Rensselaer

why not change the world?®

Data Mining Review

Ahmed Eleish

Data Science – ITWS/CSCI/ERTH-4350/6350,

October 27th, 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
  - Dimensionailty Reduction
- Software platforms/libraries

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.

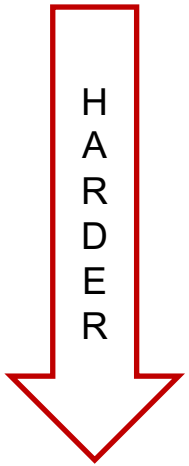
Data Mining – Types of Mining

- 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.

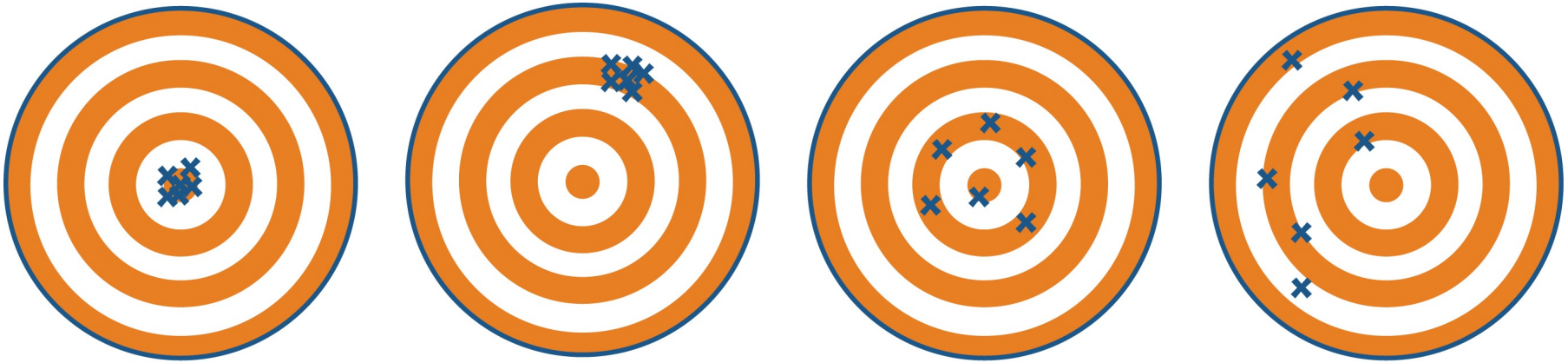
• An ***observation***, or a *case*, is a set of measurements made under similar conditions (you usually make all of the measurements in an observation at the same time and on the same object). An observation will contain several values, each associated with a different variable. **Sometimes refer to 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

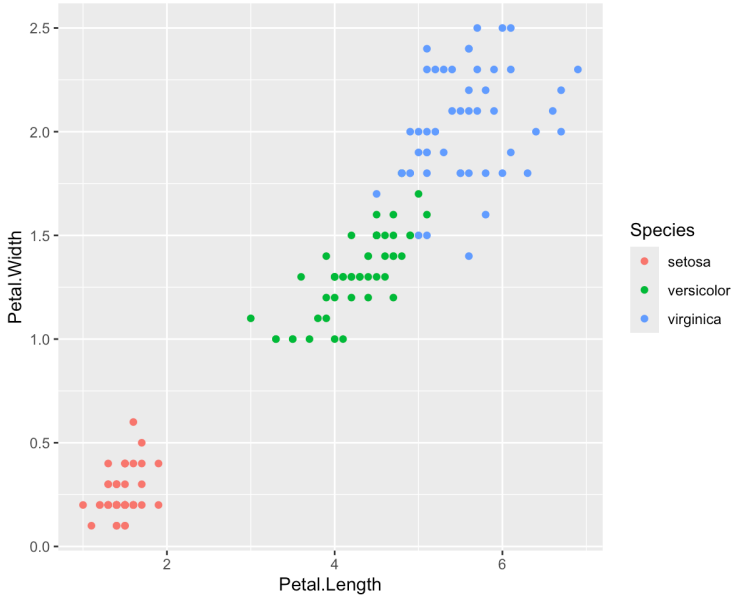
http://climatica.org.uk/climate-science-information/uncertainty

Scientific Use Case

Science Motivation: Botany

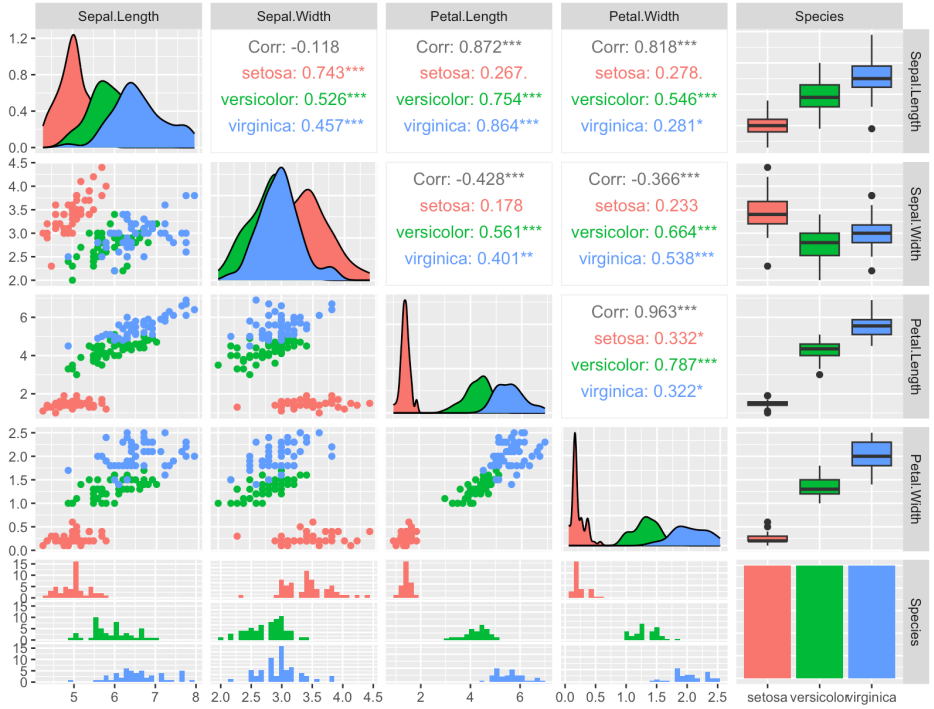
- Let's assume a botanist is interested in distinguishing the species of some iris flower samples that were collected. The botanist takes some measurements associated with each iris: length and width of the petals and length and width of sepals.
- The botanist also has access to measurements of some 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.





