



Rensselaer

why not change the world?®

Data Mining

Ahmed Eleish

Data Science – ITWS/CSCI/ERTH-4350/6350,
October 7th, 2026

Tetherless World Constellation
Rensselaer Polytechnic Institute



Contents

- Intro to Data Mining and related concepts
- A scientific use case
- Supervised Learning - Classification
- Unsupervised Learning - Clustering
- Software platforms/libraries

- Project check-in

Data Mining



Data Mining – What it is

- Extracting knowledge from large amounts of data.
- Motivation
 - Our ability to collect data has expanded rapidly.
 - It is impossible to analyze all data manually.
 - Data contains valuable information that can aid in decision making.
- Uses techniques from:
 - Pattern Recognition
 - Machine Learning
 - Statistics
 - Database Systems
- Data mining methods must be efficient and scalable (8~10 years ago, data mining could not be done on your Laptop).

Data Mining – What it isn't

- Small Scale

- Data mining methods are designed for large data sets.
- Scale is one of the characteristics that distinguishes data mining applications from traditional statistical analysis

- Foolproof

- Data mining techniques will discover patterns in any data ..
- The patterns discovered may be meaningless!
- It is up to the user to determine how to interpret the results.
- “Make it foolproof and they'll just invent a better fool.” - someone

- Magic

- Data mining techniques cannot generate information that is not present in the data.
- They can only find the patterns that are already there.

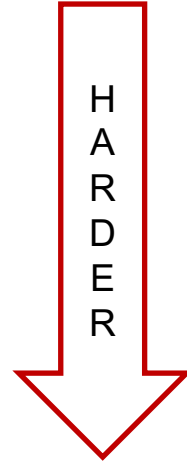
Data Mining – Types of Mining

- Classification (Supervised Learning)
 - Classifiers are trained using labeled training samples.
 - Training samples created by ground truth / experts.
 - Classifiers are then used to classify unknown samples.
- Clustering (Unsupervised Learning)
 - Grouping observations into clusters so that similar objects are in the same cluster and dissimilar objects are in different clusters.
 - Discover structure in a dataset (tendency of data points to cluster).
- Association Rule Mining
 - Initially developed for market basket analysis.
 - The goal is to extract rules from relationships between attributes in the dataset.
 - Uses include decision support, recommender systems, etc.
- Other Types of Mining
 - Outlier Analysis
 - Time Series Analysis

Models/ types

- Trade-off between Accuracy and Understandability
- Models range from “easy to understand” to incomprehensible

- Decision trees
- Rule induction
- Multi-variate Regression models
- Neural Networks
- Deep Learning



Related Concepts

Definitions

- **Data** - are encodings that represent the qualitative or quantitative attributes of a variable or set of variables.
- A **variable** is a quantity, quality, or property that you can measure.
- A **value** is the state of a variable when you measure it. The value of a variable may change from measurement to measurement.
- An **observation**, or a case, is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable.
Sometimes refer to an observation as a data point.
- **Dataset** – a set of data points (observations) collected for a particular purpose.



Types of Data

Type of Data	Level of Measurement	Examples
Categorical	Nominal (categories with no inherent order)	Eye color, ethnicity, academic major
	Ordinal (categories with inherent order)	Grade, academic year
	Binary (2 categories; special case of above)	Electric charge
Quantitative	Discrete (whole numbers)	Size of household (ratio)
	Continuous (can in theory take on any value in a range, although necessarily recorded to a predetermined degree of prevision.	Temperature C°/F° (no absolute zero - interval)
		Age, height (ratio)



Accurate vs. Precise



**High Accuracy
High Precision**



**Low Accuracy
High Precision**



**High Accuracy
Low Precision**



**Low Accuracy
Low Precision**

<http://climatica.org.uk/climate-science-information/uncertainty>

Scientific Use Case

Science Motivation: Botany

- Let's assume a botanist is interested in distinguishing the species of some iris flower samples that were collected. The botanist takes some measurements associated with each iris: length and width of the petals and length and width of sepals.
- The botanist also has access to measurements of some iris flowers that have been previously identified by an expert botanist as belonging to the species:
 - Setosa, Versicolor, and Virginica.
- Our goal is to build a machine learning model that can learn from the measurements of samples whose species is known, to predict the species of a new sample.

A First Application: Classifying Iris Species



https://en.wikipedia.org/wiki/Iris_flower_data_set

Classification Problem..

- Because we have measurements for which we know the correct species of iris, this is a *supervised* learning problem.
- The goal is to assign a class label to an unknown sample making this is a *classification* problem.

Supervised Learning

Supervised Learning - Classification

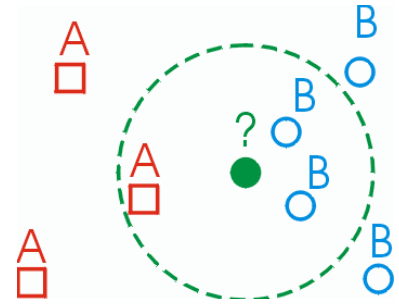
- **Supervised Learning** is a machine learning paradigm where algorithms learn patterns from labeled data, where each observation is associated with a ground truth class label.
- **Classification** is a type of supervised learning with the goal of classifying unknown samples by using patterns learned from **training** data.
- Classifiers are predictive models whose performance is evaluated through a process of **validation**, hence the "supervised" in supervised learning.

k-nearest neighbors (kNN)

