Data driven business analytics
MGT-302
Homework 1

- This assignment can be solved individually or in groups of two students. You should mention the name of your partner in your report file. Note that both of the students in a group will get the same grade.
- Deadline: 30 March 2023, 23:59 (No late submissions will be accepted)
- Upload a single zip file on Moodle containing your report and code. You can use any programming language.

1 Linear Regression [30 pts]

A scientist studying weight fluctuation has collected data from some people. She is specifically interested in studying the influence of some features on the weight fluctuations of people. She collects data $\{x^{(i)}, y^{(i)}\}_{i=1}^{m}$, where $y^{(i)}$ is a continuous variable indicating the weight change, and $x^{(i)} = (x_{i1}, x_{i2}, \ldots, x_{id})$ are the variables which record different properties of people. These features are weight, height, age, amount of exercise time per day, and incoming calories per day. She also proceeds to collect new data to predict the weight change by using sound statistical methods. The dataset contains both the training and testing data. **Note that you need to normalize the features.**

Assume that sample size is $m = 200$ and number of features is $d = 5$. We model the problem as linear regression with cost function

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)})^2, \tag{1}$$

where $h_\Theta(x^{(i)}) = \Theta_0 + \Theta_1 x_1^{(i)} + \cdots + \Theta_d x_d^{(i)}$.

Derive the gradient descent

1. **Gradient Descent (GD)**
   (a) State the gradient of your cost function.
   (b) Implement the gradient descent method using your derived gradient to minimize the cost function. Fit the model (θ) using training data (weighttrain.mat), and report the learned model and the cost function for the learned model. Try to tune step-size α of GD to reach the fastest converging rate of decreasing cost function in time.
   Important notes: For this question, at each iteration, do not update all the Θ_is simultaneously, instead update them one by one. Also, set the initial value to zero vector. Moreover, make sure you do enough iterations so that your model converges.
   (c) Plot cost function in number of iterations.
   (d) Use your learned model for test data (weighttest.mat) and report the cost function for it.
2. **Stochastic Gradient Descent (SGD)**

(a) Briefly state SGD. Then repeat parts (b),(c),(d) for it. Then compare the two plots of part (b) for GD and SGD. Which algorithm converges faster?

(b) In this part, the step-size of SGD is choosen as a decreasing sequence $\alpha_t = \frac{b}{1+t}$. Compare the convergence rate of SGD with constant step-size and adaptive step-size. (You should also tune b to get the fastest converging rate of SGD with adaptive step-size.)

2 PCA [15 pts]

Principal component analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Consider the matrix $\mathrm{M} = \begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, where each row is input data and each column is a feature. Find the first and second principal components of M step by step (no need to implement it. You can drive them on the paper).

3 K-means Clustering [25 pts]

Recall that in the K-means clustering problem, we are given a training set $\{x^{(1)}, ..., x^{(m)}\}$, and want to group the data into a few cohesive clusters. Here, we are given feature vectors for each data point $x^{(i)} \in \mathbb{R}^n$ as usual; but no labels $y^{(i)}$ (making this an unsupervised learning problem). Our goal is to predict k centroids and a label $c^{(i)}$ for each data point.

1. K-means is sensitive to the initial points of the clusters. What do you suggest to avoid this problem?

2. Implement K-means algorithm using "kmeans.mat" dataset and visualize the output for $K = 4$. Your code should work for any number of clusters (not just $K = 4$).

3. Try different values for K. Also, try different values of initial condition. The algorithm must converge for all possible initial conditions, otherwise, there is a problem in your implementation. Plot the elbow curve and discuss the optimal K in a K-means clustering. Observe the cost function value as K increases and explain it briefly.

$$\sum_{i=1}^{K} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2 \tag{2}$$

4 Support Vector Machine (SVM) [30 pts]

In this part, you will see how SVM works. You will implement an SVM classifier and plot the training and validation loss. Please check out the Jupyter notebook file and complete the cells step by step.