

# Computing Methods for Experimental Physics and Data Analysis

## Data Analysis in Medical Physics

Lecture 9: Predictive models, Machine-Learning for feature classification

Alessandra Retico

[alessandra.retico@pi.infn.it](mailto:alessandra.retico@pi.infn.it)

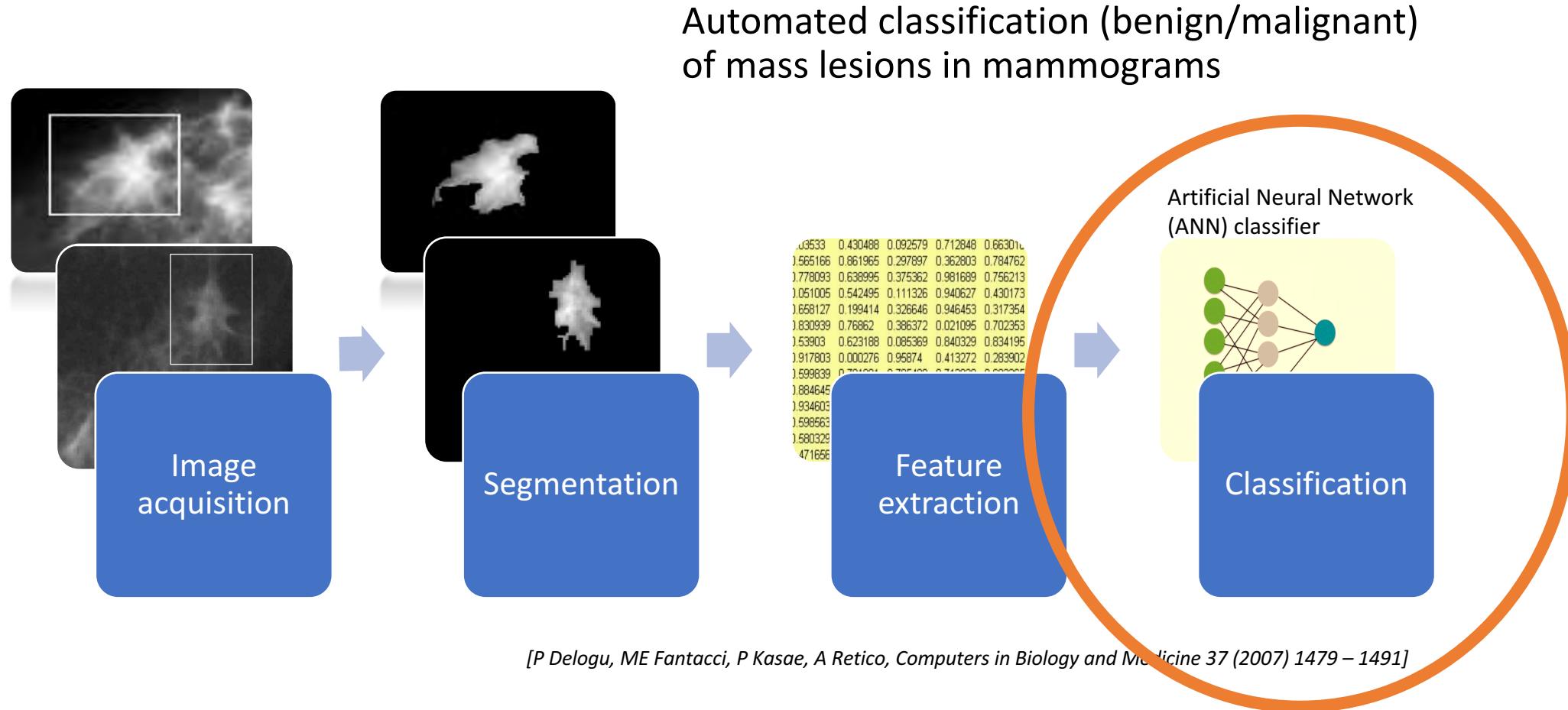
INFN - Pisa

# Todays' objectives

---

- Explore the use of predictive models in medical data analysis
  - Regression models
  - Supervised classification (categorization)
  - Unsupervised learning (clustering data)
- Quantification of model performance
  - Figures of merits: Sensitivity, Specificity, AUC , Precision, Recall, F1
- Estimate of model robustness
  - Cross validation methods: k-fold cross validation, leave-one out cross validation

# Hand-crafted feature + Machine Learning classification

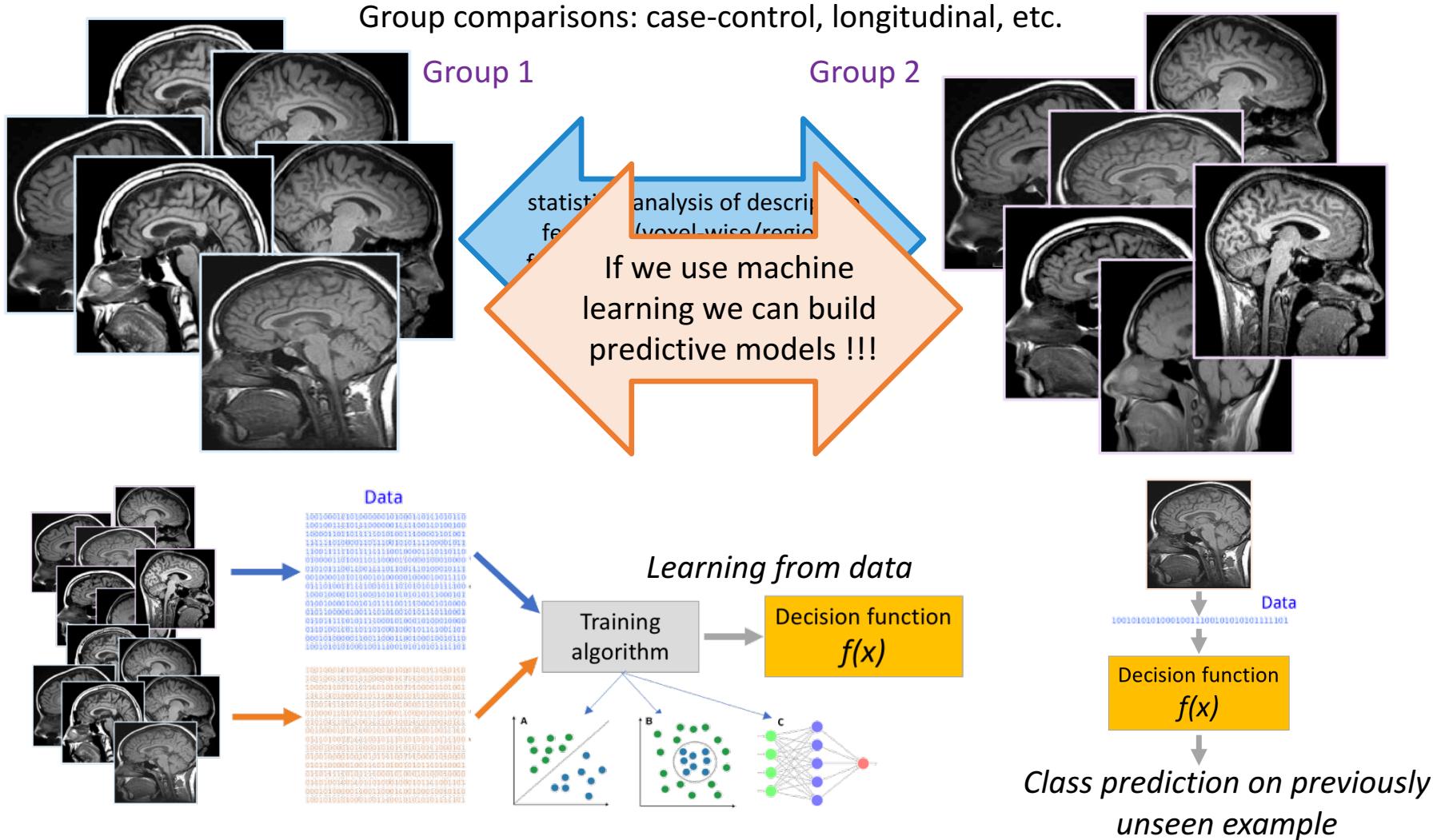


# Machine learning in Medical Imaging

---

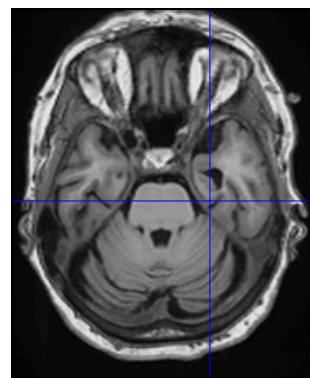
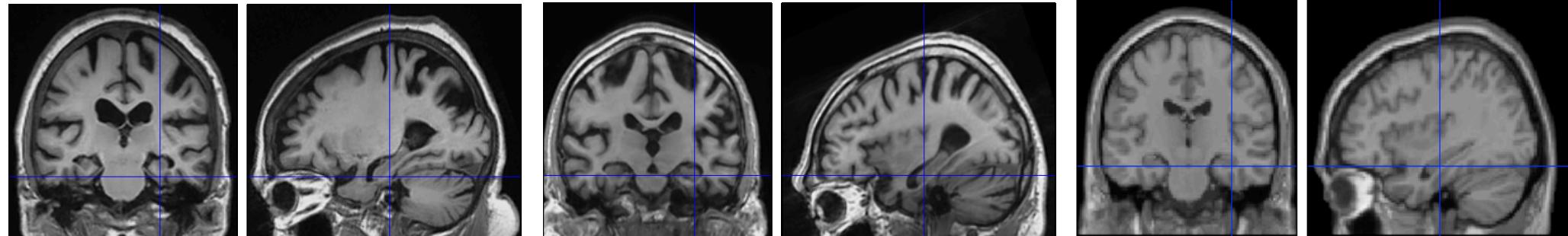
- Machine learning approaches are gaining popularity in the Medical Imaging community
- In addition to group characterization, they allow categorization of individual's previously unseen data: predictive diagnosis.
- Interesting studies have already been carried out in tumor segmentation and classification, in the study of neurological and psychiatric disorders (Alzheimer's disease, Parkinson's disease, autism spectrum disorders, schizophrenia, bipolar disorder, ... ), etc.
- International research consortia are collecting more and more data sample, including, MRI, PET, RX, CT, US data, to enable the training of machine-learning algorithms

