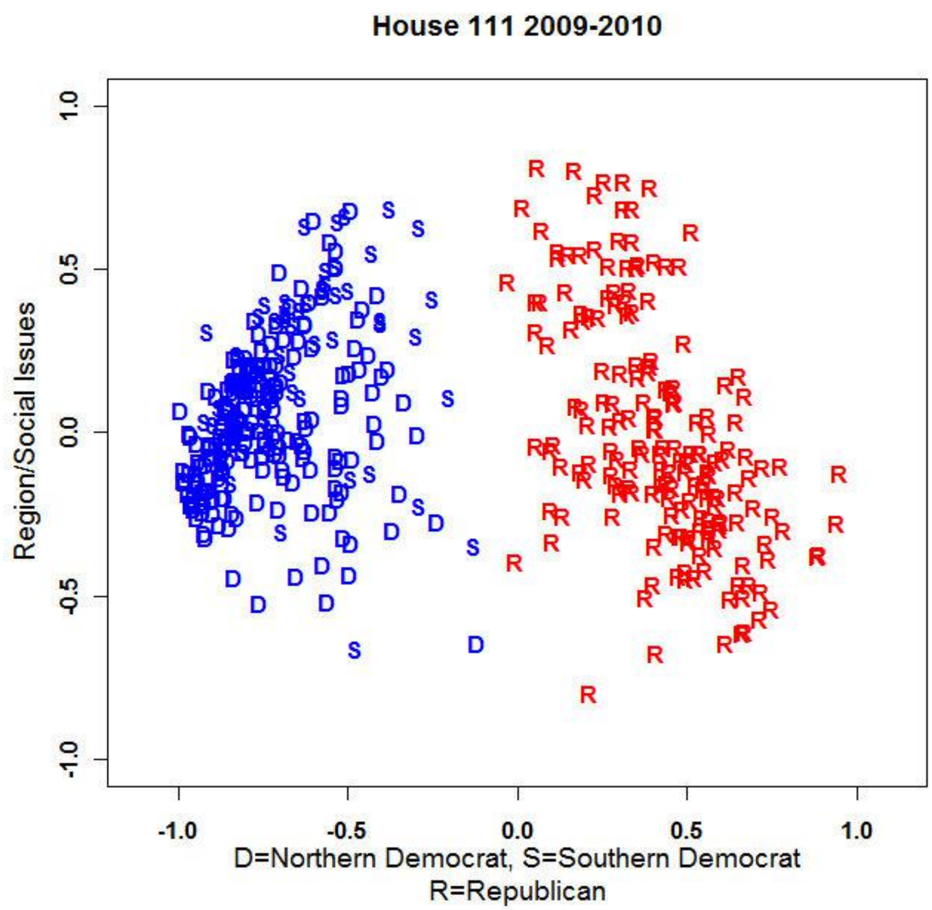
Compelling cases for PCA

- With the height and weight example, PCA did not contribute to understanding
  - With only 2 dimensions, can get a complete understanding of the data from visualization

- PCA works well for datasets with high dimensionality:
  - Political data (each politician has cast hundreds of votes during career)
  - Medical data (each patient has tens or hundreds of status markers describing conditions and treatments)
  - Survey data (each respondent answers tens of questions)

Example: identifying unique voting patterns

- DW-NOMINATE uses a technique similar to PCA on vote history from members of US Congress
- The vast majority of the information is carried by the first component shown on the x-axis, which seems to represent political party
  - Political party was not known a priori!
- The second component is not as informative about voting patterns as party, but it seems to represent positions on social policy/privacy/civil rights (differences within parties)