- Can be used in both regression and classification (“non-parametric”)
 - Supervised learning, i.e. model is trained
- kNN is a method for classifying objects based on the closest training examples in the feature space.
- **An object is classified by a majority vote of its neighbors. k is always a positive integer.** The neighbors are taken from a set of objects for which the correct classification is known.
- It is usual to use the Euclidean distance, though other distance measures such as the Manhattan distance could in principle be used instead.

kNN Algorithm

- The algorithm on how to compute the k-nearest neighbors is as follows:
 - Determine parameter k = number of nearest neighbors, beforehand. This value **is up to you**.
 - Calculate the distance between the unknown sample and all the training samples. You can use **any distance** algorithm.
 - Sort the distances for all the training samples and determine the nearest neighbors based on the k shortest distances.
 - Since this is supervised learning, get the classes for the k nearest neighbors from the training set.
 - Use the **majority** of nearest neighbors as the prediction value.



Decision Tree

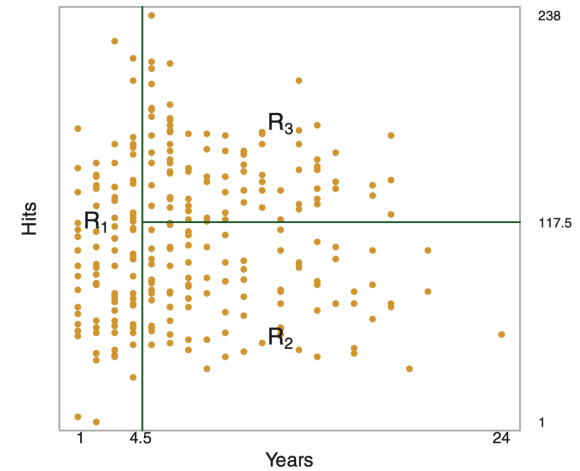
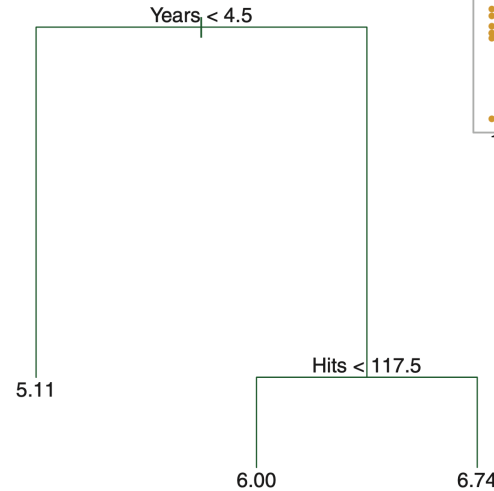
- A decision tree has a hierarchical structure with “nodes” and “directed edges”
- The top node is defined as the “root node” and the nodes at the bottom are called as “leaf nodes”
- Nodes that are neither root node nor the leaf nodes are identified as “internal nodes” in the decision tree.
- There is a “class label” or numerical value associated with each leaf node
- Decision trees can be applied to both regression and classification problems



Image credit: [User:Gokul Jadhav](#)

Decision Tree - Regression

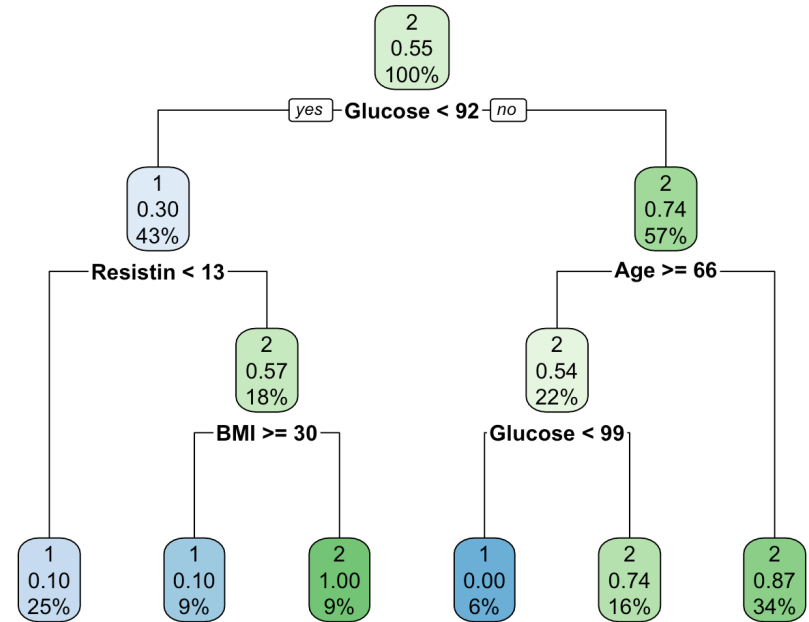
e.g. predicting a player's figure salary (order of magnitude) based on their experience, stats, etc.



Reference/Resources: Introduction to Statistical Learning with R -7 Edition: Chapter 8

Decision Tree – Classification

e.g. predicting an individual's cancer status
based on physiological measurements ..



