# 336546 – Machine Learning in Healthcare
# HW3

Ariella Chauvart 337917850

## 1) Clustering

**a.** The K-means algorithm tries to minimize the total squared error. It is based on the mean which is known to be very sensitive to extreme values: noise (or outliers) will influence a lot the values of the centroids.
In contrary, K-medoids minimizes the sum of general pairwise dissimilarities between points of the same cluster: a point labeled to be in the cluster and another datapoint designated as its center (medoid). Therefore, similarly to the median, extreme values won't affect much the values of the medoids.
→ K-medoids is more robust to noise or outliers than the K-means algorithm.

**b.** We know: $\mu = \min \sum_{i=1}^{m}(x_i - \mu)^2$.
In order to minimize $\sum_{i=1}^{m}(x_i - \mu)^2$, we will derive the expression and make it equal to zero:

$$\frac{\partial}{\partial \mu}\left(\sum_{i=1}^{m}(x_i - \mu)^2\right) = 0$$

$$\Rightarrow\ -2 \cdot m \cdot \sum_{i=1}^{m}(x_i - \mu) = 0$$

$$\Rightarrow\ \sum_{i=1}^{m}(x_i - \mu) = 0\ \ (m \neq 0)$$

$$\Rightarrow\ \sum_{i=1}^{m} x_i - m \cdot \mu = 0$$

$$\Rightarrow\ \mu = \frac{\sum_{i=1}^{m} x_i}{m} = mean\ of\ m\ examples$$

## 2) SVM

- First, we can easily identify which figures are representing SVM with a linear kernel: A and D (the separation is a line). The difference between the two is the value of the parameter C, which tells the SVM optimization how much you want to avoid misclassifying. For larger C, the penalty is higher and the optimization will choose smaller-margin hyperplane. And vice versa, a smaller C will cause the optimizer to look for larger margins even if that hyperplane misclassifies more points (lower penalty/higher tolerance).
  Therefore:
  1. Linear kernel with C = 0.01 → D
  2. Linear kernel with C = 1 → A

- Similarly, we can easily identify the separation made with polynomial kernels: C and F. We know that the higher the degree of the kernel, the more flexible the decision boundary and the higher the risk of overfitting: both characteristics match figure F (complex classifier, complex kernel). Moreover, figure C shows a separation quite close to a linear fitting which confirms the low degree (closer to 1) of polynomial kernel.
  Therefore:
  3. $2^{nd}$ order polynomial kernel → C
  4. $10^{th}$ order polynomial kernel → F

- Lastly, figure B and E correspond to an RBF kernel. RBF is a polar kernel and the hyper-parameter Gamma tells the SVM optimization how much influence the feature data points will have on the decision boundary: the higher Gamma is, the higher the influence and thereby the more fitted the boundary, and vice versa.
  Therefore:
  5. RBF kernel with Gamma = 0.2 → E
  6. RBF kernel with Gamma = 1 → B

## 3) Capability of generalization

**a.** The scientific term of the balance in machine learning aspect is *Generalization*. Generalization is the balance between goodness of fit and complexity so that a trained model obtains the best performance. The goodness of fit is the best trade-off between underfitting and overfitting (variance-bias trade-off). And the model complexity is also a determinant factor: when a model is too complex, it is usually prone to overfitting.
The simpler model is often (but not always) easier to understand and maintain and is more robust. In practice, we want to choose the best performing simplest model.

**b.** $AIC = 2p - 2\ln(\hat{L})$

- $2\ln(\hat{L})$: $\hat{L}$ is the maximum likelihood estimation and is a measure of the goodness of fit (i.e. performance) of the model. Likelihood is basically a measure of how likely one is to see their observed data given a model. The model with the maximum likelihood is the one that fits the data the best, i.e. the higher the likelihood, the higher $2\ln(\hat{L})$, the better the model fits the data, the lower AIC.
- $2p$: $p$ is the total number of learned parameters and is representative of the complexity of the model. Indeed, more parameters means a more complex model that is more likely to overfit. The more learned parameters, the higher $2p$, the higher the complexity and the higher AIC.

**c.** As we said before, the risks if the balance were to be disturbed would be overfitting or underfitting.
An excessively low complexity will most likely cause underfitting: it would mean that the model learned a small number of parameters (thus low

complexity) which would inevitably imply low goodness of fit and low performance.
On the opposite, an excessively high complexity would cause overfitting. Indeed, if the model learned from a large number of parameters (thus high complexity), the model would be highly performant on the training dataset (overfitting) but probably wouldn't provide satisfying results on any other dataset.

**d.** As we said in a., we want to choose the *best performing simplest* model.
On the one hand, best performing meaning good fitting means high likelihood and so high $2\ln(\hat{L})$.
On the other hand, simplest model means lowest complexity (but not too low) meaning small number of learned parameters and so low $2p$.
Therefore, we are aiming to minimize AIC.