Key points from lesson

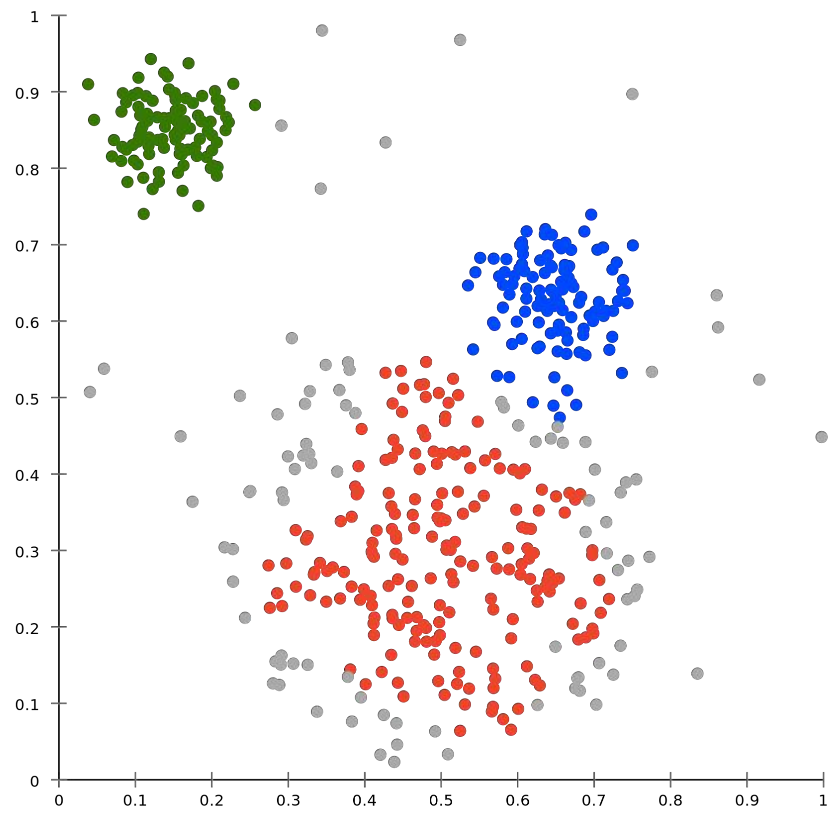
- Principal component analysis can be used to programmatically reduce dimensionality when there are too many dimensions to visualize or understand

- PCA is used to identify relationships or patterns among the features in a dataset by calculating the most informative combinations of variables that account for most of the variance in a dataset

Clustering

Introduction to clustering

- Another way of thinking about dimensionality reduction: how close is each point to each other point?
- Idea: separate data points into a number of clusters that have less distance between the points internally than to other clusters



Source: Wikimedia

k-means clustering

- *k*-means clustering starts with selecting the number of clusters, *k*
- *k* cluster-centers are placed randomly in the data space and then the following stages are performed repeatedly until convergence:
  - Data points are classified by the center to which they are nearest
  - The centroid of each cluster is calculated
  - Centers are updated to the centroid location

Source: Wikimedia

k-means demonstration

- Visualizing k-Means.html

k-means caveats

- *k*-means does not determine the appropriate number of clusters, this is set by the user based on intuition or previous knowledge of the data

- Algorithm can terminate with multiple solutions depending on initial random positions of cluster-centers and some solutions are better than others

Key points from lesson

- Clustering can be helpful to identify groups of records that have similar characteristics to one another
- When data is unlabeled, clustering can be used to group records together for deeper inspection
- Upon deeper inspection of the records in each cluster, users can understand the patterns that lead to records being grouped together, and also identify reasons for records being grouped separately