Decision Tree

- The **depth** is measured from the root node and the depth at the **root** node is **zero**.
- Tree Depth is calculated by counting the number of edges in the longest path from the root node to a leaf node.
- The number of nodes in the decision tree determines the size of the tree.
- **The decision tree constructing algorithm is referred to as a tree induction algorithm.**

Decision Tree

- Classification decisions are made by traversing the decision tree
- Traversing starts from the root node (from the top of the tree).
- The root node and the internal nodes have test conditions. Those test conditions determine which path to traverse on the tree.
- **When a leaf node is reached through traversing, the category of the leaf node determines the classification.**

Advantages and Disadvantages of Decision Trees

Advantages:

- Trees are very easy to explain to people, in fact, they are even easier to explain than linear regression.
- Some people believe that decision trees are more closely mirror human decision-making than other regression and classification techniques.
- Trees can be displayed graphically, and easily interpreted even non-experts (especially if the tree is small) can understand.

Reference/Resources: Introduction to Statistical Learning with R -7 Edition: Chapter 8

Advantages and Disadvantages of Decision Trees

Disadvantages:

- Unfortunately, trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches
- Additionally, trees can be very non-robust. In other words, a small changes in the data can cause a large change in the final estimated tree
- However, by aggregating many decision trees, using methods like *bagging*, *random forest* and *boosting*, the predictive performance of trees can be substantially improved.

Tree Induction Algorithm

The tree induction algorithm recursively partitions the dataset along its features using a measure like **Gini impurity** to determine the best split at each step. The goal is to define a set of tests (decisions) which if followed reach a leaf node with a specific class label for classification (or response variable for regression).

The algorithm will stop expanding nodes when:

- All samples in the node have **the same class label**.
- A **certain percentage** of the samples in the node. i.e. **95%** of have the same class label.
- The number of samples in the node falls below a certain **minimum number**.
- When the improvement in the **impurity measure (e.g. Gini)** is way too small to measure (too small to make a much difference in classification result).

https://en.wikipedia.org/wiki/Decision_tree_learning#Gini_impurity

Random Forest(s)

- Random Forest is a machine learning algorithm based on decision trees.
- In Random Forest the algorithm builds a large number of trees, where each tree is based on a bootstrap sample (random sample taken during training).
- After all the trees are built, the model classifies an unknown sample by averaging the predictions across all trees.

Random Forest

Disadvantages:

- Speed (it can be slow; it has to build large numbers of trees)
- Interpretability (it can be hard to interpret in the sense that you have large number of trees that are averaged together and those trees represent the bootstrap samples and are complicated to understand)

Random Forest

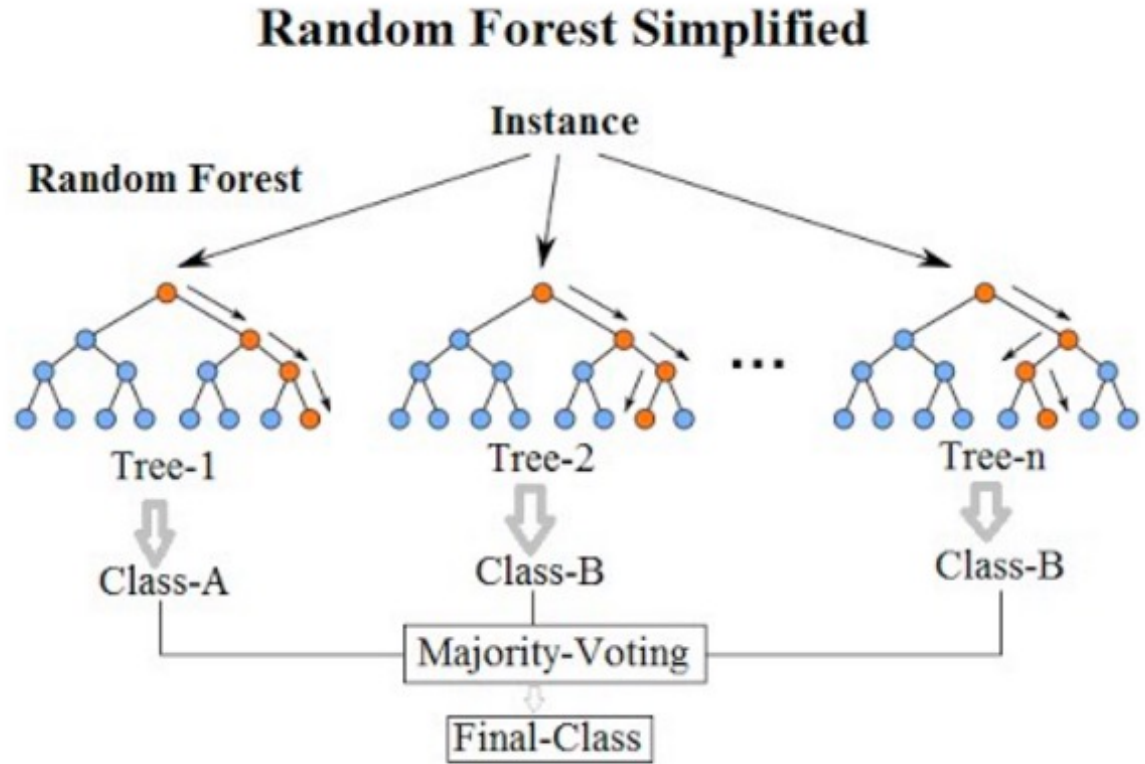


Image Resource: https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png

Random Forest

The original algorithm was created in 1995 by Tin Kam Ho.

