

## Code for Question 2:

```
clc;
clear all;
close all;

% Open the data files
load x_train.mat;
load y_train.mat;
load x_test.mat;
load y_test.mat;

% Plot the scatter plot
figure(1)
scatter(x_train,y_train);
title('Scatter plot between x train and y train')
xlabel('x train')
ylabel('y train')

% Size of data points of training set
row = ((size(x_train,1)));
col = ((size(x_train,2)));

% Augment the training data and make polynomials of degree [1, 2, 3, 7, 10]
ON = ones(row,col);

X_1dim = [ON x_train];
X_2dim = [ON x_train x_train.^2];
X_3dim = [ON x_train x_train.^2 x_train.^3];
X_7dim = [ON x_train x_train.^2 x_train.^3 x_train.^4 x_train.^5 x_train.^6 x_train.^7];
X_10dim = [ON x_train x_train.^2 x_train.^3 x_train.^4 x_train.^5 x_train.^6 x_train.^7 x_train.^8 x_train.^9
x_train.^10];

% Make the Weight Vectors

Weight_1dim = pinv(X_1dim'*X_1dim)*X_1dim'*y_train;
Weight_2dim = pinv(X_2dim'*X_2dim)*X_2dim'*y_train;
Weight_3dim = pinv(X_3dim'*X_3dim)*X_3dim'*y_train;
Weight_7dim = pinv(X_7dim'*X_7dim)*X_7dim'*y_train;
Weight_10dim = pinv(X_10dim'*X_10dim)*X_10dim'*y_train;

% Define the Mean Square Error Variables
MSE_1 = 0;
MSE_2 = 0;
MSE_3 = 0;
MSE_4 = 0;
MSE_5 = 0;

% Compute the Mean Square Error

for i = 1 : row
```

```

MSE_1 = MSE_1 + ((y_train(i) - Weight_1dim*X_1dim(i,:)))^2;
MSE_2 = MSE_2 + ((y_train(i) - Weight_2dim*X_2dim(i,:)))^2;
MSE_3 = MSE_3 + ((y_train(i) - Weight_3dim*X_3dim(i,:)))^2;
MSE_4 = MSE_4 + ((y_train(i) - Weight_7dim*X_7dim(i,:)))^2;
MSE_5 = MSE_5 + ((y_train(i) - Weight_10dim*X_10dim(i,:)))^2;

```

end

```

MSE_1 = MSE_1/25;
MSE_2 = MSE_2/25;
MSE_3 = MSE_3/25;
MSE_4 = MSE_4/25;
MSE_5 = MSE_5/25;

```

% Plot the graph of MSE vs Polynomial Degree

```

deg = [1;2;3;7;10];
MSE = [MSE_1;MSE_2;MSE_3;MSE_4;MSE_5];
figure(2)
plot(deg,MSE,'-o')
title('Plot of MSE vs polynomial degree for training sample')
xlabel('Polynomial degree')
ylabel('Mean Square Error value')
txt1 = num2str(MSE_1);
text(deg(1),MSE(1),txt1);
txt2 = num2str(MSE_2);
text(deg(2),MSE(2),txt2);
txt3 = num2str(MSE_3);
text(deg(3),MSE(3),txt3);
txt4 = num2str(MSE_4);
text(deg(4),MSE(4),txt4);
txt5 = num2str(MSE_5);
text(deg(5),MSE(5),txt5);

```

% Size of data points of test set

```

row1 = ((size(x_test,1)));
col1 = ((size(x_test,2)));

```

% Augment the test data and make polynomials of degree [1, 2, 3, 7, 10]

```

ON1 = ones(row1,col1);

```

```

XT_1dim = [ON1 x_test];
XT_2dim = [ON1 x_test x_test.^2];
XT_3dim = [ON1 x_test x_test.^2 x_test.^3];
XT_7dim = [ON1 x_test x_test.^2 x_test.^3 x_test.^4 x_test.^5 x_test.^6 x_test.^7];
XT_10dim = [ON1 x_test x_test.^2 x_test.^3 x_test.^4 x_test.^5 x_test.^6 x_test.^7 x_test.^8 x_test.^9 x_test.^10];

```

% Define the Mean Square Error Variables

```

MSE_t1 = 0;
MSE_t2 = 0;
MSE_t3 = 0;
MSE_t4 = 0;