# Between-group comparisons

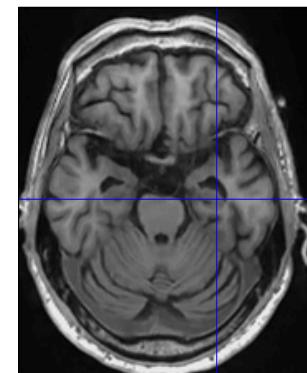


# Studying brain atrophy

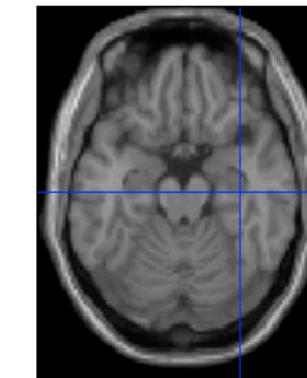
---



AD

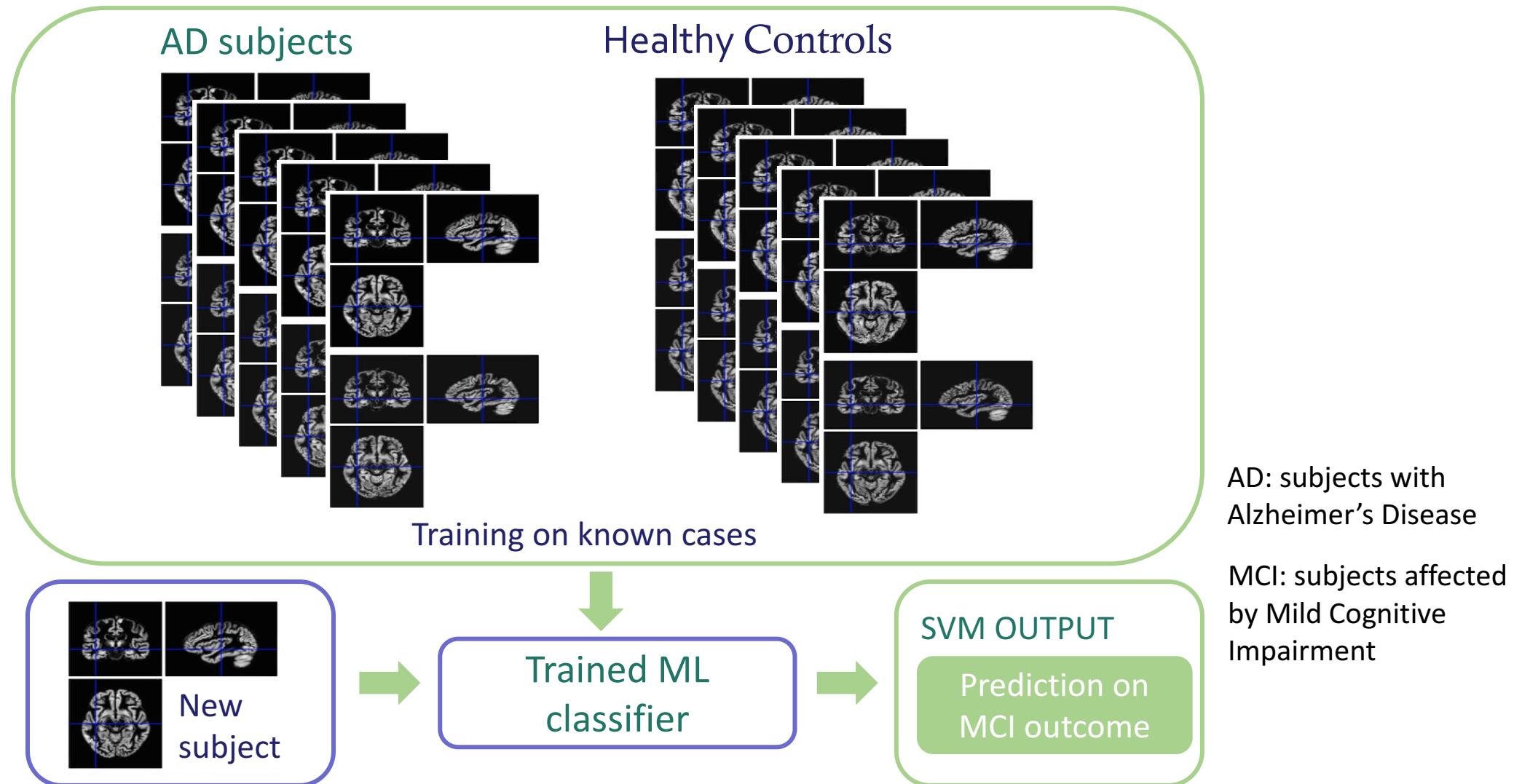


CTRL



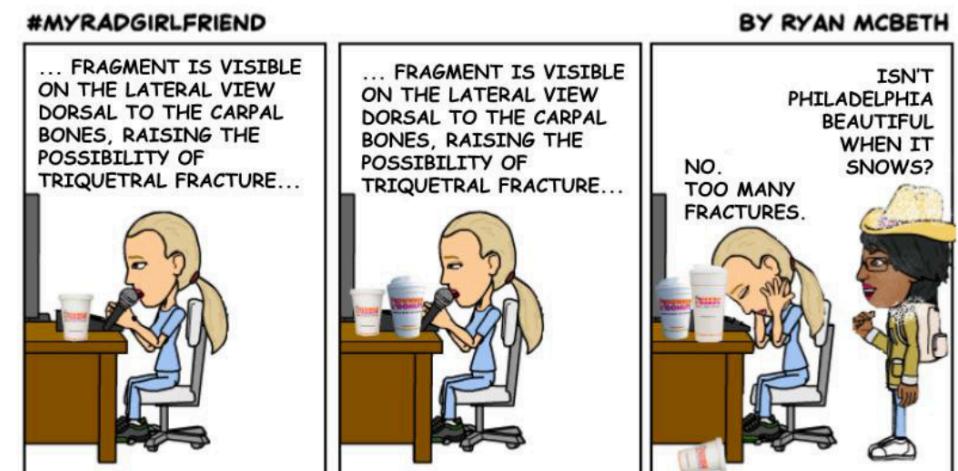
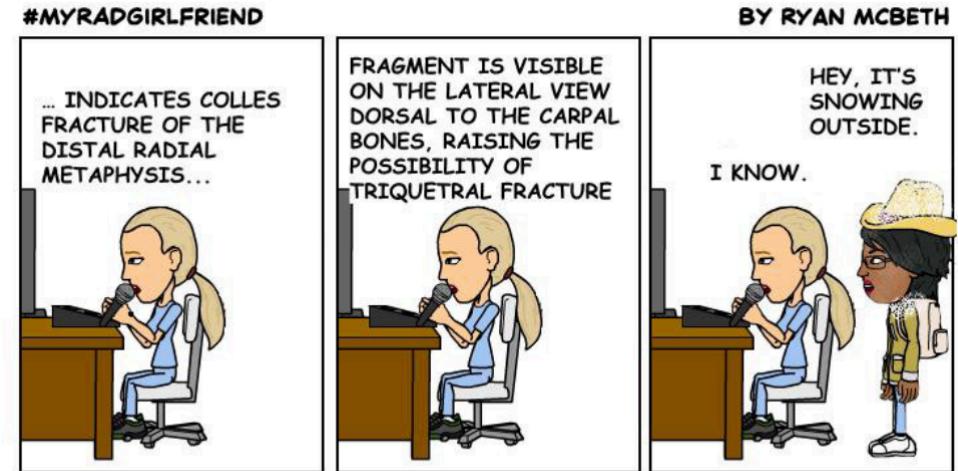
Healthy  
younger subject

# Example of use of machine learning



# Why using machine-learning techniques?

- Some tasks cannot be defined well, except by examples.
- Relationships and correlations can be hidden within large amounts of data.
  - Machine Learning/Data Mining may be able to find these relationships.
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
  - *Learning from examples*



