Daniel Cherniavsky 205847817

1. Clustering

a. Is K-medoid more robust to noise (or outliers) than the K-means algorithm? Explain your answer.

K-means attempts to minimize the total squared error , when the error is calculated between the mean of each cluster and the points that are labeled to be in that cluster. However k-medoids minimizes the sum of dissimilarities between points labeled to be in a cluster and a point designated as the center of that cluster. In contrast to the *k*-means algorithm, k-medoids chooses datapoints as centers**.** Moreover, it could be more robust to noise and outliers as compared to k-means because it minimizes a sum of general pairwise dissimilarities instead of a sum of squared Euclidean distances of the points that are belong to a cluster from the mean of the cluster. As learned in the lectures, a mean value is sensitive to noise, and due to that it is not robust such as median value. For example, let us look at this simple example: `[0, 1, 2, 3, 100]` Both the median and medoid of this set are `2` . The mean is `21.2`. While, mean has the lower squared error, but assuming that there might be a measurement error in this data set the medoid is much more representative in this case. Another fact that makes K -medoid more robust than K- means is that in oppose to K-medoid algorithm, The algorithm of K -means does not guarantee convergence to the global optimum. The result may depend on the initial clusters, which makes it more sensitive to noise.

1.b. Prove that for 1D case ($x \in R^1$) of k-means, the centroid (μ) which minimizes the term of $\sum_{i=1}^{m}(x_i-\mu)^2$ is the mean of m examples.

Proof:

For finding the μ which minimizes the term, let us look at:

$$\arg\min_{\mu}\left(\sum_{i=1}^{m}(x_i-\mu)^2\right)$$

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

(derivative is linear operator)

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

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

$$\Rightarrow \mu_0=\frac{\sum_{i=1}^{m}x_i}{m}$$ ∎

Let us check that μ_0 is a minimum point:

$$\frac{\partial}{\partial\mu^2}\left(\sum_{i=1}^{m}(x_i-\mu)^2\right)^2 = 2m\cdot\mu\Big|_{\mu=\mu_0} > 0$$

$\Rightarrow \mu_0=\frac{\sum_{i=1}^{m}x_i}{m}$ is a minimum.

Bonus: Prove that the centroid (practicly the medioid) which minimize the term $\sum_{i=1}^{m} |X_i - \mu|$ is the median of m examples given that μ belongs to the dataset.

Proof:

For finding the μ which minimizes the term, let us look at.

$$\underset{\mu}{\operatorname{argmin}} \left(\sum_{i=1}^{m} |X_i - \mu| \right)$$

$$\Rightarrow \frac{\partial}{\partial \mu} \left(\sum_{i=1}^{m} |X_i - \mu| \right) = \sum_{i=1}^{m} \frac{\partial}{\partial \mu} \left(|X_i - \mu| \right) = \sum_{i=1}^{m} \operatorname{Sign}(X_i - \mu) \equiv 0$$

(derivative is a linear operator; demand)

Let us consider the expression $\operatorname{Sign}(X_i - \mu)$ and which values it might get:

$X_i < m \Rightarrow \operatorname{Sign}(X_i - \mu) = -1$

$X_i > \mu \Rightarrow \operatorname{Sign}(X_i - \mu) = 1$

$X_i = \mu \Rightarrow \operatorname{Sign}(X_i - \mu) = 0$

$\Rightarrow$ There must be equal amount of ± 1 inside the sum in order the derivative to equal 0.

$\Rightarrow$ There must be equal amount of X_i that $X_i > \mu$ and $X_i < \mu$

$\Rightarrow$ Because μ belongs to the dataset, μ must be the median ■

2. SVM

In the following figures you can see a visualization of SVM running with different settings (kernels and parameters) as follows:

SVM method based on creating a decision boundary which makes the distinction between two or more classes. The linear, polynomial and RBF or Gaussian kernel are simply different by the hyperplane decision boundary. The non-linear kernel functions are used to map the original dataset into a higher dimensional space in order of making it linearly separable. In each one of this method we use a hyperparameter which helps us to balance between bias and variance and thus, prevent the model from overfitting or underfitting. Soft margin SVM, which is not overfitted, allows some examples to be misclassified or be on the wrong side of decision boundary. Soft margin SVM often results in a better generalized model.

Now I will classify the figures:

Linear SVM: this method uses the hyperparameter "C" which adds a penalty for each misclassified data point. If C is small, the penalty for misclassified points is low so a decision boundary with a large margin is chosen, so we allow greater number of misclassifications. If C is large, SVM tries to minimize the number of misclassified examples due to high penalty which results in a decision boundary with a smaller margin.

A - 1. Linear kernel with C = 0:01. The boundary decision line is the locate in the middle of the margin. As we can see, two purple points located inside the margin (on the boundary decision line). Hence it is a "soft margin" with low c.

D - 2. Linear kernel with C = 1. As we can see, the margin is not including points of any group which meets the definition of large c parameter.

