STATISTICS & ML WITH R

Intro to Machine Learning

2024

M. Chiara Mimmi & Luisa M. Mimmi

WORKSHOP SCHEDULE

- Modules
  - 1. Intro to R and data analysis
  - 2. Statistical inference & hypothesis testing
  - 3. Modeling correlation and regression
  - 4 Mapping causal & predictive approaches
  - 5. Machine Learning
  - 6. Bonus topics:
    - MetaboAnalyst;
    - Power Analysis

- Each day will include:
    - Frontal lecture (MORNING)
    - Practical training with R on the same topics (AFTERNOON)

MODULE 5 – LECTURE OUTLINE

- Intro to Machine Learning (ML)
- Classification of ML algorithms
- ***Supervised*** ML examples
  - Logistic Regression
  - Decision trees*
  - PLS-DA
- ***Unsupervised*** ML examples
  - PCA
  - K-means clustering*

* = coming soon!

Different goals of statistical modeling *(before)*

1. **ASSOCIATION/CORRELATION** → observational studies
   - aimed at **summarizing or representing the data structure**, *without* an underlying causal theory
   - may help **form hypotheses** for explanatory and predictive modeling

2. **CAUSAL EXPLANATION** → experimental studies
   - aimed at **testing "explanatory connection"** between treatment and outcome variables
   - prevalent in "**causal theory-heavy**" fields (economics, psychology, environmental science, etc.)

- **Note**:
  - ✓ The **same modeling approach** (e.g., fitting a regression model) can be used for **different goals**
  - ✓ While they shouldn't be confused, **explanatory power** and **predictive accuracy** are complementary goals: e.g., in bioinformatics (which has little theory and abundance of data), predictive models are pivotal in generating avenues for causal theory.

3. **EMPIRICAL PREDICTION** → algorithmic machine learning and data-mining modeling

Different goals of statistical modeling *(today!)*

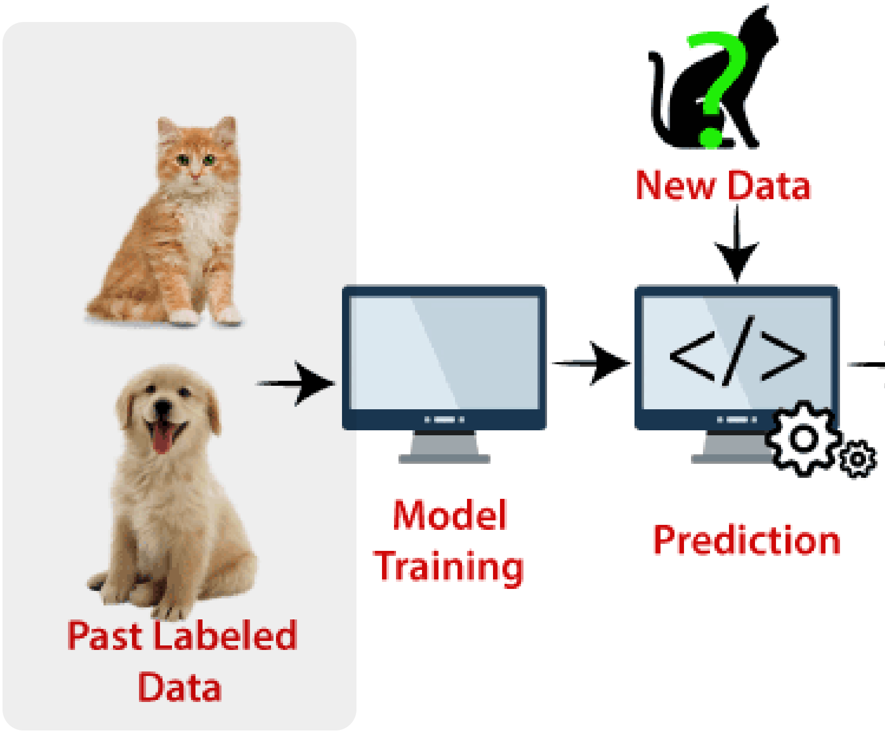
1. **ASSOCIATION/CORRELATION** → observational studies
2. **CAUSAL EXPLANATION** → experimental studies
3. **EMPIRICAL PREDICTION** → algorithmic machine learning and data-mining modeling
    - aimed at **predicting new or future observations** (without necessarily explaining how)
    - relies on **big data**
    - prevalent in fields like natural language processing, bioinformatics, etc.. In epidemiology, there is more of a mix causal explanation & empirical prediction

- **Notes**:
    - ✓ "Prediction" does not necessarily refer to future events, but rather to *future* datasets that were previously unseen to the algorithm

MACHINE LEARNING

Defining Machine Learning (ML)

> "At its core, **Machine Learning** is just a **"thing-labeler"**, taking something and telling you what label it should get."
>
> (Cassie Kozyrkov)



Source: Image from https://entri.app/blog/what-is-svm-algorithm-in-machine-learning/

Defining Machine Learning (ML)

- **Machine Learning** is a broad and highly active research field. (In the life sciences, "*precision medicine*" is an application of machine learning to biomedical data)

- The **general idea** is to predict or discover outcomes from measured predictors, in problems like:
  - *Can we discover new types of cancer from gene expression profiles?*
  - *Can we predict drug response from a series of genotypes?*
  - *How do we classify a set of images/spectrometry outputs, etc.*
  - *Given various clinical parameters, how can we use them to predict heart attacks?*

- The **ML is a data-driven (inductive) approach**, where a machine **learns** the rules/patterns from a set of **training data** and (then) **validates** findings on a set of **testing data**

- In contrast with inferential statistics, **ML *doesn't worry* about assumptions on parameters** (probability distribution, error, correlation, etc.), **nor the causal nexus** between specific predictor(s) and response, **nor the data collection strategy**

- In contrast with hypothesis-driven statistics, **in ML the rules are not necessarily specified**… hence ML = a subfield of AI

Stylized comparison between statistics and machine-learning

	Standard (causal inference) Statistics	Machine Learning
Typical Goal	Explanation, uncovering causal relationships	Prediction of an outcome, maximizing model’s accuracy
Typical Task	Theory-driven research to identify the causal effect (better: pre-register your hypothesized model).	Data-driven exploration done with different algorithms to learn patterns and test accuracy on new and unseen data.
Parameters of interest:	Causal effect size and statistical significance, p-value of treatment X for outcome Y	Model’s accuracy (%), precision/recall, sensitivity/specificity, in predicting Y
Data generating process	Designed ex-ante based on study goal (e.g. randomized control trial, or observational study with statistical control variables)	Useful but not strictly necessary, and often not available (usually applied on high dimensional data)
Dataset	Use **ALL AVAILABLE DATA** to calculate effect of interest (it was designed to be representative of a population).	It is critical to **SPLIT THE DATA** (usually 75% for training and 25% for testing the algorithms) leaving aside a sub-sample to test the model with unseen new data

Source: Adapted from https://forloopsandpiepkicks.wordpress.com/2022/02/10/beginners-guide-to-machine-learning-in-r-with-step-by-step-tutorial/

Performance of a ML model

- What are some important questions to ask in order to understand the true performance and utility of the model?

  - *What were the **metrics** used to determine the model performance?*
  - *Were there any patients/cases from the training set also in the validation or test sets?*
  - *Was the model task benchmarked with human expert performance? Is the task possible for human experts and what is the agreement between human experts for the same task? (**interpretability**)*
  - *What is the performance on a dataset from a different population/institution/device than that used in the training data (in other words how does the model **generalize** to new data)?*
  - *What factors led to the decisions for the **operating point thresholds (cut-off)**? Does that fit the medical/clinical use case for optimal utility?*
  - *Based on the output of the model what are the consequences of accurate model decisions? What about model errors (false positives/false negatives)?*
  - *Does the input data require processing?*
  - *What is the data source for the model and are data **available** when needed in a deployment setting?*

Predictive performance of a ML model: metrics

Upon executing a test (or ML predictive model), we will evaluate:

- **Sensitivity** (True Positive Rate): the proportion of actual positives that are correctly identified.
- **Specificity** (True Negative Rate): the proportion of actual negatives that are correctly identified.
- **Accuracy**: the proportion of both true results (% true positives & true negatives) in the population.

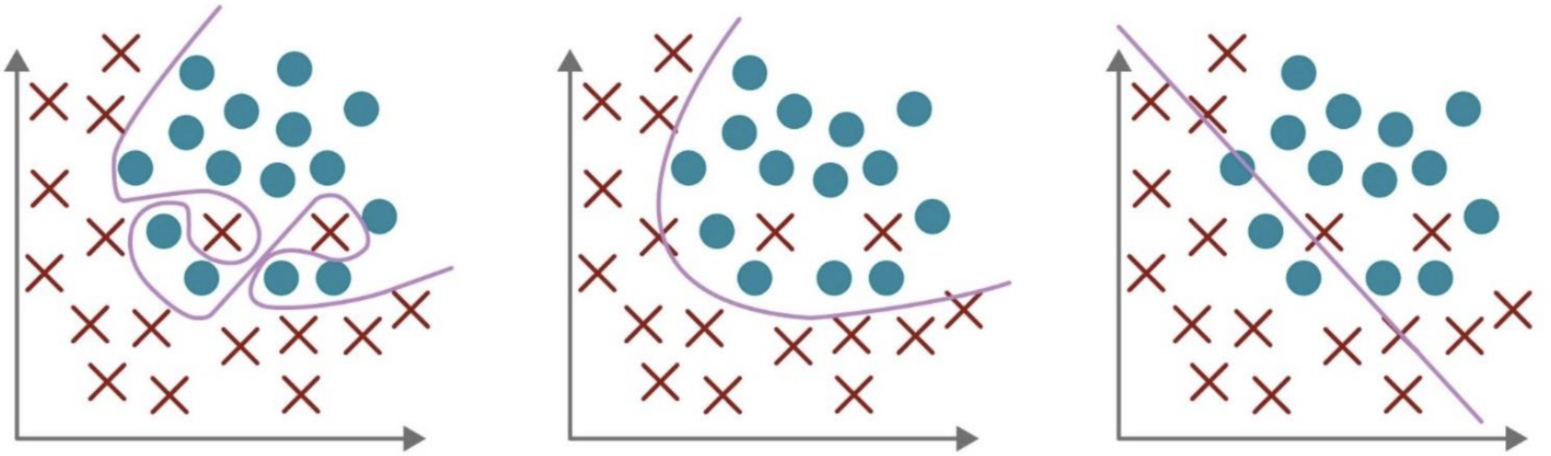
This performance results will vary according to the **cut-off** value: 0.5, 0.75, etc.

- lower cutoff (50%) → MAXIMIZE **sensitivity** (~ take in all possibly sick)
- higher cutoff (75%) → MAXIMIZE **specificity** (~ exclude all likely healthy)

Generalizability of a ML model: *overfitting* risk!

