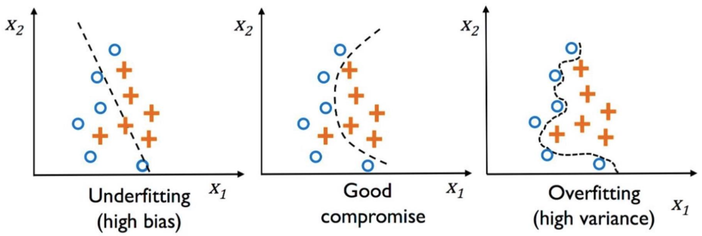
Overview and final words

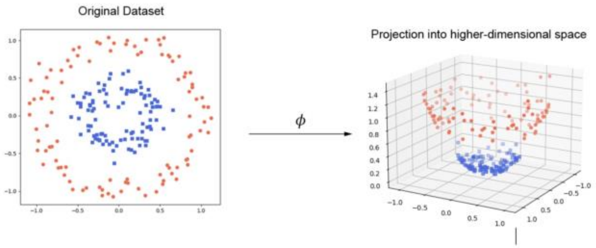
DAT200

Overview

Overfitting



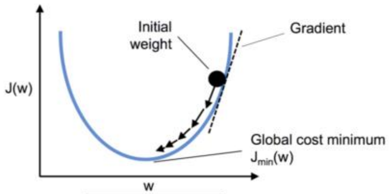
Kernels / feature mapping



ϕ feature mapping, here:

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

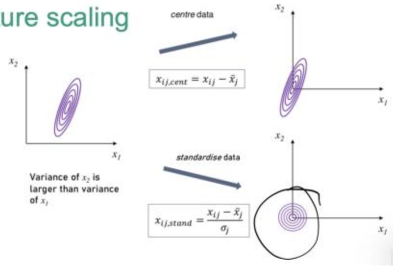
Basic Learning Algos



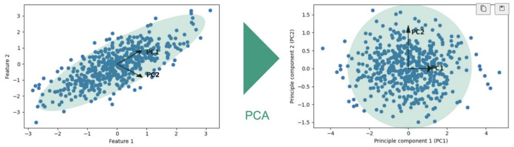
$$\Delta \boldsymbol{w} = -\eta \nabla J(\boldsymbol{w})$$

Gradient descent

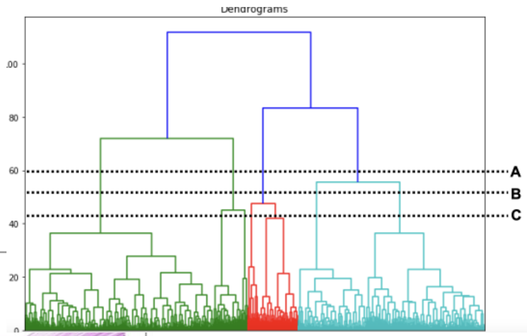
Feature scaling


centre data
$x_{ij,cent} = x_{ij} - \bar{x}_j$
standardise data
$x_{ij,stand} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$
x_2
x_1
Variance of x_2 is larger than variance of x_1

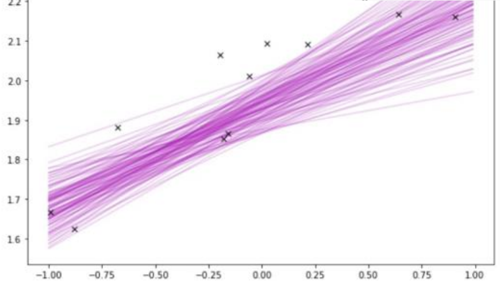
Feature extraction / selection

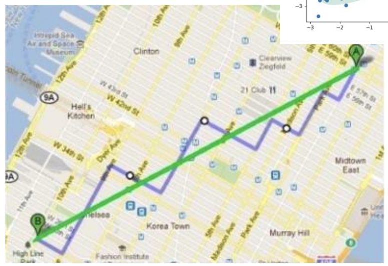


Clustering

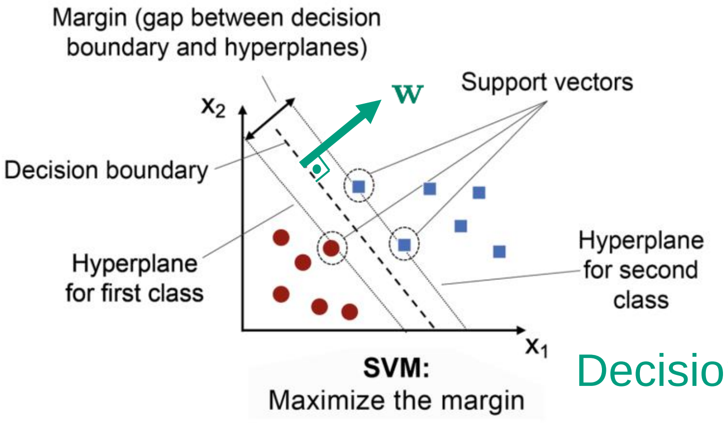


Regression



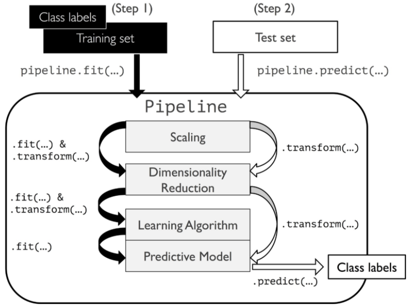


K-NN

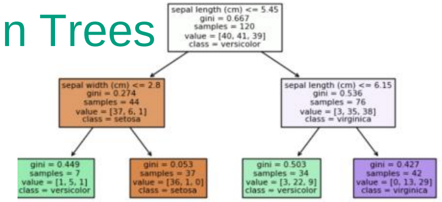

Margin (gap between decision boundary and hyperplanes)
x_2
w
Support vectors
Decision boundary
Hyperplane for first class
Hyperplane for second class
x_1
SVM:
Maximize the margin

