**MTH655: Assignment 1**

*Due: September 25, 9pm*

*Please submit your solutions through D2L*

**Regression:**

In class we discussed linear regression model applied to “Salary vs. Age” data. In particular, we estimated the regression

**Y=a+bX**

where Y is salary and X is age. In this assignment you will learn how to select the best model from a set of models.

**Problem formulation**

Estimate the 5th (see p.10 in slides for Chapter 1) and the 2nd (see p.15 in slides for Chapter 1) degree polynomial regressions. Test the performance of these models similarly to how it is done in “Linear regression” (errTrain and errTest variables) to obtain the table on p.17 in slides for Chapter 1.

**To submit:**

1. Python codes that implement each regression. The code must be structured similarly to “Linear regression” code published on D2L (for example, the block that plots the results must be present in your code as well) (**50 points**)
2. Your explanation of which model is best (in other words, you will have to explain why a particular model is best and why you should choose that model over the other models that you estimated). It would be great if in your answer you use the words that are common for Machine Learning “overfitting”, “underfitting”, “generalization”, etc. (**20 points**)

**Optimization:**

Gradient descent algorithm is a very popular numerical optimization technique. Training an Artificial Neural Network is essentially applying the gradient descent algorithm. It is based on the fact that a function increases in the direction of its gradient. Consider the function

f(x,y) = x2+y2

which has the gradient (2x, 2y)T. Loosely speaking, the gradient is the vector of first partial derivatives. Let’s fix the point (x0,y0), then the gradient at this point is (2x0, 2y0)T.

The gradient descent algorithm is as follows:

1. Choose the starting point (x0,y0) and specify the step size *a*.
2. Update the point according to the equation

(xk,yk) = (xk-1,yk-1) – *a* \* gradient at (xk-1,yk-1)

1. Repeat step 2 until the difference between (x1,y1) and (x0,y0) is very small.

**Problem formulation**

Write a Python script that implements the gradient descent algorithm for function f(x,y) = x2+y2 staring at the point (1,1) and using the step size 0.01. Did you reach the minimum function value of 0? How many steps are required to get to the point where the function has minimum value?

**To submit:**

The Python code that implements the gradient descent algorithm. (**30 pts**)