Here is another common challenge to be aware of:

- **overfitting** = when the model memorizes the random fluctuations, anomalies, and noise in the training set
- this will give HIGH ACCURACY in the training data, but the model outcomes won't be GENERALIZABLE to test or external data



Over-fitting: fitting too strongly to the training data

Appropriate-fitting

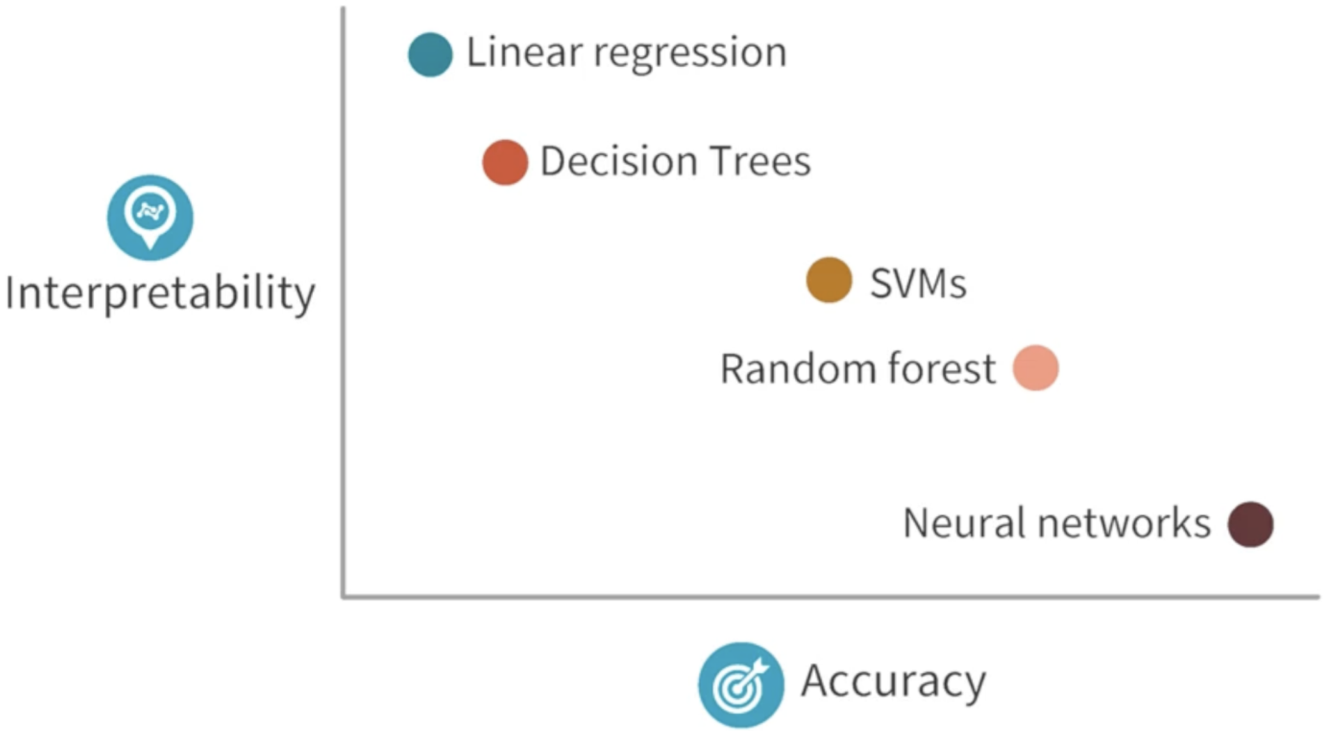
Under-fitting: unable to capture underlying trend of data

Source: https://www.coursera.org/learn/fundamental-machine-learning-healthcare/

Interpretability of ML algorithms

Understanding the domain and being aware of the possible spurious correlation affecting the outcome is key.

There is an ongoing tension between so called *"black box"* versus *interpretable model algorithms:* each may be more or less suitable for different applications



Source: https://www.coursera.org/learn/fundamental-machine-learning-healthcare/

ML common terms

- *features* = inputs
- *labels* = outputs
  - *real numerical values*
  - *categories or classes*
- *dataset* = a collection of examples
  - *training set* = a a set of examples or input-output pairs
  - *validation set* = set of examples to periodically assess the generalization performance of the model
  - *test set* = set of examples that we hold out until the very end of the model development process
- *example* = 1 input-output pair

- *training (fitting) the model* = minimizing the *loss* (prediction – output) of the training set (i.e. map inputs to labels as accurately as possible)
- *hyperparameters* = meta-level design choices on the program that trains the model
  - these are adjusted using training /validation sets in a "*training loop*"
- *hyperparameters tuning* = repeated adjustments of the steps to train and validate the model
- *(for classification) decision boundary (or cutoff point)* = probability number used to separate categories (e.g. sick/healthy, cancer/no cancer)

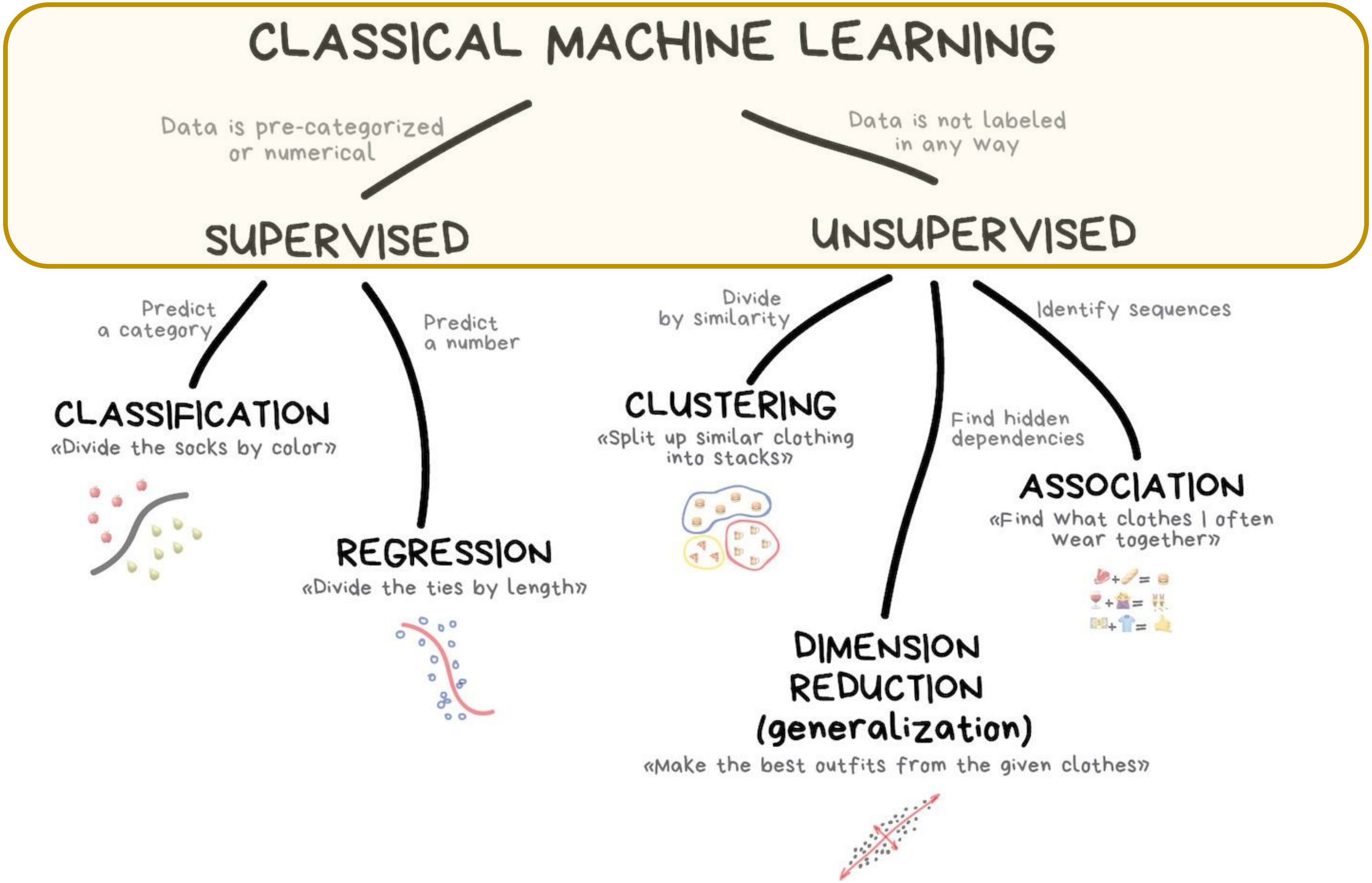
MODULE 5 – LECTURE OUTLINE

- Intro to Machine Learning (ML)
- Classification of ML algorithms
- *Supervised* ML examples
  - Logistic Regression
- *Unsupervised* ML examples
  - PCA
  - PLS-DA

Supervised or Unsupervised ML algorithms?

….another conceptual framework

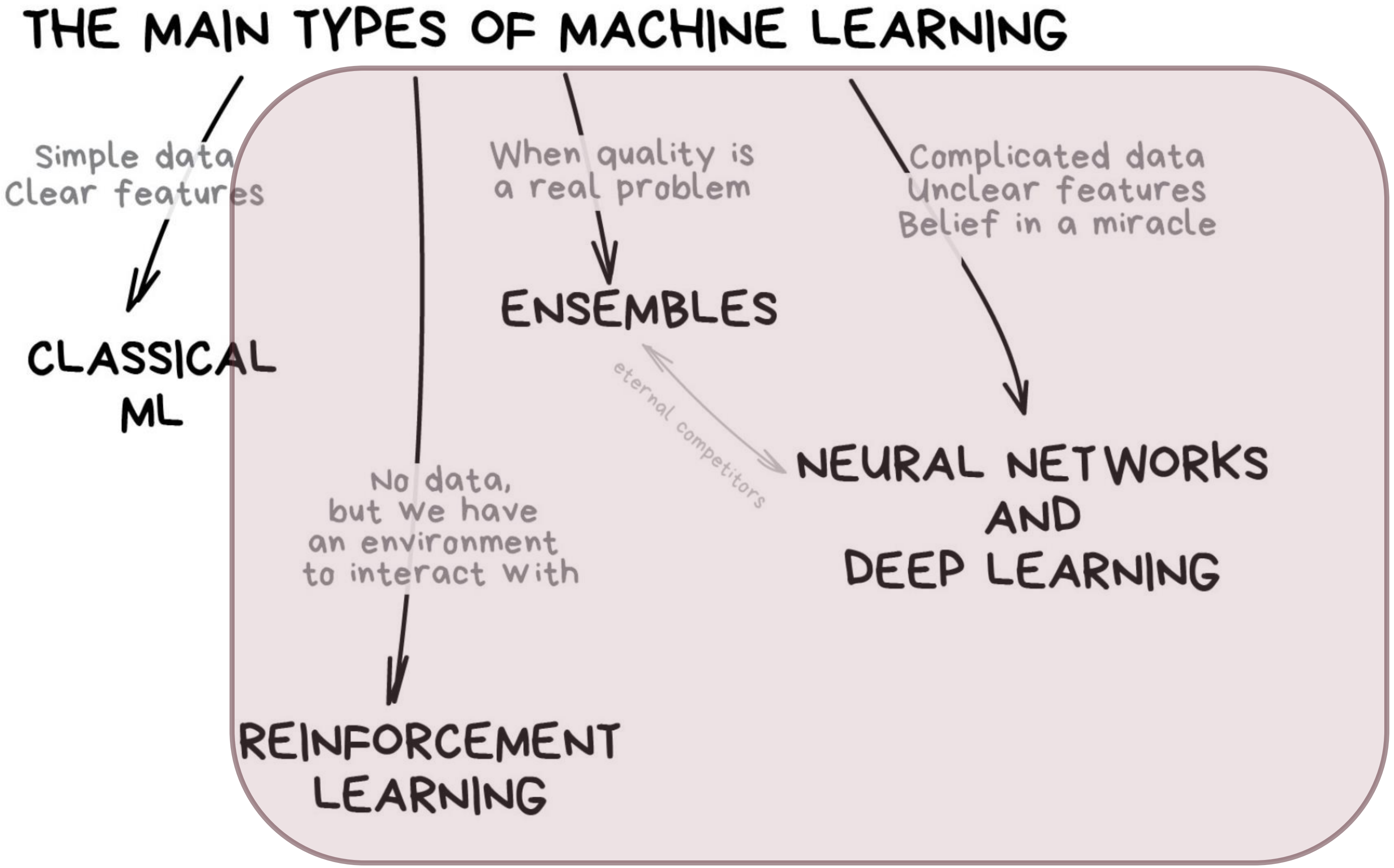
A fundamental distinction: supervised and unsupervised ML


