Machine Learning 2016/2017: Assignment 1

1. Suppose that we have historical data of result of soccer matches of teams playing against Ajax. We want to use this information to learn to predict at a certain moment whether a team will win, lose or draw against Ajax. Our approach will be based on Machine Learning.
(a) Define the given and the goal of the prediction task and of the learning task that best matches our goal. Classify the learning task as supervised, unsupervised, reinforcement learning, and if supervised as classification or regression.

Prediction Given: A club
Prediction Goal: The correct ouput: win, lost, draw
Learning Given: The historical data
Learning Goal: Producing a working hypothesis

(b) What would be the form of training data for the learning task? Give a small training set.

X	Y
Club1	win
Club2	win
Club3	lose
Club4	draw
Club5	draw
Club6	lose

2. Given the following data:

X	**Y**
3	**6**
5	**7**
6	**10**

(a) Manually (using only a calculator) calculate two iterations of the gradient descent algorithm forunivariate linear regression function. Initialize the parameters such that the regression function passes through the origin (0, 0) and has an angle of 45 degrees. Use a learning rate of 0. Give the intermediate results of your calculations

and also compute the mean-squared error of the function after 2 iterations.

$\phi 0 := \phi 0 - \alpha * (1/m) * \sum (h\phi(x(i)) - y(i))$
$\phi 1 := \phi 1 - \alpha * (1/m) * \sum (h\phi(x(i)) - y(i)) * x(i)$

Itteration 1 $\phi 0 = 0$ $\phi 1 = 1$ $\alpha = 0.1$

$\phi 0 := 0 - 0.1 * (1/3) \sum x(i) - y(i)$
$\phi 0 := - 0.1 * (1/3) [(3 - 6) + (5 - 7) + (6 - 10)]$
$\phi 0 := - 0.1 * (1/3) * -9$
$\phi 0 := 0.3$

$\phi 1 := 1 - 0.1 * (1/3) \sum (x(i) - y(i)) x(i)$
$\phi 1 := 1 - 0.1 * (1/3) [(3 - 6)3 + (5 - 7)5 + (6 - 10)6]$
$\phi 1 := 1 - 0.1 * (1/3) (-9 - 10 - 24)$
$\phi 1 := 1 - 0.1 * -14(1/3) = 2.43$

Itteration 2 $\phi 0 = 0.3$ $\phi 1 = 2.43$ $\alpha = 0.1$

$\phi 0 := 0.3 - 0.1 * (1/3) \sum (0.3 + 2.43 * x(i) - y(i))$
$\phi 0 := 0.3 - 0.1 * (1/3) [(1.59 + 5.45 + 4.88]$
$\phi 0 := 0.3 - 0.1 * 3.97$
$\phi 0 := -0.10$

$\phi 1 := 2.43 - 0.1 * (1/3) \sum [0.3 - 2.43 * x(i) - y(i)] * x(i)$
$\phi 1 := 2.43 - 0.1 * (1/3) [$ $(0.3 - 2.43 * 3 - 6) * 3 +$
$(0.3 - 2.43 * 3 - 7) * 5 +$
$(0.3 - 2.43 * 3 - 10) * 6]$
$\phi 1 := 2.43 - 0.1 * (1/3) (-38.97 - 94.25 - 145.68)$
$\phi 1 := 11.73$

Mean Squared Error

$MSE = \sqrt{} (\sum((\phi 0 + \phi 1 * x(i)) - y(i))^2 / m)$
$MSE = \sqrt{} (\sum((-0.1 + 11.73 * x(i)) - y(i))^2 / 3)$
$MSE = \sqrt{} (\sum$ $((-0.1 + 11.73 * 3 - 6)^2$
$((-0.1 + 11.73 * 5 - 7)^2$
$((-0.1 + 11.73 * 6 - 10)^2$ $/ 3)$
$MSE = \sqrt{} ((846.2281 + 2657.4025 + 3633.6784) / 3)$

MSE = 48.776049 = 48.78

(b) Convert the data to z-scores (with mean = 0, sd = 1) and repeat the calculations above. Compare the results with those for the original data

The data will be the exact same because the formula for calculating z-scores is

z = (x - mean) / standard deviation

so if the mean is 0, and the sd is 1, nothing changes. The results will therefor not differ from the original data.

3. Suppose that X1 predicts Y, with some (mean squared) error MSE. We now extend the data with an additional variable X2 and use a learning algorithm that uses both X1 and X2 to predict Y. What will be the effect on the mean squared error of Y compared to just using X1 if X2 is equal to:

Original MSE:
$\sqrt{(\sum((\phi0 + \phi1 * x(i)) - y(i))^2 / m)}$

(a) a + b.X1
Updated MSE:
$\sqrt{(\sum((2\phi0 + 2\phi1 * x(i)) - y(i))^2 / m)}$

(b) a + b.X1^2
Updated MSE:
$\sqrt{(\sum((2\phi0 + \phi1 * x(i)) + \phi1\, x(i)^2 - y(i))^2 / m)}$

4. Derive an equation that can be used to find the optimal value of the parameter θ1 for univariate linear regression without doing gradient descent. This can be done by setting the value of the derivative equal to 0.

The cost function is:

$J(\theta0, \theta1) = (m/2) \sum (\theta0 + \theta1 * x(i) - y(i))^2$

Because $\theta 0$ is a constant we can treat it as such and write:

$J(\theta 1) = (m/2) \sum (\theta 0 + \theta 1 * x(i) - y(i))^2$

When taking the derivative of a sum, we take only the derivative of the inside of the sum, which will look like this:

$J(\theta 1)' = (m/2) \sum ((2 \theta 0 - 2y(i)) * x(i) + 2x(i)^2 * \theta 1$

When setting that equal to 0, we lose the (1/2)*m and are left with

$\sum ((2\theta 0 - 2y(i)) * x(i) + 2x(i)^2 * \theta 1) = 0$

This is a function that can then be solved for any data set, and will lead to the optimal $\theta 1$.