Istanbul Technical University- Spring 2018-2019
BLG454E Learning From Data, Homework 1
Purpose: Linear Regression, Classification
Total worth: 100 points
Handed out: Thursday, March 14, 2019.
Due: Wednesday, March 31, 2019 23:00. (through ninova!)
Instructors: Yusuf Yaslan, Islem Rekik
Assistants: Fulya Çelebi Sarıoğlu (sarioglu16@itu.edu.tr),
İsmail Bilgen(ibilgen@itu.edu.tr)

You should use MATLAB or Python.

PART 1 [60 points]

Train regression model **y = x w + b** using *regression_data.txt*. The first column represents samples (each sample has a single feature which is 'head size') and the second column represents the target score to predict (i.e., brain weight). The aim is to train a linear regression model from **Head Size(x)** to **Brain Weight(y).**

a) Implement Linear Regression using Gradient Descent (Don't use built-in function for linear regression!) [30 points]

b) In order to fit (training stage) and evaluate (testing stage) your model use 5-fold cross validation (Don't use built-in functions!) [20 points]

c) Report or explain overall MSE (Mean Square Error) of 5-fold cross validation. [10 points]

PART 2 [40 points]

a) Examine the training dataset (*classification_train.txt*), the number of features, the number of classes and their probabilities (**$P(c_i)$**). Each sample has two features (first two columns) and one label (third column). Plot the training dataset (feature 1 on x-axis and feature 2 on y-axis). [5 points]

b) Assume that each class is generated from a multivariate Gaussian distribution.

c_1(class 1)	**c_2(class 2)**
Mean Vector ($\mu_1 = [? \ ?]$)	Mean Vector ($\mu_2 = [? \ ?]$)
Covariance Matrix $\Sigma_1 = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$	Covariance Matrix $\Sigma_2 = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$

Estimate the mean vector and covariance matrix for each class using the **training** data (*classification_train.txt*). [10 points]

c) Compute the discriminant function $g_i(x)$ for each class (c_i). [10 points]

d) Predict each testing sample x^{tst} class label by computing its $g_i(x^{tst})$ for all classes. [**Hint:** If $g_1(x^{tst}) \geq g_2(x^{tst})$ the predicted testing label is c_1 else c_2] [10 points]

e) Compute classification accuracy using the testing set (*classification_test.txt*) by comparing ground truth label and predicted labels. [5 points]

For example; accuracy of below table is **3/5.**

Test instance	**Feature1**	**Feature2**	**Actual Label**	**Predicted Label**
x_1	1.73802424	3.72105508	1	1
x_2	2.9007531	1.15054811	1	1
x_3	-0.07302643	7.54612352	1	0
x_4	-2.13195888	-0.71787816	0	0
x_5	1.49967502	6.85820584	1	0