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

Descriptive Analysis of Iris



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:

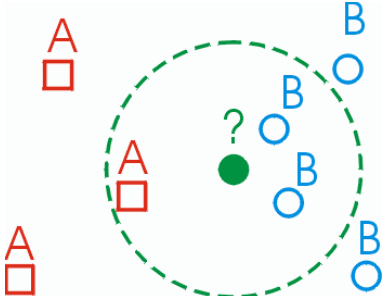
– Determine parameter *k* = number of nearest neighbors, beforehand. This value **is up to you**.

– 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.



k-Nearest Neighbors

Pros:

- Simple and intuitive
- No training phase (lazy learning)
- Non-parametric (no assumptions about data)
- Effective with sufficient data

Cons:

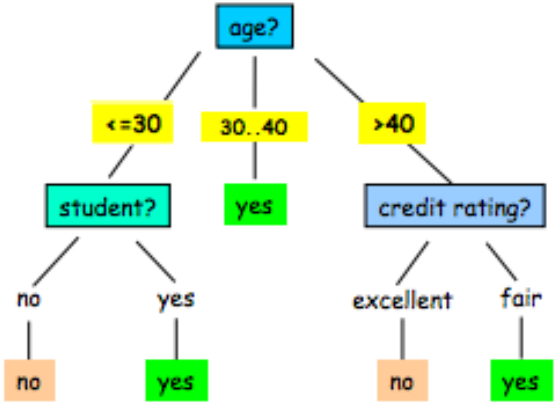
- Computationally expensive for large datasets
- Sensitive to feature scaling
- Requires choosing k
- Storage intensive (must keep all training data)

Decision Trees

Decision tree classifier

Classification by Decision Tree Induction

age	income	student	credit_rating	buys_computer
<=30	high	no	fair	no
<=30	high	no	excellent	no
31..40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31..40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31..40	medium	no	excellent	yes
31..40	high	yes	fair	yes
>40	medium	no	excellent	no



buys_computer ?

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

Decision Tree - Classification

• When we implement decision trees for classification, the idea is to split the data into subsets. So that each subset belongs to one particular class.

• In other words, splitting the data into regions, that are separated by decision boundaries, where each region's samples have only one class.

• Classification decisions are made by traversing the decision tree

• Traversing starts from the root node (from the top of the tree).

• **When a leaf node is reached through traversing, the category of the leaf node determines the classification**.

Decision Tree

Pros:

• Trees can be displayed graphically and easily interpreted by non-experts (especially if the tree is small) can understand.

• Some people believe decision trees are more closely mirror human decision-making.

Cons:

• Trees generally do not have the same level of predictive accuracy as some of the other regression and classification approaches.

•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

Exercise: Decision Tree with Iris dataset

```
iris.data <- iris

s.train <- sample(150,100)

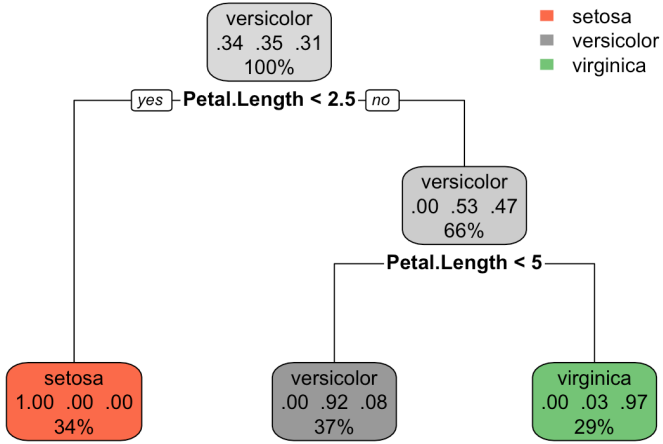
# creat training and testing sets
iris.train <-iris[s.train,]
iris.test <-iris[-s.train,]

# Decision tree model
tree.model <- rpart(Species~., iris.train, method = "class")

#plotting the decision tree model using rpart.plot() function
rpart.plot(tree.model)

tree.model.predicted <-  predict(tree.model, iris.test, type = "class")

## confusion matrix/contingency table
table(tree.model.predicted, iris.test$Species, dnn=list('predicted','actual'))
```