# Ingredients for classification problems

---

## Training a classifier

- Train, test and validation sets
- Choose the model

## Classifier performance evaluation

- Figures of merit
  - Sensitivity, Specificity and Receiver Operative Characteristic curve (ROC)
  - Area under the ROC curve (AUC)
- Cross validation procedure

## Dimensionality problems

- Feature reduction

# Train, Test, Validation sets

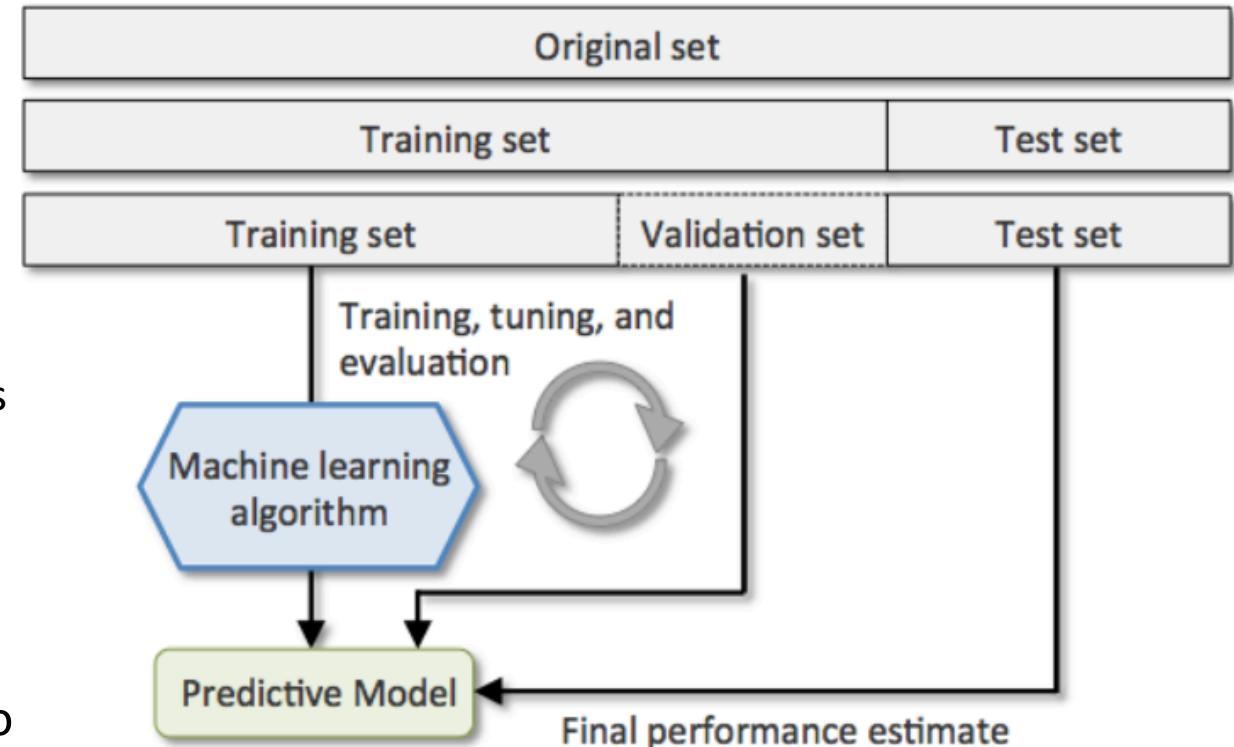
The data sample should be partitioned into subsets for training, validation and testing the classification algorithms.

**Training:** allowing the algorithm to set weights and “learn” how to classify data, using the training set

- **Training set:** a subset of items, with known classification
- This produces **classifier**, i.e. a **mapping** from features to class.
- **Validation set:** a subset of items, with known classification, that can be used to optimize the system performance

**Classification:** using these training-set weights to classify data of previously unknown category

- **Independent validation set (test set):** a subset of items, with known classification, where to evaluate the classifier performance

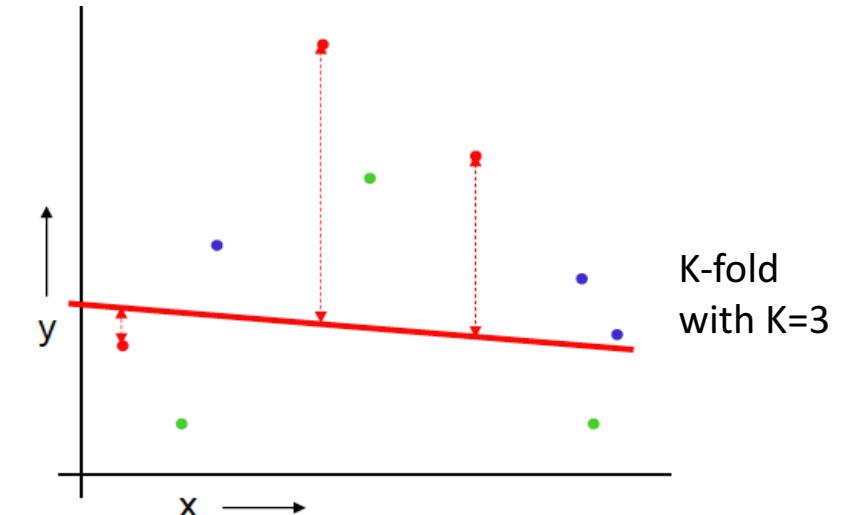
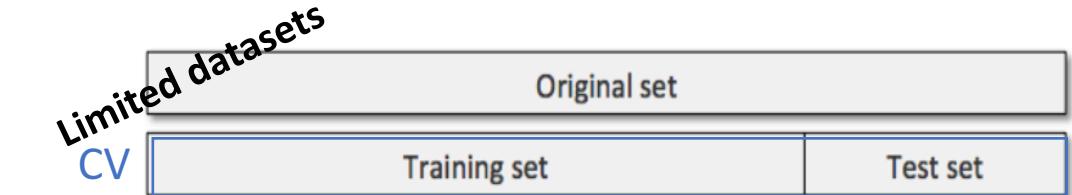
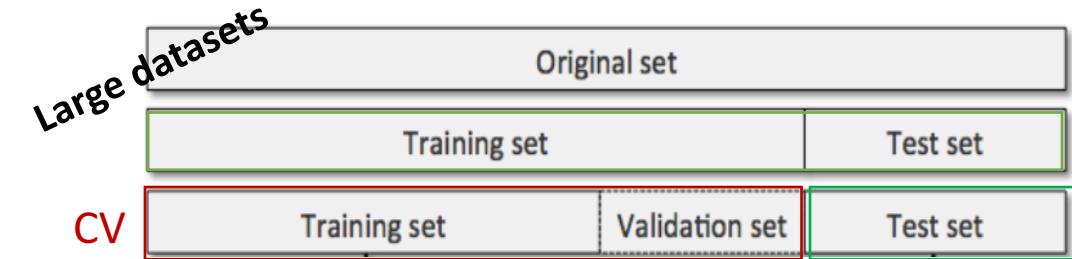


# Cross validation procedures

Cross-validation: data resampling procedures are used to evaluate machine-learning models on a limited data sample.

## K-fold cross-validation:

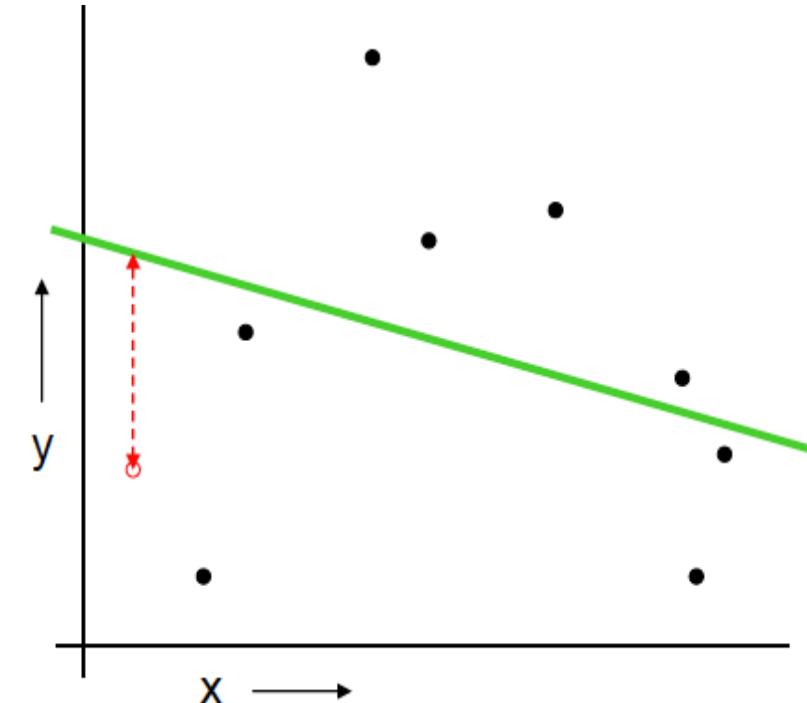
- Original sample partitioned into K subsamples. Of K subsamples, one is retained as validation data while remaining ( $K - 1$ ) subsamples are used as training data.
- Cross-validation process is repeated K times (the folds), with each of the K subsamples used exactly once as the validation data.
- The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation.