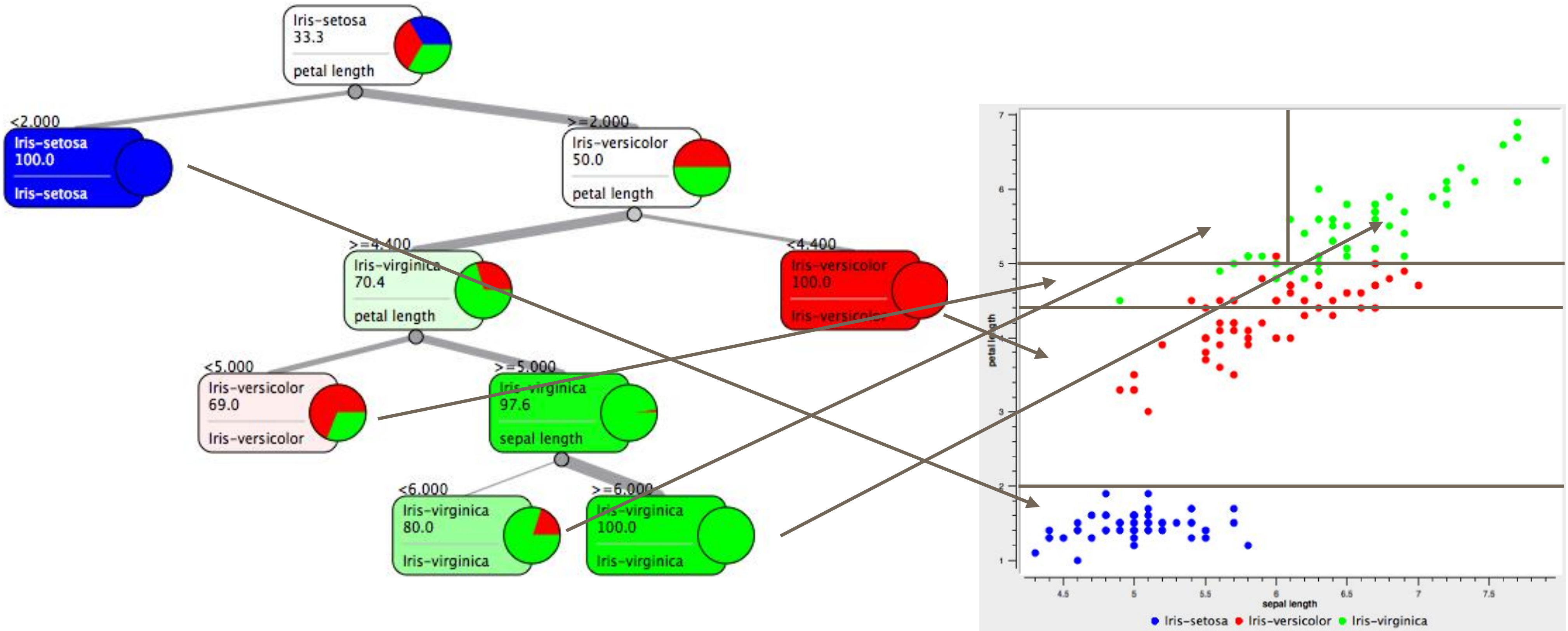
Classification

Recall supervised learning

- Clustering and PCA allow users to see patterns in the data, which is the best that can be done because there are no labels to guide the analysis
- With supervised learning, the label is included in the learning process:
  - Unsupervised: what features are most important or interesting?
  - Supervised: what features are most informative about the differences between these groups?
- Classification methods: each record falls into some category or class, predict the category of a new record based on values of other features in the record
- Regression methods: one variable depends on some or all of others, predict the value of the dependent variable based on the values of the independent variables