SVM

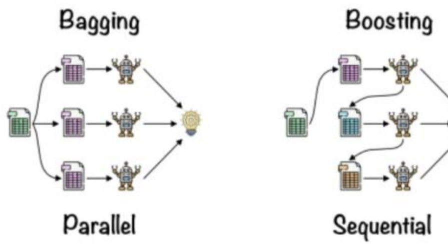
Pipelines / Model evaluation



Decision Trees



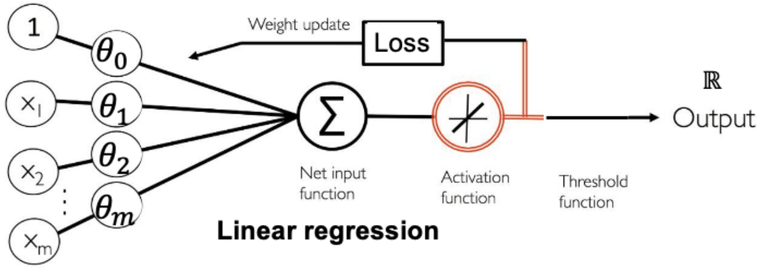
Ensembles

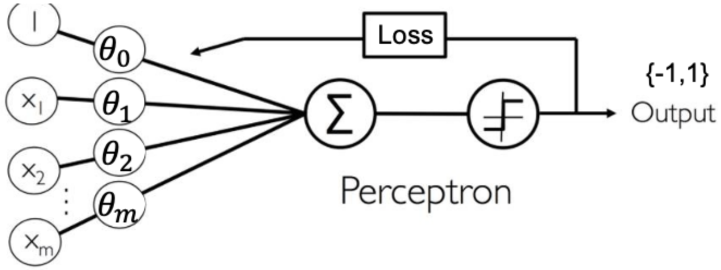


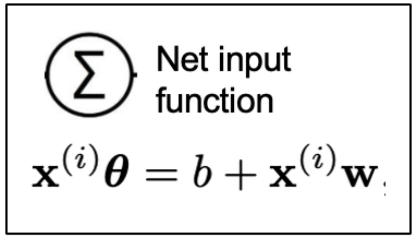
Basics of machine learning, classification/regression

- Simple artificial neurons
  - The Perceptron
  - Adaptive linear neuron (Adaline)
- Simple linear regression, least squares, normal equations

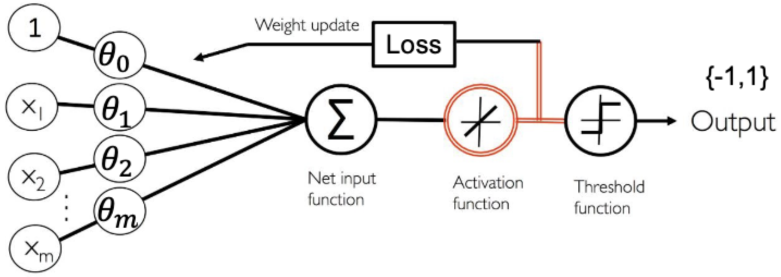
- Net input
- Activation function
- Threshold function
- Cost/Loss function
- Gradient descent
- Newton’s method







$$\mathbf{x}^{(i)}\boldsymbol{\theta} = b + \mathbf{x}^{(i)}\mathbf{w}$$



Adaptive Linear Neuron (Adaline)

Example task 1

Below you see the equation for the perceptron learning rule.

$$\Delta w_j = \eta\left(y^{(i)} - \hat{y}^{(i)}\right)x_j^{(i)}$$

What does w_j represent?

- The learning rate
- [✓ A feature weight]
- The prediction error
- A feature value

Example task 2

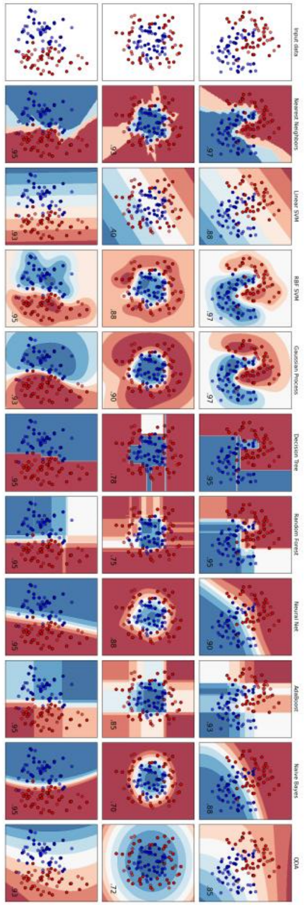
Which activation function does Adaline use?

- The sigmoid function
- [✓ The identity function]
- A threshold function
- The radial basis function

Classification algorithms / classification analysis with sci-kit learn

- Binary to multi-class: **One-vs-all** (one classifier per class)
- **Logistic regression** (outputs probabilities, nonlinear loss function, regularization, …)
- **Support vector machines** (margin hard/soft (slack variables), kernels)
- **Decision trees** (information gain, impurity, random forests, bagging, …)
- **K-NN** (lazy learner, parametric vs. non-parametric models, distances)

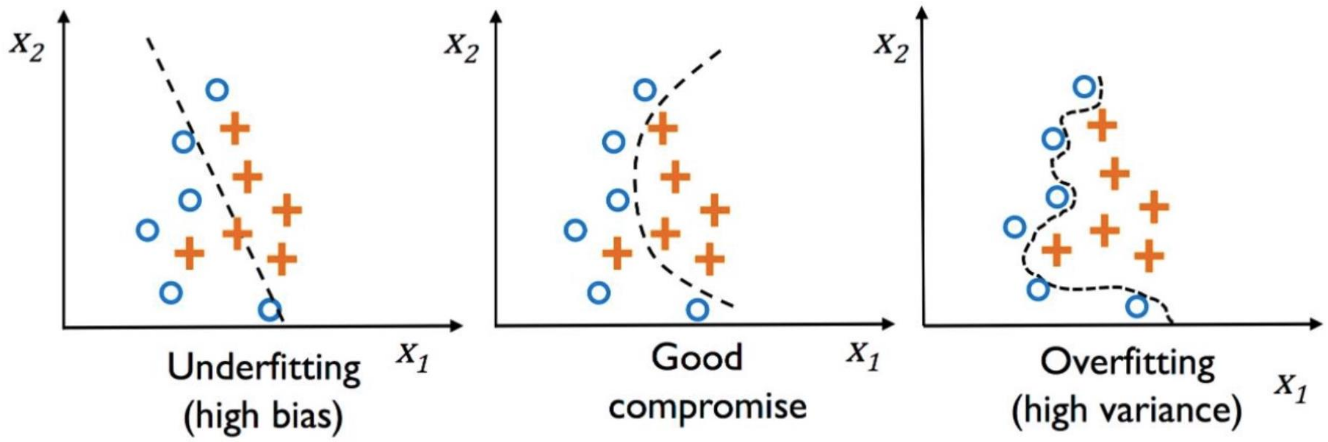
- Associated main **hyperparameters** and their **effects**
- **Algorithmic understanding** in the detail level provided in the lectures (e.g. know the steps of PCA, majority voting, bagging/boosting, …)