# Cross validation procedures

---

- Leave-one-out cross-validation (LOOCV):
  - It is a  $k$  fold CV with  $k=N$ , where  $N$  is the number of subjects.
  - Uses a single observation from original sample as validation data, and the remaining observations as training data.
  - This is repeated such that each observation in the sample is used once as the validation data.
  - The results obtained on the single observations should be combined to produce a the model estimation.
  - We cannot assign an error to the performance we obtained.



# Figures of merit to quantify classification performance

**Sensitivity = True Positive Rate (TPR)**

**Specificity = $1 - \text{False Positive Rate (FPR)}$**

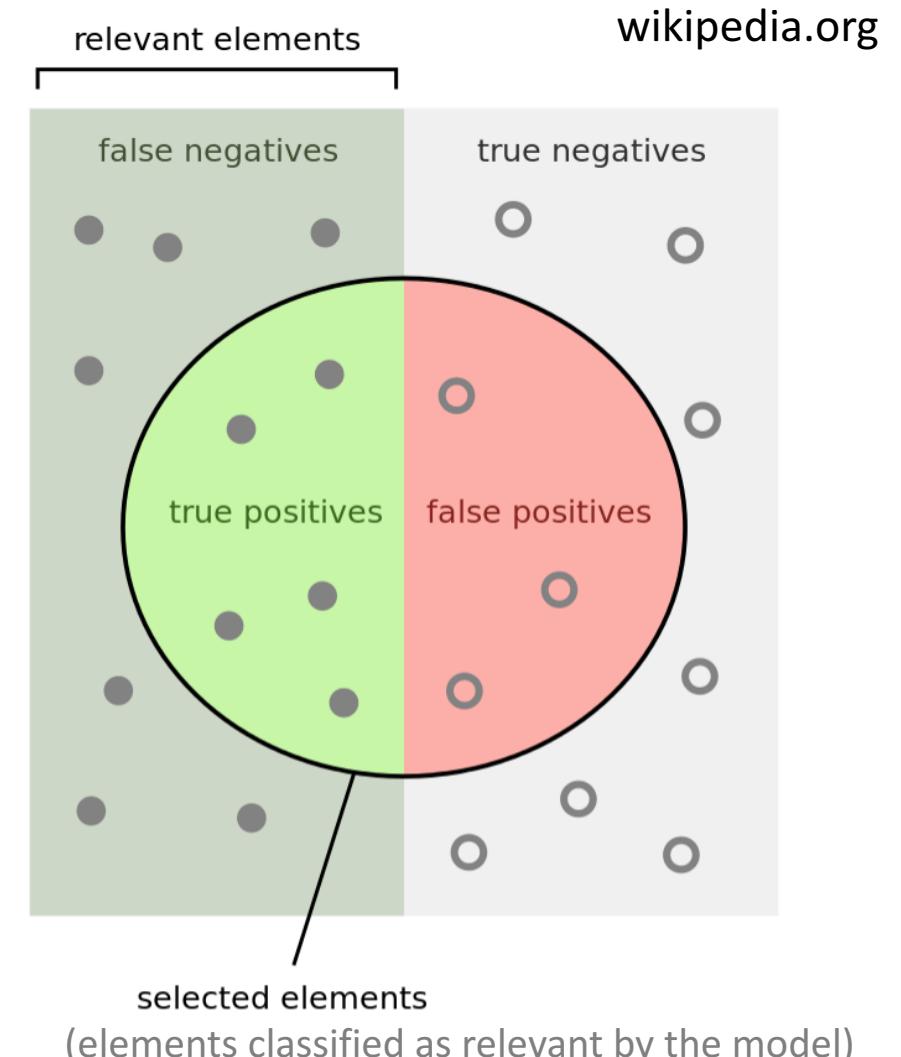
$$TPR = \frac{TP}{TP + FN} \quad (\text{true positive rate})$$

$$FPR = \frac{FP}{FP + TN} \quad (\text{false positive rate})$$

**Accuracy**

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

| CONFUSION MATRIX |     | Predicted class     |                     |
|------------------|-----|---------------------|---------------------|
|                  |     | YES                 | NO                  |
| Actual class     | YES | True Positive (TP)  | False Negative (FN) |
|                  | NO  | False Positive (FP) | True Negative (TN)  |



# Figures of merit to quantify classification performance

## Precision and recall

How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

or Positive Predictive Value (PPV)



How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

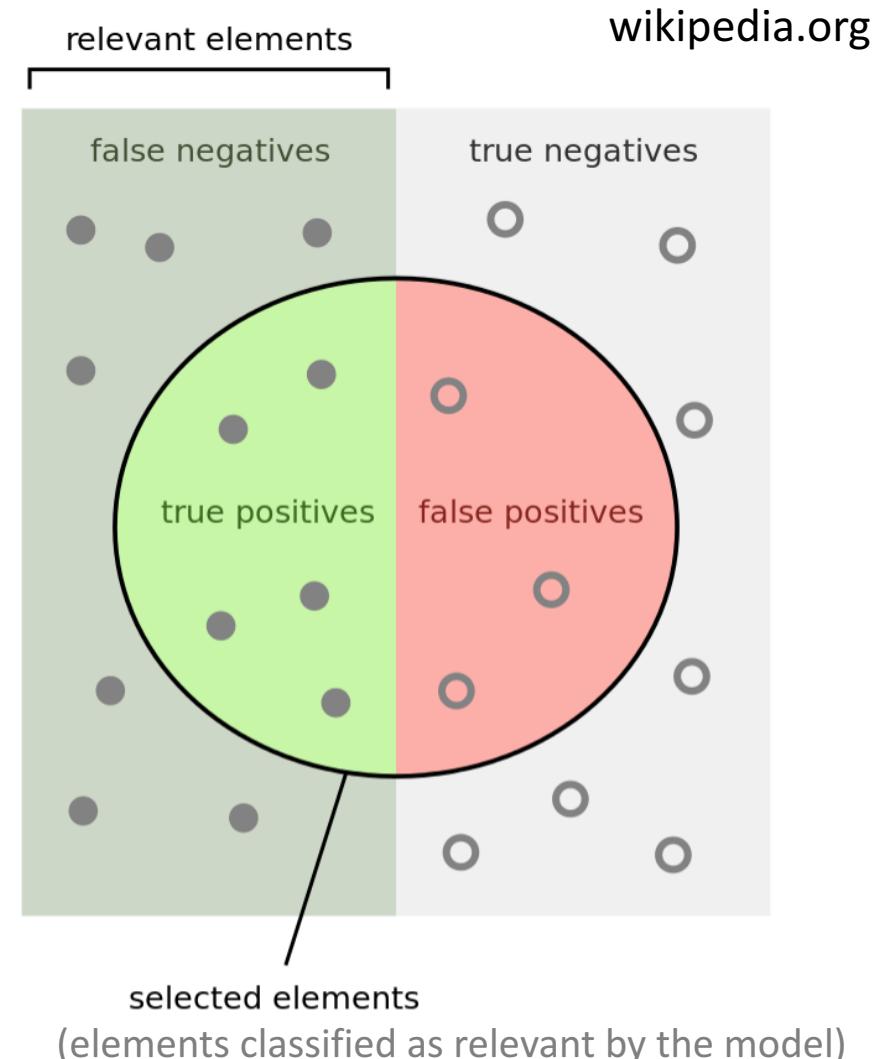
or True Positive Rate (TPR), or Sensitivity



## $F_1$ metric

The  $F_1$  score is the harmonic mean of the precision and recall

$$F_1 = \left( \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} \right) = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$



