ChE 694: Multivariate Data Analysis Term Project

Course Instructor

Prof. Vinay Prasad

Nirav Raiyani
St ID: 1557017

Contents

List of Figures

List of Tables

1 Introduction

1.1 Near Infrared Reflectance (NIR) Spectroscopy data

The data set consist of Near Infrared Spectroscopy of 108 soil samples collected from 36 different plot exposed to different climate conditions.Three samples taken from each site consist of two samples from 0 -5 cm depth and 1 sample are from 5 - 10 cm depth. The aim is to predict soil organic matter and fungal biomass using rapid and inexpensive technique of NIR Spectroscopy compared to other conventional methods. The data analytics tools are used to process NIR spectroscopy data and to develop the relationship between these data and soil chemical and microbiological properties.

1.2 Methods

The Spectroscopy data set contains 1050 variables and 108 observations. PCA was used to reduce the number of variables. Linear regression and PLS(Partial Least Squares) was performed between reduced number of regressors and observed data. The Hierarchical and Non-Hierarchical clustering technique were used to separate two different set of samples. The PCR and PLS were performed on the clustered data sets to see if there is any improvement of fit compared to the unclusterred data. The results were justified by comparing the values of coefficient of determination(R^2). In each regression analysis, 70 percent of the total data was used to train the model and remaining 30 percent data was used to test the model.

2 Pre-processing of data

The data in its raw form can not be used directly into data analysis. The pre-processing of the data is necessary to insure that all the variables are in the same range. To achieve that the training data was standardized around to its mean and variance. The test data set was standardized with respect to mean and variance of training data.

3 Principle Component Analysis

Principal component analysis (PCA) is a method of exploring the directions of maximum variability in th regressor space. These new directions (which are hidden) are extracted to potentially reduce the dimensionality of the data and predict outliers. These directions are called "Principal Components". The data is then projected onto these directions to obtain scores.Principal components can then be used to perform regression which is known as "Principle component Regression"(PCR). PCA is sensitive to the scaling of regressors hence standardization of data is highly recommended. Spectroscopy data set had 1050 variables, which, after performing PCA, were reduced to just four. As shown in figure 1, first four principle components explains almost 98 percent of the total variance.

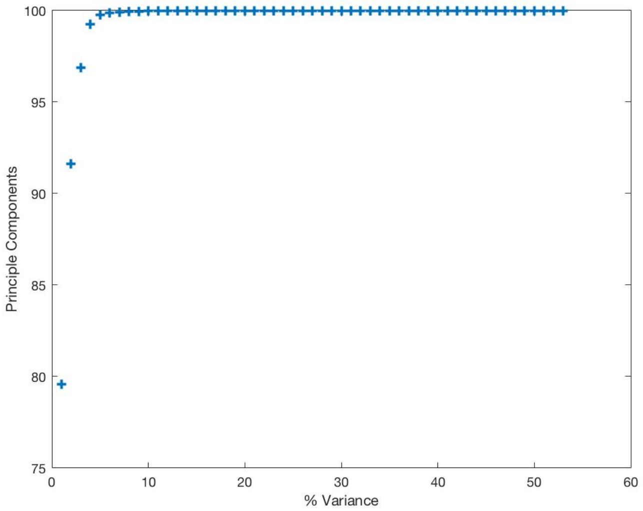

Figure 1: Scree plot.

	Y_1	Y_2
$R^2_{training}$	0.8138	0.7505
R^2_{test}	0.7796	0.1489

Table 1: PCR: R^2.