Overfitting, Bias/variance trade-off

Techniques to **reduce overfitting:**

- **Reduce model complexity**
  - Reduce the number of parameters
  - Reduce the number of input features
- **Regularization** (L1 / L2)
- **Dimensionality reduction**



Example task 3

A model that is overfitted has…

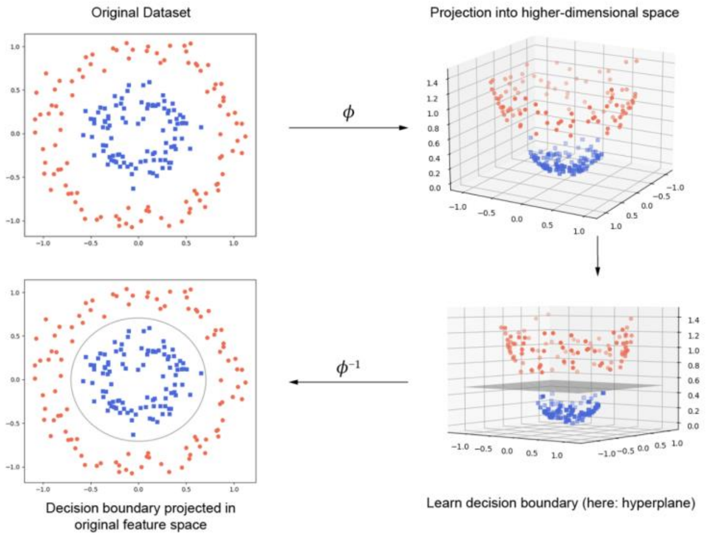
- High bias and low variance
- [✓ High variance and low bias]
- Both high variance and high bias

Kernels, kernel trick, feature maps

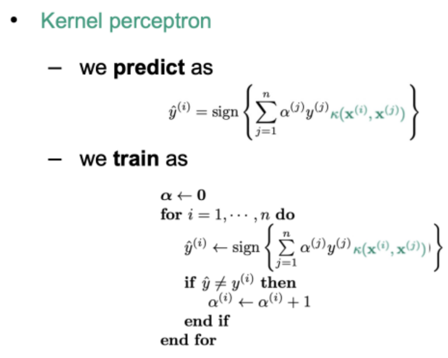
- Nonlinear decision boundaries
- Properties of the kernel function/matrix
- Feature maps
- Kernel trick
- Replace inner/dot products by kernel function calls

$$\mathbf{x}^{(i)}\mathbf{x}^{(j)^T} \longrightarrow \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \phi(\mathbf{x}^{(i)})\phi(\mathbf{x}^{(j)})^T$$

- Kernel SVM, Kernel perceptron, Kernel PCA


Original Dataset
Projection into higher-dimensional space
ϕ
ϕ^{-1}
Learn decision boundary (here: hyperplane)
Decision boundary projected in original feature space

$$\kappa(\mathbf{x}, \mathbf{x}') : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

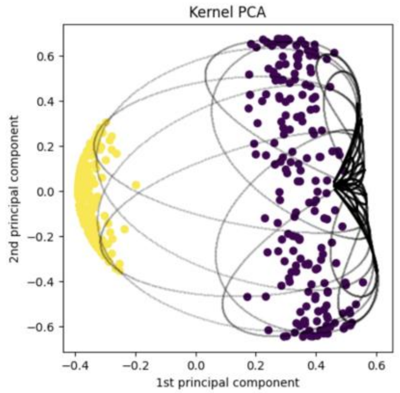


- Kernel perceptron
  - we **predict** as

$$\hat{y}^{(i)} = \text{sign}\left\{\sum_{j=1}^{n} \alpha^{(j)} y^{(j)} \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})\right\}$$

  - we **train** as

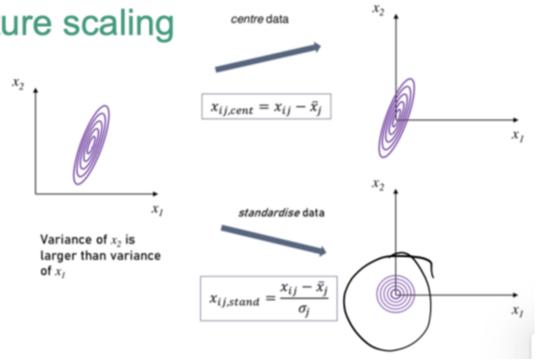
```
$\boldsymbol{\alpha} \leftarrow \mathbf{0}$
for $i = 1, \cdots, n$ do
    $\hat{y}^{(i)} \leftarrow \text{sign}\left\{\sum_{j=1}^{n} \alpha^{(j)} y^{(j)} \kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})\right\}$
    if $\hat{y} \neq y^{(i)}$ then
        $\alpha^{(i)} \leftarrow \alpha^{(i)} + 1$
    end if
end for
```



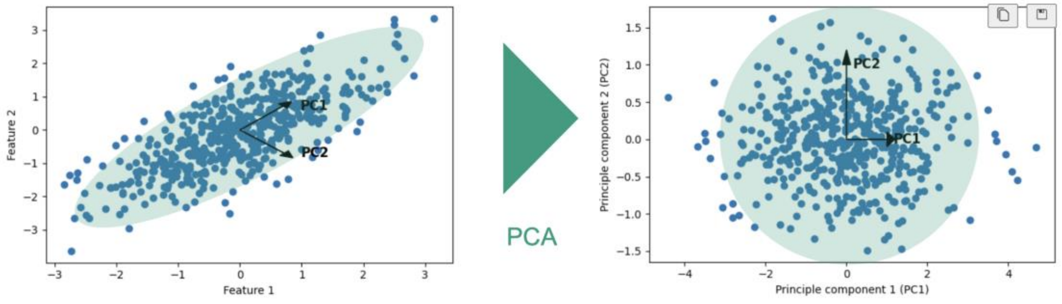
Pre-processing and feature selection/extraction

- Techniques used on various data types (e.g. **encoding**)
- **Removing** or **replacing / imputing** missing values
- **Scaling**
- **Feature selection** (SFS/SBS, feature importances from random forests, regularization (L1/L2))
- **Feature extraction** / Dimension reduction
  - PCA (unsupervised), LDA (supervised)

Feature scaling



x_2
x_1
Variance of x_2 is larger than variance of x_1
centre data
$x_{ij,cent} = x_{ij} - \bar{x}_j$
standardise data
$x_{ij,stand} = \frac{x_{ij} - \bar{x}_j}{\sigma_j}$