# Figures of merit to quantify classification performance

## Area under the Receiver Operating Characteristic (ROC) curve

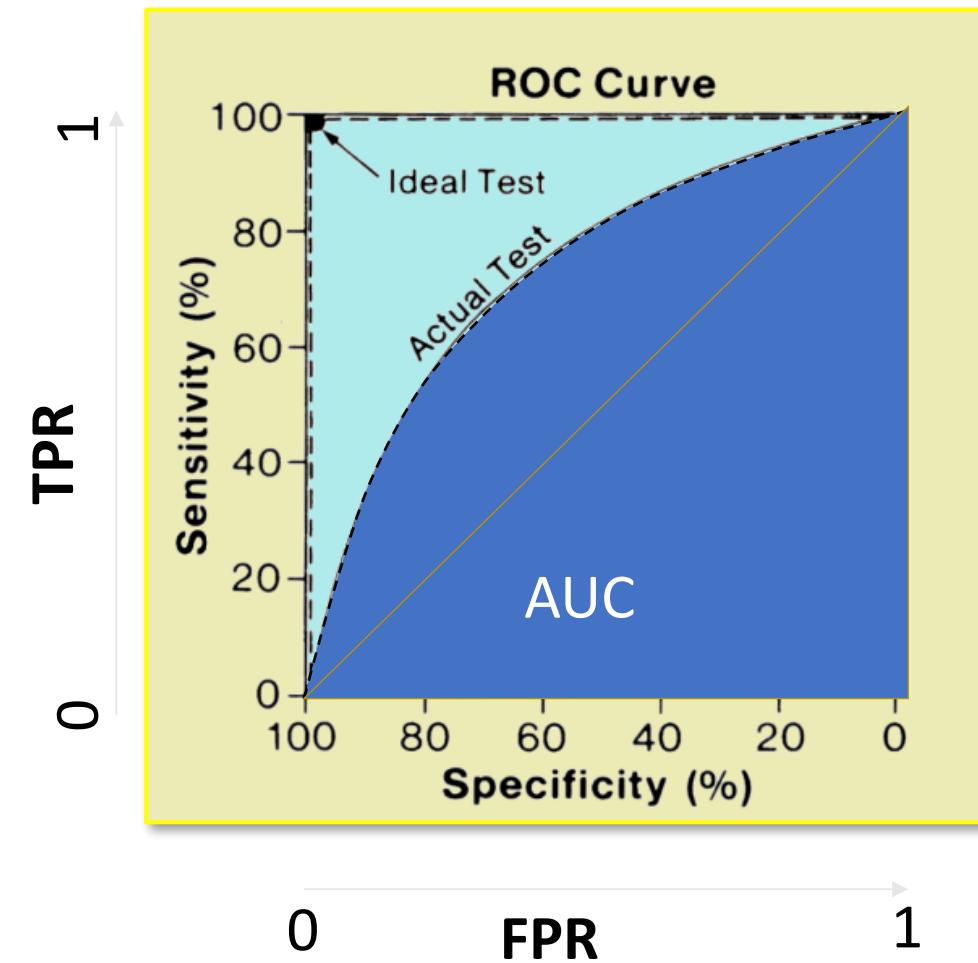
For each value of the decisional threshold (operative point) on the classifier output (generally in the  $[0,1]$  range), sensitivity and specificity values can be computed to obtain:

- the Receiver Operating Characteristic (ROC) curve

The classifier performance can thus be expressed in terms of a single number:

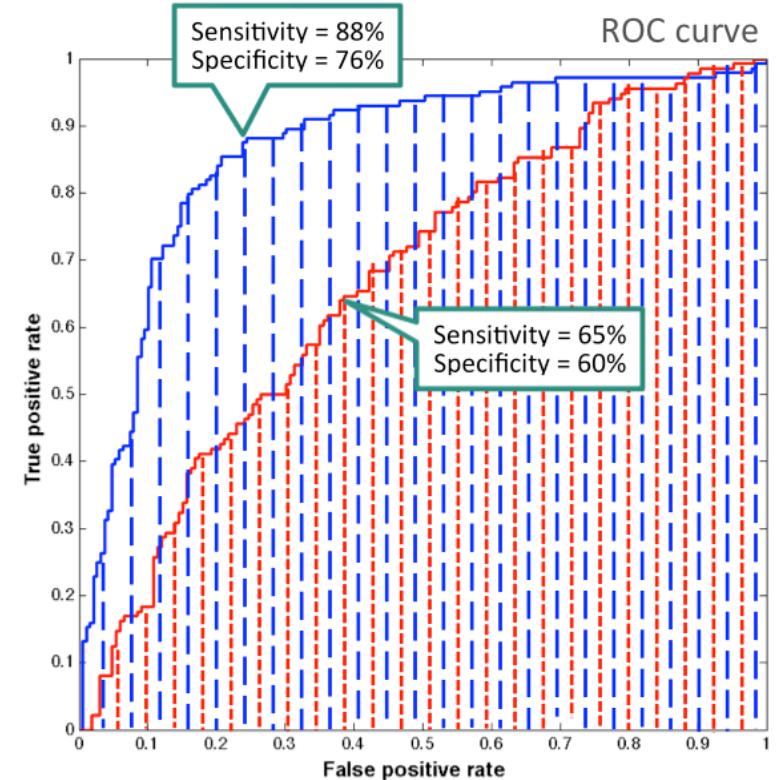
### Area Under the ROC Curve (AUC)

It is very useful to compare classifiers working at different operative points.



# Area under the ROC curve (AUC)

- To compare classifiers we may want to reduce the ROC performance to a single scalar value representing expected performance  
→ Calculate the AUC
- Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1
- However, because random guessing produces the diagonal line between  $(0, 0)$  and  $(1, 1)$ , which has an area of 0.5, no realistic classifier should have an AUC less than 0.5
- An ideal classifier has an area of 1
- **Important statistical property:** AUC is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance



Comparing two ROC curves:

- The graph represents the areas under two ROC curves, A and B. Classifier B has greater area and therefore better average performance

# Result reporting

---

Mind out:

- the test set should representative of the whole sample (e.g. with respect to demographic characteristics)

For large cohorts, you should compute:

- Sensitivity & specificity / accuracy / AUC on the test set (independent validation)

For small cohorts, you should compute:

- Average sensitivity & specificity / accuracy / AUC and standard deviation over N repetitions of the k-fold cross validation on the test set

# Dimensionality reduction

---

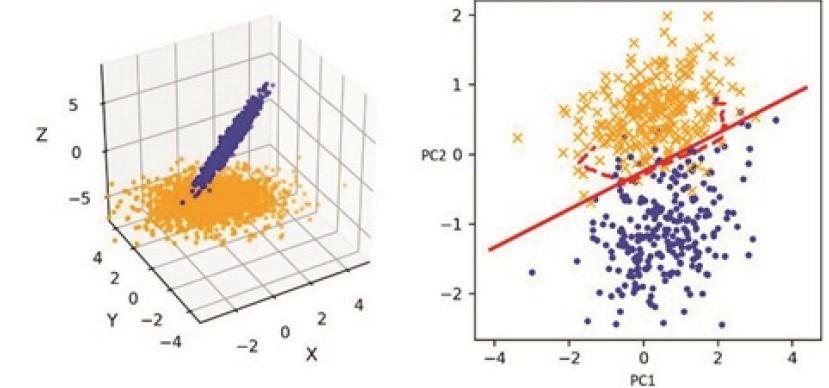
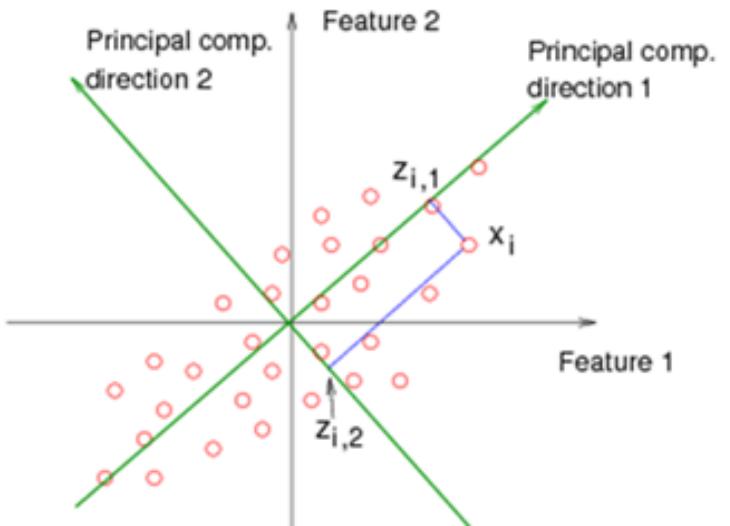
- The image feature extraction procedures may generate a high amount of features.
- As the number of available subjects to analyse in medical imaging studies is very often limited to ~100 or ~1000 at most, it is important to identify the most informative input variables (feature selection).
- What is the best number of features to consider?
  - A rule of the thumb say that almost 10 examples should be available for each model weight to be trained.
- Dimensionality reduction procedures aim to reduce the number of variables to a set of principal variables.
  - **Feature selection** (a set of reduced number of features is retained)
  - **Feature transformation** (a set of new features is generated by the combination of the existing features)

# Principal Component Analysis (PCA)

- PCA is an unsupervised technique which converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.
- It uses an orthogonal transformation and finds a sequence of linear combinations of the variables that have **maximal variance** and are **mutually uncorrelated**.
- The principal components of a set of features  $X_1, X_2, \dots, X_p$  are the normalized linear combinations of the features. The first principal component is:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \dots + \phi_{p1}X_p \quad \sum_{j=1}^p \phi_{j1}^2 = 1$$

- PCA is very effective in dimensionality reduction when variables are highly correlated
- PCA also serves as a tool for data visualization, as it finds a lower-dimensional representation of a dataset.
- The information about the relevance of the original imaging feature can be retrieved