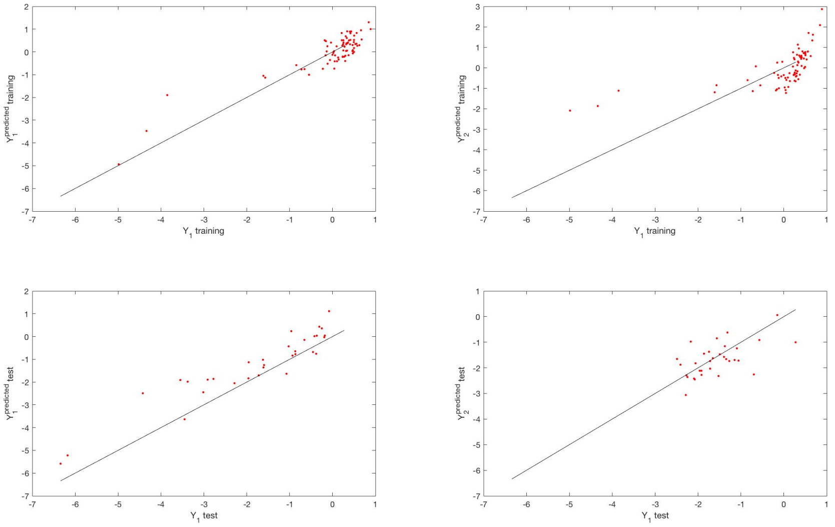

Figure 2: PCR: predicted vs actual response plot .

Table 1 gives the values of the coefficient of determination for training and test data set. It is evident that performance of regression model is very poor in predicting Y_2 for test data set. Figure 2 shows the predicted vs actual response plot for training and test data set for both variables.

4 Partial Least Squares (PLS)

Partial least squares regression has similarities to principal components regression, it finds a linear regression model by transforming the predicted variables and the responses to a new space. PLS explores the relations between X and Y. A PLS model will try to find direction in the X space that explain the maximum variance direction in the Y space. PLS regression is particularly suited when there is multicollinearity among predictors. Multicollonearity is a special case where regressors are correlated with each other. In such case, standard regression will generally speaking fail. PLS is also known as **Projection to latent structure**.

	Y_1	Y_2
$R^2_{training}$	0.9615	0.9510
R^2_{test}	0.8684	0.4488

Table 2: PLS: R^2.

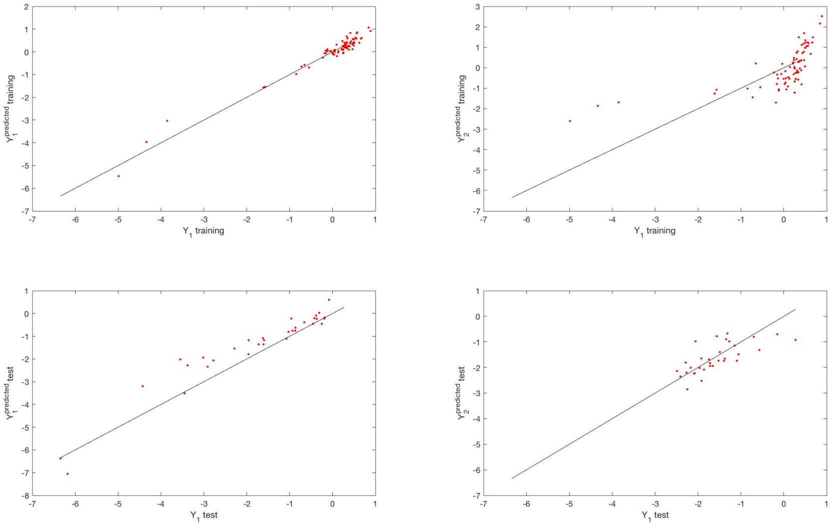

Figure 3: PCR: Predicted vs actual response plot .

The R^2 values for PLS regression are given in table 2 and figure 3 shows corresponding predicted vs actual response plot. The results shows that regression model trained by PLS have improved the results significantly compared to PCA.

5 Clustering Analysis

Clustering is the exercise of grouping objects on basis of their similarities. Objects which are deemed to be similar on the basis of defined similarity measures are grouped into clusters, and most often clustering is not a means to an end but an intermediate step to look for similarities that are not apparent. Broadly clustering is categorized as: hierarchical and non-hierarchical.