Classification trees

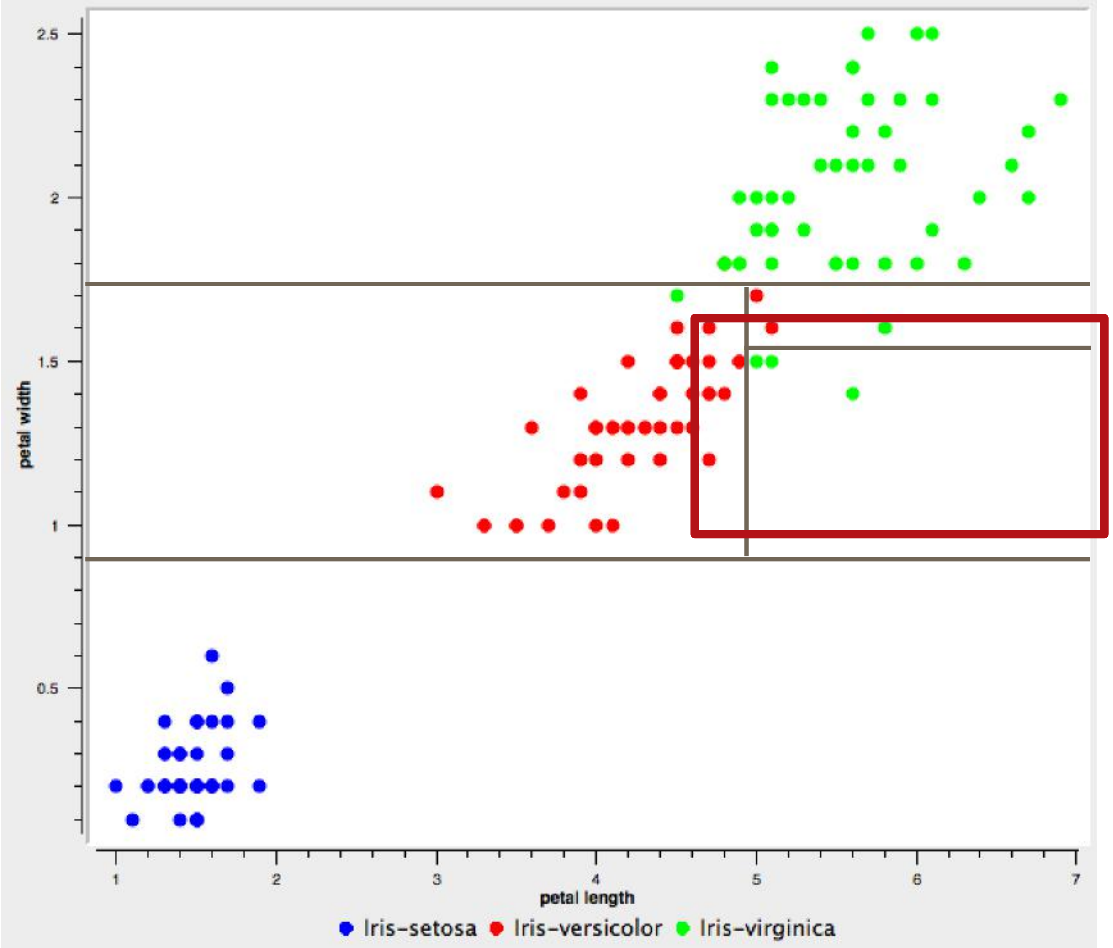
- Classification trees split data find optimal values for features, used to split data by class, recall the San Francisco and New York housing example
- Tree diagram shows the class makeup of each node, and the relative number of data points that reach each node



Overfitting Example

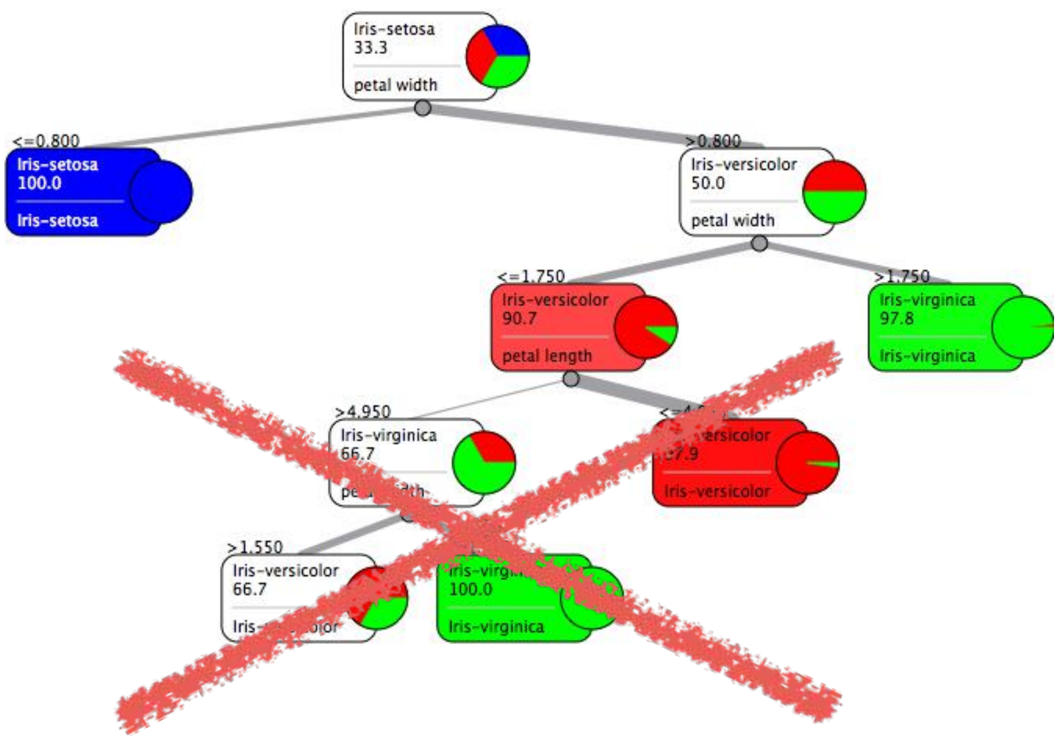
- The center region on the right is questionable, there are two rules relating to six records
  - Are these rules really meaningful?

