HW5 DM Assignment

1. (10 pts) Suppose we find K clusters using Ward's method, bisecting K-means, and ordinary k-means. Which of these solutions represents a local or global minimum? Explain briefly.

   Solution:
   In the conventional k-means clustering method, a data point is allocated to the closest centroid, and the new mean is calculated. With each cycle, K centroids are initialized at random. The SSE (Sum of Squared Error) continuously improves until it reaches a regional or global minimum.

   Ward's agglomerative technique merges the clusters from different states based on the lowest rise in SSE inside clusters. Global and local minima are unlikely to be encountered by this greedy approach, which computes the clustering only level by level.

   By combining the two techniques, bisecting K-means divides the massive initial set of points into smaller groups. This is also a greedy strategy that optimizes calculations incrementally without considering the overall situation. Considering the above points it seems that it is very unlikely that it will encounter a global minima or even a local minima.

2. (20 pts) Apply single and complete link hierarchical clustering algorithms to cluster four coronavirus genomes (with their distances shown in the table below). Show your calculations (step by step) and the dendrogram of the clustering results.

genome	*A*	*B*	*C*	*D*
A	0	20	7	10
B		0	15	8
C			0	6
D				0

SINGLE LINK

	A	B	C	D
A	0	20	7	10
B		0	15	8
C			0	6 → min
D				0

1st Dendogram

	A	B	[C,D]
A	0	20	7 → min
B		0	8
[C,D]			0

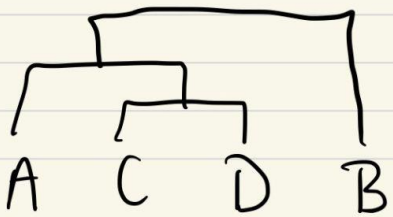
$$d(A,[C,D]) = \min(d(A,C), d(A,D))$$
$$= \min(7, 10) = 7$$

$$d(B,[C,D]) = \min(d(B,C), d(B,D))$$
$$= \min(15, 8) = 8$$

2nd Dendogram

	[A,C,D]	B
[A,C,D]	0	8
B		0

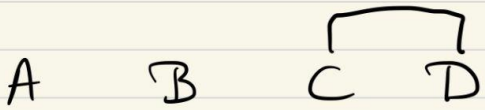
$$d(B,[A,C,D]) = \min(d(B,A), d(B,C), d(B,D))$$
$$= \min(20, 15, 8)$$
$$= 8$$



COMPLETE LINK

	A	B	C	D
A	0	20	7	10
B		0	15	8
C			0	(6) → min
D				0

1st dendogram



	A	B	[C,D]
A	0	20	(10) → min
B		0	15
[C,D]			0

$$d(A,[C,D]) = \max(d[A,C), d(A,D)) = \max(7, 10) = 10$$

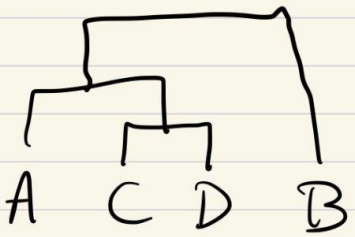
$$d(B,(C,D)) = \max(d(B,C), d(B,D))$$
$$= \max(15, 8) = 15$$

2nd dendogram

	[A,C,D]	B
[A,C,D]	0	20
B		0

$$d(B,[A,C,D]) = \max(d(B,A), d(B,C), d(B,D))$$
$$= \max(20, 15, 8) = 20$$

3rd Dendogram



3.

a)
When the data includes clusters of dramatically different sizes, sampling will become troublesome. There are undoubtedly several much larger clusters and some smaller clusters in the sample space. The larger clusters will unavoidably dominate and lead to an overrepresentation of those classes, despite the fact that the smaller clusters will not be adequately represented. Particularly in hierarchical clustering, it will be increasingly difficult to detect the smaller clusters from within the larger ones.

b)
When the data has a disproportionately large number of dimensions, the curse of dimensionality will make sampling challenging. The sampling process often ignores important relationships between data points, such as variance and correlations. The lack of important background could lead to a number of incorrect judgments.

c)
Sampling will be challenging because of unusual points and outliers. Any outliers that are taken into consideration for a sample may be included and skew the cluster, producing an inaccurate result, or they may be deleted and cause the sample to separate, either creating its own cluster or joining another cluster improperly, disrupting it.

d)
Extremely erratic placements could be problematic and have an impact on the sample process. The irregular zones face the danger of being missed or disregarded during sampling because of their complex construction and hazy borders. As a result, each data point won't be completely captured when clusters form. Concurrently, due to varying border definitions, certain data points may be incorrectly categorized.