PCA

Example task 4

What does the following equation represent?

$$\kappa(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \exp\left(-\frac{||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||_2^2}{2\sigma^2}\right) := \exp\left(-\gamma||\mathbf{x}^{(i)} - \mathbf{x}^{(j)}||_2^2\right)$$

- The Linear kernel
- The polynomial kernel
- The sigmoid kernel
- [✓ The radial basis function kernel]

Feature extraction: Principal Component Analysis (PCA)

Summary of steps (centering but not standardization is included in `PCA` from `sklearn`)

1. **Center** (covariance matrix) or **standardize** (correlation matrix) the data. [→ Always at least center (subtract the mean), but don't standardize if the scale carries significant meaning, e.g. several features measuring similar things (e.g. signal intensity at different locations)]
2. Compute the **covariance/correlation matrix ($\boldsymbol{\Sigma} = \frac{\mathbf{1}}{\boldsymbol{n}-\mathbf{1}} \boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$)**
3. Compute the **eigendecomposition** of $\boldsymbol{\Sigma}$ [→ 2./3. may be replaced by one step by performing a singular value decomposition directly on the centered data]
4. Rank the **eigenvectors** according to the values of the corresponding **eigenvalues** of $\boldsymbol{\Sigma}$
5. Keep $k \leq m$ features, and construct a projection matrix $\boldsymbol{W}$ from the "top" k eigenvectors
6. **Transform the** d -dimensional **input data set** with m features to the new k features (**principal components**), using the projection matrix $\boldsymbol{W}$ to obtain the new $\boldsymbol{k}$ **-dimensional feature subspace**

$$\boldsymbol{T} = \boldsymbol{XW}$$

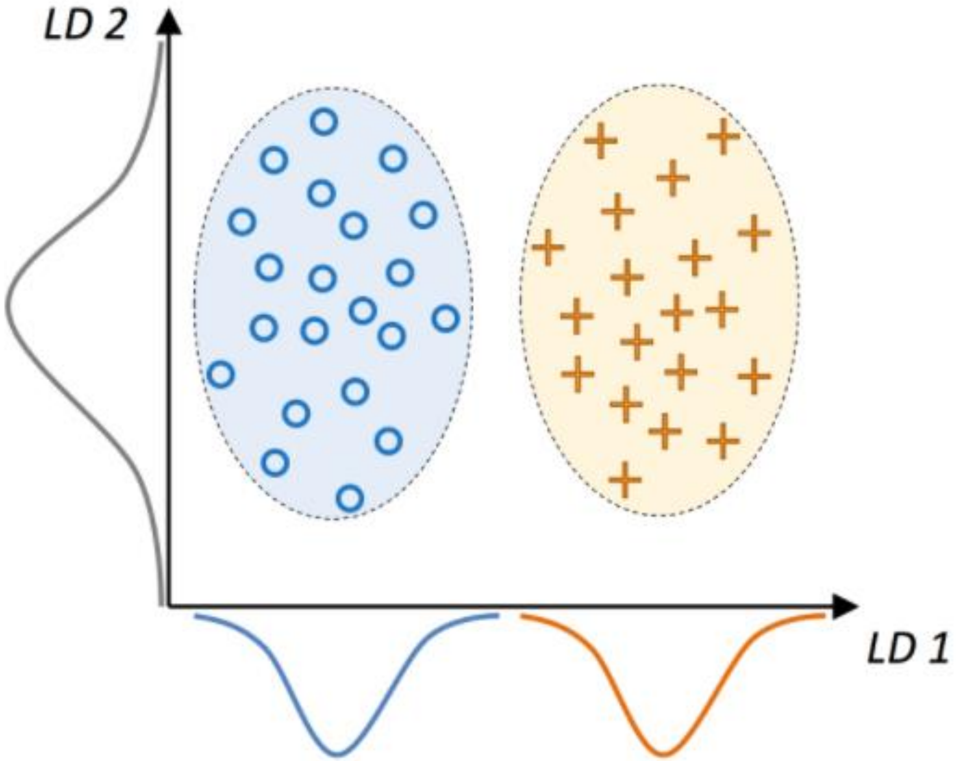
Example task 5

PCA is often used for… (multiple correct answers)

- Dimensionality reduction
- Classification
- Feature extraction
- Data visualisation
- Prediction
- Interpretation of patterns in the data
- Clustering

Feature extraction using Linear Discriminant Analysis

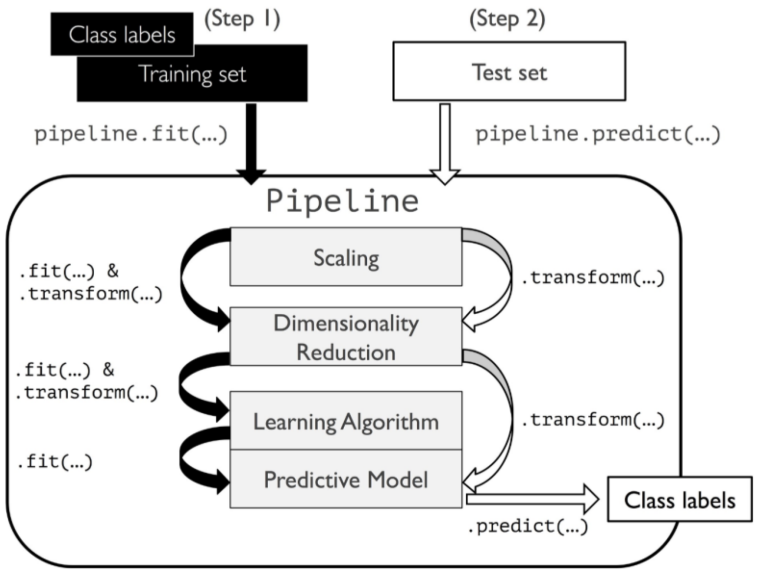
- The concept of LDA
- In the binary classification example to the right a dataset has been projected onto two *linear discriminants*.
- Which of the **LD**’s would you choose to classify the dataset?