- There are too few data points in this region to find a true pattern



Tree pruning

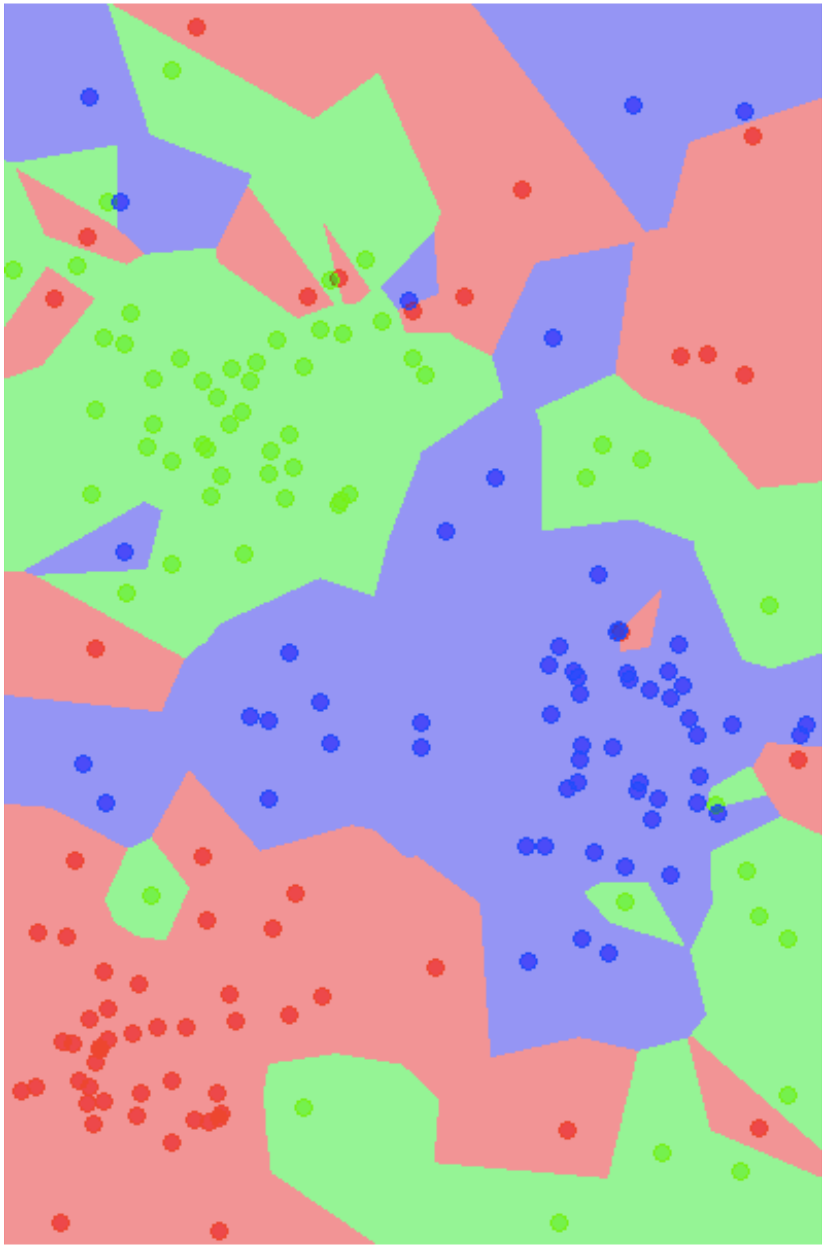
- Tree pruning removes rules associated with overfitting from the tree
- The new tree misses a few points classified correctly, but contains only meaningful rules, more generalizable to new data



k-nearest neighbors

- Another way to classify a data point is by taking a vote among its closest neighbors in data space

- In *k*-nearest neighbors: a new data point is assigned the class of the plurality of its nearest neighbors in the training set, considering the nearest *k* neighbors