```

```

MSE_t5 = 0;

% Compute the Mean Square Error

for i = 1 : row1

MSE_t1 = MSE_t1 + ((y_test(i) - Weight_1dim'*XT_1dim(i,:)))^2;
MSE_t2 = MSE_t2 + ((y_test(i) - Weight_2dim'*XT_2dim(i,:)))^2;
MSE_t3 = MSE_t3 + ((y_test(i) - Weight_3dim'*XT_3dim(i,:)))^2;
MSE_t4 = MSE_t4 + ((y_test(i) - Weight_7dim'*XT_7dim(i,:)))^2;
MSE_t5 = MSE_t5 + ((y_test(i) - Weight_10dim'*XT_10dim(i,:)))^2;

end

MSE_t1 = MSE_t1/25;
MSE_t2 = MSE_t2/25;
MSE_t3 = MSE_t3/25;
MSE_t4 = MSE_t4/25;
MSE_t5 = MSE_t5/25;

% Plot the graph of MSE vs Polynomial Degree

deg = [1;2;3;7;10];
MSE = [MSE_t1;MSE_t2;MSE_t3;MSE_t4;MSE_t5];
figure(3)
plot(deg,MSE,'-o')
title('Plot of MSE vs polynomial degree for test sample')
xlabel('Polynomial degree')
ylabel('Mean Square Error value')
txt1 = num2str(MSE_t1);
text(deg(1),MSE(1),txt1);
txt2 = num2str(MSE_t2);
text(deg(2),MSE(2),txt2);
txt3 = num2str(MSE_t3);
text(deg(3),MSE(3),txt3);
txt4 = num2str(MSE_t4);
text(deg(4),MSE(4),txt4);
txt5 = num2str(MSE_t5);
text(deg(5),MSE(5),txt5);

% Ridge Regression

% Values of lambda
lambda = [1e-5, 1e-3, 0.1, 1, 10];

% Size of data points of training set
row = ((size(x_train,1)));
col = ((size(x_train,2)));

% Augment the training data and make polynomial of degree 7
ON = ones(row,col);

X_7dim = [ON x_train x_train.^2 x_train.^3 x_train.^4 x_train.^5 x_train.^6 x_train.^7];

```

% Define the MSE variable

MSE = zeros(size(lambda));

% Compute the Mean Square Error

```
for j = 1:size(lambda,2)
    for i = 1:row
        weight = pinv(lambda(j)*eye(8) + X_7dim'*X_7dim)*X_7dim'*y_train;
        MSE(j) = MSE(j) + ((y_train(i) - weight'*X_7dim(i,:'))^2 + lambda(j)*(weight'*weight))/25;
    end
    weight;
end
```

MSE = MSE/25;

% Plot the curve for training dataset error

```
figure(4)
loglog(lambda, MSE)
title('Plot of MSE vs log(lambda)')
xlabel('lambda')
ylabel('Mean Square Error value')
```

```
figure(5)
lambdas = log10(lambda);
scatter(lambdas,MSE)
hold on;
plot(lambdas,MSE)
title('Plot of MSE vs log(lambda)')
xlabel('Log(lambda)')
ylabel('Mean Square Error value')
```

% Size of data points of test set

row1 = ((size(x\_test,1)));

col1 = ((size(x\_test,2)));

% Augment the test data and make polynomials of degree [1, 2, 3, 7, 10]

ON1 = ones(row1,col1);

XT\_7dim = [ON1 x\_test x\_test.^2 x\_test.^3 x\_test.^4 x\_test.^5 x\_test.^6 x\_test.^7];

% Define the MSE variable

MSE = zeros(size(lambda));

% Compute the Mean Square Error

```
for j = 1:size(lambda,2)
    for i = 1:row1
        weight = pinv(lambda(j)*eye(8) + XT_7dim'*XT_7dim)*XT_7dim'*y_train;
        MSE(j) = MSE(j) + ((y_test(i) - weight'*XT_7dim(i,:'))^2 + lambda(j)*(weight'*weight))/25;
    end
end
```

```
end  
weight;  
end
```

```
MSE = MSE/48
```

```
% Plot the curve for training dataset error
```

```
figure(6)  
loglog(lambda, MSE)  
title('Plot of MSE vs log(lambda)')  
xlabel('lambda')  
ylabel('Mean Square Error value')
```

```
figure(7)  
lambdas = log10(lambda);  
scatter(lambdas, MSE)  
hold on;  
plot(lambdas, MSE)  
title('Plot of MSE vs log(lambda)')  
xlabel('Log(lambda)')  
ylabel('Mean Square Error value')
```