5.1 Hierarchical Clustering

Hierarchical Clustering groups objects on a measure of similarity at levels (hierarchies). This method gives a choice of number of clusters desired and the results are generally presented on a dendrogram. Different linkage measures used are: Single, Average and Complete.

5.1.1 Single Linkage

Distance is defined as the smallest separation among clusters (or objects), and this measure tries to pull clusters apart.

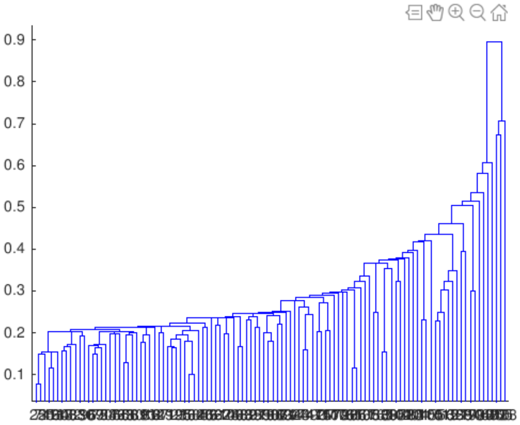

Figure 4: Hierarchical Clustering: Single Linkage dendrogram.

5.1.2 Complete Linkage

Distance is defined as the largest separation among clusters, and quite often complete linkage reults in concurrent clusters.

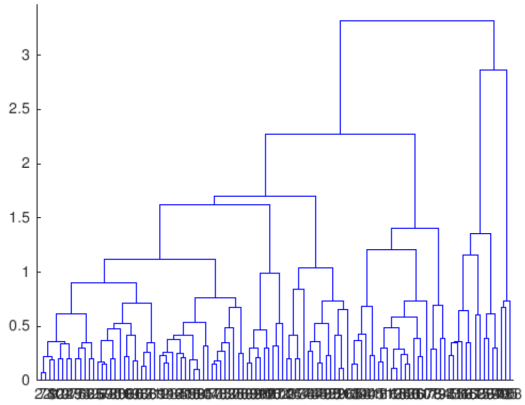

Figure 5: Hierarchical Clustering: Complete Linkage dendrogram.

5.1.3 Average Linkage

A measure of distance which estimates the averaged separation between clusters.

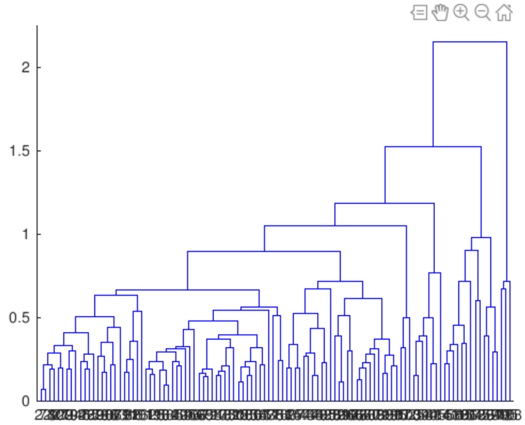

Figure 6: Hierarchical Clustering: Average Linkage dendrogram.

Figure **??**f4) , 5 and 6 shows the deprograms for all three class of hierarchical clustering for soil data set.

5.2 Non-hierarchical

Non hierarchical approach(specifically k-means discussed here) starts out with a randomly made partition in the data set and makes reassignments based on similarity measures. This iterative procedure is stopped when no further reassignments happen in a complete cycle. In addition, It was observed that the method gives different results based on the initial guess. k-means clustering was performed on the data and two clusters of size 74 and 34 were obtained. This reaffirms the prior knowledge about existence of two clusters(soil samples from two different depths 0-5cm and 5-10 cm). Figure 7 demonstrates the 4 principle components plotted against each other for class 1 and class 2.