Source: Wikimedia

k-nearest neighbors demonstration

- KNN Visualization.html

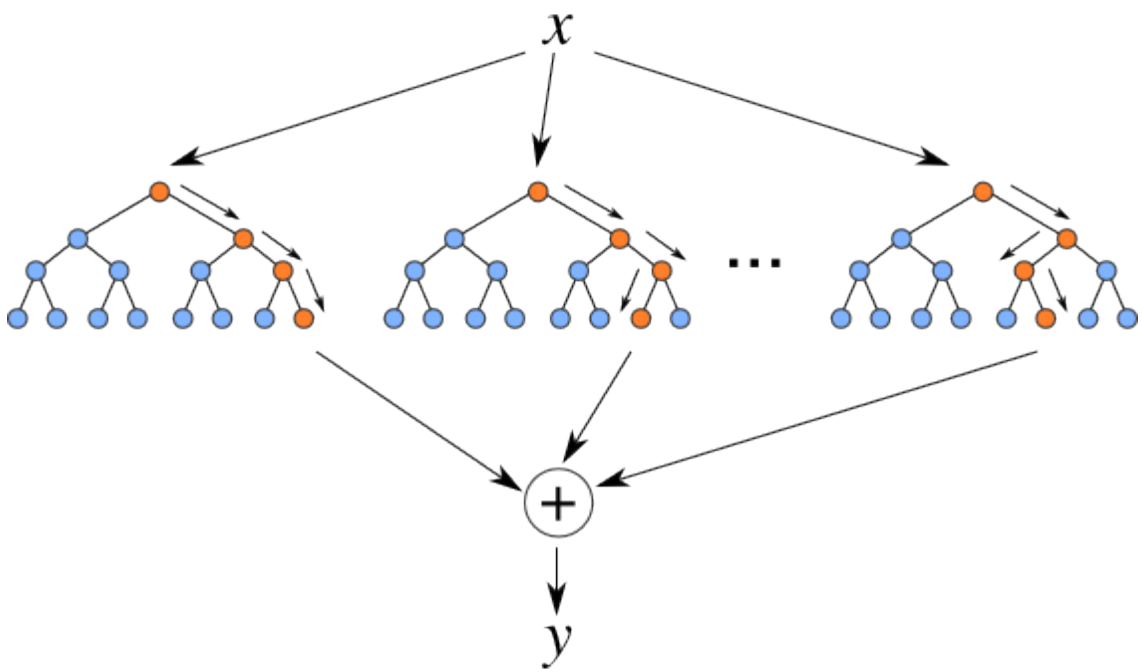
Naïve Bayes classifier

- The Naïve Bayes algorithm considers the value of each feature independently, for each record, and computes the probability that a record falls into each category
  - What is the probability that the member of congress is a democrat given that they voted Yes on issue 1? And what is the probability that the member of congress is a republican given that they voted Yes on issue 1?

- Next, the probabilities associated with each feature are combined for each class according to Bayes' rule to determine the most likely category for each new record
- Almost completely immune to overfitting
  - Individual points have minimal influence
  - Very few assumptions are made about the data