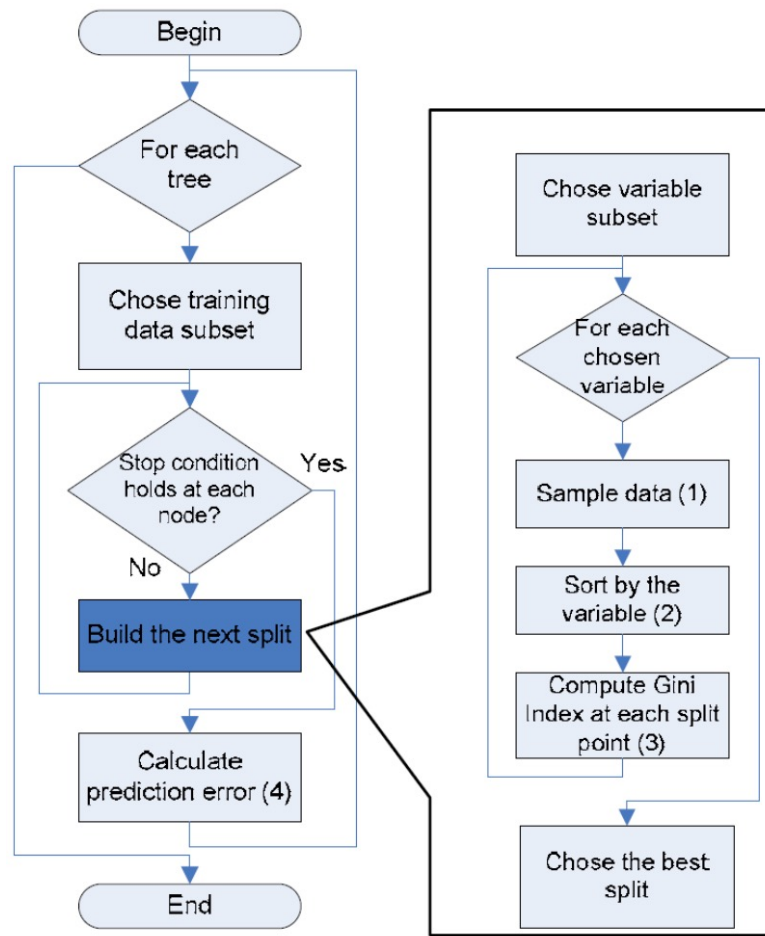
An extension of the algorithm was developed by Leo Breiman and Adele Cutler, who registered "Random Forests" as a trademark in 2006

- <http://www.stat.berkeley.edu/~breiman/RandomForests/>

Random Forest Algorithm

- Let N_{trees} be the number of trees to build for each of N_{trees} iterations:
 1. Select a new random bootstrap sample from training set.
 2. Grow a tree on this bootstrap.
 3. At each internal node, randomly select m predictors and determine the best split using only these predictors.
 4. Save tree as is, alongside those built thus far.
- To make a prediction, compute the average response (regression) or majority vote (classification) from all individually trained trees.

Random Forest



Image/ Photo Credit: Albert A. Montillo

Evaluating Classification Models

- Confusion Matrix / Contingency Table

		<i>Predicted Value</i> ↓	
		Positive	Negative
<i>Real Value</i> →	Positive	TP	FN
	Negative	FP	TN

Classification Accuracy

- *Accuracy = (Number of correct predictions) / (Total number of data points)*

$$= \frac{TP+TN}{N}$$

		<i>Predicted Value</i>	
		Cancer	Non-cancer
<i>Real Value</i>	Cancer	43	13
	Non-cancer	9	51

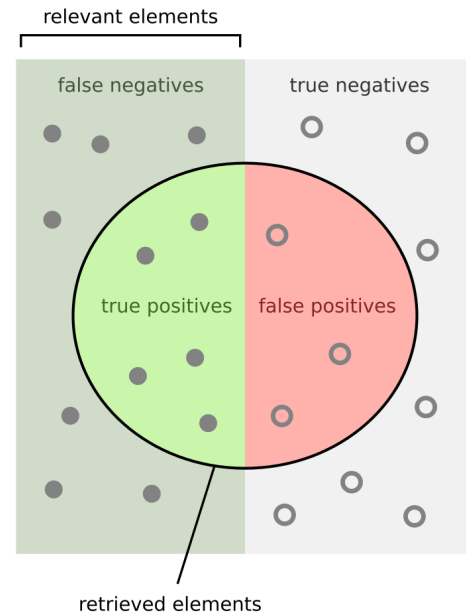
Classification accuracy = 0.81

81%

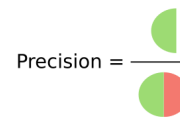
Per Class Evaluation

$$\text{Recall} = \frac{\text{instances **correctly** classified as positive}}{\text{all positive instances}}$$

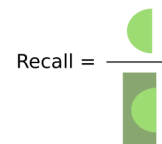
$$\text{Precision} = \frac{\text{instances **correctly** classified as positive}}{\text{all instances classified as positive}}$$



How many retrieved items are relevant?



How many relevant items are retrieved?