Code: https://rpi.box.com/s/uhobs1a0ces4lji04e7ollse11j4noav

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

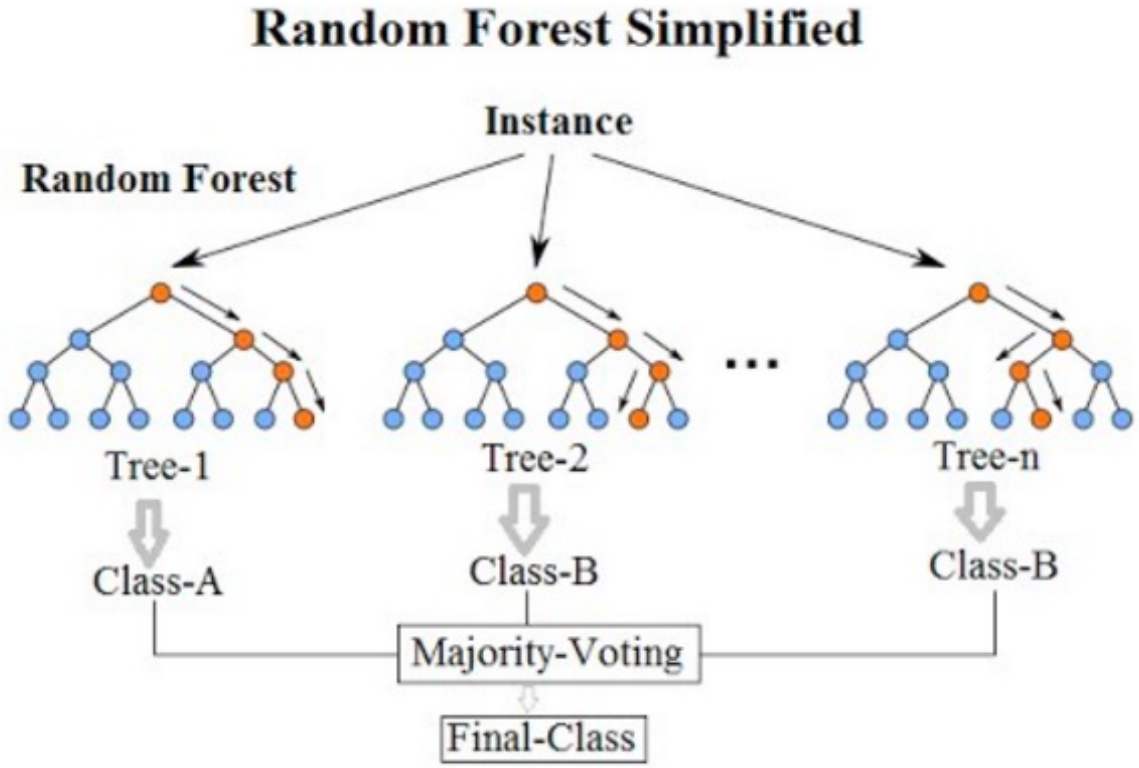


Image Resource: https://commons.wikimedia.org/wiki/File:Random_forest_diagram_complete.png

Random Forest

Disadvantages:

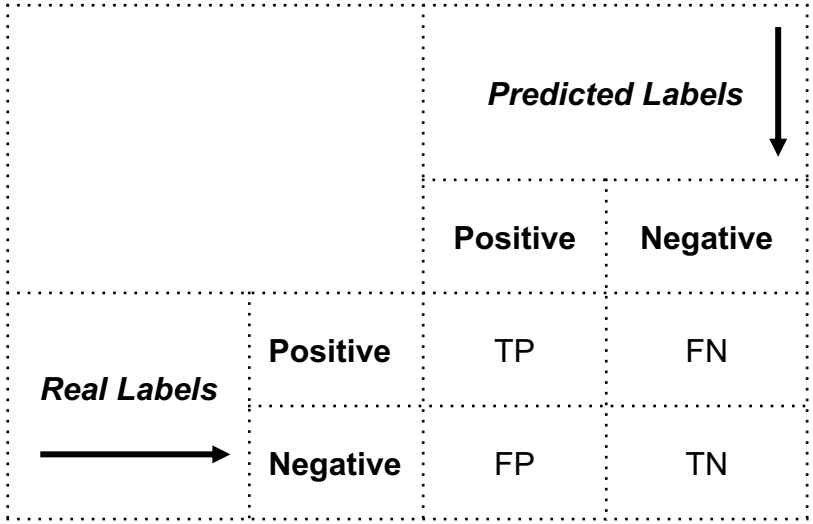
- Speed (it can be slow; it has to build large numbers of trees)

- 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)

Evaluating Classifiers

Evaluating Classification Models

- Confusion Matrix / Contingency Table



Classification Accuracy

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

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

		Predicted Value	
		Cancer	**Non-cancer**
Real Value	**Cancer**	43	13
	Non-cancer	9	51

