# Custom Implementation of Linear Regression using gradient descent

In this assignment, we will be implementing linear and logistic regression on a given dataset without using any available implementation of it.

In addition,  
we will experiment with design and feature choices.  
We will be using the Appliances Energy Prediction dataset available for download at  
https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction  
Goal:Implement a linear regression model on the dataset to predict the energy usage of appliances. You are  
not allowed to use any available implementation of the regression model. You should implement the  
gradient descent algorithm with batch update (all training examples used at once). Use the sum of  
squared error normalized by 2\*number of samples [J(β0, β1) = (1/2m)[∑(yᶺ(i) – y(i))2] as your cost and  
error measures, where m is number of samples. You can use any number of features greater than 15.  
You can decide on which features to use using some experimentation and exploratory analysis.  
Also implement a logistic regression model as described in Part 4. For logistic regression, you can write  
your own implementation similar to linear regression, or can use any R or Python library. You should use  
the logistic regression cost/error function from the class. In addition you can also use accuracy/ROC/etc.

Tasks: **Part 1:** Download the dataset and partition it randomly into train and test set using a good train/test  
split percentage.  
**Part 2**: Design a linear regression model to model the energy usage of appliances. Include your  
regression model equation in the report.  
**Part 3:** Implement the gradient descent algorithm with batch update rule. Use the same cost function as  
in the class (sum of squared error). Report your initial parameter values.  
**Part 4:** Convert this problem into a binary classification problem. The target variable should have two  
categories. Implement logistic regression to carry out classification on this data set. Report  
accuracy/error metrics for train and test sets.

Experimentation:

**1**. Experiment with various parameters for linear and logistic regression (e.g. learning rate ∝) and  
report on your findings as how the error/accuracy varies for train and test sets with varying  
these parameters. Plot the results. Report the best values of the parameters.

**2.** Experiment with various thresholds for convergence for linear regression. Plot error results for  
train and test sets as a function of threshold and describe how varying the threshold affects  
error. Pick your best threshold and plot train and test error (in one figure) as a function of  
number of gradient descent iterations.

**3.** Pick ten features randomly and retrain your models only on these ten features. Compare train  
and test error results for the case of using your original set of features (greater than 15) and ten  
random features. Report the ten randomly selected features.

**4.** Now pick ten features that you think are best suited to predict the output, and retrain your  
models using these ten features. Compare to the case of using your original set of features and  
to the random features case. Did your choice of features provide better results than picking  
random features? Why? Did your choice of features provide better results than using all  
features? Why?