Pipelines, cross-validation and model selection

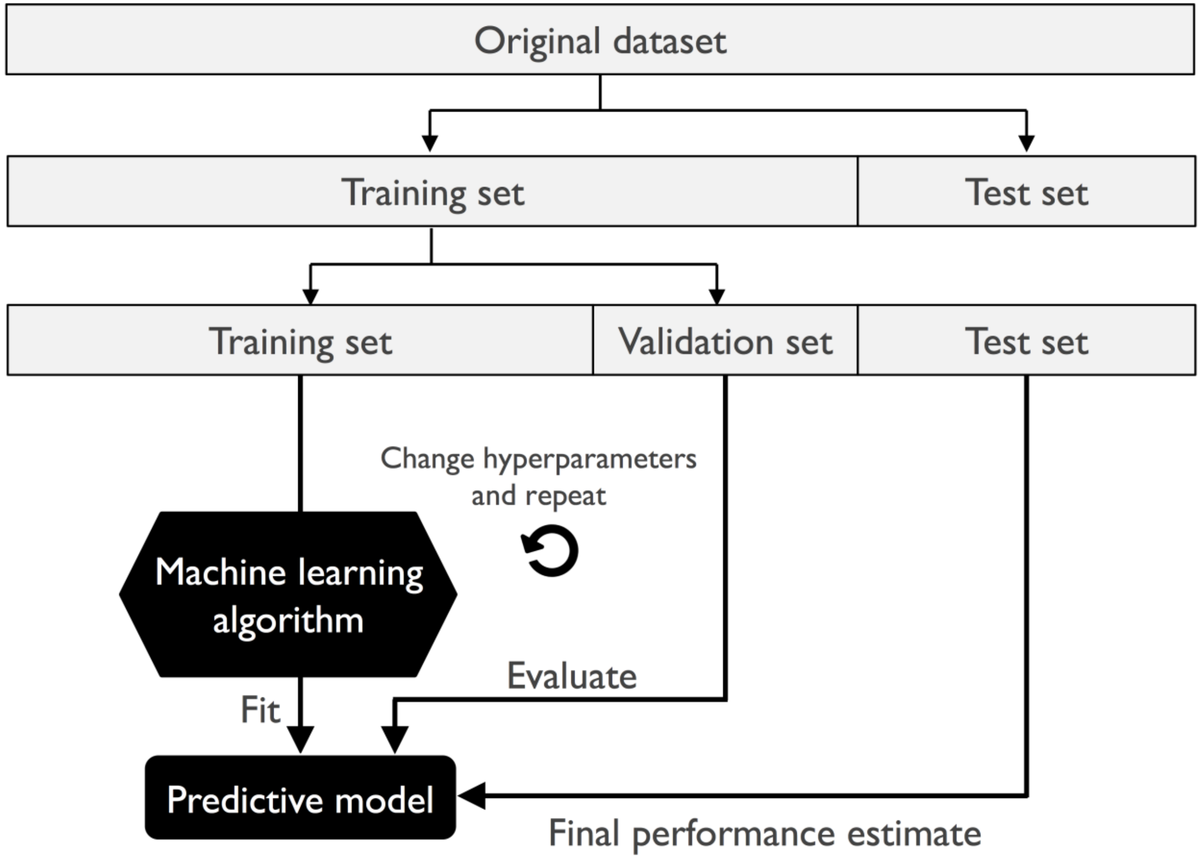
- Pipelines
- Holdout cross-validation
- K-fold cross-validation
- Learning & validation curves
- Grid search
- Randomized search
- Nested cross-validation

- Confusion matrix
- Important metrics (precision, recall, F1)
- ROC-curve and ROC-AUC
- Multi-class classification (micro- vs macro-averaging)



The holdout method (“validation” partition)

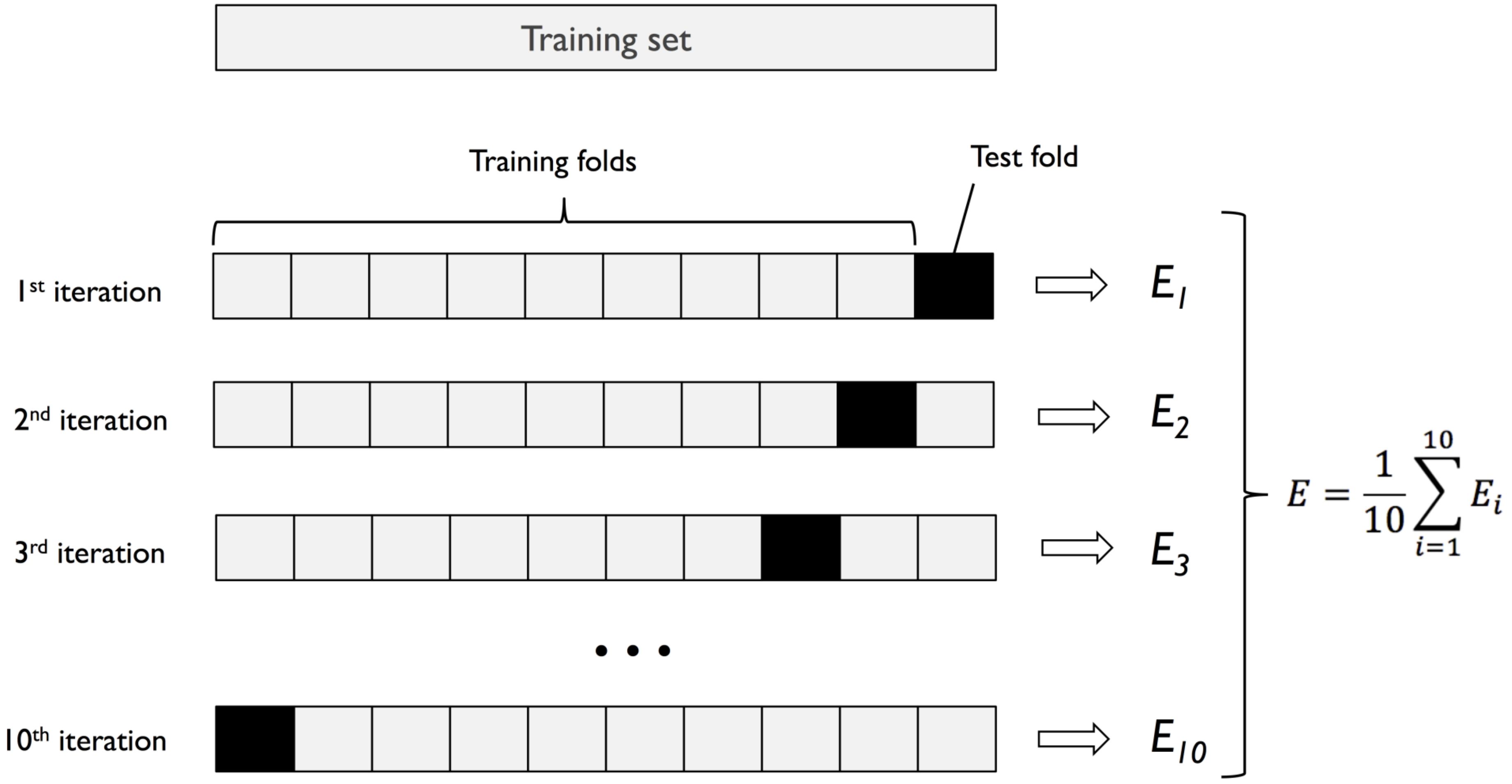
- The method of splitting the full dataset into training partition and an evaluation partition is referred to as the *hold out method*