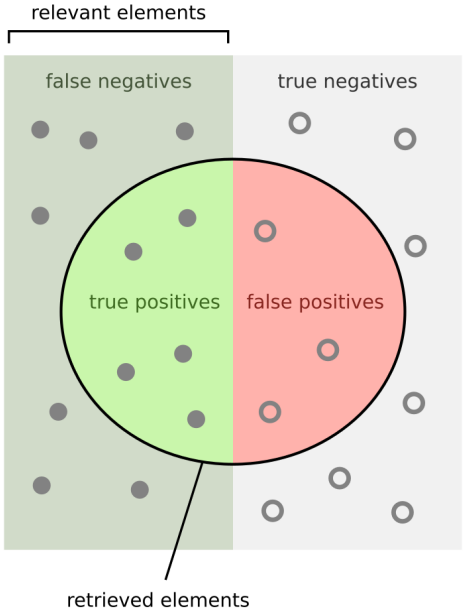
Classification accuracy = 0.81

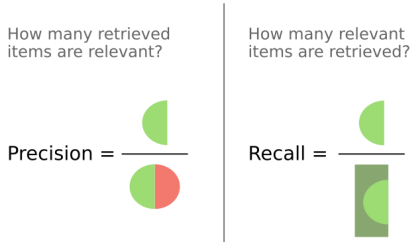
81%

Per Class Evaluation

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

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

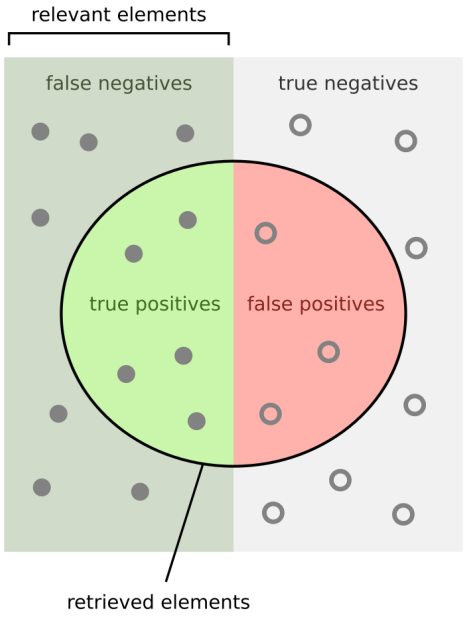


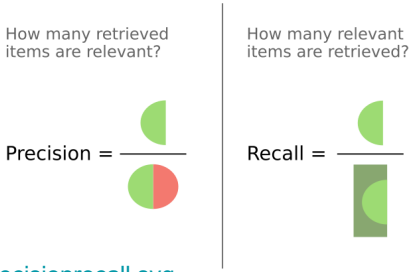


https://en.wikipedia.org/wiki/Precision_and_recall
Credit (unmodified): Walber (own work) - CC BY-SA 4.0 - https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg

Evaluation Metrics – Per Class

- ***Recall = (True Positive) / (True Positive + False Negative)***
  - ***Proportion of positive class correctly identified***
- ***Precision = (True Positive) / (True Positive + False Positive)***
  - ***Proportion of positive predictions that are correct***
- ***F1 = 2 [(Recall * Precision) / (Recall + Precision)]***
  - ***F1 = (2*True Positive) / [2*True Positive + False Positive + False Negative]***
  - ***Harmonic mean (weighted average) of 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

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

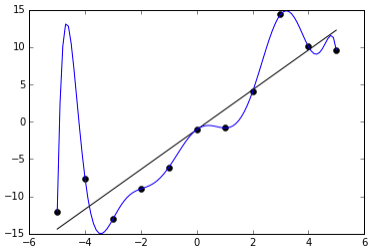
Errors

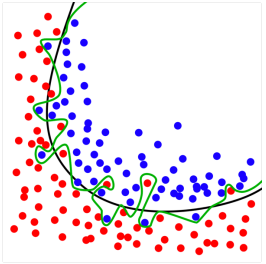
- The error on the training (validation set) data is called as the “**training error**”
- The error on the test data is referred to as the **“test error”**
- **The error on the test data is a good indication of how well the classifier will perform on new data and this is known as the generalization**.
- 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

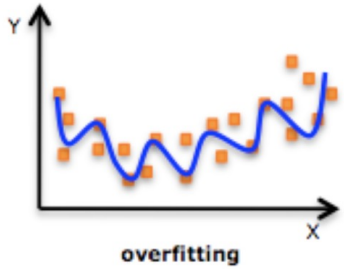
- This is a good indication that the model may have learned to ***model the noise*** in the training data, instead of the learning from the underlying structure of the data.

- Overfitting is an indication of poor generalization.

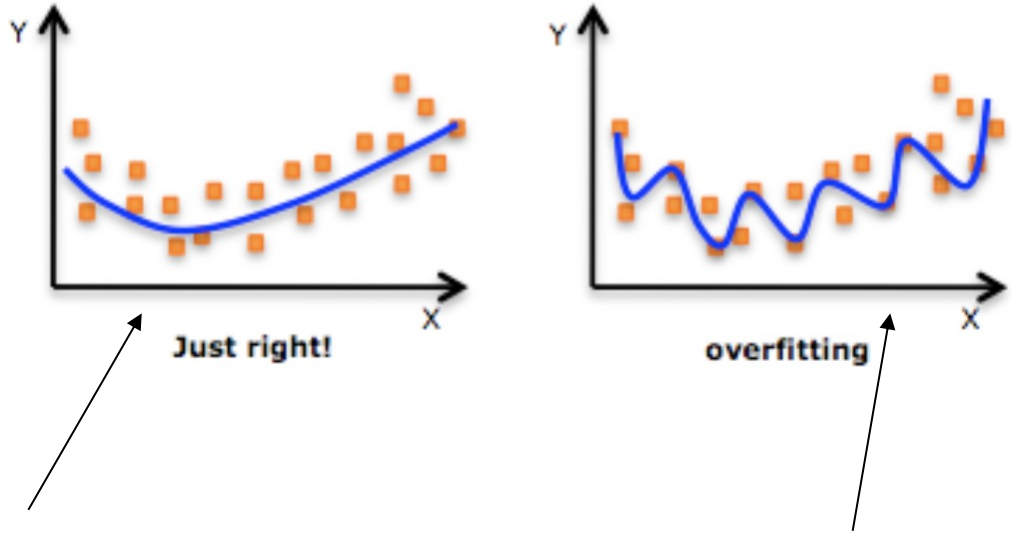




Image/Photo Credit:
https://en.wikipedia.org/wiki/Overfitting#/media/File:Overfitted_Data.png