CLASSICAL MACHINE LEARNING
Data is pre-categorized or numerical
Data is not labeled in any way
SUPERVISED
UNSUPERVISED
Predict a category
Predict a number
Divide by similarity
Identify sequences
CLASSIFICATION
«Divide the socks by color»
REGRESSION
«Divide the ties by length»
CLUSTERING
«Split up similar clothing into stacks»
Find hidden dependencies
ASSOCIATION
«Find what clothes I often wear together»
DIMENSION REDUCTION (generalization)
«Make the best outfits from the given clothes»

Source: Image from https://vas3k.com/blog/machine_learning/index.html

More *AI-ish* ML: reinforced learning, deep learning, etc.…

outside the scope of this course…



Source: Image from https://vas3k.com/blog/machine_learning/index.html

A fundamental distinction: supervised and unsupervised ML

- ML includes many different algorithms that can be used for understanding data. These algorithms can be classified as:

- **Supervised Learning Algorithms**:
  - building a model to estimate or predict an output based on one or more inputs
    - **Regression**: Modeling a relationship, the typical output variable is continuous (e.g. weight, height, time, etc.) or dichotomous.
    - **Classification**: Splits objects based on one of the attributes known beforehand. The the typical output variable is categorical (e.g. male or female, pass or fail, benign or malignant, etc.)

- **Unsupervised Learning Algorithms**:
  - finding structure and relationships among inputs. There is no "supervising" output
    - **Clustering:** Finding "clusters" of observations in a dataset that are similar to each other (*based on unknown features*).
    - **Association:** Finding "rules" that can be used to draw associations. For example, if a patient has a high biomarker X, he will have a low biomarker Y.
    - **Dimension reduction**: Assembling specific features into more high-level ones (e.g. PCA)

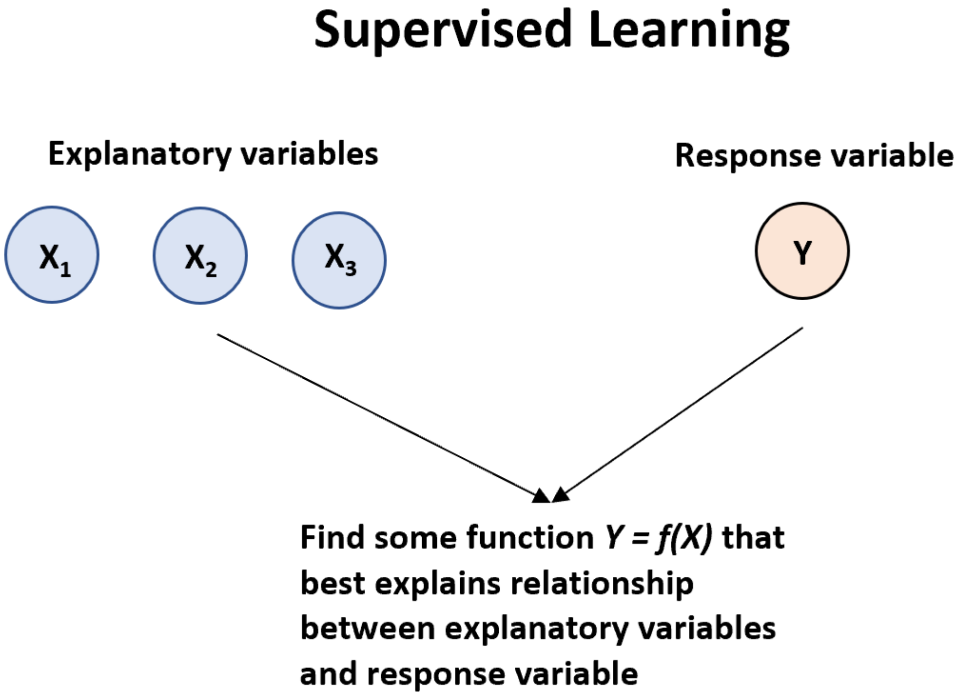
MODULE 5 – LECTURE OUTLINE

- Intro to Machine Learning (ML)
- Classification of ML algorithms
- *Supervised* ML examples
  - Logistic Regression
- *Unsupervised* ML examples
  - PCA
  - PLS-DA

Supervised ML algorithms

Supervised Learning Algorithms mechanics

- A supervised learning algorithm can be used when we have **one or more explanatory variables** (X_1, X_2, X_3, …) and a **response variable** (Y) and we would like to find some function that describes the relationship between the explanatory variables and the response variable:
- $Y = f(X) + \varepsilon$
- where
  - f () represents **systematic information that X provides about Y** and where
  - ε is a random error term independent of X with a mean of zero.



Supervised Learning

Explanatory variables

Response variable

X_1 X_2 X_3

Y

Find some function *Y = f(X)* that best explains relationship between explanatory variables and response variable

Source: https://www.statology.org/supervised-vs-unsupervised-learning/

Supervised Learning Algorithms purpose

There are two main reasons to use supervised learning algorithms:

1. **Prediction**: We often use a set of explanatory variables to predict the value of some response variable (e.g. using square footage and number of bedrooms to predict home price)

2. **Inference**: We may be interested in understanding the way that a response variable is affected as the value of the explanatory variables change (e.g. how much does home price increase, on average, when the number of bedrooms increases by one?)

- *Depending on whether our goal is inference or prediction (or a mix of both), we may use different methods for estimating the function f. For example, linear models offer easier interpretation but non-linear models that are difficult to interpret may offer more accurate prediction.*

Supervised Learning: commonly used algorithms

Most commonly used supervised learning algorithms:

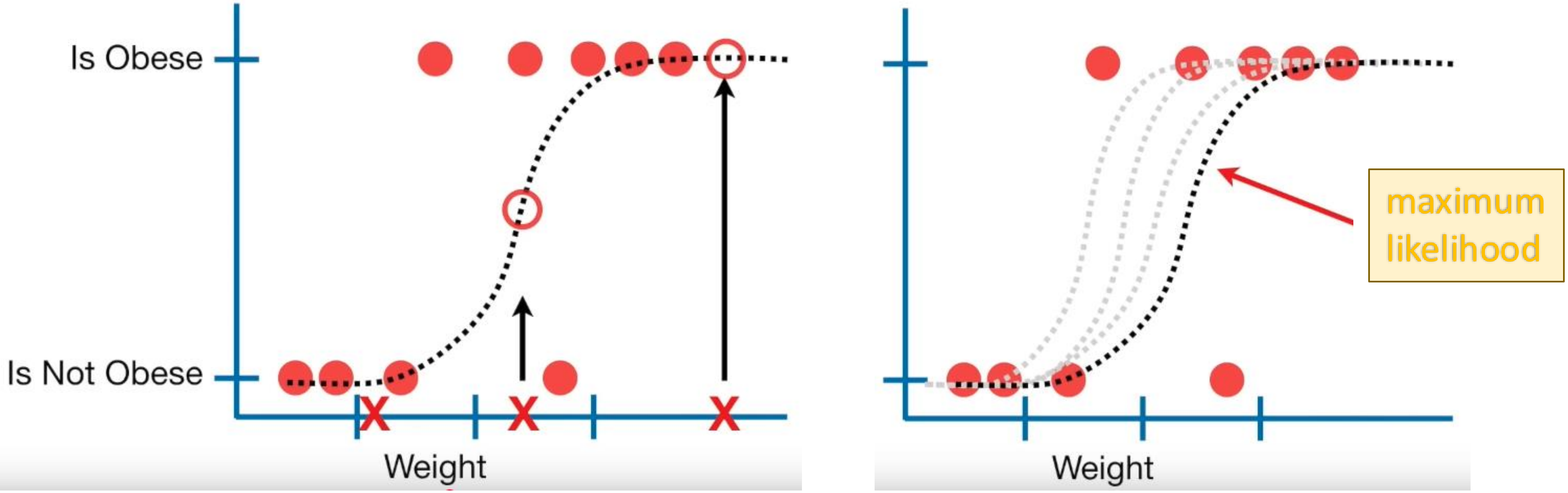
- Linear regression
- Logistic regression
- Linear discriminant analysis
- Quadratic discriminant analysis
- Decision trees
- Naive bayes
- Support vector machines
- Neural networks

Logistic Regression for classification

An example of **supervised** ML algorithm

Purpose of logistic regression

- **Logistic regression** tells the **PROBABILITY** of some phenomenon (e.g. being obese) and is normally used for **binary classification**:
  - e.g. if the probability a mouse is obese > 50%, we will classify it as obese
- Unlike linear regression (that uses least square) the logistic regression line is fit using the **maximum likelihood** criterion