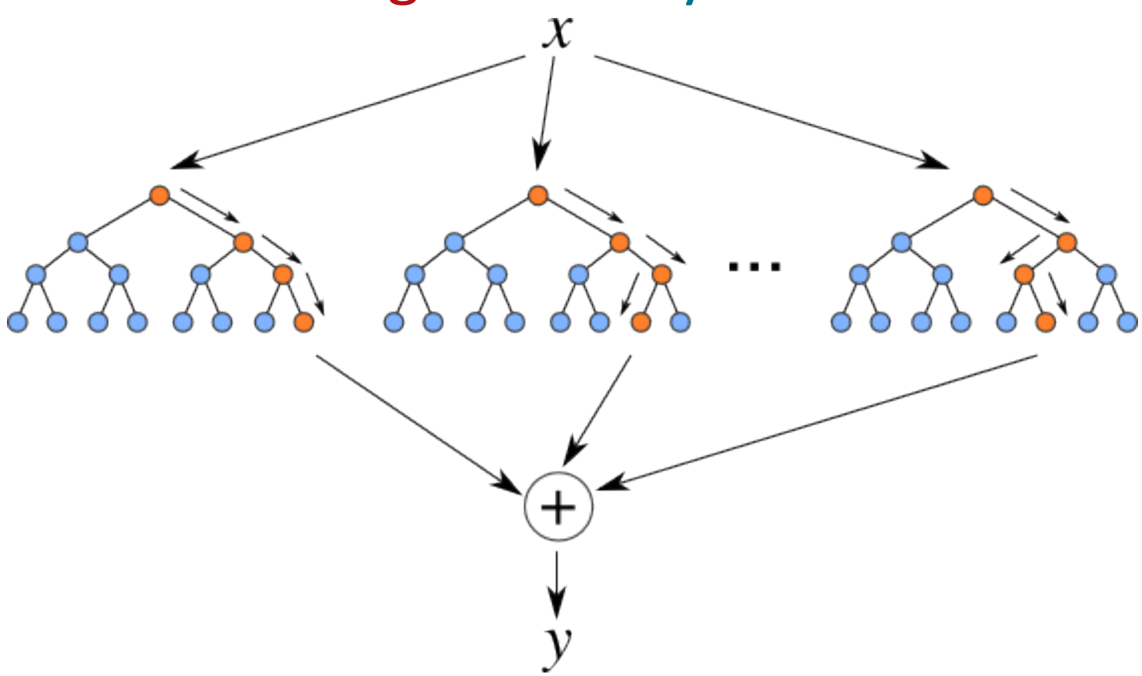
Random forest

- Random forest is an ensemble classifier that uses multiple different classification trees
  - Trees are generated using random samples of records in the original training set
- Accuracy and information about variable importance is provided with the result



Random forest

- Similar to decision tree with a few differences including:
  - No pruning necessary
  - Trees can be grown until each node contains very few observations
  - Better prediction than classification trees, generally
  - No parameter tuning necessary with random forest



Source: https://kgpdag.wordpress.com/

Key points from lesson

- Some methods of classification are more prone to overfitting than others
- The best methods are those that are least influenced by individual data points, however it is important to not miss important information
- Tradeoff between being insensitive to noise and being sensitive to signal, variance vs. bias

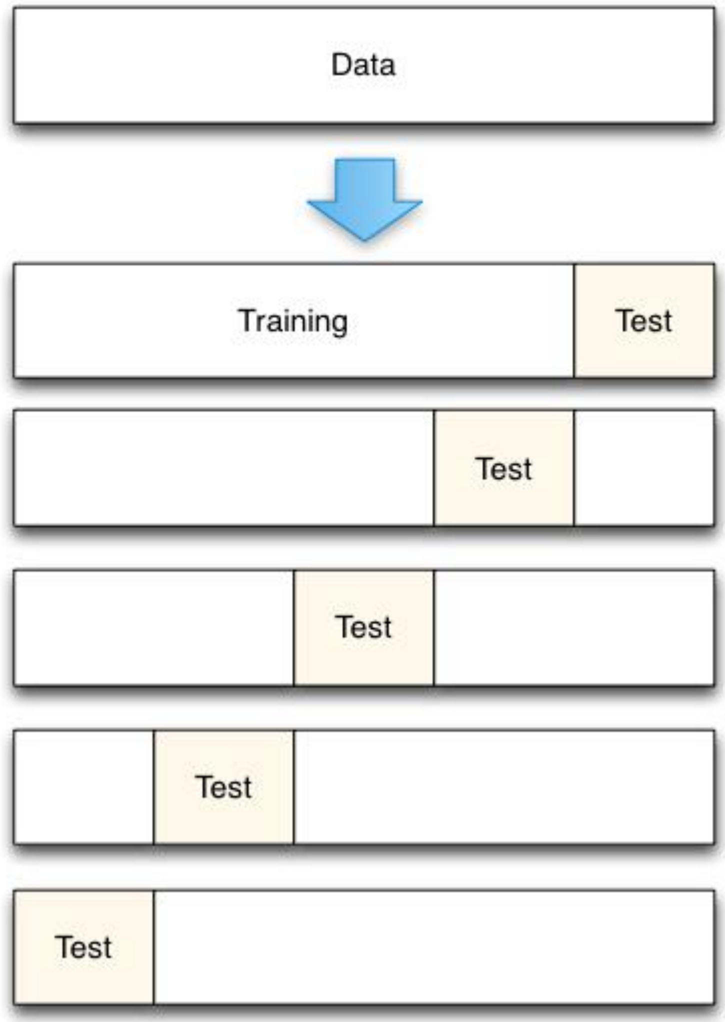
Comparing predictor accuracy

Cross-validation

- Models should be good at making classifications of unlabeled data, not describing data that is already classified