Stop here for holdout validation

When splitting training set again it’s called **holdout cross-validation**

K-fold cross-validation (CV)



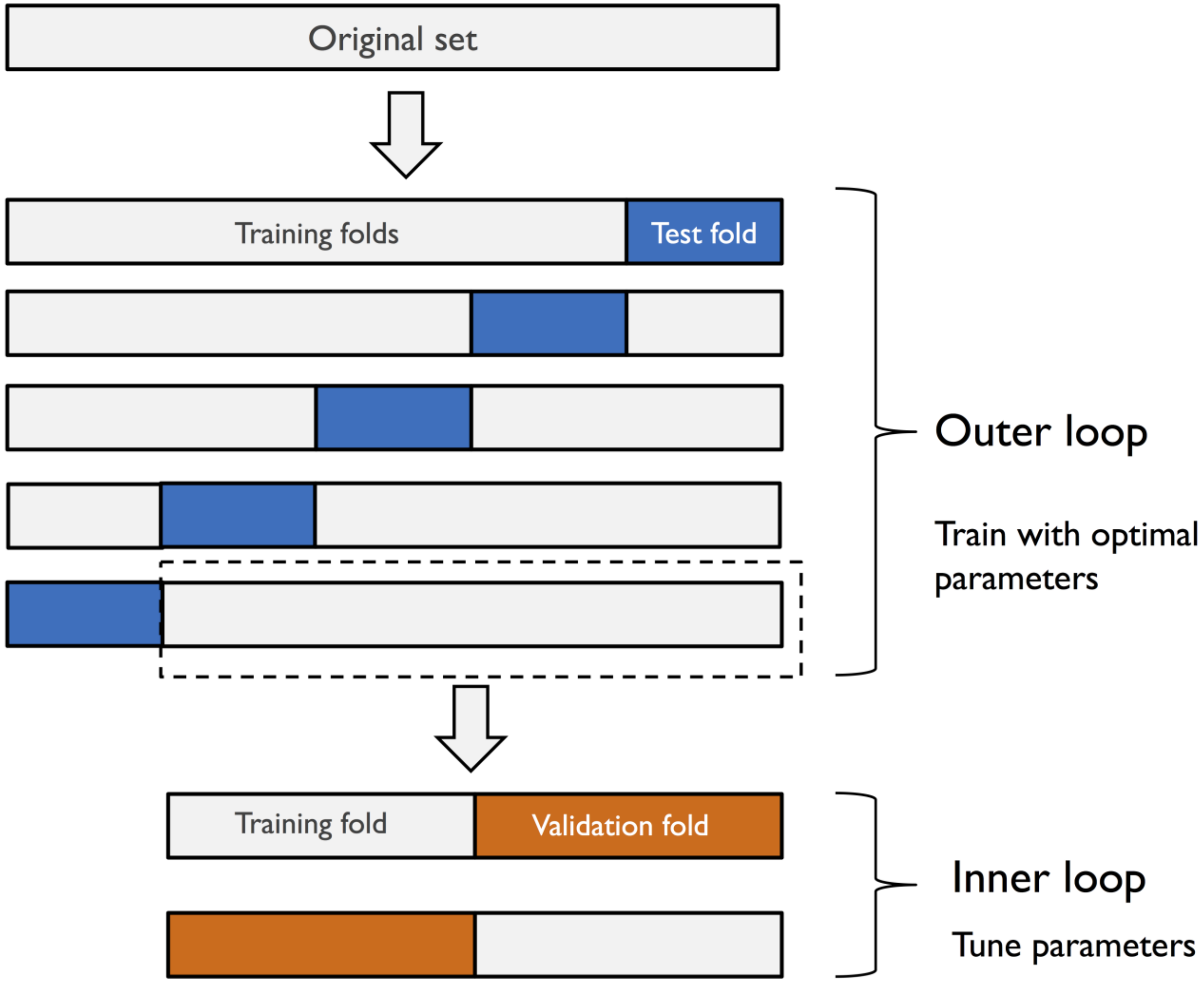
$$E = \frac{1}{10}\sum_{i=1}^{10} E_i$$

Grid search & randomized search

- Grid search:
  - Brute-force exhaustive search through grid of specified set of hyperparameters
- Randomized search:
  - We **don’t** specify a grid of hyperparameter combinations to search exhaustively
  - Instead we specify
    - A range of possible hyperparameter values (could be continous)
    - Parameter specific probability distributions
    - Max set of iterations

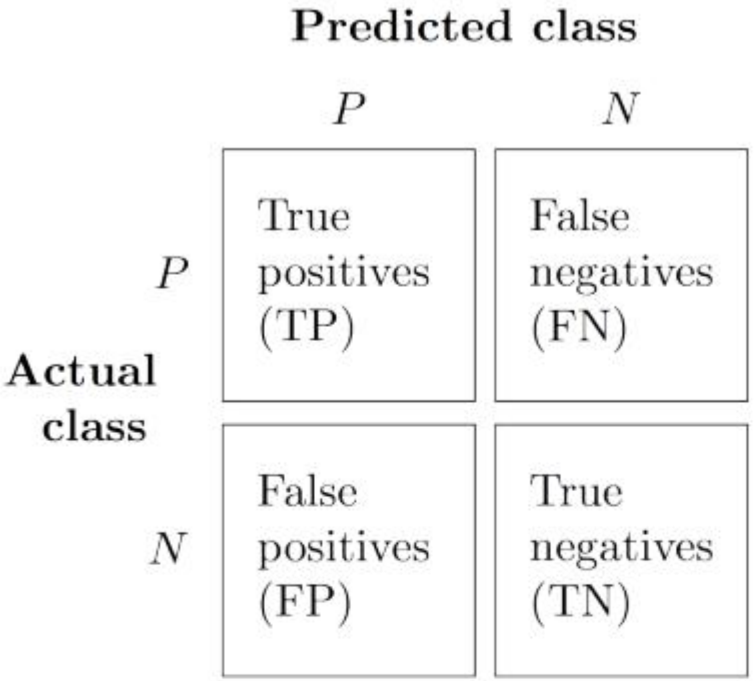
Nested cross-validation

- Cross-validation loop within a cross-validation loop
- Addresses the fact that the initial split between the training/val set and test set is also sensitive to how the split is done
- Becomes very computationally expensive
- Is rarely done when you are working with datasets of over a certain size