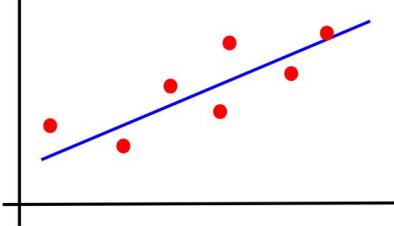


# Three canonical learning problems

---

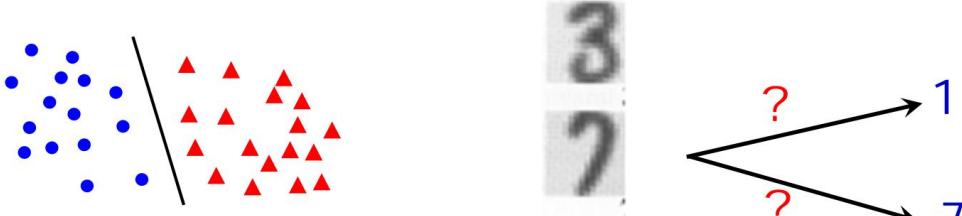
## 1. Regression - supervised

- estimate parameters, e.g. of weight vs height



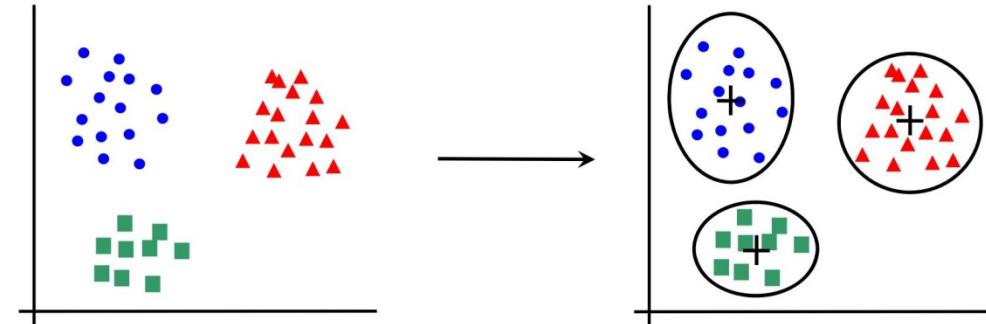
## 2. Classification - supervised

- estimate class, e.g. handwritten digit classification

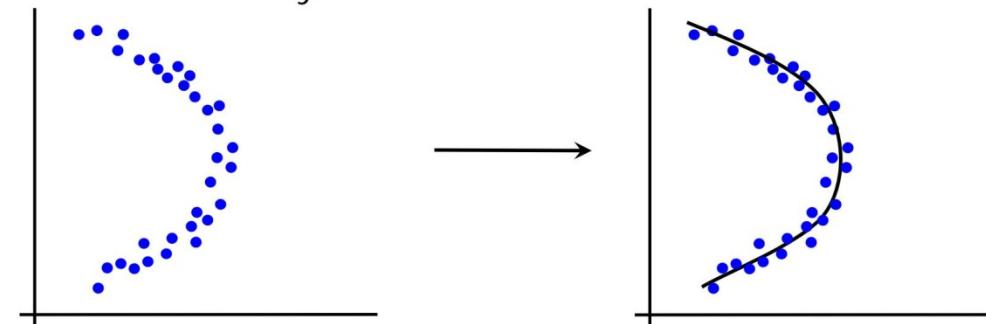


## 3. Unsupervised learning – model the data

- clustering



- dimensionality reduction



# Labeled data for supervised classification/regression

In supervised learning an algorithm is employed to learn the mapping function

$$y_i = f(x_i)$$

- In the machine learning framework, each image  $x_i \in \mathbb{R}^n$  is considered as a point in a  $n$ -dimensional space ( $n$  is the number of voxels/features in the image).

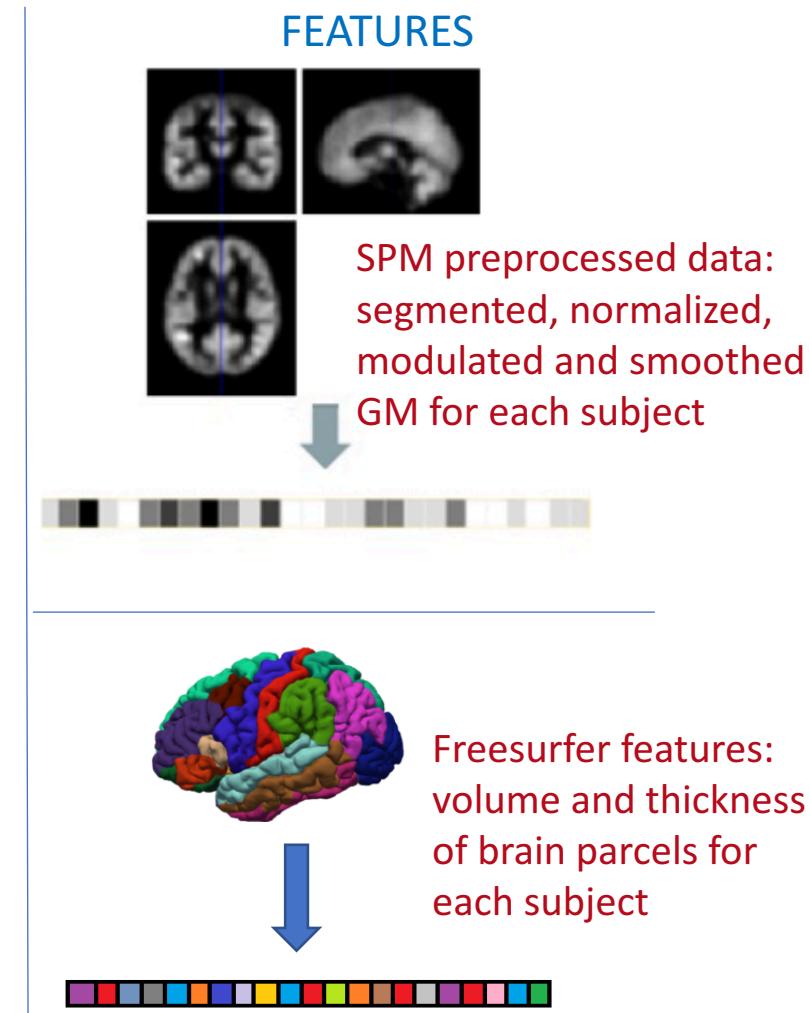
## Binary classification

- In a two-class classification (e.g. patients vs. controls) the  $i$ -th image can be labelled with  $y_i$ :

$$y_i \in \{-1, 1\} \quad \text{where } i = 1, \dots, I.$$

## Regression

- In a regression to estimate continuous parameters (e.g. age of subjects) the  $i$ -th image can be labeled with  $y_i \in \{\text{age}_{\min}, \text{age}_{\max}\}$



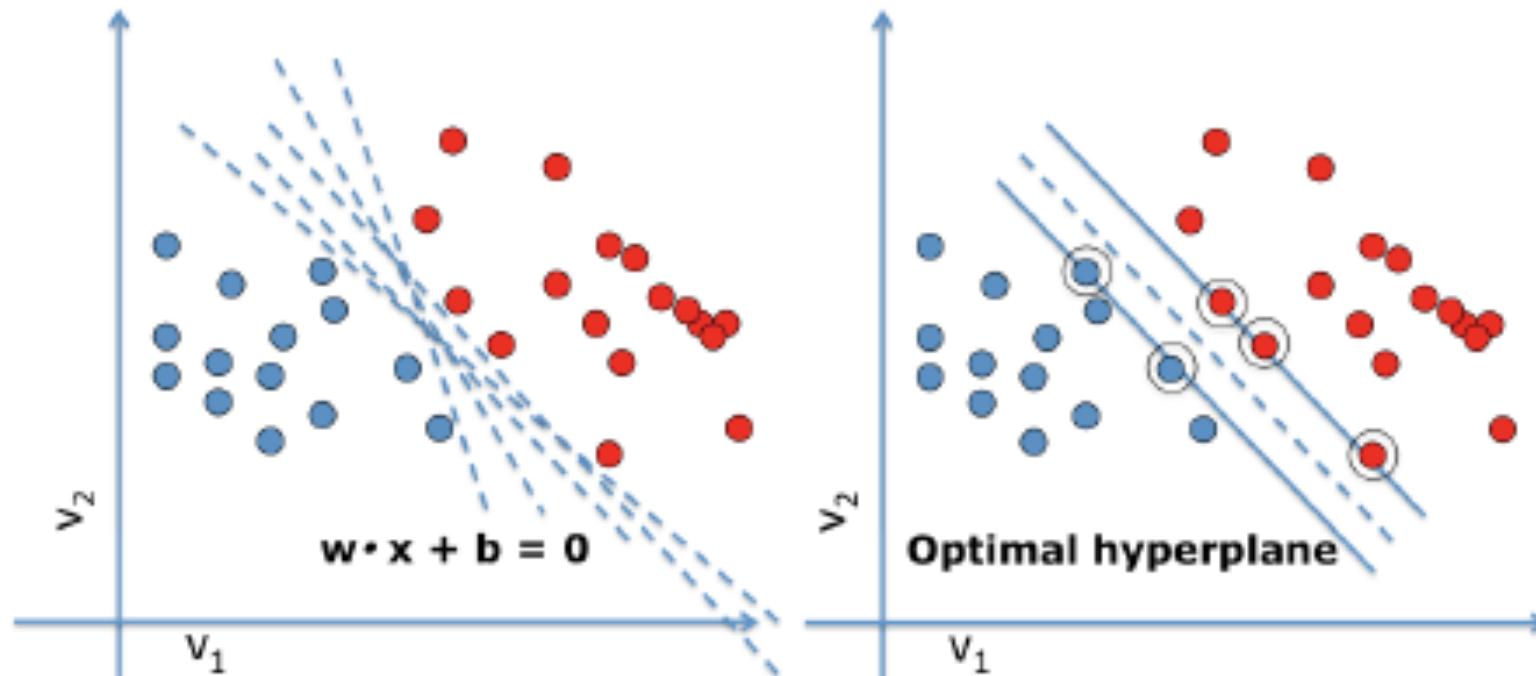
<http://freesurfer.net>

# Support Vector Machines (SVM)

---

[Vapnik VN, The Nature of Statistical Learning Theory. New York: Springer (1995)]

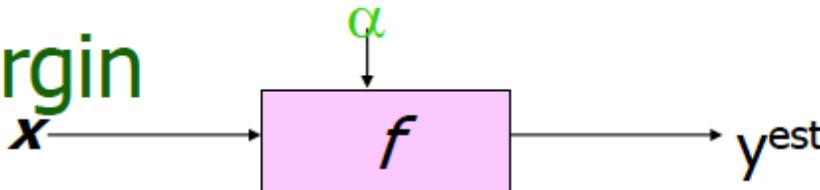
- The SVM linear classification method aims to estimate the separating hyperplane between positive and negative examples characterized by the largest margin.