- Randomly divide data into a training set and a test set
  - Hide test set while building the tree
  - Hide training set while calculating accuracy
  - Computed accuracy represents accuracy on unseen data

- Techniques are available to do this multiple times, ensuring each record is in the test set exactly once, e.g. k-folds



Source: blog.kaggle.com

Comparing models

- Several standard measure of performance exist, can run multiple models and compare metrics:
  - Accuracy
  - Precision
  - Recall
  - And more

- Application drives which performance metrics are most important for a given task
  - It is more important to detect all people with cancer than to correctly classify people without cancer

Sensitivity and specificity

Test	Disease Present	n	Disease Absent	n	Total
Positive	True Positive (TP)	*a*	False Positive (FP)	*c*	*a + c*
Negative	False Negative (FN)	*b*	True Negative (TN)	*d*	*b + d*
Total		*a + b*		*c + d*	

Sensitivity $\frac{a}{a + b}$

Specificity $\frac{d}{c + d}$

Positive Likelihood Ratio $\frac{\text{Sensitivity}}{1 - \text{Specificity}}$

Negative Likelihood Ratio $\frac{1 - \text{Sensitivity}}{\text{Specificity}}$

Positive Predictive Value $\frac{a}{a + c}$

Negative Predictive Value $\frac{d}{b + d}$

Source: medcalc.org

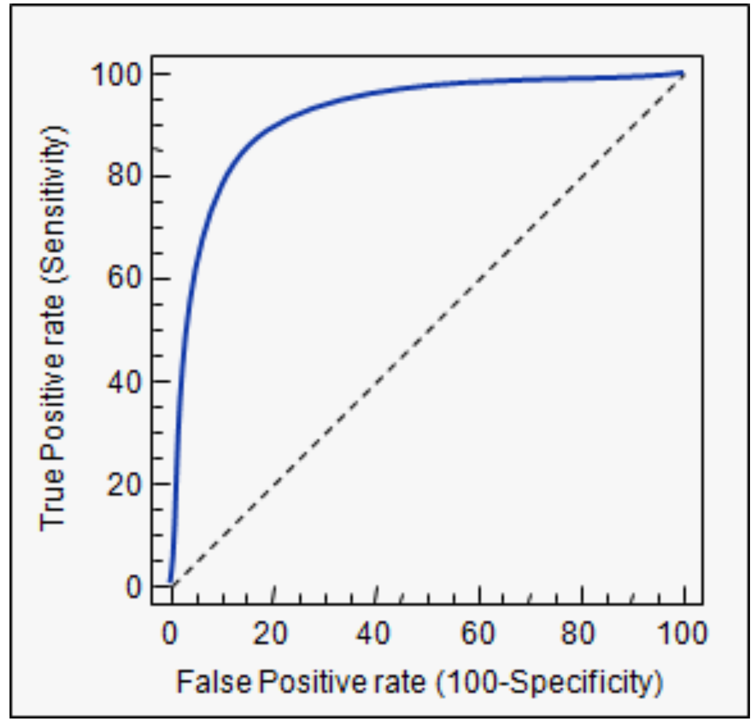
The ROC curve

- The Receiver Operating Characteristic (ROC) curve plots the true positive rate (Sensitivity) versus the false positive rate (100 - Specificity) for different cut-off points

- Each point on the curve represents a pair of sensitivity/specificity values corresponding to a particular decision threshold

- A test with perfect discrimination (no overlap in the two distributions) has an ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity)

- The closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test



Source: medcalc.org

Key points from lesson

- Each model has different properties and is best for different types of tasks —compare them with performance metrics
- Beware of overfitting!
- If tweaking a tuning parameter results in a dramatic increase in accuracy, the model is probably overfit