Polynomial SVM: In this case the hyperparameter is used to set the degree of the polynomial kernel. As greater the degree of the polynomial kernel the more we tend to overfitting.

C - 3. 2nd order polynomial kernel. By looking at the shape of the boundary decision, we can see the shape of the paraboloid which suits 2nd degree polynomial. In addition, we can assume that the model is not overfitted in that case due to quite large margin.

F - 4. 10th order polynomial kernel. In that case we can see a much more overfitted model, due to small margin and really low chance for misclassification.

RBF SVM: The radial basis function is based on the gaussian distribution in multidimensional space. Gamma parameter of RBF controls how much we fit the training data. Low values of gamma results in more points being grouped together. For high values of gamma, the points need to be very close to each other in order to be considered in the same group. Therefore, models with very large gamma values tend to overfit.

E - 5. RBF kernel with gamma = 0:2. We can that this model is based on the RBF kernel due to it round shape which suits to multidimensional gaussian. In addition, we can see here quite large margin which suits to low gamma.

B - 6. RBF kernel with gamma = 1. As explaind for the previous figure, we can see here a RBF model. However this model is much more overfitted and the margin is much smaller, which suits to high gamma.

3. Capability of generalization

a. What is the scientific term of the balance that Einstein meant to in machine learning aspect?

"Everything should be made as simple as possible but not simpler" – Albert Einstein. In other words, it means that the best theory is the simplest that makes quite good observations. In my opinion, the scientific term that represents this concept is the term of generalization. We try to achieve that concept by using a proper cost function. In supervised machine learning, models are trained on training data. The goal is to find the label of each training example from the training data, by that create a learning model. Then, we apply the created model on the test set. The Cost Function quantifies the error between predicted values and expected values for the test set. The purpose of Cost Function is on the one hand is to be minimized – so the predicted outcomes of the model will be as closer as possible to the expected values. On the other hand, we prefer that our model won't learn about outliers that in many cases consider to be a noise. So, we do not want our model to perform too well. To avoid this overfitting we use regularization term, which helps us to get the generalization. Regularization basically adds the penalty as model complexity increases, by that avoiding overfitting.
As explained in the tutorial, we want our model to learn the data and not memorize it. So, when it meets new examples that are not exists in the data set it will know how to classify them. Basically, this is the meaning of the generalization – creating a model that is quite simple and not overfitted, yet good enough to classify correctly the examples in high accuracy. Another way to describe that balance is by looking at the bias variance tradeoff. Bias tells us about the difference between the average prediction of our model and the correct value which we are trying to predict. Variance tells us about the spread of our data. The tradeoff between them means that if our model is too simple then it may have high bias and low variance. On the other hand, if our model is too complex then it's going to have high variance and low bias. So, we need to tune our model and find balance without overfitting and underfitting the data.

b. How does each of the terms $(\boldsymbol{2p};\ \boldsymbol{2ln}(\hat{\ }\ \boldsymbol{L}))$ in AIC affect the terms of the balance you defined in (a)?

The AIC lets us test how well our model fits the data set without over-fitting it. Its rewards models that achieve a high score and penalizes them if they become overly complex. The formula of AIC is: $AIC = 2p - 2ln(\hat{L})$. Actually, we can deduce that the best-fit model according to AIC is the one that explains the greatest amount of variation using the fewest possible independent variables. P representing the number of independent variables used to build the model. So as greater P is, as we have more parameters in our model, meaning a more complex model that will tend to overfitting. The $\boldsymbol{ln}(\hat{L})$ part of the equation represents the maximum likelihood estimate of the model. Meaning that this part explains how good our model performs. The ln function is a monotonic increasing function, hence as the likelihood is greater as this part of the equation increasing, causing decrease in AIC. So in total, for a generalized model we want to use as less parameters as possible (low p) and still be accurate by our classification – high $ln(\hat{L})$, and in total a low AIC.

c. What are the two options that are likely to happen if this balance was violated?

Actually, the balance that is described is on the one hand a generalized model that learns the data and doesn't memorize it, meaning not overfitted model. On the other hand we don't want to simple model that misclassifies. Given that, the first problem that may appear if the balance is violated is a too complex model which includes high number of parameters (great P), causes overfitting and incompatibility to generalize. The second problem that may occur is too simple model that does not predict well (low $\ln(\hat{L})$).

d. What are we aiming for with the AIC? Should it be high or low? Explain.

We are aiming with the AIC to find the balance that is described in this question. Meaning that we want to find a model that describes our phenomenon as simple and generalized as possible and still accurate enough. As explained, we strive to have **low p** – (simple model with low amount of parameters) and **high** $\mathbf{ln}(\widehat{\boldsymbol{L}})$ (high likelihood estimation) , and by looking at the AIC formula: $AIC = 2p - 2ln(\widehat{L})$, we want AIC score as low as possible.