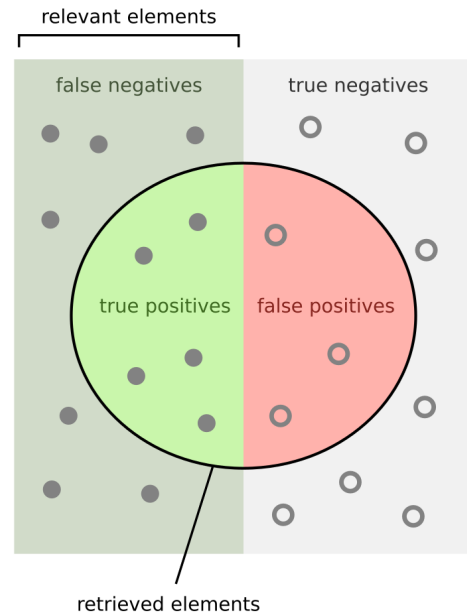


https://en.wikipedia.org/wiki/Precision_and_recall

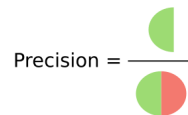
Credit (unmodified): Walber (own work) - [CC BY-SA 4.0](https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg) - https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

Evaluation Metrics – Per Class

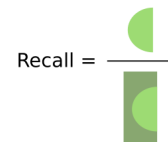
- **Recall** = $(\text{True Positive}) / (\text{True Positive} + \text{False Negative})$
 - Proportion of positive class correctly identified
- **Precision** = $(\text{True Positive}) / (\text{True Positive} + \text{False Positive})$
 - Proportion of positive predictions that are correct
- **F1** = $2 [(\text{Recall} * \text{Precision}) / (\text{Recall} + \text{Precision})]$
 - $F1 = (2 * \text{True Positive}) / [2 * \text{True Positive} + \text{False Positive} + \text{False Negative}]$
 - Harmonic mean (weighted average) of precision and recall



How many retrieved items are relevant?



How many relevant items are retrieved?



Credit (unmodified): Walber (own work) - [CC BY-SA 4.0](https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg) - https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

Iris Classification Code (R)

<https://rpi.box.com/s/mfzhlhncrubhr0ce1ui5zdx9ueuorblv>

Model Validation

Evaluation of Model Training

- To robustly evaluate predictive models the training process is repeated multiple times according to commonly used sampling strategies.
- The goal is for model training to be exposed to as much of the variation in structure in the dataset as is reasonably possible.
- Each training iteration is evaluated separately, with the average performance of the model over the number of training iterations considered an indicator of training success.

Training, Validation and Test sets

- **Training:** subset of dataset used as input to the model's training algorithm
- **Validation:** subset used to evaluate models during training
- **Test:** subset used to test the final model

e.g.

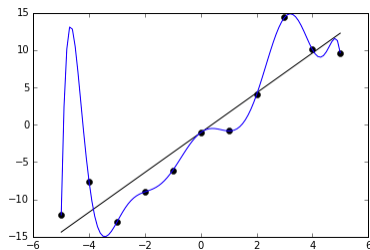
- The Iris dataset is initially split into a training set (90% - 135 obs) and a test set (10% - 15 obs) ~ this depends on the size of the dataset.
- Over 10 iterations, the training set is split into training (100 obs) and validation (35 obs), and after training, the average training error is calculated
- The final model is tested on the test set (15 obs) and the test error is calculated

Errors

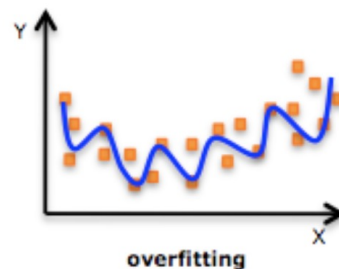
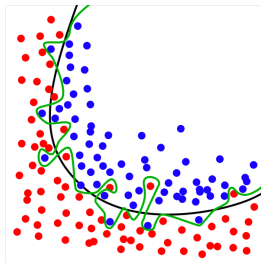
- The error on the training (validation set) data is called as the **“training error”**
- The error on the test data is referred to as the **“test error”**
- **The error on the test data is a good indication of how well the classifier will perform on new data and this is known as the generalization.**
- If the classifier performs well on the new data, then it is a good generalization. Generalization refers to how well the model is performing on the new data **(data not used to train the model)**

Overfitting

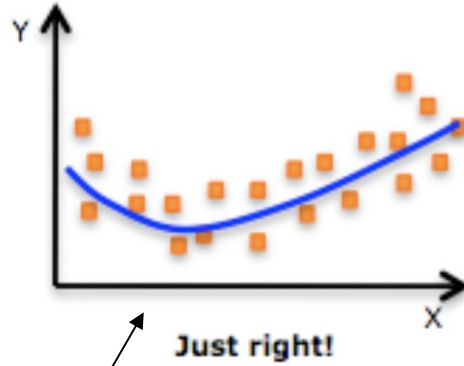
- This is a good indication that the model may have learned to ***model the noise*** in the training data, instead of the learning from the underlying structure of the data.
- Overfitting is an indication of poor generalization.



