HW3

Name: Wajdi Nicola
ID: 315479352

1) Clustering

a. Both the k-medoid algorithm and the k-mean algorithm work on the same concept which is to find a set of k cluster representatives and assign other items to them (this by trying to minimize the Euclidian metric between the examples and selected point).
The k-mean algorithm seeks for the clusters in the whole space trying to minimize the Euclidian metric to zero. On the other hand, the k-medoid algorithm seeks for the clusters in the data itself that minimize the Euclidian metric. We see that the k-mean algorithm chooses the clusters based on the mean value of the data (as we will prove in question b below as well) where the k-medoid algorithm chooses the clusters based on the median value of the data (as mentioned in the bonus question below). As we know, the median value is more robust to outliers and noise (the mean is dramatically affected by outliers), thus we would expect the K-medoid to be more robust to noise and outliers.

b. Let us assume that the expression is a function f of μ → $f(\mu) = \sum_{i=1}^{m}(x_i - \mu)^2$. To get the minimum of this function we derive the function (by the parameter μ) and equal to zero:

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

$$m\mu = \sum_{i=1}^{m} x_i$$

$$\mu = \frac{\sum_{i=1}^{m} x_i}{m}$$

We got that the mean of x is a critical point, to check if it is max/min we derive one more time to check the sign:

$$f''(\mu) = 2\,m > 0$$

Thus, the minimum for f is achieved with the value of μ that is the mean of X.

Bonus question:

Let us assume that the expression is a function f of μ → $f(\mu) = \sum_{i=1}^{m}|(x_i - \mu)|$. To get the minimum of this function we derive the function (by the parameter μ) and equal to zero:

$$f'(\mu) = sign\big(f(\mu)\big) = sign\left(\sum_{i=1}^{m}|(x_i - \mu)|\right) = 0$$

The sign(z) function equals to zero when the z=0, thus in this case we will get the critical point in $\sum_{i=1}^{m}|(x_i - \mu)| = 0$. We have here the absolute value function of X byt centered in the value of μ and not 0. Thus, half of the values of x are greater that μ and half of the values are less than μ, so μ have to be the media in order to get the critical point which will be the minimum.

2) SVM:

Here is the matching between the images and the settings with explanation:

- Linear kernels: the linear kernels will be identified in the features' domain by a linearly separable data (will be separated by a linear plain). The images that is suitable for the linearly separable data are A and D. The parameter c implicates the margin violation of the classification. The higher c is the stricter we are about our classification. In the image A we can see that some of the data is on the separating lain (the data in purple), which indicates that we are not that strict about the classification in this case and some of the data is in the margin domain. Thus, we can conclude that the small c describes the classification in the image A, while the big c describes the more violated margin in the image D.
    - **1. Linear kernel with C = 0:01 → A**
    - **2. Linear kernel with C = 1 → D**
- Polynomial kernels: polynomial kernel is seen by a separating curve that is suitable to the polynomic degree in the features' domain (nonlinear if the polynomic degree is greater than 1). The images C and F are suitable to describe polynomial kernels. The higher the plonomic degree the more "complicated and branched" the separating curvature is. Thus, image C is suitable for 2nd order polynomial kernel, and the branched image F is suitable for the 10th order polynomial kernel.
    - **3. 2nd order polynomial kernel → C**
    - **4. 10th order polynomial kernel → F**
- RBF kernels: RBF kernels is identified by transformation of the features' domain to a higher degree domain for the sake of the separating and classification. When transformed back to the original domain the data will be separated by closed domain that is combination of gaussians from the different axis in the higher degree domain as we can see in the images B and E. The bigger the closed domain, the wider the gaussians that separated the data, the smaller the parameter γ is. Thus, image B with the narrow close area will suit the RBF with the big γ, and E with the wide close area will suit the RBF kernel with the small value of γ.
    - **RBF kernel with γ = 1 → B**
    - **RBF kernel with γ = 0.2 → E**

3) Capability of Generalization

a. The scientific term of the balance in ML is generalization. Generalization is the ability to handle unseen data. The capability of generalization is determined by the system complexity and goodness of fit. A good generalization is the key for a good machine learning. Over training and high system complexity might reduce the quality of the generalization, thus balance is needed between good model fitness and simple model to achieve the "as simple as possible model but not simpler" model which is a model with just the suitable generalization.

b. The expression that describes AIC is given by $AIC = 2p - 2\ln(\hat{L})$, let us understand what each parameter means and what does it contribute to AIC:

- 2p: gets higher as our data gets bigger (p is the number of estimated parameters). This part of the expression plays a rule in reducing the fitness to avoid over fitting by increasing model complexity.

- $-2\ln(\hat{L})$: is an indicator of the goodness of the fitness of the model (L is the likelihood of the model with the data). The higher the fitness the higher this expression the lower AIC (because of the minus in the expression).

The expression is balanced to allow good fit but not over fitting.

c. As explained above, imbalance between the parameters that controls the model complexity and goodness of fit may lead to overfitting or underfitting. This might damage the generalization of the model and we will not achieve a good performance of the ML.

d. The AIC is defined as the estimator of out-of-sample prediction error. So we aim to have a low AIC in our model in order to have a good fitting model. We should be carful about how low the AIC in our model is to avoid overfitting, so balance is needed as we explained before.