Student ID: 06996263 Student name: Fang Zhu

Q1:

Code:

```
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
import math
import textwrap
import numpy as np
import sklearn
from sklearn.model_selection import KFold

upper = 1
lower = 0
Num = 100*1
gVar = 0.1
split_num = 4
scalar=1
step = (upper-lower)/(Num)

x = np.arange(lower+step/2,upper+step/2,step)
ntr = np.random.normal(0,gVar,Num-25*scalar)
nte = np.random.normal(0,gVar,Num-75*scalar)
m_num = 10

kf = KFold(split_num)
```

```
kf.get_n_splits(x)
print(kf)
E_train=np.zeros(m_num)
E_test=np.zeros(m_num)
M1=np.zeros(m_num)
E=np.zeros(m_num)

for train_index, test_index in kf.split(x):
    x_train, x_test = x[train_index], x[test_index]



#print(x_train)
#print(x_test)

for M in range(0,m_num):

    X_train = np.array([x_train**m for m in range(M+1)]).T
    w = np.linalg.inv(X_train.T@X_train)@X_train.T@ntr

    X_tr=np.array([x_train**m for m in range(M+1)]).T
    X_te=np.array([x_test**m for m in range(M+1)]).T

    y_test=X_te@w
    y_train=X_tr@w

    y_trn=np.sin(2*math.pi*x_train)+ntr
    y_ten=np.sin(2*math.pi*x_test)+nte
```

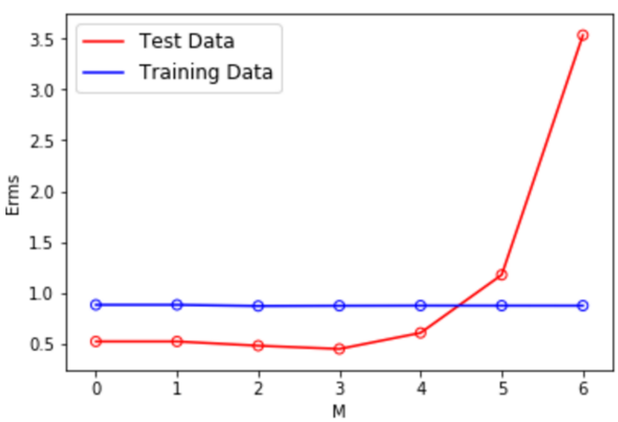
```
    y1=np.sin(2*math.pi*x_test)
    y2=np.sin(2*math.pi*x_train)

    e_1=(y_train-y_trn)**2
    E_1=(sum(e_1)/50)**0.5
    E_train[M]=E_1

    e_2=(y_test-y_ten)**2
    E_2=(sum(e_2)/50)**0.5
    E_test[M]=E_2

    M1[M]=M
#print(E_train)
p1 = plt.plot(M1,E_test,'r')
p2 = plt.plot(M1,E_train,'b')
plt.scatter(M1,E_test,facecolors='none',edgecolors='r')
plt.scatter(M1,E_train,facecolors='none',edgecolors='b')
plt.ylabel('Erms') #label x and y axes
plt.xlabel('M')
plt.legend((p1[0],p2[0]),('Test Data', 'Training Data'), fontsize=12)
```

Picture: (m=3 according to the picture)



Discussion:

In this example I can take the more training data and different test validation data set to form different shape, and comparing different types of line chart, we can find a proper number.

Q2:

1. XY size: $M \times N$
2. YX size: not defined
3. YX^T size: $N \times M$
4. aX size: not defined
5. a^TX size: $1 \times N$
6. aX^T size: not defined
7. a^Tb size: not defined
8. b^Tb size: is constant
9. bb^T size: $N \times N$
10. $sX+Y$ size: not defined

Q3:

Suppose the size of X is a*b, and because of definition of XH, the size of XH is b*a.

In order to make the computation meaningful, the size of v should be a*1 and the size of u should be b*1.

Then:

$X^HXv = \lambda v$: a*b @ b*a @ a*1 = a*1

$X^HXu = \lambda u$: b*a @ a*b @ a*1 = b*1

I found here " b*a @ a*1 = b*1 ", so I try to define " $X^Hv=n$ ".

Afterwards:

$X^HX (X^Hv) = X^H \lambda v = \lambda (X^Hv)$

$X^H X n \quad = \quad \lambda n$

Therefore, they have the same eigenvalues but different eigenvectors.

Q4:

1. $\frac{\partial f}{\partial x} = 3x^T + 3x^T + 4y^T = 6x^T + 4y^T$

2. $\frac{\partial^2 f}{\partial x^2} = 6$

Q5:

1. $\frac{\partial f}{\partial x} = - 10(QX)^T - 10X^TQ + 4y^T = -10\,(X^TQ^T + X^TQ) + 4y^T =$
$-10X^T(Q^T + Q) + 4y^T$

2. $\frac{\partial^2 f}{\partial x^2} = -10\,(Q^T + Q)^T = -10Q^T - 10Q$