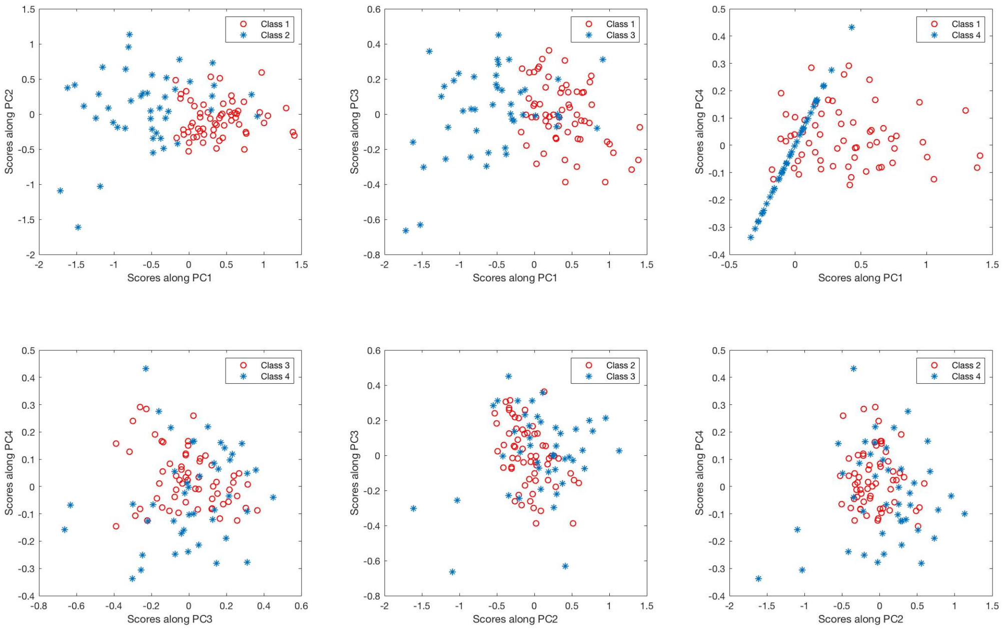

Figure 7: Principle components for class 1 and class 2

6 PCR after clustering

The basic idea behind this section was to check that after separating the data set into two different classes does the regression analysis performed on each class improves the results? In the two classes, Class 1 has 74 observations and class 2 contains 34 variables.

6.0.1 Class 1

	Y_1	Y_2
$R^2_{training}$	0.7142	0.7328
R^2_{test}	0.7523	0.6112

Table 3: PCR Class i: R^2 .

6.0.2 Class 2

	Y_1	Y_2
$R^2_{training}$	0.9548	0.8838
R^2_{test}	0.8549	0.8156

Table 4: PCR Class ii: R^2 .

The values of the coefficient of determination (R^2) for class 1 and class 2 are shown in Table 3 and 4. In this case, as earlier, the first four principle component were taken as regressors and 70 percent of the total data were used to train the model.

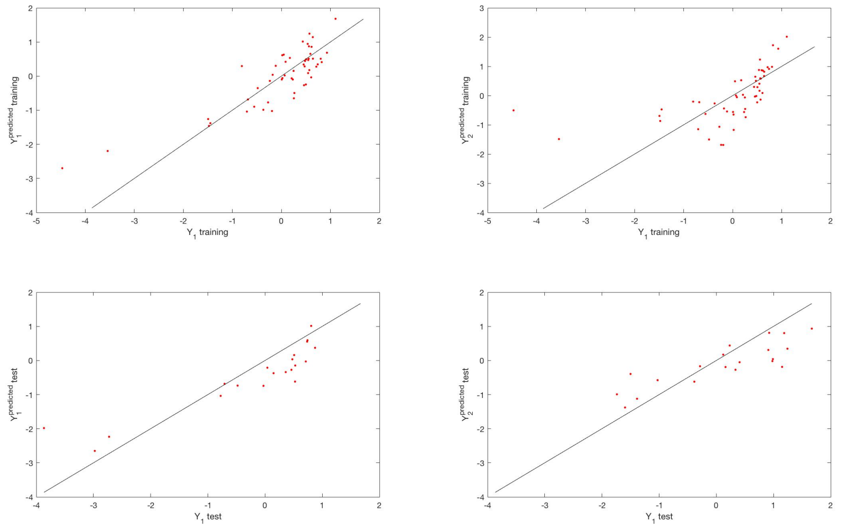

Figure 8: PCR Class i: Predicted vs actual response plot .