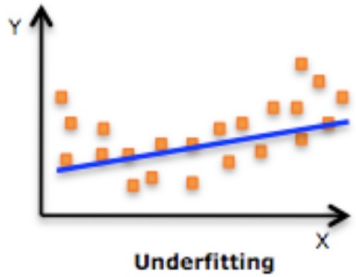
Image/Photo Credit:
http://pingax.com/regularization-implementation-r/



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)

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 used all the others for training; test one 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

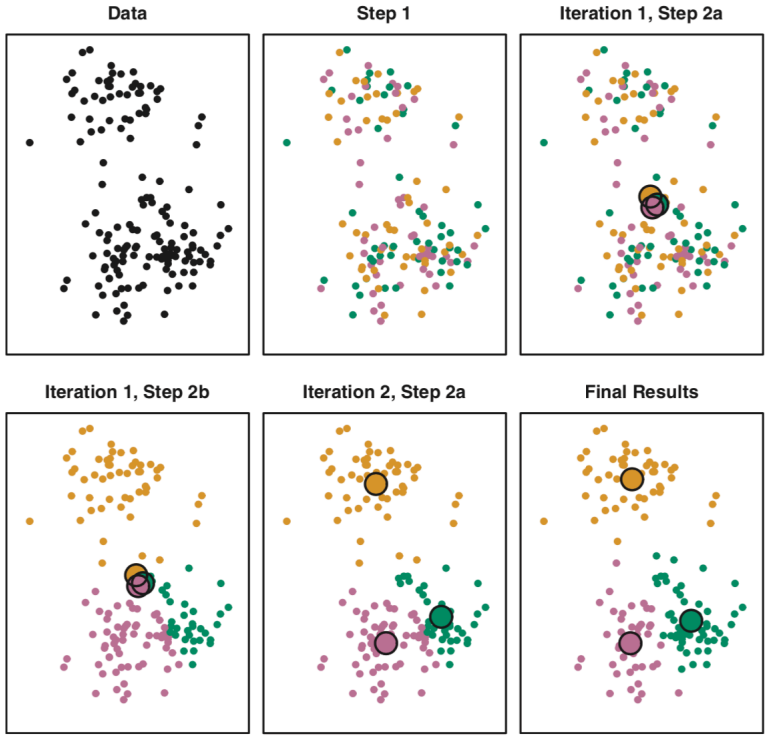
Unsupervised Learning - Clustering

- **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 Algorithm

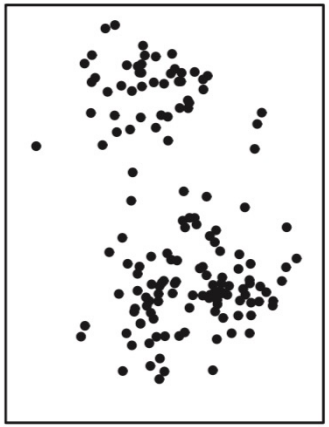
Algorithm 10.1 *K-Means Clustering*

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 kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
   (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).

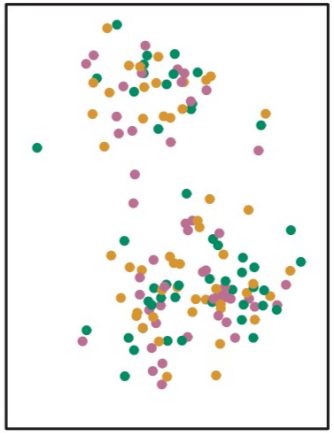


Reference: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

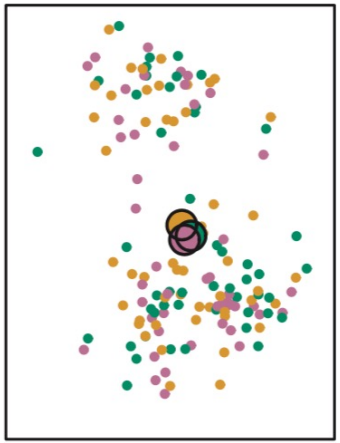
K-Means Algorithm



Observations (data) is shown



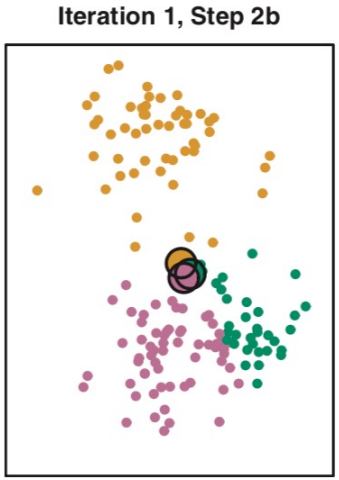
Step 1 of the algorithm: each observation is randomly assigned to a cluster



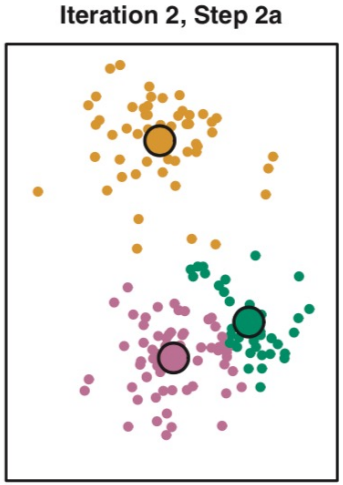
Iteration1 Step 2(a): The cluster centroids are computed; these are shown in large colored disks. Initially centroids are almost completely overlapping because the initial cluster assignment were chosen at random

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

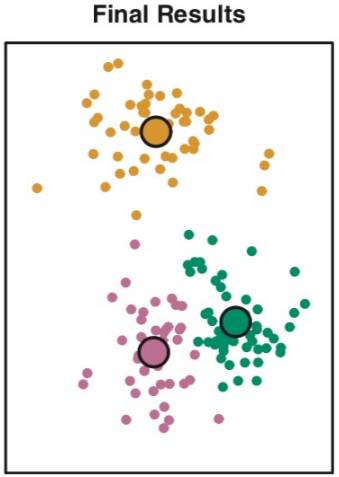
K-Means Algorithm



Iteration 1 Step 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.