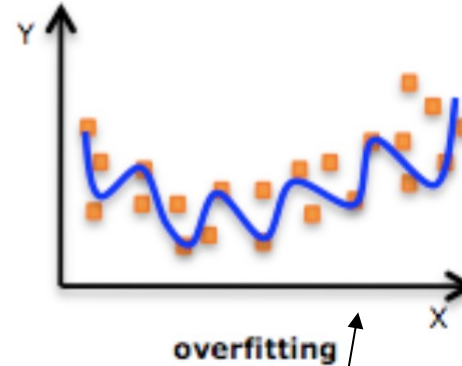
Image/Photo Credit:
https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitted_Data.png



Image/Photo Credit:
<http://pingax.com/regularization-implementation-r/>



Model is fitting to
the structure of the data

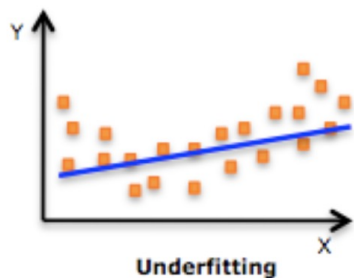


Model is fitting to
the noise of the data

Image/Photo Credit: <http://pingax.com/regularization-implementation-r/>

Underfitting

- **Underfitting** occurs when a statistical model cannot adequately capture the underlying structure of the data.
- In other words, **underfitting take place when the model has not properly learned the structure of the data.**



Image/Photo Credit: <http://pingax.com/regularization-implementation-r/>

Robustly Validating Models (Cross Validation)

- There are several ways to robustly evaluate/validate models
 - K-fold Cross validation
 - Monte Carlo Cross validation
 - Leave-One-Out Cross validation

[https://en.wikipedia.org/wiki/Cross-validation_\(statistics\)](https://en.wikipedia.org/wiki/Cross-validation_(statistics))

K-fold Cross Validation

- In k-fold cross validation, the data are segmented in to k number of **disjoint partitions**.
- During each iteration, one partition is used as the test set and the remaining $k-1$ (combined) for training; The process is repeated k times.
- Each time using a different partition for testing, so that each partition is used exactly one time for the validation.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Monte Carlo Cross Validation (Repeated random sub-sampling)

- In Monte Carlo cross validation, the dataset is split into training/test sets over n iterations with the samples in each selected at random.
- The size of each partitions may be constant or vary over the iterations.
- Commonly used in research, considered robust because of the averaging effect over multiple iterations.
- Downside: since selection is random, some observations may not end up in test sets and some may be oversampled

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Leave One Out Cross Validation (LOOCV)

- For as many iterations as there are observations, drop one observation and use all the others for training; test on the 1 observation and average at the end.
- Depending on the size of the dataset, may be computationally expensive.

Resource/Reference: Introduction to Statistical Learning with R, 7th Edition - Chapter 5

Unsupervised Learning

Unsupervised Learning - Clustering

- **Unsupervised Learning** is a machine learning paradigm where algorithms learn patterns from unlabeled data.
- **Clustering** is a type of unsupervised learning with the goal of finding structure in a dataset by identifying natural clusters of data points based on a similarity criterion (usually distance).
- The observations in these clusters are generally more similar (closer) to each other than they are to points in other clusters.

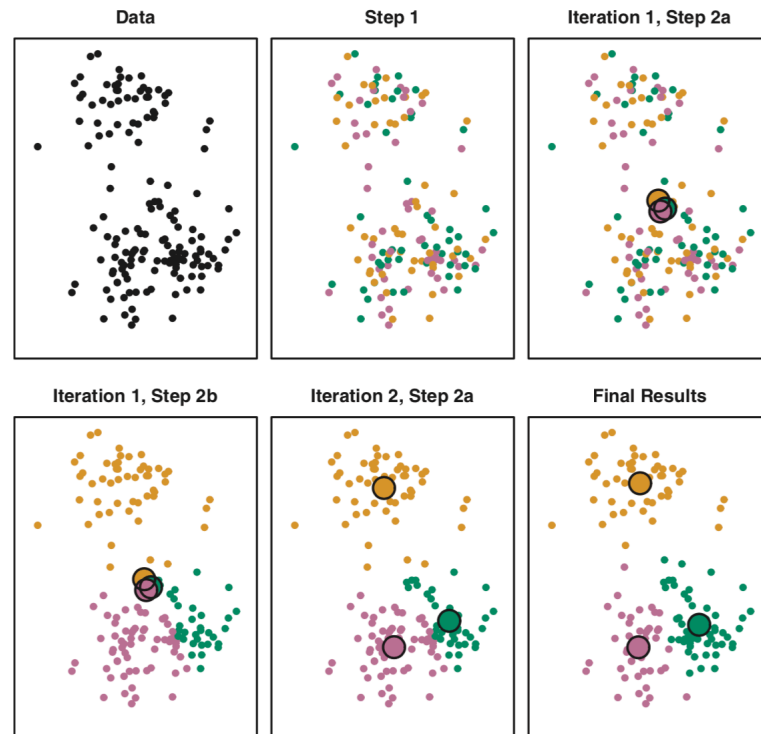
K-Means

- k-Means clustering is an unsupervised learning algorithm that, as the name hints, finds a fixed number (k) of clusters in a set of data.
- A *cluster* is a group of data points that are grouped together due to similarities in their features. When using a K-Means algorithm, a cluster is defined by a *centroid*, which is a calculated point at the center of a cluster.
- Every point in a data set is part of the cluster whose centroid is most closely located in feature space. To put it simply, K-Means finds k number of centroids, and then assigns all data points to the closest cluster while minimizing the sum of squared distances from the points to their assigned centroid.
- K-Means assumes spherical clusters.
- K-Means is sensitive to outliers.

