

assignment07

November 14, 2018

Name : Do-Hyun Kim

Student ID : 2018120193

github : https://github.com/ppooiiuuyh/datamining_assignments/tree/master/assignment07

```
In [7]: #=====
# import modules
#=====
import numpy as np
import matplotlib.pyplot as plt

In [8]: #=====
# my equation
#=====
def fun(x):
    # f = np.sin(x) * (1 / (1 + np.exp(-x)))
    f = np.abs(x) * np.sin(x)
    return f

In [9]: #=====
# my polyfit
#=====
def mypolyfit(x,y,p):
    X = np.array([ [x[j]**i for i in range(p+1)] for j in range(x.shape[0]) ])
    #X*Xt
    Xt_X = np.matmul(X.T,X)
    #print(Xt_X)
    #(X*Xt)-1
    Xt_X_inv = np.linalg.inv(Xt_X)
    #print(Xt_X_inv)

    # XF = Y
    # XtXF = XtY
    # F = (XtX)-1XtY
    fits = np.matmul( Xt_X_inv, np.matmul(X.T,y))
    #print(fits.shape)
    return fits
```

```

In [10]: #=====
# my polyval
#=====
def mypolyval(x,fits):
    X = np.array([ [x[j]**i for i in range(fits.shape[0])] for j in range(x.shape[0])])
    vals = np.matmul(X,fits)
    return vals

In [11]: #=====
# variables
#=====
num      = 1001
std      = 5

n        = np.random.rand(num)
nn       = n - np.mean(n)
x        = np.linspace(-10,10,num)
y1       = fun(x)
y2       = y1 + nn * std

In [12]: #=====
# plotting
#=====
for i in range(2,10):
    popt = mypolyfit(x, y1, i)
    out = mypolyval(x,popt)
    plt.plot(x, y1, 'b.', x, y2, 'k.')
    plt.plot(x,out,'r',label="p="+str(i))
    plt.legend()
    plt.show()

```