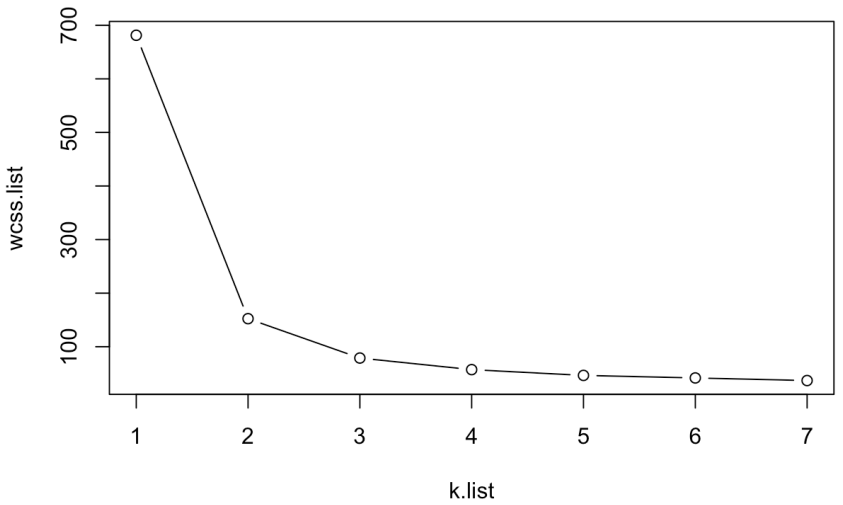
Image/Photo Credit: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans
Reference: Introduction to Statistical Learning with Applications in R, 7th Edition, Chapter 10 – KMeans

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 clsuter

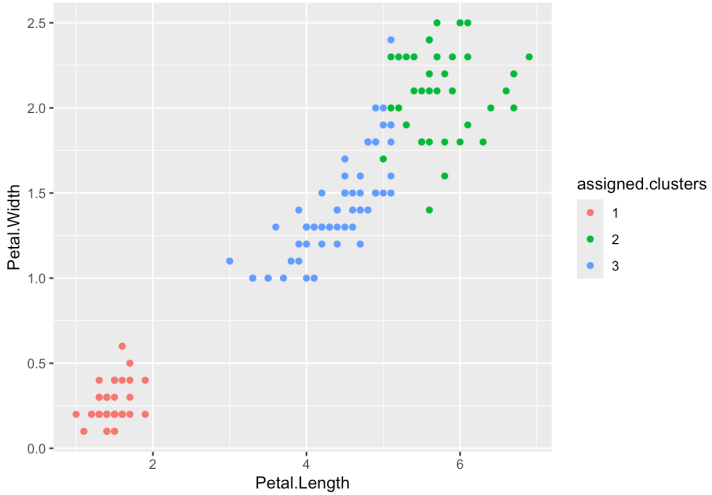
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \text{ if } |C_i| > 1$$

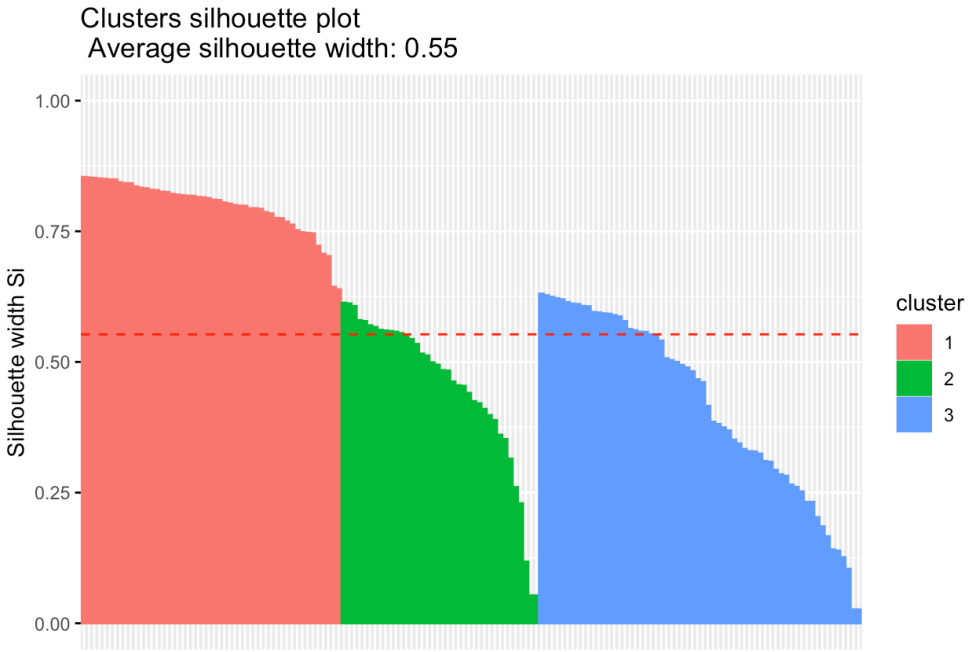
$$s(i) = 0, \text{ if } |C_i| = 1$$

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

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)

Dimensionality Reduction

Definitions

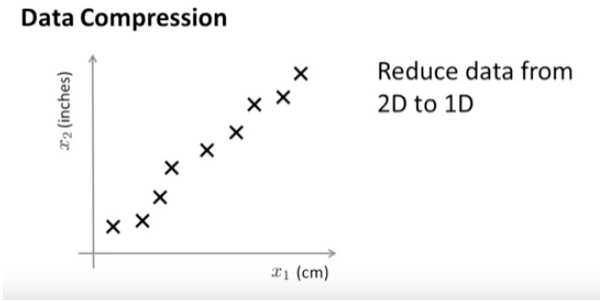
- **Dimensions of a dataset:** The number of features (variables, columns) in a dataset (usually denoted as *p*)

- **Dimensionality Reduction (of a dataset):** the process of reducing the number of features of a dataset through some transformation that preserves the patterns or structure in the data.