K-Means Algorithm

Algorithm 10.1 *K-Means Clustering*

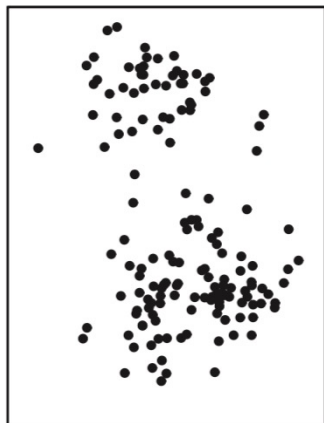
1. Randomly assign a number, from 1 to K , to each of the observations. These serve as initial cluster assignments for the observations.
 2. Iterate until the cluster assignments stop changing:
 - (a) For each of the K clusters, compute the cluster *centroid*. The k th cluster centroid is the vector of the p feature means for the observations in the k th cluster.
 - (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).
-



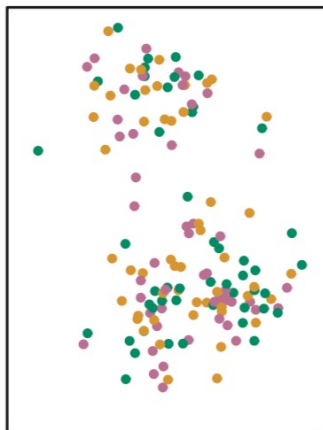
Reference: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

K-Means Algorithm

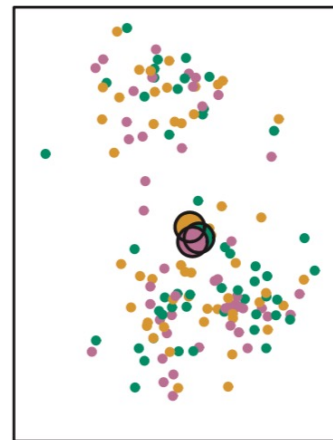
Data



Step 1



Iteration 1, Step 2a



Observations (data) is shown

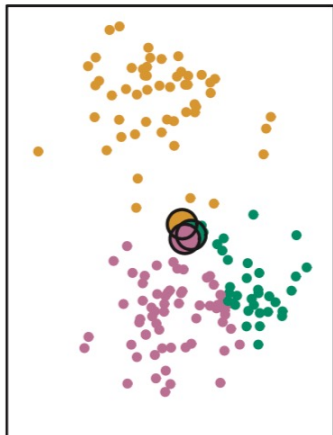
Step 1 of the algorithm: each observation is randomly assigned to a cluster

Iteration1 Step 2(a): The cluster centroids are computed; these are shown in large colored disks. Initially centroids are almost completely overlapping because the initial cluster assignment were chosen at random

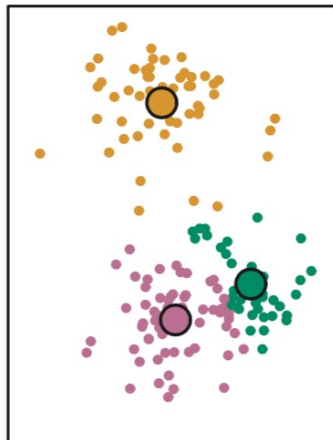
Reference: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

K-Means Algorithm

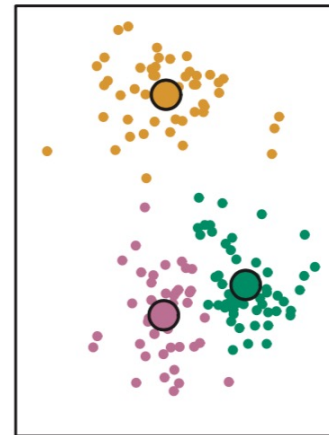
Iteration 1, Step 2b



Iteration 2, Step 2a



Final Results



Iteration 1 Step 2(b) : each observation is assigned to the nearest centroid

Iteration 2, Step 2(a): the step 2(a) is once again performed, leading to new cluster centroids.

Final Results: the results obtained after ten iterations. You can see the distinct clusters with their centroids.

Image/Photo Credit: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

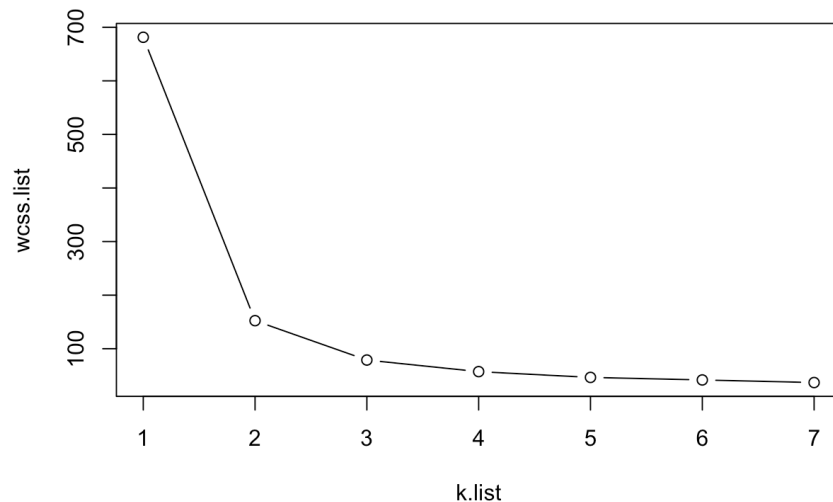
Reference: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

Evaluating K-Means Models (Elbow Method)

Within-Cluster Sum of Squares: sum of squared Euclidean distances between all points in a cluster and the cluster's centroid.

$$WCSS = \sum_{i=1}^k \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2$$

- The elbow method is a heuristic that can be subjective and unreliable.