See the Andrew Moore's SVM tutorial <http://www.cs.cmu.edu/~awm/tutorials>

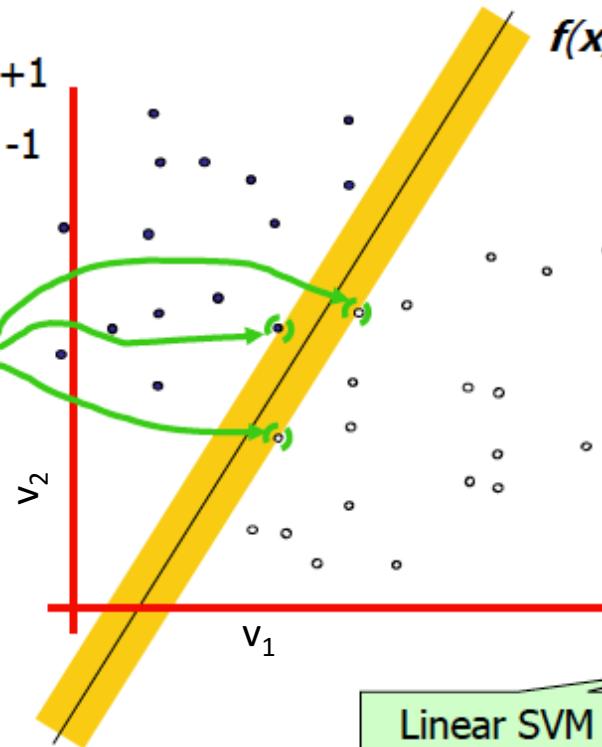
# Support Vector Machines (SVM)

## Maximum Margin



- denotes +1
- denotes -1

Support Vectors are those datapoints that the margin pushes up against



Linear SVM

$$f(x, w, b) = \text{sign}(w \cdot x - b)$$

The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

This is the simplest kind of SVM (Called an LSVM)

Linear kernel

$$K(x, y) = x^T y, \quad x, y \in \mathbb{R}^d$$

Gaussian kernel (RBF Kernel):

$$K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \quad x, y \in \mathbb{R}^d, \sigma > 0.$$

[Vapnik VN, The Nature of Statistical Learning Theory. New York: Springer (1995)]

Slides from Andrew Moore's SVM tutorial <http://www.cs.cmu.edu/~awm/tutorials>

# Discrimination maps

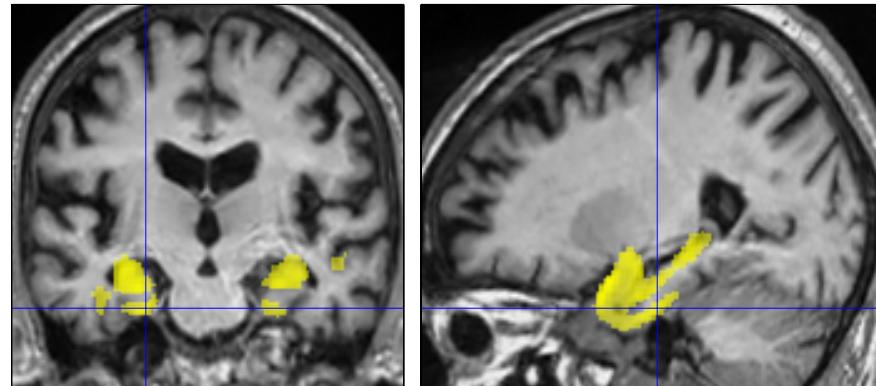
Linear-kernel SVMs allow direct extraction of the weight vector as an image.

- During the SVM training the separating hyperplane is identified so that

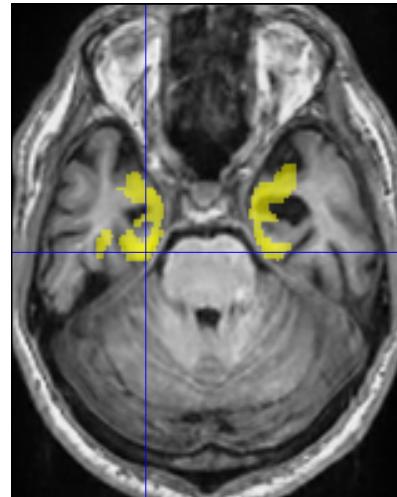
$$\underline{w} \cdot \underline{x} + b = 0$$

where  $\underline{x}$  is a data pattern,  $\underline{w}$  is the weight vector and  $b$  is an offset

- $\underline{w}$  can be used to generate a map of the most discriminating voxels/regions



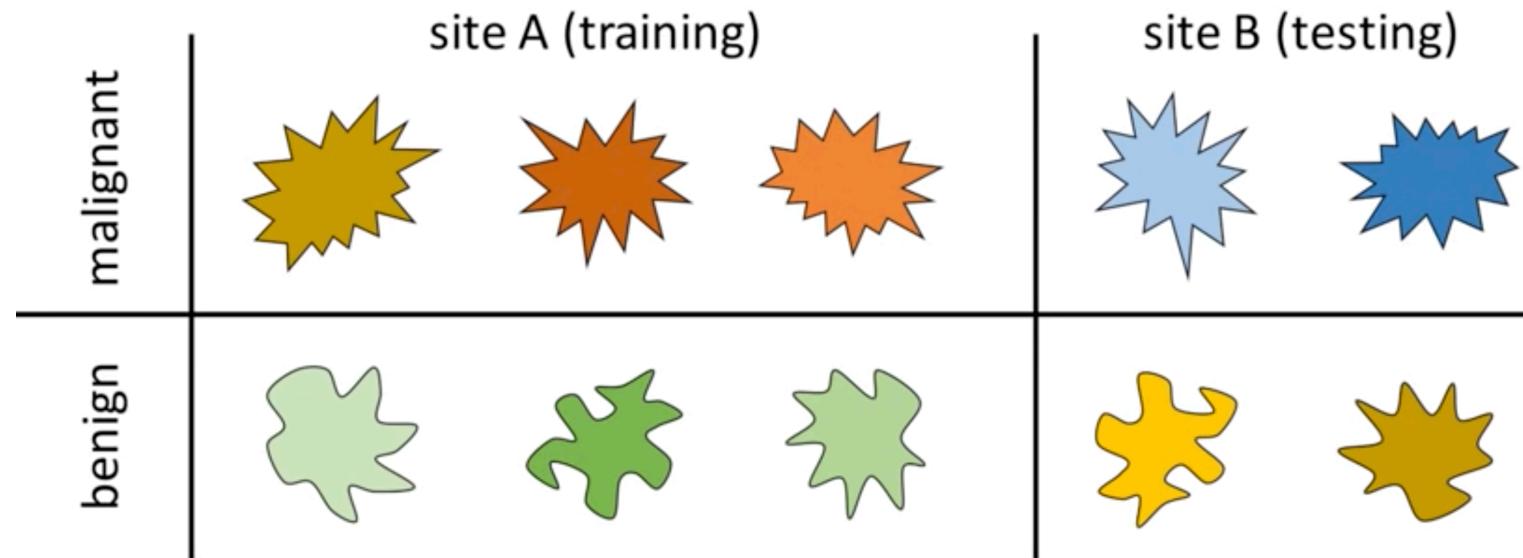
$$|\underline{w}_i|$$



Discrimination map obtained in the classification of subjects with Alzheimer's Disease vs. Control  
The map indicates the regions of local gray matter atrophy in subjects with Alzheimer's Disease