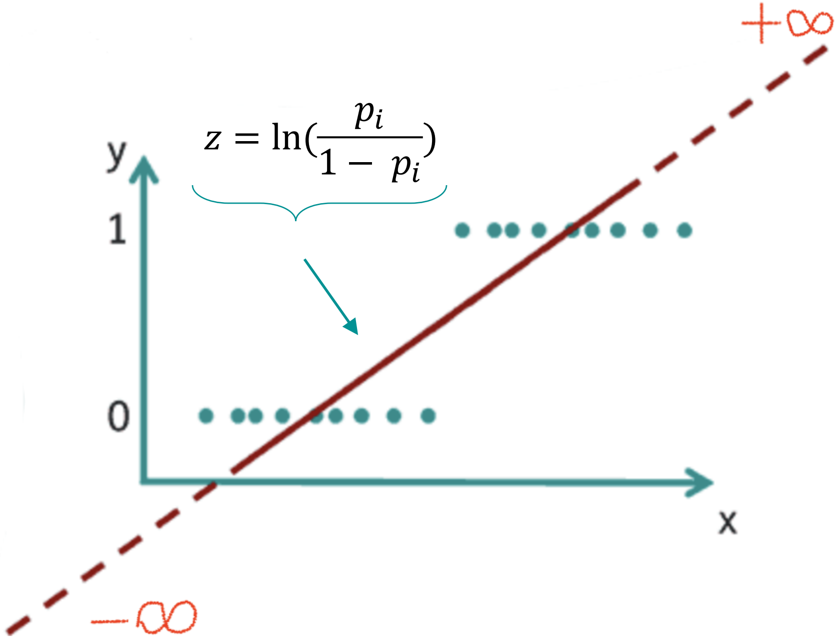
maximum likelihood

Source: https://www.youtube.com/watch?v=yIYKR4sgzI8&t=204s

The “*logic*” of logistic regression

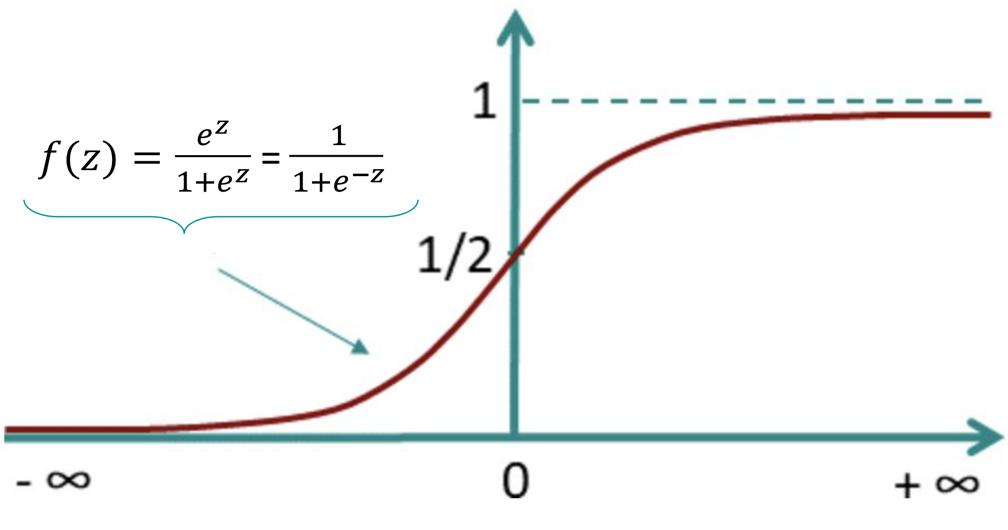
- **Logistic regression** is similar to **linear regression**, but… now the response variable **y** can only take **values 0 and 1**
- So we need **some** function to restrict the value range for the prediction between 0 and 1: the ***logistic function*** (or ***sigmoid*** function).

Logit = Log(odds) function



$$\underbrace{z = \ln(\frac{p_i}{1 - p_i})}$$

Logistic function (*sigmoid*)



$$\underbrace{f(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}}$$

Source: https://datatab.net/tutorial/logistic-regression

The logit and inverse-logit functions

If $\boldsymbol{p_i} = P(Y = 1 \mid X_k)$ is the probability of $\mathbf{Y = 1}$ given a vector of independent variables X_k, **linear regression** $\boldsymbol{y_i} = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}$ is not appropriate.

We can use **logistic regression**, where:

- 1) Instead of directly predicting $\boldsymbol{y_i}$, we predict the ***logit*** of the probability $\boldsymbol{logit(p_i)}$ which maps probabilities *[0, 1]* to real numbers *[−∞ , +∞]*

logit $$\boldsymbol{logit(p_i)} = \ln\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{\beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}}$$

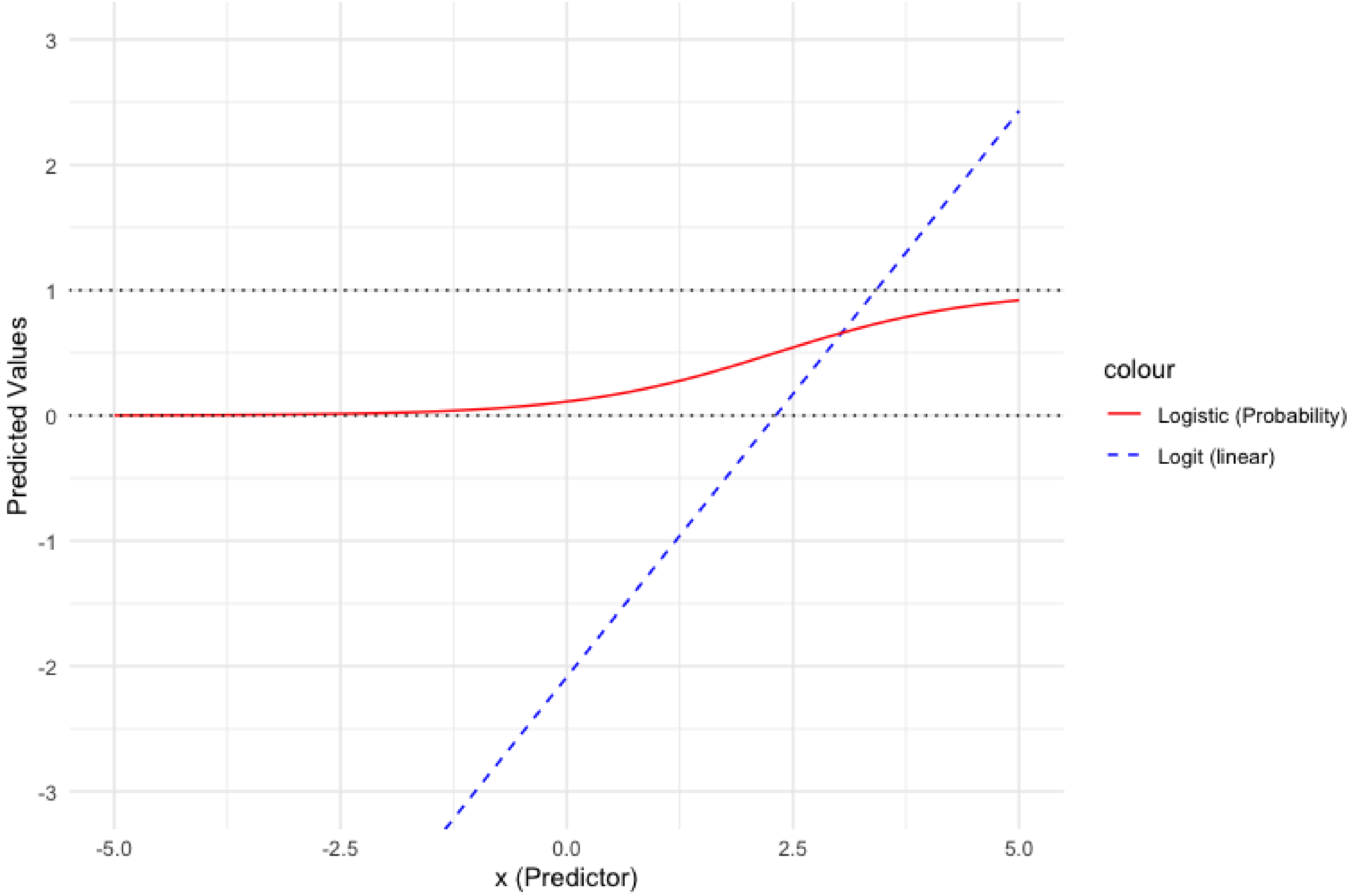
- 2) To obtain a **meaningful** probability value for $\mathrm{y} = 1$, we solve for $\boldsymbol{p}$ with the *logistic()* function that maps values back to the range *[0, 1]*:

inverse-logit $$\boldsymbol{p_i} = \frac{1}{1+e^{-(\beta_0+\beta_1 x_{1,i}+\cdots+\beta_k x_{k,i})}}$$

(or “**link**” function,
aka **sigmoid** function)

- 3) Finally, the result from the *link()* function (*[0,1)]*) is then passed through a decision rule (i.e. a ***threshold***) to divide the outcome into classes as required

Comparing predictions of Y from LOGIT [−∞ , +∞] and LOGISTIC function [0 , 1]

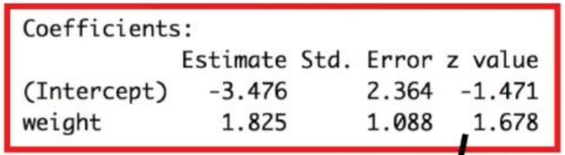


Interpreting a coefficient of Y (disease) for continuous X (weight)

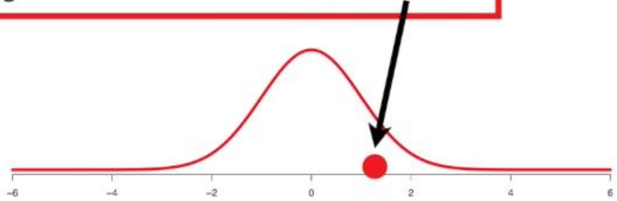
- In logistic regression, **coefficients** are in terms of the *log(odds)*
  - **INTERCEPT**: when weight = 0, log(odds) of disease = -3.476
  - **SLOPE**: for every *additional* 1 unit of weigh, log(odds of disease) increases by 1.825
- Sometimes **coefficients** are converted to odds, simply by $odds = e^{\beta_j}$