Plot of total within cluster sum of squares with values of k

Evaluating K-Means Models (Silhouette Method)

Silhouette value: a measure of similarity between a point and its cluster

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

Mean distance between i and other points in the cluster

$$b(i) = \min_{J \neq I} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)$$

Minimum mean distance between i and all points in any other cluster

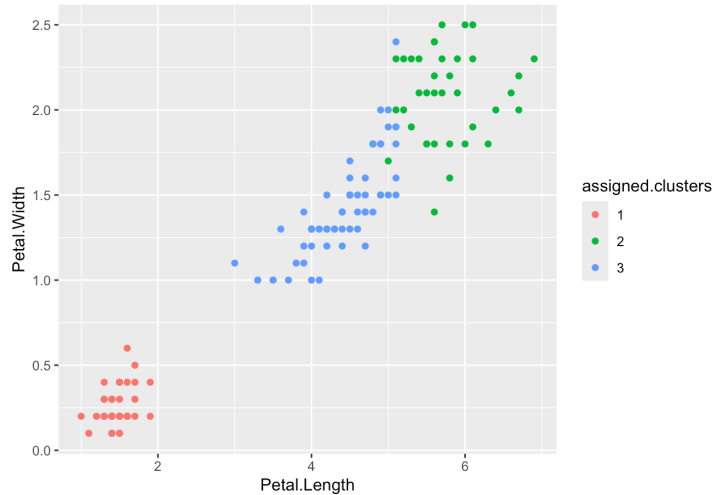
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

$$s(i) = 0, \text{ if } |C_i| = 1$$

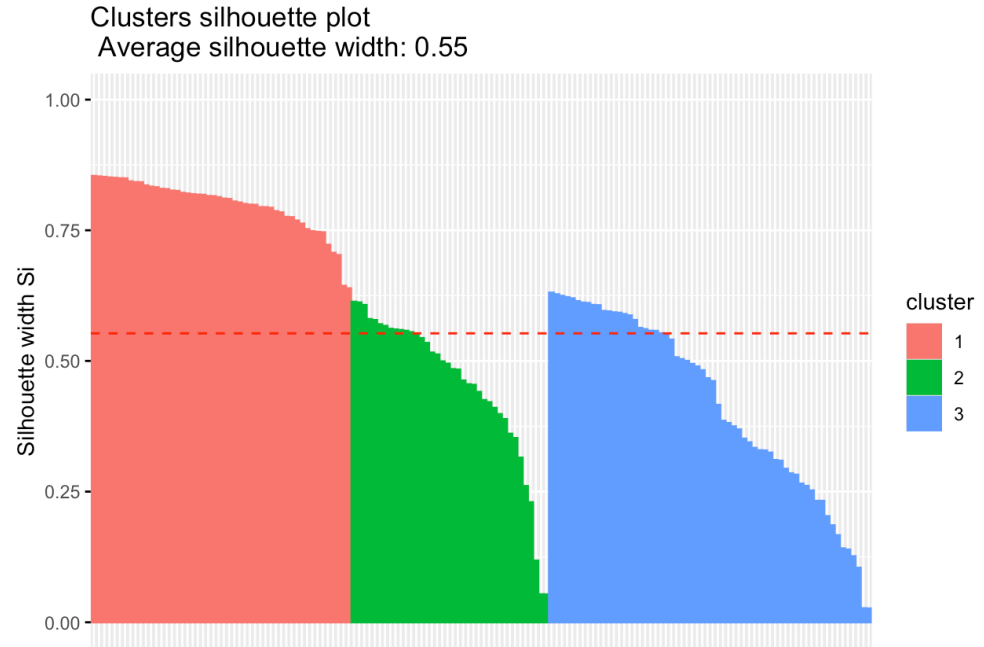
[https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering))

Evaluating K-Means Models (Silhouette Method)

Silhouette value: a measure of similarity between a point and its cluster



[https://en.wikipedia.org/wiki/Silhouette_\(clustering\)](https://en.wikipedia.org/wiki/Silhouette_(clustering))



Software Platforms for Data Mining

Platforms / Libraries

- R
 - R: (<https://www.r-project.org/>)
 - Rstudio: (<https://posit.co/downloads/>)
 - Notable Packages:
 - Tidyverse (<https://tidyverse.org/>) ~ collection of R packages for data science
 - ggplot2 (<https://ggplot2.tidyverse.org/>) ~ visualizations
 - iGraph (<https://igraph.org/>) ~ graph/network visualization & analysis
 - dplyr (<https://dplyr.tidyverse.org/>) ~ data manipulation
 - forcats (<https://forcats.tidyverse.org/>) ~ categorical variables

...



Platforms / Libraries

- Python

- Jupyter Notebook / Lab (<https://jupyter.org/>) ~ IDE for notebooks, code, and data
- Anaconda (<https://www.anaconda.com/download>) ~ Package manager for Python/Jupyter
- Notable libraries:
 - scikit-learn (<https://scikit-learn.org/stable/index.html>) ~ machine learning in Python
 - NumPy (<https://numpy.org/>) ~ numerical data structures / computing methods
 - SciPy (<https://scipy.org/>) ~ scientific computing methods
 - Matplotlib (<https://matplotlib.org/>) ~ visualizations
 - Pandas (<https://pandas.pydata.org/>) ~ data analysis and manipulation

Thanks!