HW3

Q1

a. k-medoid and k-mean algorithms are very similar clustering algorithm. The noticeable difference is using median instead of mean in the iterative step to assign the data point into a cluster. As median is less affected by noise or outlier compered to mean so is k-medoid is more robust in compression to K-means.

b. $\bar{x}$ is the mean and calculated by

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

Now for the proof

$$\sum_{i=1}^{m}(x_i - \mu)^2 = \sum_{i=1}^{m}(x_i - \bar{x} + \bar{x} - \mu)^2 =$$

$$= \sum_{i=1}^{m}(x_i - \bar{x})^2 + 2 * \sum_{i=1}^{m}(x_i - \bar{x})(\bar{x} - \mu) + \sum_{i=1}^{m}(\bar{x} - \mu)^2 =$$

$$= \sum_{i=1}^{m}(x_i - \bar{x})^2 + 2 * (\bar{x} - \mu)\sum_{i=1}^{m}(x_i - \bar{x}) + n\,(\bar{x} - \mu)^2$$

$$** \sum_{i=1}^{m}(x_i - \bar{x}) = 0$$

So

$$= \sum_{i=1}^{m}(x_i - \bar{x})^2 + m\,(\bar{x} - \mu)^2$$

And $\mu = \bar{x}$ minimize the equation.

Proof for **

$$\sum_{i=1}^{m}(x_i - \bar{x}) = \sum_{i=1}^{m} x_i - \sum_{i=1}^{m} \bar{x} = \sum_{i=1}^{m} x_i - m\bar{x} = \sum_{i=1}^{m} x_i - \sum_{i=1}^{m} x_i = 0$$

By definition

$$m\bar{x} = \sum_{i=1}^{m} x_i$$

Q2

D and A:

In both D and A, we can see linear separation between the two groups which appropriate for Linear Kernel. The C hyperparameter penalize the model for misclassification, so it may choose a smaller-margin hyperplane if it does a better job of getting all the training data classified correctly. In this case A match C=0.01 and D match C=1.

C:

In C the separation between the two groups is done clearly by 2-order polynomial.

F:

In F we can see the danger of using too complex model. Instead of learning from the train data the model exhibit overfitting. This is typical for use of high order polynomial kernel.

B and E

In B and E, the separation between the two groups is done by RBF kernel. The γ hyperparameter determine how much the model will try to fit to the data. high Gamma can lead to overfitting. So E matches $\gamma = 0.2$ and B for $\gamma = 1$.

1-A 2-D 3-C 4-F 5-E 6-B

Q3

a. The machine learning term for the balance is Generalization. In machine learning we want to find the "sweet spot" between bias ("not too simple") to Variance and overfitting ("too complex"). We can train our model endlessly, but after a point it well lose its ability to Generalize and give useful prediction for new (unseen) data.
b. When our model become more and more complex (has more parameters) it tends to overfit to the train data set. So large p (which represent the number of parameters in our model) can indicate our model will overfit. The $\ln(\hat{L})$ represents the Log-Likelihood function of our model (which is calculated using the values of the learned parameters). This function measures our model fit to the data. high value indicates high model accuracy.
c. Overfitting, or high bias. One option is to train inaccurate model with high disparity between the model prediction and the data (on training and test data). The second option is to train a complex model which overfit to the train data set. The model will get good result on the train set but will have poor performance on the test set.
d. We should aim to a low AIC score. We want our model to be as simple as possible (so low value of p) and also want an accurate model with good fit to the data set so high value of $\ln(\hat{L})$, which translate to low AIC.