e)
As hierarchical clustering is ideally adapted to manage globular clusters due to its repeated refining process, sampling in this case will only become problematic if the resulting clusters are exceedingly small or dense. The clusters might not, however, always adequately represent the dataset.

f)
Sampling would surely be challenging for a dataset with varying densities. The high-density data points may be under- or over-sampled as a result of this. It's also possible that the low-density clusters won't be seen at all.

g)
As there aren't as many noise spots present in this case, sampling won't encounter any major issues. However, if the few noise points deviate significantly from the existing data points, they may skew some clusters or perhaps produce new clusters.

h)
Non-Euclidean data will surely present issues for sampling in hierarchical clustering. Given that hierarchical clustering implicitly relies on the Euclidean distance between two points, choosing representative data points would be difficult and could lead to inaccurate or distorted groups.

4. (20 pts) Compute the entropy and purity for the confusion matrix (comparing a clustering result with external knowledge) below. Does it represent a good clustering result?

Cluster	Financial	Science	Sports	Politics	Total
#1	1	0	200	99	300
#2	50	50	50	150	300
#3	200	5	5	0	210
#4	0	2	250	8	260

(Solution)

First, let's calculate the entropies for all the clusters, then the purities and finally compare the two

Entropy Formula:

$$e_i = -\sum_{j=1}^{L} \frac{m_{ij}}{m_i} \log_2 \frac{m_{ij}}{m_i}$$

Entropy for cluster #1

$$\Rightarrow -\frac{1}{300}\log_2\frac{1}{300} - \frac{0}{300}\log_2\frac{0}{300} - \frac{200}{300}\log_2\frac{200}{300} - \frac{99}{300}\log_2\frac{99}{300} = 0.945$$

Entropy for cluster #2

$$\Rightarrow -\frac{50}{300}\log_2\frac{50}{300} - \frac{50}{300}\log_2\frac{50}{300} - \frac{50}{300}\log_2\frac{50}{300} - \frac{150}{300}\log_2\frac{150}{300} = 1.792$$

Entropy for cluster #3

$$\Rightarrow -\frac{200}{210}\log_2\frac{200}{210} - \frac{5}{210}\log_2\frac{5}{210} - \frac{5}{210}\log_2\frac{5}{210} - \frac{0}{210}\log_2\frac{0}{210} = 0.032$$

Entropy for cluster #4

$$\Rightarrow -\frac{0}{260}\log_2\frac{0}{260} - \frac{2}{260}\log_2\frac{2}{260} - \frac{250}{260}\log_2\frac{250}{260} - \frac{8}{260}\log_2\frac{8}{260} = \boxed{0.262}$$

Total Entropy

$$\Rightarrow \frac{-251}{1070}\log_2\frac{251}{1070} - \frac{57}{1070}\log_2\frac{57}{1070} - \frac{505}{1070}\log_2\frac{505}{1070} - \frac{257}{1070}\log_2\frac{257}{1070} = \boxed{0.837}$$

Purity Formula

$$P_i = \max_j\left(\frac{m_{ij}}{m_i}\right)$$

Purity for cluster #1

$$P_1 = \frac{200}{300} = 0.667$$

Purity for cluster #2

$$P_2 = \frac{150}{300} = 0.5$$

Purity for cluster #3

$$P_3 = \frac{200}{210} = 0.952$$

Purity for cluster #4

$$P_4 = \frac{250}{260} = 0.961$$

Total Purity

$$\Rightarrow \frac{300}{1070} \times 0.667 + \frac{300}{1070} \times 0.5 + \frac{210}{1070} \times 0.952 + \frac{260}{1070} \times 0.961 = 0.747$$

Because we can see that total entropy is less than total purity, we can say that the given matrix does represent a good clustering.

5.

Solution:

a) The contingency tables for A and B are:

	A = T	A = F	Total
+	4	0	4
-	3	3	6
Total	7	3	10

	B = T	B = F	Total
+	3	1	4
-	1	5	6
Total	4	6	10

Calculating the entropy below we have:

Entropy pre-split,

Ei = -1 * [(count(A=T)/total) * $\log_2$(count(A=T)/total) + (count(B=T)/total) * $\log_2$(count(B=T)/total)]= -[0.4 * $\log_2$ 0.4 + 0.6 * $\log_2$ 0.6] = 0.97