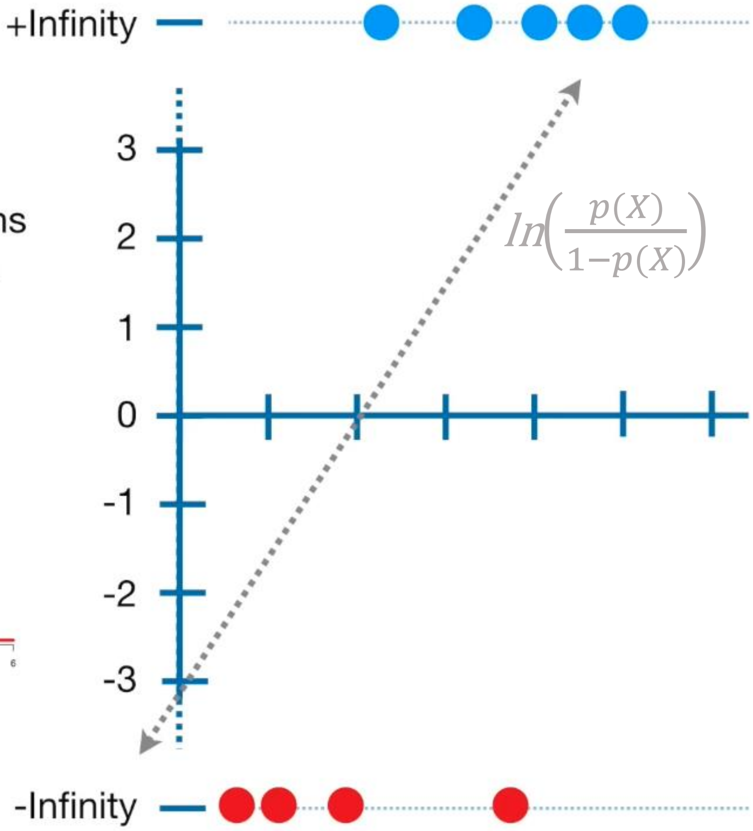
y = **-3.48** + **1.83** × weight

Again, the z-value is the number of standard deviations the estimate is from 0 on a standard normal curve…

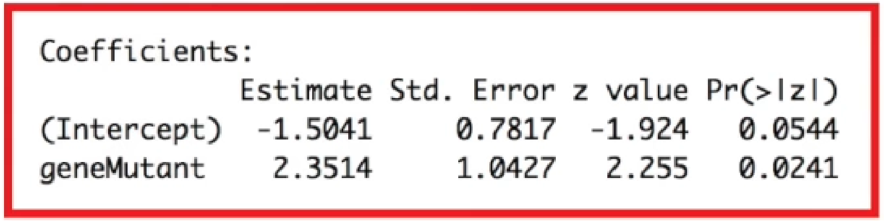


```
Coefficients:
            Estimate Std. Error z value
(Intercept)   -3.476      2.364  -1.471
weight         1.825      1.088   1.678
```





Interpreting a coefficient of Y (obesity) for discrete X (YES mutated gene)



```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.5041     0.7817  -1.924   0.0544
geneMutant    2.3514     1.0427   2.255   0.0241
```

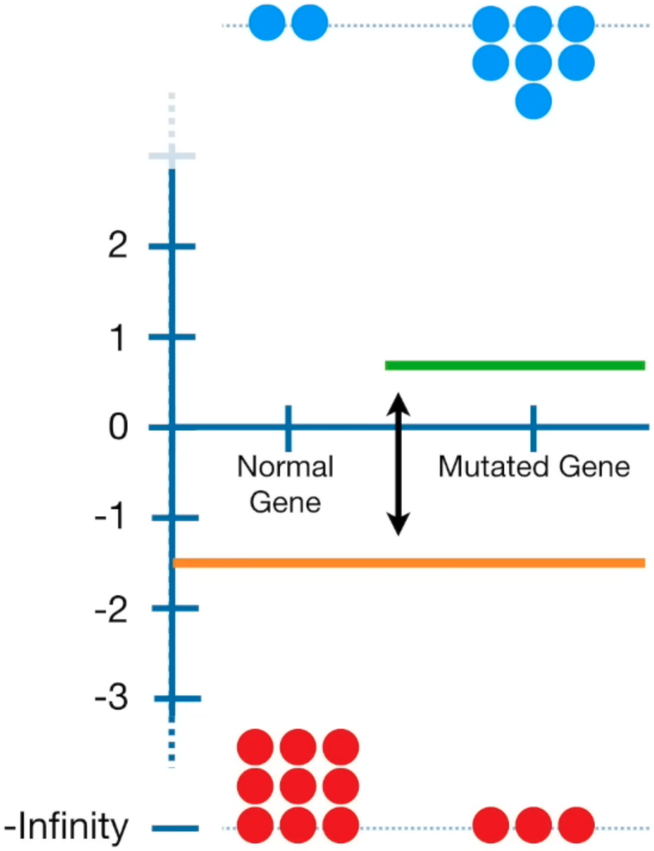
log (odds of Y) $= \log(\text{odds gene}_{\text{normal}}) \times B_1 +$

$\log\left(\frac{\text{odds gene}_{\text{mutated}}}{\text{odds gene}_{\text{normal}}}\right) \times B_2$

↑

…can be converted into division, this term is a **log(odds ratio)**.

It tells us, on a log scale, how much having the mutated gene increases (or decreases) the odds of a mouse being obese.



log (odds of Y) $= \log(2/9)\text{x } \beta_1 + \log\left(\frac{7/3}{2/9}\right)\beta_2 = -1.5\text{x } \beta_1 + 2.35\text{x } \beta_2$

MODULE 5 – LECTURE OUTLINE

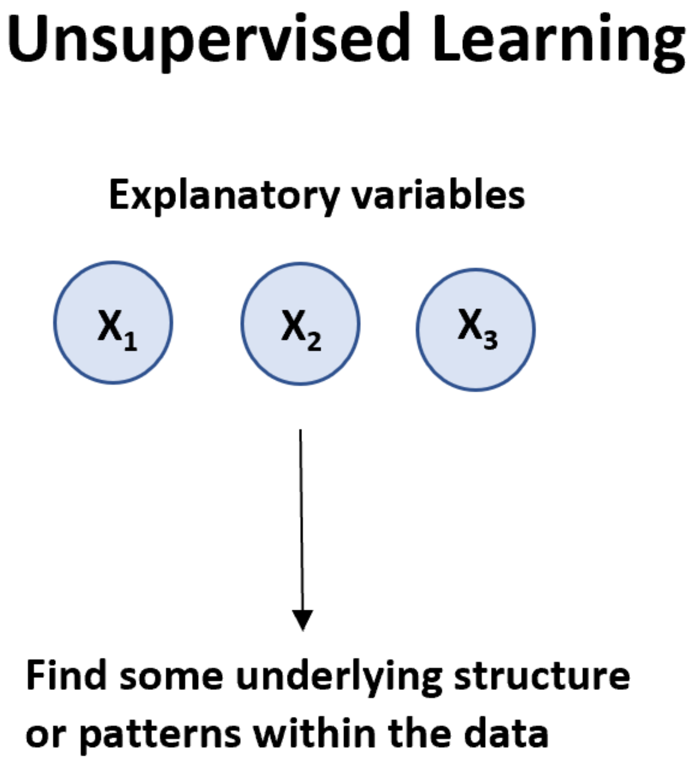
- Intro to Machine Learning (ML)
- Classification of ML algorithms
- *Supervised* ML examples
  - Logistic Regression
- *Unsupervised* ML examples
  - PCA
  - PLS-DA

PCA for dimension reduction

An example of *unsupervised* ML algorithm

Unsupervised Learning Algorithms mechanics

An unsupervised learning algorithm can be used when we have a list of variables (X_1, X_2, X_3, …, X_p) and we would simply like to find underlying structure or patterns within the data.



Unsupervised Learning

Explanatory variables

Find some underlying structure or patterns within the data

Source: https://www.statology.org/supervised-vs-unsupervised-learning/

Unsupervised Learning Algorithms typical purpose

There are two main types of unsupervised learning algorithms:

1. **Clustering:** Using these types of algorithms, we attempt to find “clusters” of observations in a dataset that are similar to each other. This is often used in retail when a company would like to identify clusters of customers who have similar shopping habits so that they can create specific marketing strategies that target certain clusters of customers.

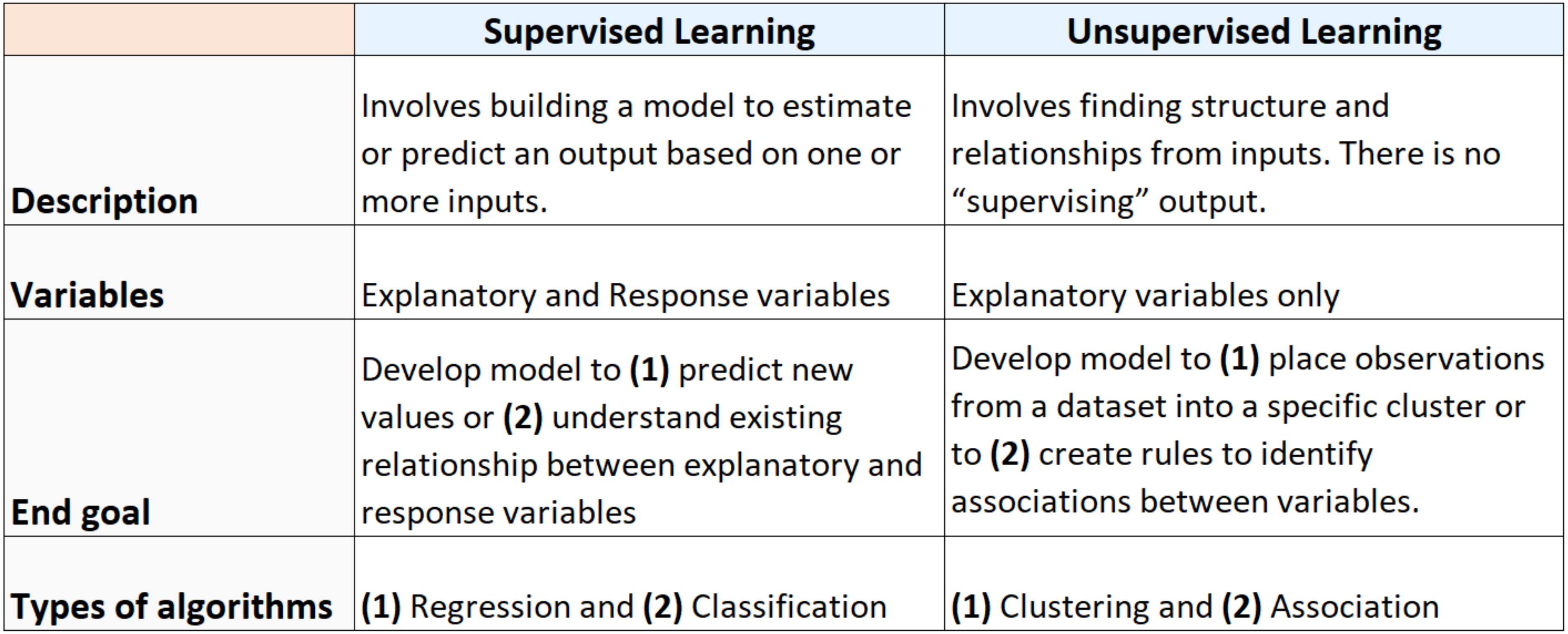
2. **Association:** Using these types of algorithms, we attempt to find “rules” that can be used to draw associations. For example, retailers may develop an association algorithm that says “if a customer buys product X they are highly likely to also buy product Y.”