Confusion matrix

- It is a square matrix showing the classes that a model predicts versus the classes of the ground truth
- Let 1 be positive and 0 be negative
- **TP**: Positive sample model predicts positive
- **FP**: Negative sample model predicts positive
- **FN**: Positive sample model predicts negative
- **TN**: Negative sample model predicts negative



		Predicted class	
		P	*N*
Actual class	*P*	True positives (TP)	False negatives (FN)
	N	False positives (FP)	True negatives (TN)

Different evaluation metrics

- **Precision** (PRE), **Recall** (REC) and **F1**-score are central metrics in this course
- Precision seeks to measure the amount of TP’s in relation to FP’s
- Recall (also known as *sensitivity* in medicine) is equivilant to TPR

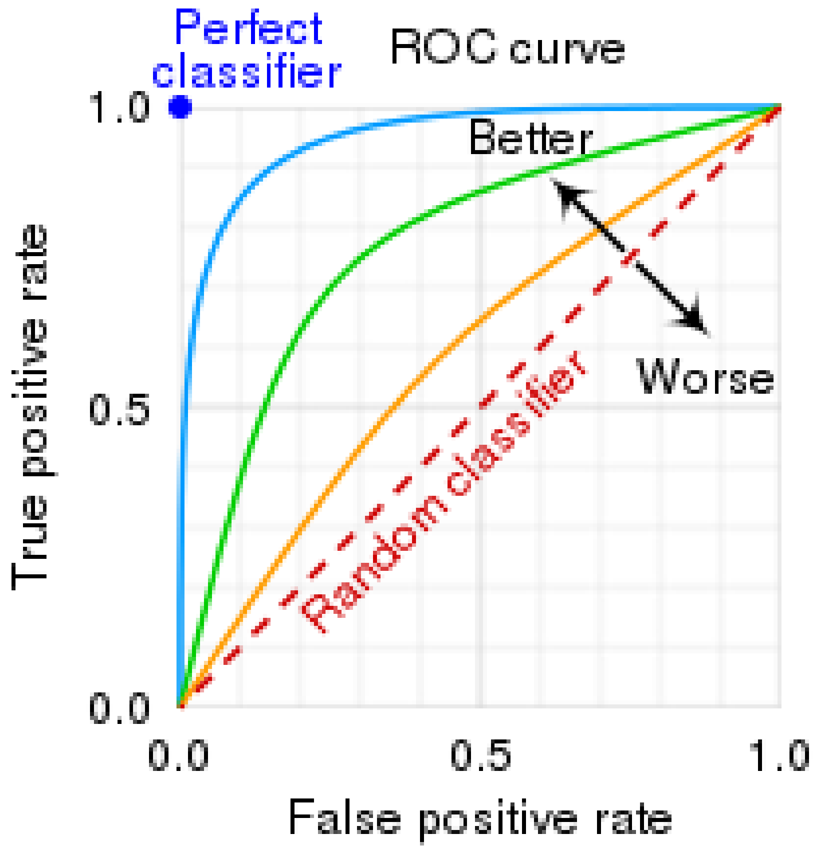
$$\mathbf{Recall} = \frac{TP}{TP + \mathbf{FN}} \quad \mathbf{Precision} = \frac{TP}{TP + \mathbf{FP}}$$

- Often, optimizing for recall might come at the cost of lowering precision
- F1-score is a metric that seeks to combine precision and recall

$$\mathbf{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

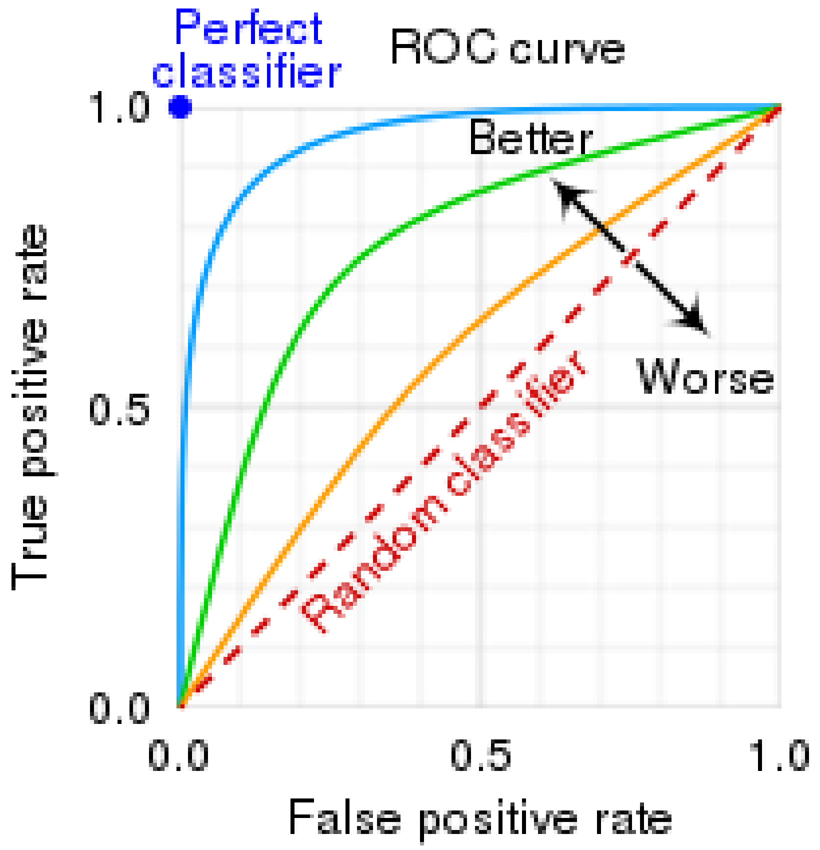
Receiver Operator Curve and Area Under Curve

- The Receiver Operator Curve (ROC) is a graphical representation of a classifier performance.
- Plots TPR versus FPR for a binary classifier at **different decision thresholds**
- Can be understood as trying to visualize how much better a model is than random guessing
- Can be used with any classifier that applies a decision boundary (the majority of classifiers)
- Online illustration