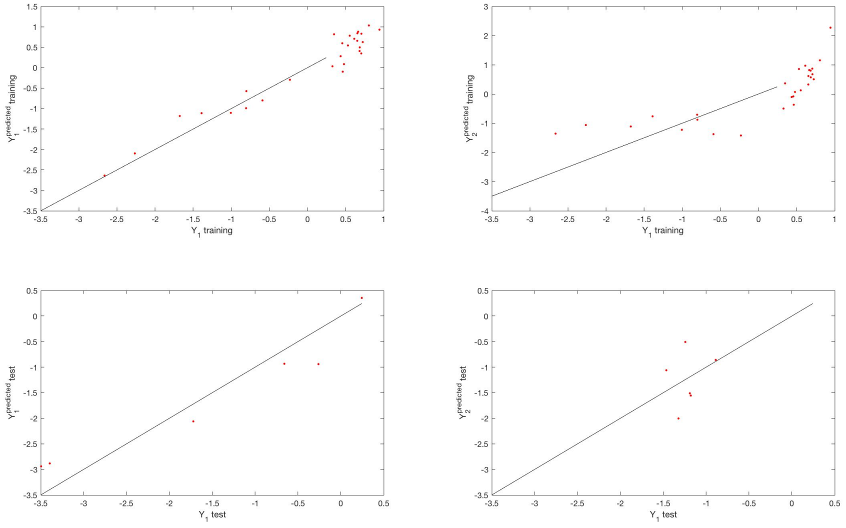

Figure 9: PCR Class ii: Predicted vs actual response plot .

6.1 PLS after clustering

6.1.1 Class 1

	Y_1	Y_2
$R^2_{training}$	0.9443	0.8784
R^2_{test}	0.9143	0.8285

Table 5: PLS Class i: R^2 .

6.1.2 Class 2

	Y_1	Y_2
$R^2_{training}$	0.9600	0.8900
R^2_{test}	0.8707	0.8307

Table 6: PLS Class ii: R^2 .

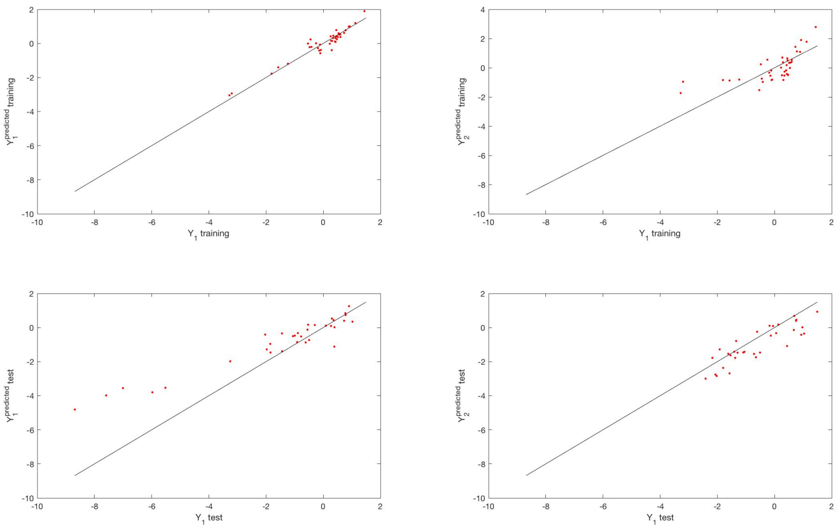

Figure 10: PLS Class i: Predicted vs actual response plot .