Information Gain with A,

E (A=T) = - (4/7) * $\log_2$ (4/7) - (3/7) * $\log_2$ (3/7) = 0.985

E (A=F) = - (3/3) * $\log_2$ (3/3) - (0/3) * $\log_2$ (0/3) = 0

=> ig(A) = Ei – [7/10 * E(A=T) + 3/10 * E(A=F)] = 0.97 – [0.689 + 0] = 0.28

Information Gain with B,

E (B=T) = - (3/4) * $\log_2$ (3/4) - (1/4) * $\log_2$ (1/4) = 0.811

E (B=F) = - (1/6) * $\log_2$ (1/6) - (5/6) * $\log^2$ (5/6) = 0.65

=> ig(B) = Ei – [4/10 * E(B=T) + 6/10 * E(B=F)] = 0.97 – [0.324 + 0.39] = 0.256

Chosen Information Gain = max[IG(A),IG(B)] = max(0.28,0.256) = 0.28

Clearly, igA is greater than igB, hence the decision tree will favor A

b)

Gini index pre split,
Gi = 1 – (0.42 + 0.62) = 0.48

Gini Gain on A,
G (A=T) = 1 – [(4/7)^2 + (3/7)^2] = 0.489
G (A=F) = 1 - [(3/3)^2 + (0/3)^2] = 0
Gini(A) = Gi - [(7/10) * G(A=T) + (3/10) * G (A=F)] = 0.48 - [(7/10) * 0.489 + 0] = 0.137

Gini Gain on B,
G (B=T) = 1 – [(1/4)^2 + (3/4)^2] = 0.375
G (B=F) = 1 - [(1/6)^2 + (5/6)^2] = 0.278
Gini (B) = Gi - [(4/10) * G(B=T) + (6/10) * G(B=F)] = 0.48 - [(4/10) * 0.375 + (6/10) * 0.278] = 0.163
Final Gini = max(Gini(A), Gini(B))
=> max(0.137, 0.163) = 0.163

The observed GINI Index for B is greater than that of A, hence the tree will favor B

c)
Classification error before split,
Ci = 1 - max(0.4,0.6) = 0.4

Error associated with A,
C (A=T) = 1 – max(4/7,3/7) = 3/7 = 0.428
C (A=T) = 1 – max(3/3, 0/3) = 0
CE (A) = Ci – [7/10 * 0.428 + 0] = 0.4 – 0.2996 = 0.1004

Error associated with B,
C (B=T) = 1 – max(1/4,3/4) = 1/4 = 0.25
C (B=T) = 1 – max(1/6,5/6) = 1/6 = 0.166
CE (B) = Ci – [4/10 * 0.25 + 6/10 * 0.16] = 0.4 – 0.1996 = 0.2004

The appropriate classification error is given by = max(A,B)
=> max(0.1004,0.2004) = 0.2

Since B is greater the decision tree favors B

d)
Every one of the three impurity metrics supports splitting based on certain distinctions they are sensitive to. Classification error calculates the percentage of incorrect classifications, entropy determines the typical amount of data needed to identify a randomly chosen example, and the Gini index calculates the likelihood of incorrect classifications.

Entropy favored characteristic A on the split, as seen in the case above, while the Gini index and classification error supported attribute B. It is therefore highly likely that they have distinct preferences for particular attributes because of the objective method in which they measure the impurity itself.

6. (10 pts) Consider a labeled data set containing 100 data instances, which is randomly partitioned into two sets of A and B, each containing 50 instances. You use A as the training set to learn two decision trees, T10 with 10 leaf nodes and T100 with 100 leaf nodes. The accuracies of the two decision trees on data sets A and B are shown in the table below.

	Accuracy	
Data Set	T_{10}	T_{100}
A	0.86	0.97
B	0.84	0.77

(a) Based on the accuracies shown in the above table, which classification model would you expect to have better performance on unseen instances?

(Solution) I would choose the model T10 because we can see that T10 has similar accuracy as the train data (A) on the test data (B). Whereas, T100 has an accuracy of 0.97 on the train data which could likely mean it has overfit the dataset. We can confirm this information by looking at it's accuracy in the test dataset which is 0.77.

(b) Now, you tested T10 and T100 on the entire data set (A + B) and found that the classification accuracy of T10 on the data set (A + B) is 0.85, whereas the classification accuracy of T100 on the data set (A + B) is 0.87. Based on this new information and your observations in the table, which classification model would you finally choose for classification.

(Solution) Again I would choose model T10 for classification here. Although the accuracies are similar in the combined dataset, it still stands that T100 has overfit the train set (A). Due to this, the accuracy we see 0.87 on (A+B) is the average of its accuracies on A and B. So the T100 model would still be a poor choice of model for our problem. However, T10 has similar performance on A and B individually and when combined will have an average between the two. Hence, T10 is a better choice for our problem here.