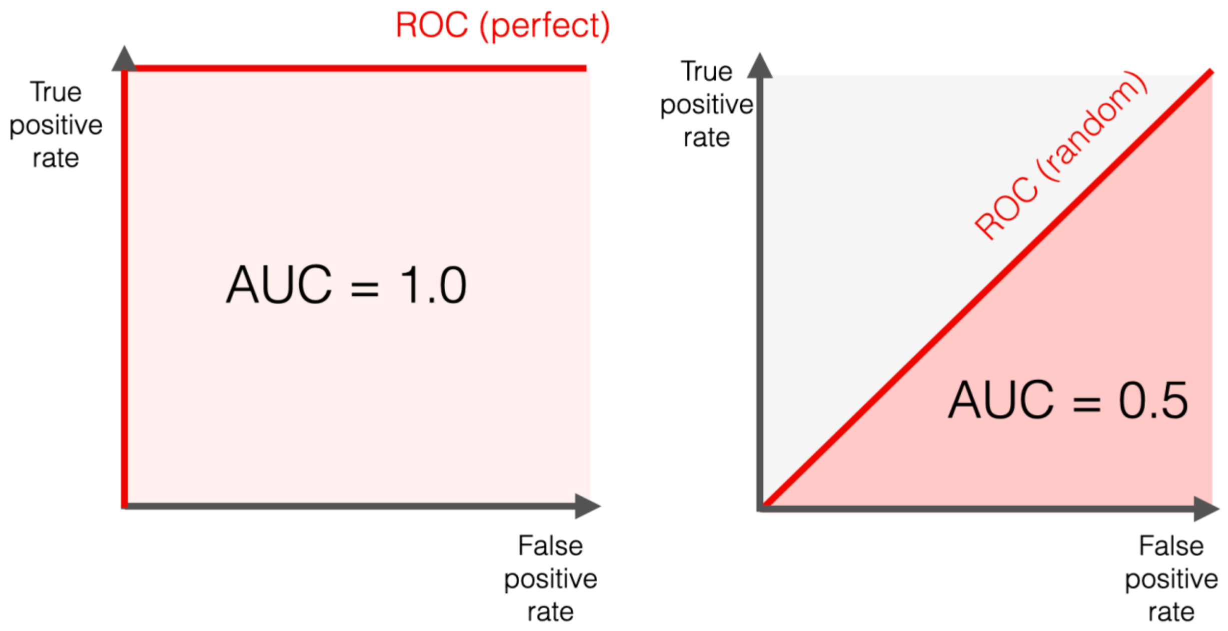
Receiver Operator Curve and Area Under Curve

- Illustrates the trade-off between FPR and TPR
- To compute the ROC classifiers must output a probabilistic output/softmax output which can be thresholded
- Diagonal curve is the performance of random guessing (worst possible score)
- Blue curve is better



Receiver Operator Curve and Area Under Curve

- ROC-AUC: Area Under the Receiver Operating Characteristic Curve
- Quantitative measure of overall classifier performance at all possible thresholds



Ensemble Learning

Majority voting

Combinatorial argument

Weight and posterior probabilities

Bagging **(parallel)**

Repeatedly sample with replacement from the training data.

Train the classifier on each sample and predict new data.

Vote for final prediction.

Random forest (decision trees)

Boosting (sequential)

Very simple individual classifiers (weak learners), usually a decision tree stump (a tree with only one split).

Focus on the samples that are hard to classify

Re-learning based on up-weighting of previous misclassifications.

Ada-Boost

Train m weak learners; compute error rate; compute weight coeff. (inverse error), update sample weights; repeat

Gradient-Boost/XG-Boost

Initial prediction (mean); train m trees on the loss from the previous tree; combine models with learning rate for final prediction

XG: ‘unique’ trees; gain & similarity score

Regression

Linear regression

Analytical solution vs OLS-regression (gradient descent)

Evaluation metrics (MSE; RMSE; R^2)

Simple, Multiple

Residuals vs fitted values

(4) Model assumptions (i. independent errors, ii. normal dist. error, iii. equal variance, iv, linearity

None-linear patterns

Outliers

RANSAC

Outliers; iterate through sub sample, fit model, determent consensus set (inliners/outliners)

Nonlinearity

Transformation; polynomial regression

Regularization

L1 vs L2 vs Elastic

Decision tree regression

Clustering

K-means clustering

Pick k random centroid; cluster based on distance; move centroids to centre; repeat

Groups represented by (Euclidean) distance to single point/object per cluster

Need to set k, simple, tends towards spherical clusters

Suitable for large data sets

K-mean ++

Initiate each centroid using probability weights according to their distance to existing centroids

Soft/fuzzy clustering

Pick k random centroid; cluster based on membership probabilities (m); move centroids to centre; repeat

Also needs k, a bit more complicated, cluster result are similar

Optimal number of clusters - determine k?

Elbow; iterate through a range of k; pick k where distortion (SSE) stops to improve

Silhouette analysis; iterate through range of k; estimate cohesion (intra cluster) and separation (closest cluster) scores; calculate silhouette values, pick k where all clusters above average

Density based clustering

Radius (ε) and minimum number of points

Identify core points; for each core points; assign cluster to density connected points; remaining points are noise

Non-spherical clusters, no need to pick a *k*-value

Hierarchical clustering

Agglomerative; top-down vs Divisive; bottom-up

Distance metric and linkage (merging criterion for clusters)

Single, complete, Wards, Average; Centroid, Median

Can be cut at desired level or number of clusters afterwards

Example task 6

What is the elbow method in clustering?

- A technique to determine the optimal number of clusters by plotting the sum of ________ against the number of ________ and selecting the elbow point.

[✓ Squared Errors]
Absolute Differences
Variances

Classes
Neighbors
[✓ Clusters]

Example task 7

What is the difference between LDA and PCA?

- LDA is a ________ method that considers the ________,

[✓ Supervised]
Unsupervised

Cluster centers
[✓ Class labels]
Feature weights

while PCA is an ________ method that does not.

Supervised
[✓ Unsupervised]

Example task 8

What is an ensemble in machine learning?

- A. A model trained on one subset
- B. Converts unsupervised to supervised
- [✓ C. Combines multiple models for better performance]

Example task 9

$$J_m = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij}^{m} \|x_i - \mu_j\|^2$$

What does the parameter m in the Fuzzy C-Means objective function control?

A. Number of clusters

B. Distance metric

[✓ C. Fuzziness of the clustering]

D. Learning rate

Example task 9

Rearrange the following steps into the correct order for the K-Means algorithm:

A. Repeat until centroids no longer change significantly.

B. Assign each data point to the nearest centroid.

C. Update each centroid based on the mean of its assigned points.

D. Randomly initialize k centroids.

Answer:

D → B → C → A