Linear Regression Learning Algorithm

1. Load the dataset and divide it into features (i.e., inputs) and actual result (i.e., output). Store features and actual outputs in two separate variables.
2. Plot each example set in a graph if possible, in order to have an understanding of the data.
3. Determine a cost function, then pass the features, actual results and a set of initial parameters to the function to calculate cost values i.e., average over the squared error between the prediction and actual output.

   $J(X, y, \theta) = \frac{1}{2m} \sum_{i=1}^{n} \left(h_\theta\left(x^{(i)}\right) - y\right)^2$
4. Calculate the optimized parameter by running gradient descent algorithm. It's recommended to check whether the value of cost function is decreasing after each update of parameter vector θ. If it increases, then there must be some problem. Plot a 3D and a contour graph to keep track of the cost value converging towards the global minima.

   $\theta = \theta - \alpha \times \frac{1}{m} \left(X^T \times \left((X \times \theta) - y\right)\right)$
5. IMPORTANT: θ should be updated all at once before we calculate the cost value for that value of parameter θ.
6. Plot cost values against number of iterations, and it should show that the cost value gradually converges towards the global minima. It should never increase. Once it reaches at global minima, it stays there as the slope becomes 0 at global minima and derivative term also becomes 0 resulting in no changes in the value of θ.
7. Once our cost value reaches the global minima, we have obtained the required optimized vector θ.