Unsupervised Learning: commonly used algorithms

- Most commonly used unsupervised learning algorithms:
  - Principal component analysis
  - K-means clustering
  - K-medoids clustering
  - Hierarchical clustering
  - Apriori algorithm

Summary: Supervised vs. Unsupervised Learning

- Here are the key differences between supervised and unsupervised learning algorithms:



	Supervised Learning	Unsupervised Learning
Description	Involves building a model to estimate or predict an output based on one or more inputs.	Involves finding structure and relationships from inputs. There is no “supervising” output.
Variables	Explanatory and Response variables	Explanatory variables only
End goal	Develop model to **(1)** predict new values or **(2)** understand existing relationship between explanatory and response variables	Develop model to **(1)** place observations from a dataset into a specific cluster or to **(2)** create rules to identify associations between variables.
Types of algorithms	**(1)** Regression and **(2)** Classification	**(1)** Clustering and **(2)** Association

Source: https://www.statology.org/supervised-vs-unsupervised-learning/

Principal Component Analysis (PCA)

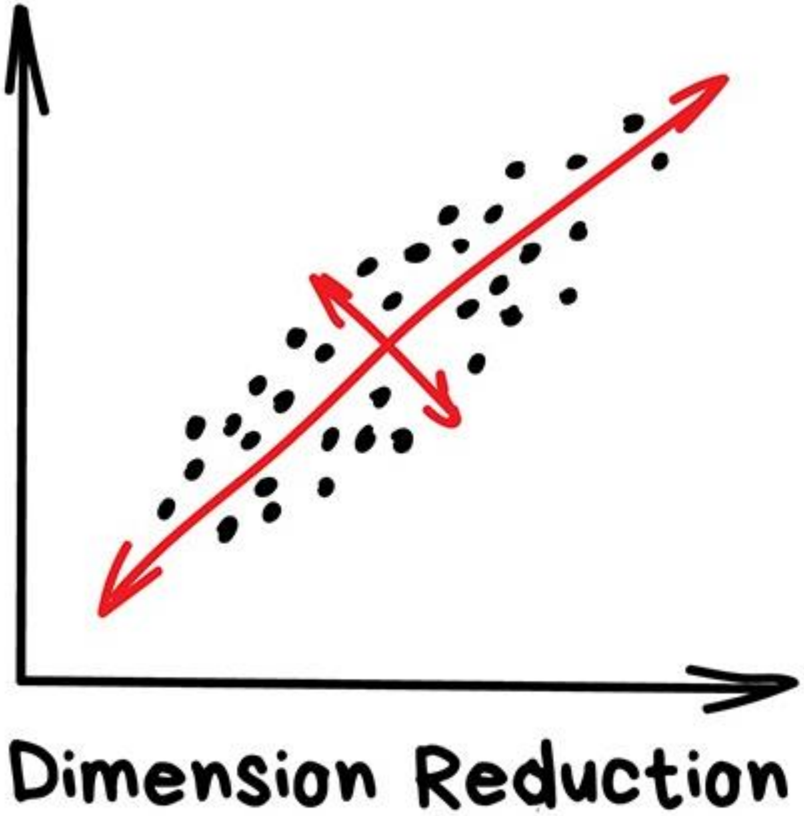
A type of *unsupervised* learning algorithm for dimensionality reduction

Purpose of PCA

- The goal of PCA is to transform a high-dimensional dataset into a lower-dimensional dataset while retaining as much of the variance in the data as possible.

- Common use cases of PCA:

  1. to reduce the dimensionality of high-dimensional datasets
  2. to visualize the structure of the data
  3. to remove noise and redundant information from the data
  4. as a preprocessing step for other machine learning algorithms



Source: Image from https://vas3k.com/blog/machine_learning/index.html

Covariance

Population mean is unknown

$$var(x) = \frac{\sum_i^n (x_i - \overline{x})^2}{N-1}$$

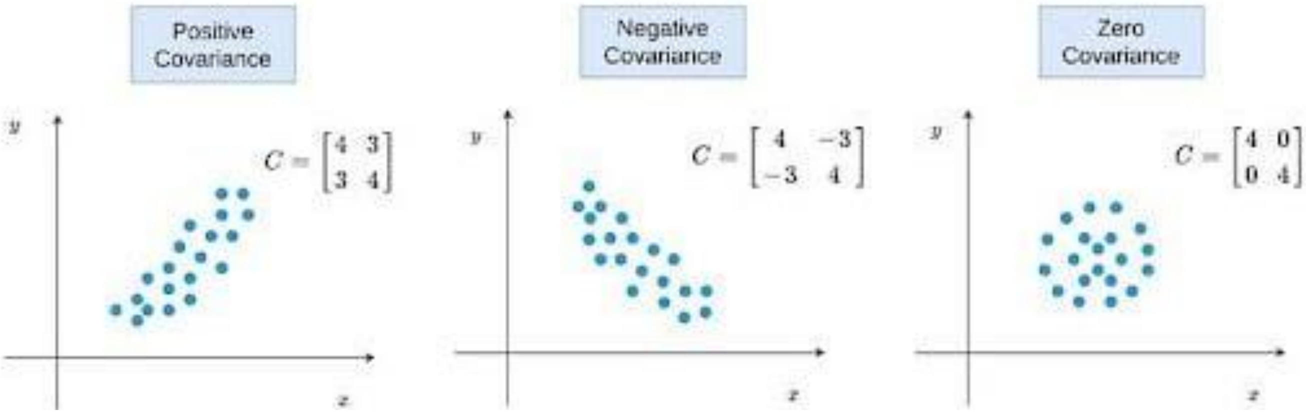
Population mean is unknown

$$cov(x,y) = \frac{\sum_i^n (x_i - \overline{x}) \cdot (y_i - \overline{y})}{N-1}$$

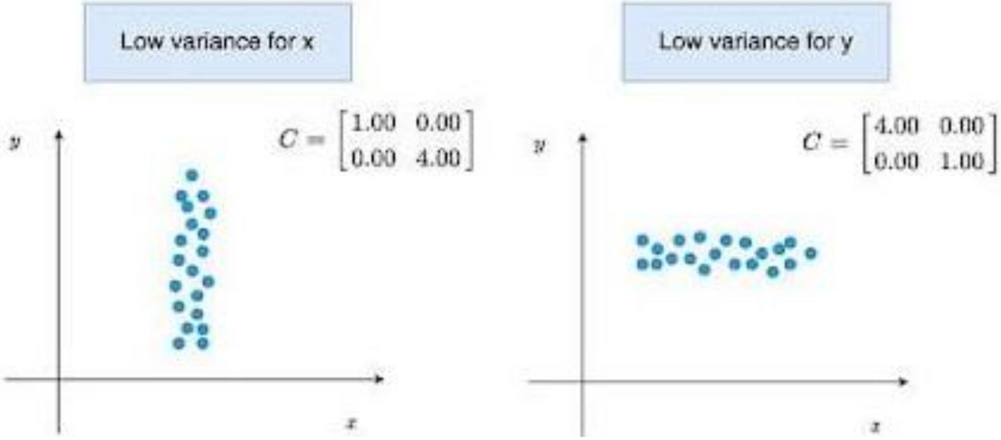
$$\begin{array}{cc} & \begin{array}{ccc} x & y & z \end{array} \\ \begin{array}{c} x \\ y \\ z \end{array} & \begin{bmatrix} var(x) & cov(x,y) & cov(x,z) \\ cov(x,y) & var(y) & cov(y,z) \\ cov(x,z) & cov(y,z) & var(z) \end{bmatrix} \end{array}$$

Variance measures how the values vary in a variable. **Covariance** measures how changes in one variable are associated with changes in a second variable.

Covariance



Positive, negative and zero covariance.



Different variances and zero covariance.

Source: https://builtin.com/data-science/covariance-matrix

PCA

PCA originally is a linear algebra operation.

It is a transformation method that creates (weighted linear) combinations of the original variables in a data set, with the intent that the new combinations will capture as much variance in the dataset as possible while eliminating correlations (i.e., redundancy).

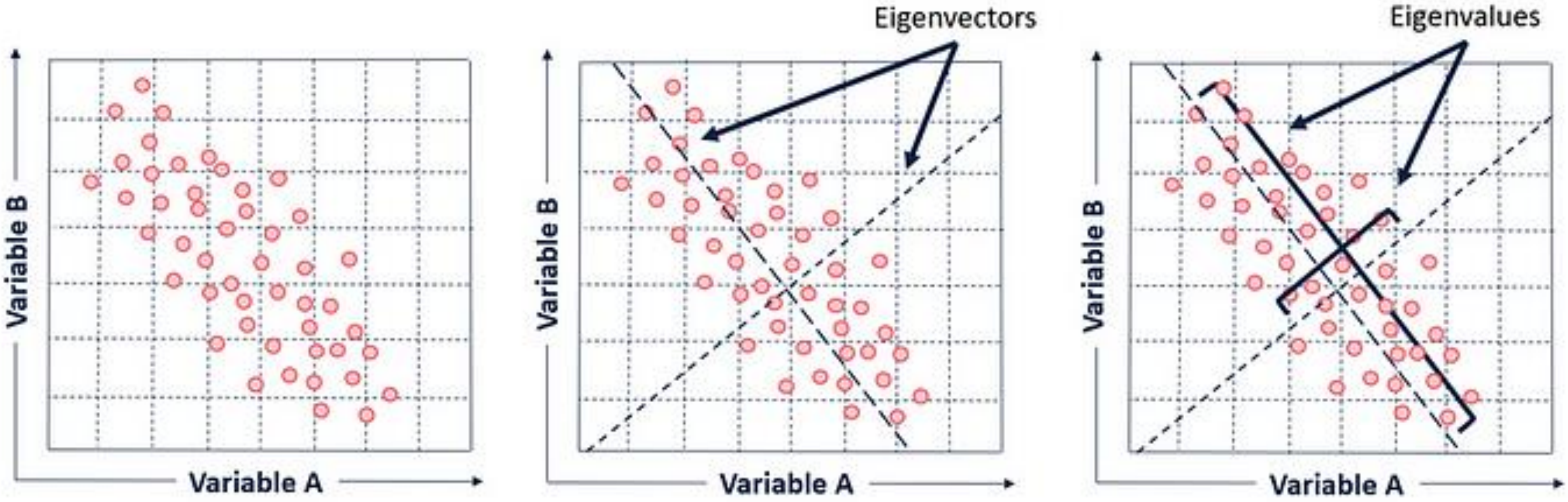
PCA creates the new variables using the eigenvectors and eigenvalues calculated from the covariance matrix of your original variables.