Dimensionality Reduction (DR)

X1: distance measured in cm
X2: distance measured in inches

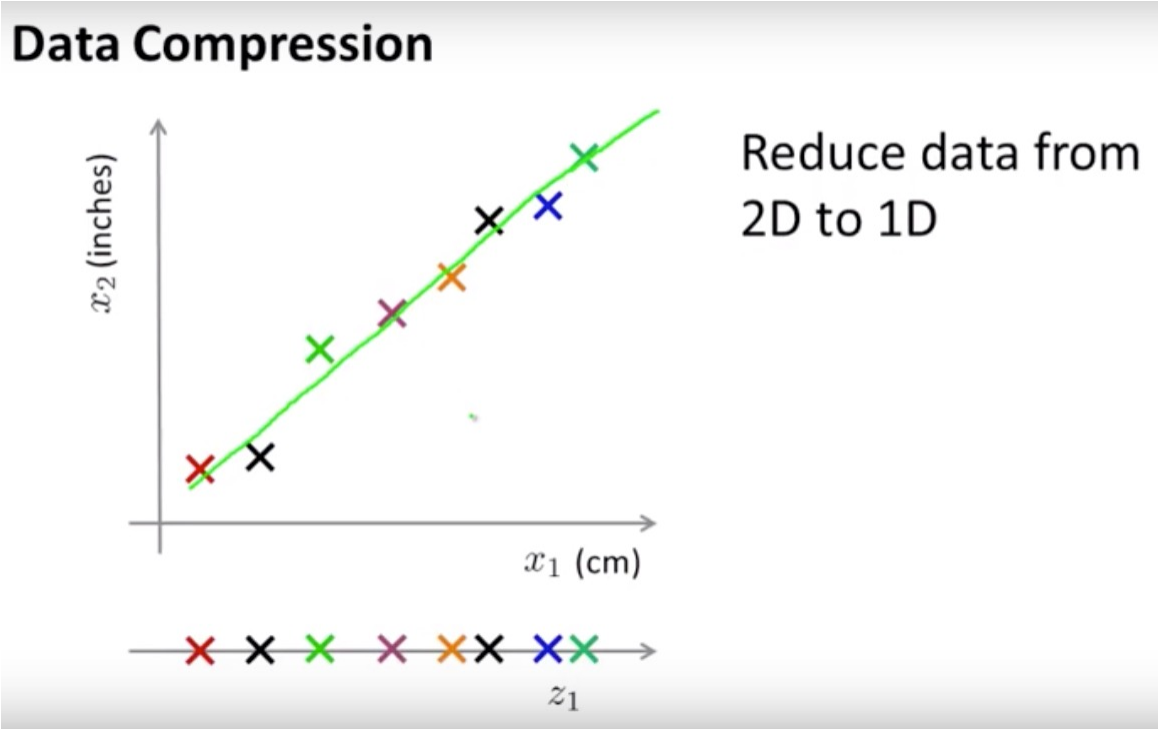
We want to reduce the data to one dimension



Length in cm is rounded off to the nearest cm and length in inches is rounded off to the nearest inch, that is why those examples do not perfectly lie on a straight line.

Image source; ML course Stanford University

Dimensionality Reduction (DR)



Dimensionality Reduction (DR)

- If a dataset contains hundreds of features, it is difficult to keep track of all those features, and there is redundancy.

-> DR is used in feature selection, reduction

- Why?
    - **Curse of dimensionality** – challenges of learning from (very) high-dimensional data, or datasets with a large number of dimensions relative to the number of observations*.

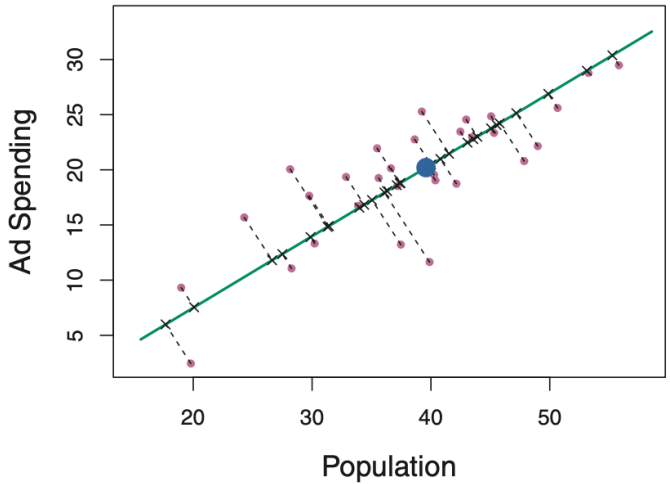
* These considerations are generalizations and are more narrowly defined per domain/dataset/analysis

