Assignment 3

***Note 1**: for answers with Python, display both codes and results clearly.*

***Note 2**: for answers with manual calculation, please display all calculation steps clearly.*

Question 1. **[30 points @ 6 points each]** A firm collected 5 training instances with 2 features $\boldsymbol{X_1}$ and $\boldsymbol{X_2}$, and their ***Type*** values.

Instance	X_1	X_2	Type
0	13.4	11.2	1
1	7.9	2.1	0
2	7.1	8.9	1
3	7.3	6.9	0
4	10.7	8.9	1

(a) Use Python to plot the 5 instances with X_1 on the x-axis and X_2 on the y-axis. Visualize instances with different color according to their ***Type*** values.

With a new instance with $(X_1, X_2) = (6.5, 2.1)$, please complete the following tasks with either Python or manual calculation. Round results to 4 decimal places if you use manual calculation. No need to round if you work with Python.

(b) Calculate the Euclidean Distance between the new instance and each training instance using both X_1 and X_2.

(c) Calculate the Cosine Distance between the new instance and each training instance using both X_1 and X_2.

(d) What is the predicted ***Type*** value for the new instance using 3-NN and *majority vote* (based on Cosine Distance)? What is the estimated class probability?

(e) What's the predicted ***Type*** value for the new instance using 3-NN and *weighted voting* (based on Euclidean Distance)? What is the estimated class probability?

Please report the results in one or two tables. For example, answers for Q1(b) -(c) can be organized as below:

Instance	X₁	X₂	Type	(b) Euclidean Distance	(c) Cosine Distance
0	13.4	11.2	1		
…	…	…	…		

Assignment 3

Question 2. **[30 points]** A firm collected 6 instances with 2 features X_1 and X_2.

Instance	X_1	X_2
0	1	4
1	1	3
2	0	5
3	5	2
4	6	3
5	4	0

With instance 0 and 3 selected as initial centroids, we'd like to simulate the k-means algorithm to separate all instances into two clusters ($k = 2$). Please complete the following tasks with Python or manual calculation. Round results to 4 decimal places if you use manual calculation. No need to round if you work with Python.

(a) **[5 points]** Compute Euclidean distance from each instance to the initial centroids.

(b) **[5 points]** Assign instances to the two clusters by finding their closest centroids.

(c) **[5 points]** Compute the clustering quality with $SSE = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, m_i)^2$.

(*Note:* $d(p, m_i)$ *is Euclidean Distance between instance* p *& its centroid* m_i*.*)

(d) **[5 points]** Compute the mean feature values for instances in the two clusters respectively, in the format of (X_1, X_2).

(e) **[10 points]** Update the cluster centroids with the mean feature values calculated in step (d), then repeat step (a) – (d) once. Will the clustering result (i.e., cluster labels) change? Any improvement in terms of SSE?

Please report the results in one or two tables. For example, answers for Q2(a)-(d) can be reported in the following table.

Instance	X₁	X₂	(a) Distance to Instance 0	(a) Distance to Instance 3	(b) Cluster Label	(d) Updated Centroid
0	1	4				
…	…	…				
5	4	0				
(c) SSE:						

Assignment 3

Question 3. **[24 points]** A bank trained a classification model to predict the likelihood of default for each customer. There are 1000 customers in the database: the “No Default” cases take up 80% of the data while the “Default” cases take up 20%. Applying this classifier on this dataset yields the following confusion matrix.

		Predicted Class	
		Default	**No Default**
Actual Class	**Default**	150	50
	No Default	100	700

As the average lending amount is \$100 and interest rate is 15%, the cost-benefit matrix (negative numbers means cost) is:

		Predicted Class	
		Default	**No Default**
Actual Class	**Default**	0	-\$100
	No Default	0	\$15

(a) **[4 points]** Which group (“*Default*” or “*No Default*”) will you consider as the positive class?

(b) **[8 points @ 2 points each]** Calculate the followings score for this model:

(i) Accuracy

(ii) True positive rate (*Sensitivity/recall*)

(iii) True negative rate (*Specificity*)

(iv) Precision (*for the positive class only*)

(c) **[4 points]** Calculate the expected value (per person) for this model.

(d) **[4 points]** Assume we aim to target the same proportion of customers as in the first table, with only positive predictions will be targeted. Write down the confusion matrix for a random classifier.

(e) **[4 points]** Calculate the overall expected value (per person) for the random classifier in step (d).

Question 4. **[16 points]** Two classifiers (Model A and B) are used to predict whether the Fed Funds rate will increase or not (class label: 1= increase, 0 = no increase), with each quarter considered as an instance. The estimated probabilities of increase over the past 6 quarters by model A and B respectively are displayed in the following table:

Quarter	Actual Class	Model A	Model B
0	1	0.43	0.63
1	1	0.52	0.53
2	1	0.85	0.56
3	1	0.69	0.71
4	0	0.03	0.18
5	0	0.31	0.76

Please complete the following tasks with either Python or manual work.

(a) **[12 points]** Plot the ROC curve for the 2 classifiers together with the random classifier. Please calculate the TP and FP rates with the following cutoff values [0, 0.2, 0.4, 0.5, 0.6, 0.8, 1] before plotting the ROC curve.

(*Note: you may need to calculate each model's TP and FP rates at each cut-off first. The visualization can be done with either manually or with Python.)*

(b) **[4 points]** Which model is better? Why?