Eigenvectors & Eigenvalues

In the context of PCA

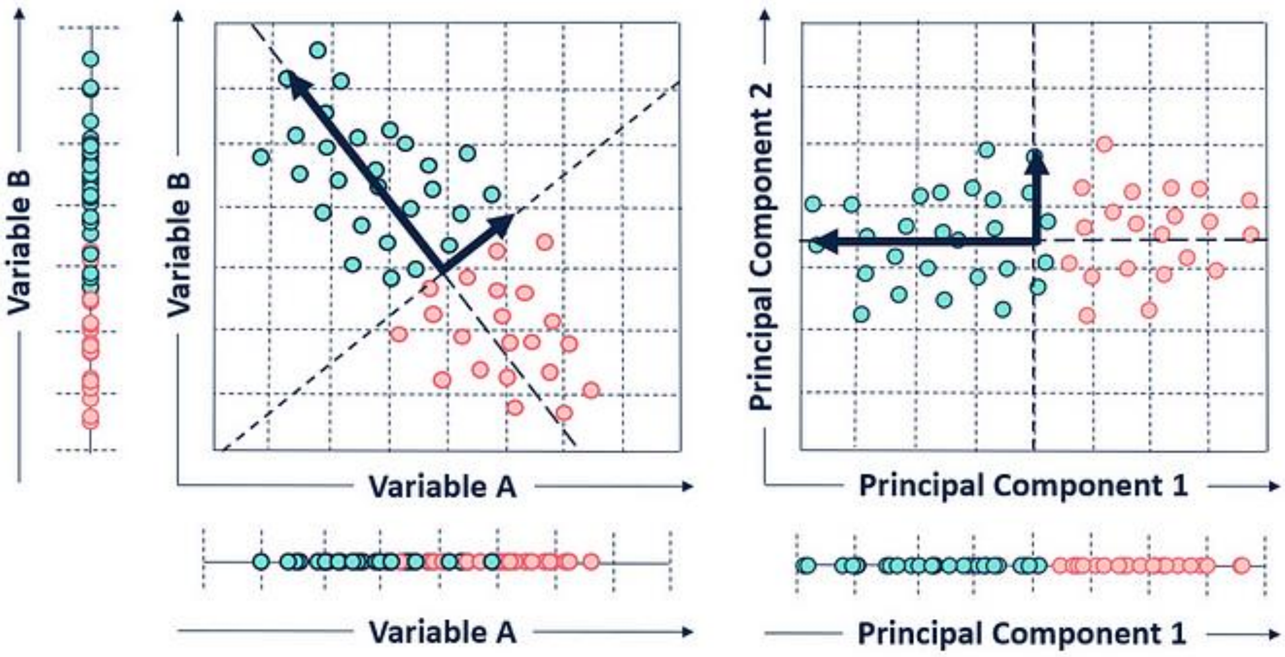
- The **eigenvectors** of the covariance matrix define the directions of the principal components calculated by PCA.

- The **eigenvalues** associated with the eigenvectors describe the variance along the new axis.



Source: https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383

Principal components

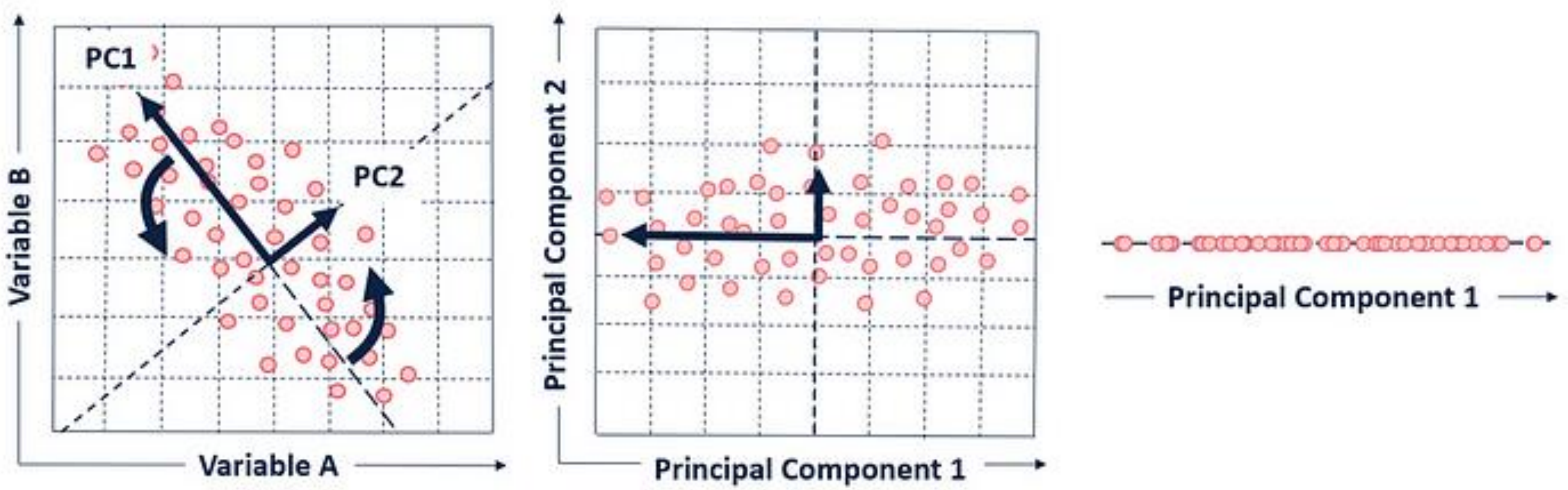


Principal Component 1 accounts for variance from both variables A and B. (dimension reduction)

The principal components (eigenvectors) are sorted by descending eigenvalue. The principal components with the highest eigenvalues are "picked first" as principal components because they account for the most variance in the data.

Source: https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383

Principal components



To convert our original points, we create a projection matrix.
This projection matrix is just the selected eigenvectors concatenated to a matrix.
We can then multiply the matrix of our original observations and variables by our projection matrix.
The output of this process is a transformed data set, projected into our new data space — made up of our principal components!

Source: https://towardsdatascience.com/tidying-up-with-pca-an-introduction-to-principal-components-analysis-f876599af383

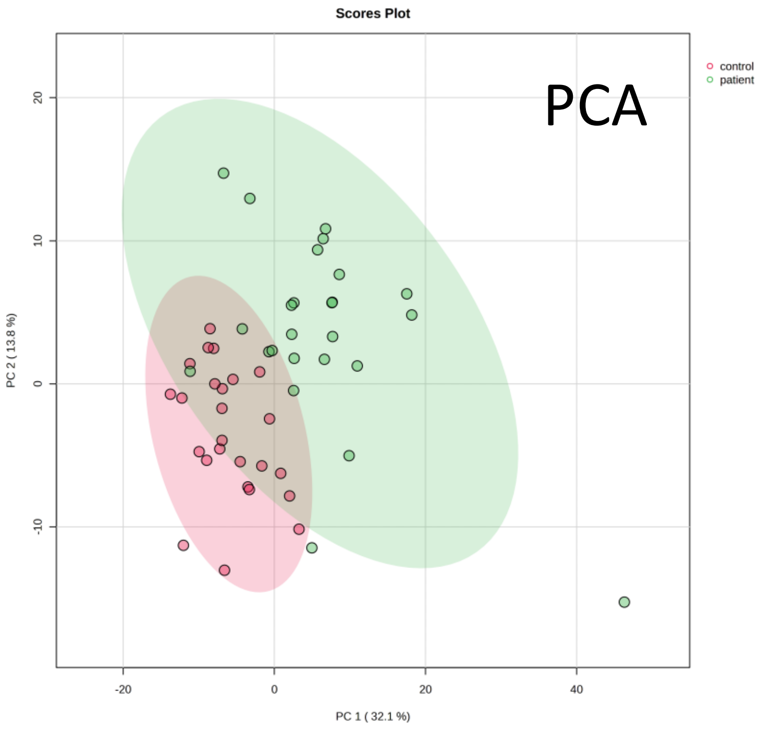
PLS Discriminant Analysis (PLS-DA)

A *supervised* alternative to PCA performing simultaneous dimensionality reduction and classification

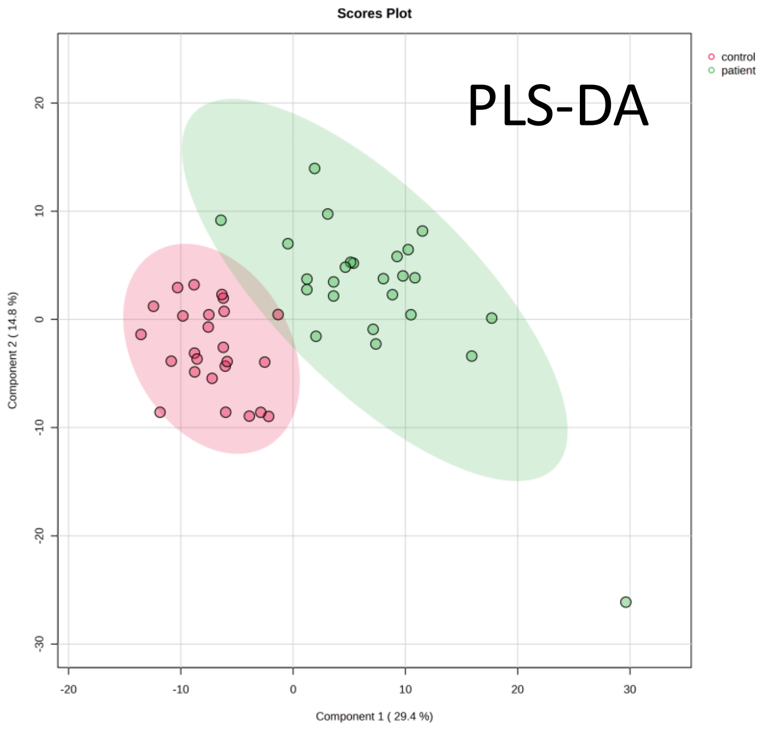
Purpose: PLS-DA vs PCA

- PCA is completely unsupervised (i.e. you don’t know in advance if there are classes in your dataset)
- In PLS-DA you know how your dataset is divided in classes from the response vector Y. The goal here is then to project the predictors into a space, while maximizing the ratio $= \frac{Between\ group\ distances}{Within\ group\ distances}$
- Common scenarios for using PLS-DA: omics sciences.