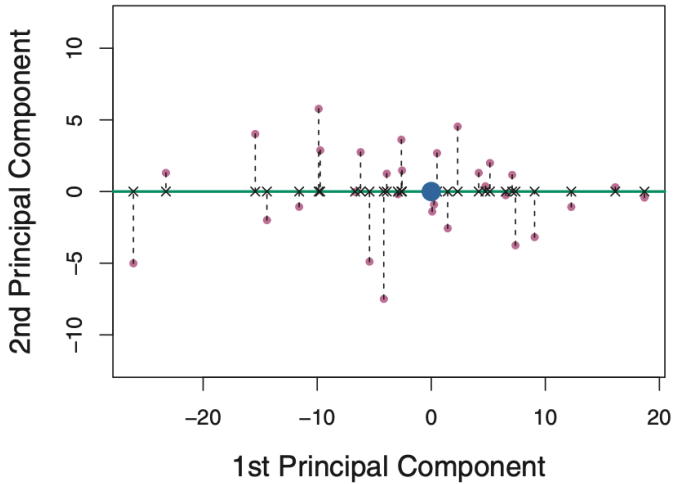
Dimensionality Reduction Methods

- Linear:
  - Principal Component Analysis (PCA)
  - Linear Discriminant Analysis (LDA)
  - Factor Analysis (FA)
- Non-linear:
  - Uniform Manifold Approximation and Projection (UMAP)
  - Autoencoders (Neural Networks)
  - Kernel PCA

Principal Component Analysis

- The dataset is projected onto the Principal Components for visualization or further analysis:





Finding the Principal Components

- The first principal component of a set of features X1, X2, . . . , Xp is the normalized linear combination of the features that has the largest **variance**.

$$Z_1 = \phi_{11} X_1 + \phi_{12} X_2 + \cdots + \phi_{p1} X_p$$

By normalized, we mean that

$$\sum_{j=1}^{p} {\phi_{j1}}^2 = 1$$

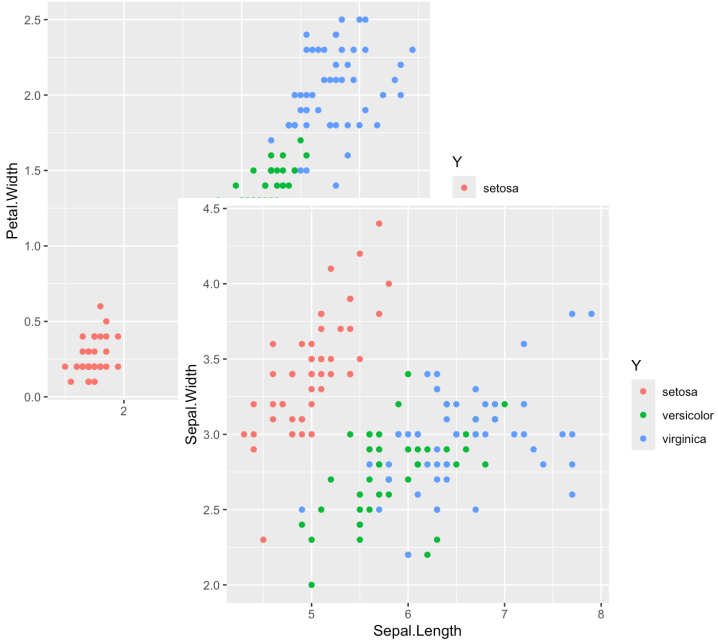
- We refer to the elements $\varphi_{11},\ldots,\varphi_{p1}$ as the **loadings** of the first principal component; together, the loadings make up the principal component loading vector, $\varphi_1 = (\varphi_{11}, \varphi_{21}, \ldots, \varphi_{p1})^T$.

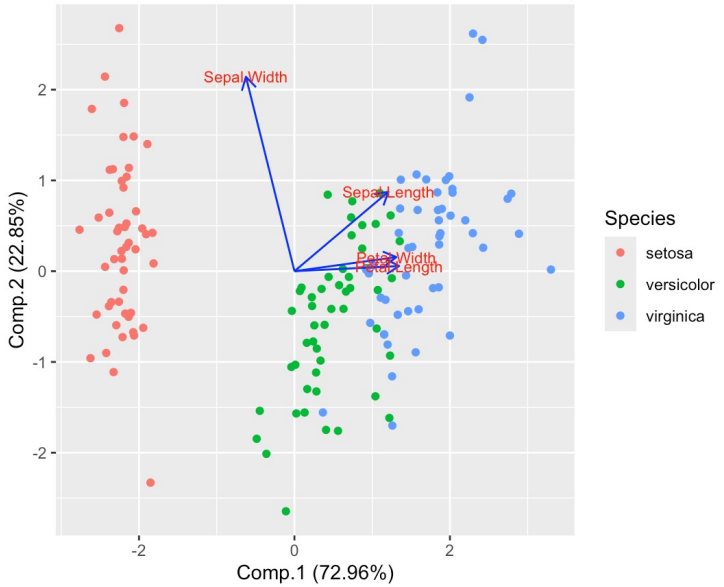
Once the loading vectors for all principal components are obtained, the dataset can be transformed (projected) to obtain new values according to the new axes (PCs).

$$Z = XV$$

where *Z* is the transformed matrix, *X* is the original matrix, *V* is the matrix of loading vectors ($\varphi_{1, \ldots} \varphi_p$)

PCA with Iris





Dimensionality Reduction with PCA

PCA is useful in many different scenarios, including:

- **Data exploration:** PCA can help you to visualize high- dimensional data in a lower dimensional space.
- **Data compression:** PCA can reduce the number of variables in a dataset, which can make it easier to work with.
- **Feature selection:** PCA can help to identify the most important variables in a dataset.
- **Data pre-processing:** PCA can be used to remove noise from a dataset and to standardize variables so that they have a similar scale.
- **Downstream Machine learning:** PCA can be used as a pre-processing step before applying machine learning algorithms to a dataset, to improve their performance and reduce overfitting.

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
    - dplyer (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!