Assignment #1
Assigned 9/9/2025, due on 9/20/2025, 11:59PM

DO NOT USE AI TO WRITE YOUR CODE. This assignment uses the Wine Quality Dataset, which will be uploaded in Canvas. Detailed information on this data set is found in:

https://archive.ics.uci.edu/dataset/186/wine+quality

For this data set, the following **gradient descent algorithms** are to be applied**:**

- **Batch:**
  without regularization
  with L2 regularization
  with L1 regularization

- **Mini batch (stochastically selected subset of the training data):**
  without regularization
  with L2 regularization
  with L1 regularization

Inspect / analyze the data set before you start the ML portion:

Divide your data randomly, into three subsets as follows:

(a)Train (say, approx. t%): X_{train}, an $m \times (n+1)$ matrix.
(b) Validation (approx. v%): X_{valid}, an $m \times (n+1)$ matrix
(c)Test (remaining subset $(100 - t - v)$%): X_{test}, an $m \times (n+1)$) matrix

Save these subsets to use for all the tasks for comparison purposes.

Y_{train}: the ground truth for the training set
Y_{valid}: the ground truth for the validation set
Y_{test}: the ground truth for the test set
Y_{hat}: the model output for the training set.

Train and test each model without regularization.
Recall the steps:

1. Let m be the size of the training set: Set up the weight vector (let's call it $\boldsymbol{w}$), of dimension n+1, where n is the number of attributes.
2. Randomly initialize $\boldsymbol{w}$ **= [w0, w1, …., wn]**
3. Calculate the output vector $\mathbf{Y_{hat}}$ of dimension $\boldsymbol{m} \times 1$, for the training data points, using the usual formula: for each data point (row vector in $\mathbf{X_{train}}$) compute the dot product with the vector $\boldsymbol{w}$**:** $\boldsymbol{x_i} \cdot \boldsymbol{w}$
4. Training MSE is the mean of $(\boldsymbol{Y_{train}} - \boldsymbol{Y_{hat}})$.^ **2**
5. Calculate **∇MSE** the vector of dimension n+1 according to the formula:

$$\boxed{\nabla \mathbf{MSE}\ (\mathbf{w}) = (2/m) \cdot (\mathbf{Xtrain})' \cdot (\mathbf{Xtrain} \cdot \mathbf{w} - \mathbf{Ytrain})}$$

6. Update **w**:

$$\mathbf{w} = \mathbf{w} - \lambda\ \nabla \mathbf{MSE}(\mathbf{w}),$$

   where λ is the learning rate.
7. Repeat Steps 4 – 8 until the termination condition is met (error less than some small value, bound on the number of epochs, or a combination of these).

For the mini-batch gradient descent, the steps above are carried out for subsets (mini-batches) of the training set selected stochastically.

Use **∇MSE** from the box equation in each of the models below:

L2 Regularization

Here we seek the vector of dimension n+1, with components:

$$\boldsymbol{w} = \boldsymbol{argmin}\left\{\boldsymbol{MSE}(\boldsymbol{w}) + \alpha\ ||\boldsymbol{w}\ ||^2\right\}$$

To solve this minimization problem, we take the gradient of $\{\mathbf{MSE(w)}+ \alpha\ |||\boldsymbol{w}\ ||^2\}$

$$\nabla(\boldsymbol{MSE}(\boldsymbol{w}) + \alpha||\boldsymbol{w}\ ||^2) = \nabla\big(\boldsymbol{MSE}(\boldsymbol{w})\big) + \nabla(\alpha||\boldsymbol{w}\ ||^2) =$$

$$= \nabla(\boldsymbol{MSE}(\boldsymbol{w})) + \alpha\ \nabla(||\boldsymbol{w}\ ||^2) = \nabla(\boldsymbol{MSE}(\boldsymbol{w})) + \mathbf{2}\ \alpha\ \boldsymbol{w}$$

You will need to come up with a way to select the parameter α(maybe validation?).

Eliminate the attribute (column in X) corresponding to the smallest component of the weight vector.

Train again, on the reduced set of attributes, selected by the L2-regularization step, using batch gradient using Xtrain, Xtest, etc. previously saved.

L1 Regularization

$$\boldsymbol{w} \cdot \boldsymbol{x} = w0 + w1x1 + \cdots + wnxn$$

Perform L1 regularization for the batch Gradient Descent. That is, instead of **∇MSE** from step 6, use the vector of dimension n+1, with components:

$$\boldsymbol{\nabla MSE} + \mathbf{2}\ \alpha\ sign(\boldsymbol{w}).$$

You will need to come up with a way to select the parameter α(maybe validation?).

Eliminate the attribute (column in X) corresponding to the smallest component(s) (value 0) of the weight vector.

Train again, on the reduced set of attributes, selected by the L1-regularization step, using batch gradient using **Xtrain**, **Xtest**, etc. previously saved.

Provide the following figures:

Plot the regression lines for each regression for the attribute corresponding to the largest component of **w**, for batch gradient descent before and after each L1 and L2 regularizations. NOTE: most likely, the regression lines will NOT look nice.

Please do the following:

1. Turn in your homework in a zipped folder identified by the names in alphabetical order of your team members.

2. At the top of your program files the names of your team members, and either the statement "*all team members have contributed in equal measure to this effort*", or, if that is not true, write **which team members contributed: they will receive full credit**. For those who did not contribute, I will override the group grade assigned by Canvas, by assigning 0.
3. Carefully comment your program
4. Turn in an analysis of (up to one page) of your results comparing the two regularization approaches.

Hint: For those coding in Matlab, Matlab has a "publish" option which can be used to produce .doc, .pdf,.html, or .tex files, which can then be further edited.