Scores plot: PCA vs PLS-DA



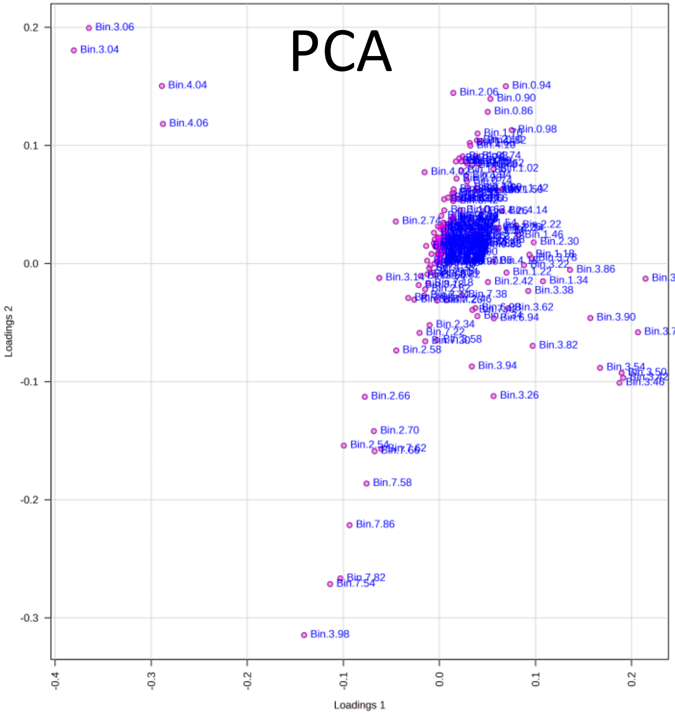
Samples projected in the space of Principal Components

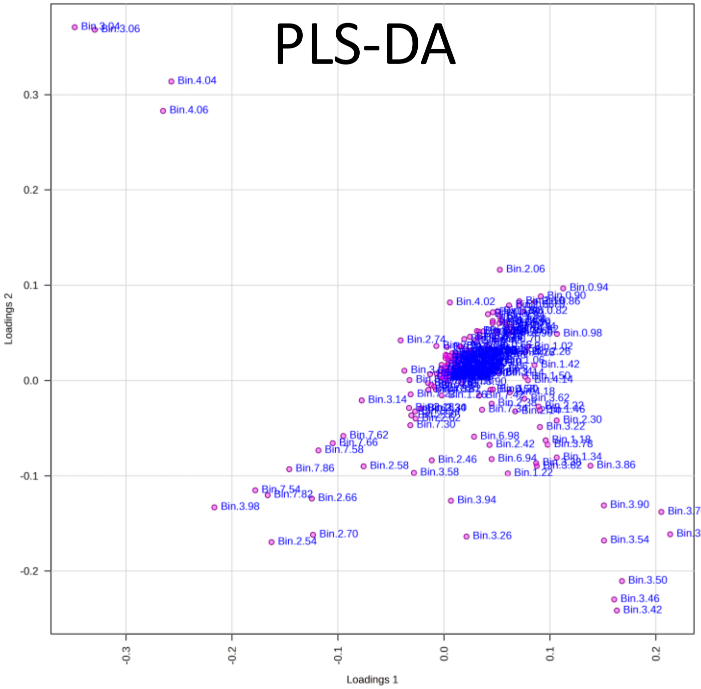


Samples projected in the space of latent variables (components) that maximize the separation between groups

*Source: Test data (**NMR spectral bins**) provided by METABOANALYST platform: https://www.metaboanalyst.ca*

Loadings plot: PCA vs PLS-DA



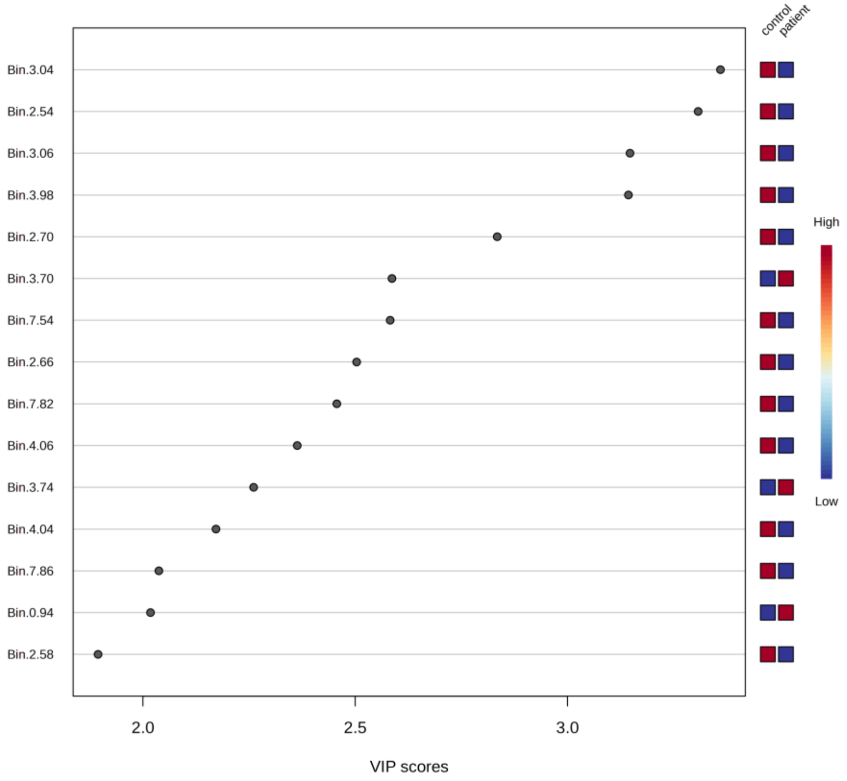


The loading vectors (here shown as points) represent the original variables in the space PCs.

The loading vectors (here shown as points) represent the original variables in the space of latent components retrieved by PLS-DA.

*Source: Test data (**NMR spectral bins**) provided by METABOANALYST platform: https://www.metaboanalyst.ca*

Feature Importance in PLS-DA

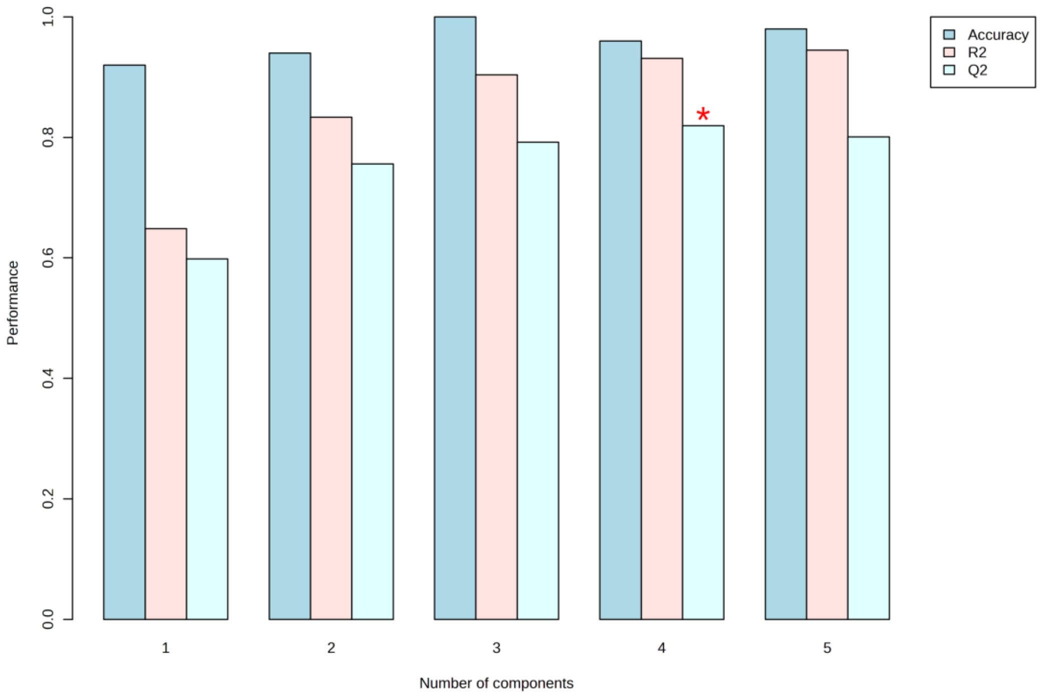


VIP (Variable Importance in Projection) scores, ranking the variables based on their significance in the PLS-DA **model of classification**.

…very useful to select potential biomarkers!

*Source: Test data (**NMR spectral bins**) provided by METABOANALYST platform: https://www.metaboanalyst.ca*

Cross validation in PLS-DA



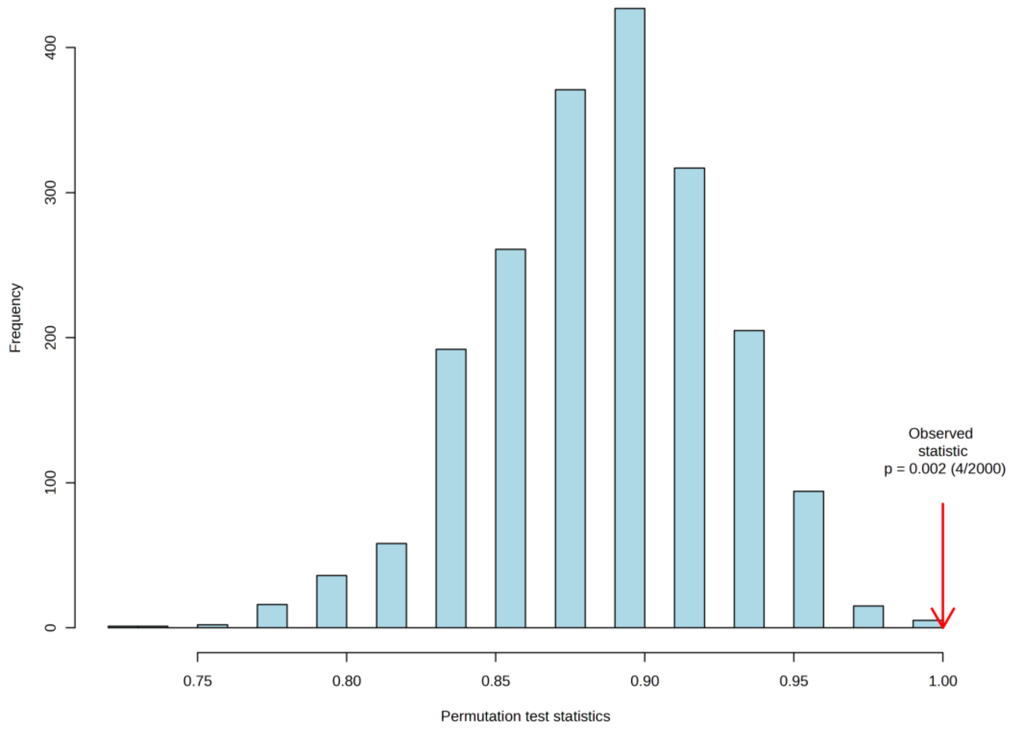
PLS-DA generate a model of classification.

By partitioning the dataset and iteratively testing the model, cross validation estimate the predictive ability of the model.

Q^2 is an analogous of R^2 in regression: the higher the better!

*Source: Test data (**NMR spectral bins**) provided by METABOANALYST platform: https://www.metaboanalyst.ca*

Permutation in PLS-DA



Permutation testing is a non-parametric approach to assess the significance of a model's results.

In the context of PLS-DA, this test helps verify whether the observed classification accuracy is better than what would be expected by chance.

*Test data (**NMR spectral bins**) provided by METABOANALYST platform: https://www.metaboanalyst.ca*