# Learning the right information: confounders

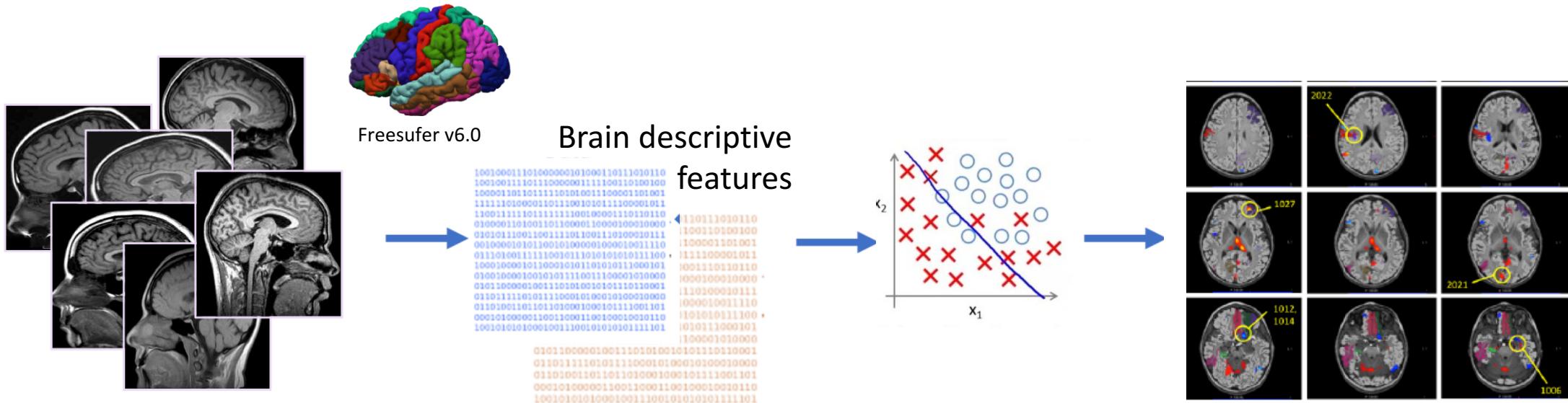
## learning the right features



- If we obtain good discrimination performance between malignant and benign masses are we sure the classifier is exploiting the right mass properties?
- A classifier trained on data from site A which learnt to distinguish masses according to color tones, will not work on data from site B.

**All possible confounder variables should be accounted for in the analysis.  
Classification results should be cross checked.**

# Analysis of MRI data: confounding parameters



- **Confounding variables** introduce bias in the classifier training phase, suggesting correlations that in fact are not there.
- Biases introduced by the **MRI acquisition site** strongly affect the classification results

# Multicenter MRI datasets: the ABIDE sample

Data gathered by different sites and/or acquisition systems carries local “fingerprint”, which can hide subtle information of interest.

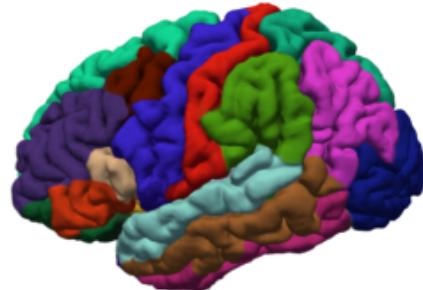
This problem is similar to the management of **systematic errors**



Autism Brain Imaging  
Data Exchange



# Site dependence of sMRI data



<http://freesurfer.net>



volume and thickness of 62  
brain parcels for each subject

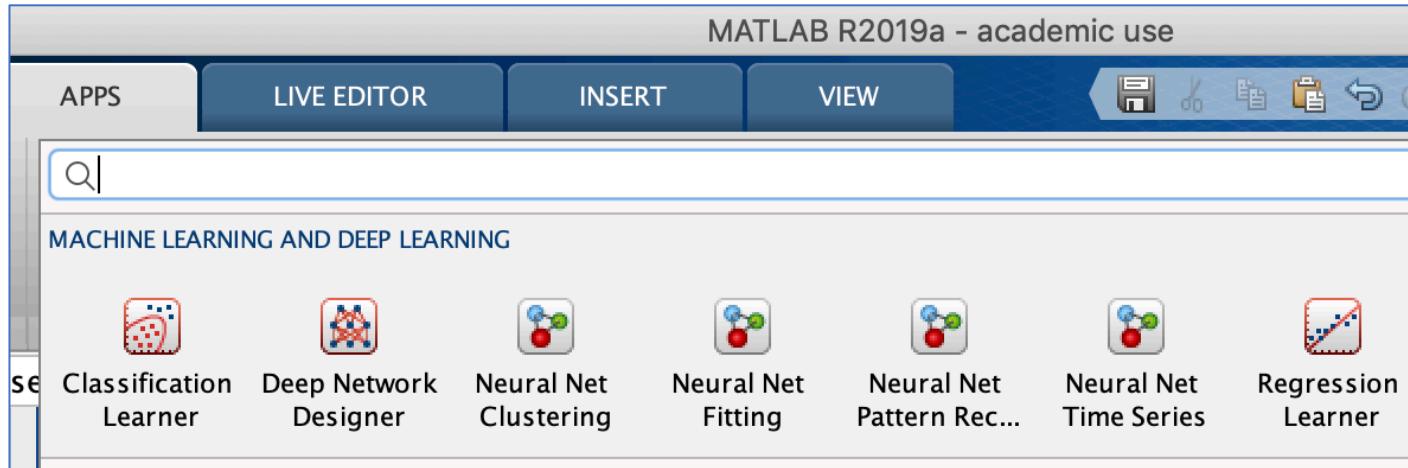
- ABIDE healthy subjects
- Site<sub>i</sub> vs. Site<sub>j</sub> binary classification

E. Ferrari et al., submitted to AImM

Areas Under the ROC Curve (AUC) obtained in two-class classification

|                 | NYU<br>ABIDE1 | NYU-1<br>ABIDE2  | NYU-2<br>ABIDE2 | OHSU<br>ABIDE1 | OHSU<br>ABIDE2 | USM<br>ABIDE1 | USM<br>ABIDE2 | UM-1<br>ABIDE1 | UM-2<br>ABIDE1 |
|-----------------|---------------|--|-----------------|----------------|----------------|---------------|---------------|----------------|----------------|
| NYU<br>ABIDE1   | -             | <b>0.78</b>  | <b>0.89</b>     | <b>0.99</b>    | <b>1.00</b>    | <b>0.99</b>   | <b>1.00</b>   | <b>0.99</b>    | <b>0.98</b>    |
| NYU-1<br>ABIDE2 |               | -  | <b>0.70</b>     | <b>0.99</b>    | <b>1.00</b>    | <b>1.00</b>   | <b>1.00</b>   | <b>0.99</b>    | <b>0.98</b>    |
| NYU-2<br>ABIDE2 |               |  | -               | <b>1.00</b>    | <b>0.98</b>    | <b>0.99</b>   | <b>0.99</b>   | <b>1.00</b>    | <b>1.00</b>    |
| OHSU<br>ABIDE1  |               |  |                 | -              | <b>0.63</b>    | <b>0.97</b>   | <b>0.96</b>   | <b>1.00</b>    | <b>1.00</b>    |
| OHSU<br>ABIDE2  |               |  |                 |                | -              | <b>0.99</b>   | <b>0.96</b>   | <b>0.98</b>    | <b>0.98</b>    |
| USM<br>ABIDE1   |               | <i>How can we<br/>eliminate/mitigate the<br/>bias due to data<br/>acquisition information?</i> |                 |                |                | -             | <b>0.75</b>   | <b>0.99</b>    | <b>0.99</b>    |
| USM<br>ABIDE2   |               |  |                 |                |                |               | -             | <b>0.97</b>    | <b>0.97</b>    |
| UM-1<br>ABIDE1  |               |  |                 |                |                |               |               | -              | <b>0.96</b>    |
| UM-2<br>ABIDE1  |               |  |                 |                |                |               |               |                | -              |

# Tools to solve classification problems



A screenshot of the scikit-learn website. The header features the scikit-learn logo, navigation links for Install, User Guide, API, Examples, and More, and a search bar with a "Go" button. The main title is "scikit-learn" with the subtitle "Machine Learning in Python". Below the title are three buttons: Getting Started, What's New in 0.22, and GitHub. To the right, there's a bulleted list of features: "Simple and efficient tools for predictive data analysis", "Accessible to everybody, and reusable in various contexts", "Built on NumPy, SciPy, and matplotlib", and "Open source, commercially usable - BSD license". At the bottom, there are three large buttons labeled "Classification", "Regression", and "Clustering".

# References and sources

---

- <http://www.cs.cmu.edu/~awm/tutorials>
- <https://www.datasciencecentral.com/profiles/blogs/roc-curve-explained-in-one-picture>
- [https://it.mathworks.com/help/stats/classification-learner-app.html?searchHighlight=classification%20learner&s\\_tid=doc\\_srcTitle](https://it.mathworks.com/help/stats/classification-learner-app.html?searchHighlight=classification%20learner&s_tid=doc_srcTitle)
- <https://scikit-learn.org/stable/>
- [https://scikit-learn.org/stable/auto\\_examples/classification/plot\\_classifier\\_comparison.html](https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html)

[https://github.com/retico/cmepda\\_medphys/tree/master/L9\\_code](https://github.com/retico/cmepda_medphys/tree/master/L9_code)

- see Lecture9\_demo\_classification mlx (use the function SVMtrainCV.m), which uses the data samples (brain features from ABIDE and features of malignant/benign mammographic masses) in [DATASETS/FEATURES/](#)
- See Lecture9\_plot\_classifier\_comparison.ipynb