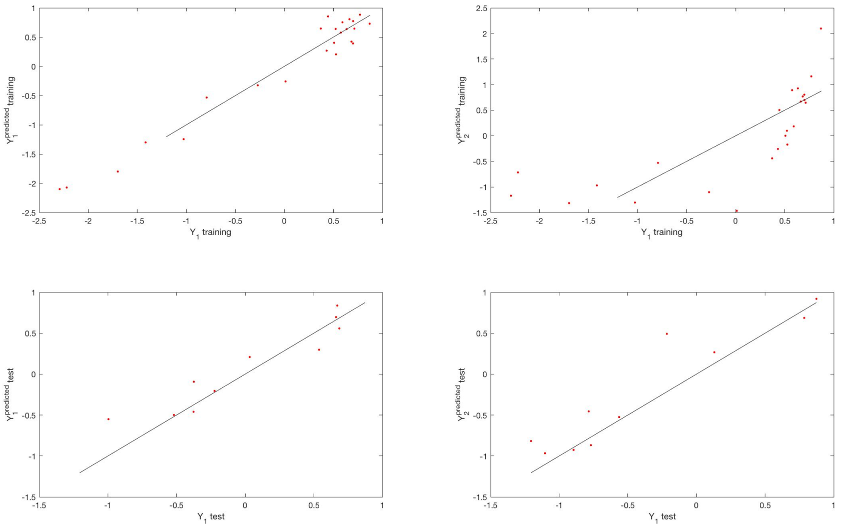

Figure 11: PLS Class ii: Predicted vs actual response plot .

Figure 10, 11 and table 5, 6 gives the results obtained after performing PLS regression on class i and class ii.

7 summary

The principle component analysis were performed on the NIR spectroscopy data set to reduce the dimensions. The PCR and PLS regression was performed on unclusterrd data which yielded poor results, particularly on soil organic matter. The coefficient of determination was very low for both aforementioned methods. On the basis of prior knowledge that the data set has the observation of two different depths, non-hierarchical clustering was used to separate them. Regression analysis was performed on both classes of separated data using the aforementioned techniques. As expected, the new model shown significant improvement of the fit. Furthermore, the number of observation were not sufficient to train a robust model as the significant change in final results was observed for a small change in training data set.

MATLAB CODES

1.PCA-PCR(Unclustered)

```
clc;
 clear all
 close all;
 %loading data

 %Responce variables
 Y_actual = xlsread('Y.xlsx');
 %Regression variables
 X_actual = xlsread('X.xlsx')';
 V1 = cond(X_actual);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 T =108;
 n= 76;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :1050
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);

 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PCA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EXTRA CALCUALTION FOR VALIDATION OF THE DATA FROM PCA FUNCTION
%cov_X= cov(X_N);
%[eg ev] = eig(cov_X);
%temp = diag(ev);
%sortv = sort(temp);
%PCA of regressors
```

```
[coeff,score,latent,tsquared,explained,mu] = pca(X_Nt);
%scatter(score(:,1), score(:,2),score(:,3))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Scree Plot%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sum = 0;
for i = 1:53
  sum = sum + explained(i,1);
  scree(i) = sum;
end
plot(scree,'+','LineWidth',2)
%First 4 principle components explains almost 99% variance of regressors
%data set
%So new set of regressors is the first four column of score matrix
X = score(:,1:4);
V3 = cond(X);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PCR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Here,the first 72 observations are used to train the model
Xr1=[ones(n,1) X]
[beta,Sigma,E,CovB,logL] = mvregress(Xr1,Y_Nt);
Y_cap_training = Xr1*beta;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)))
R2_TRAINING1 =1- ((E(:,1)'*E(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)))
R2_TRAINING2 = 1-((E(:,2)'*E(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining observations%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%determining the prinnciple score of test regressors from coefficients of
%training data
X2 = X_Nr*coeff(:,1:4);
Xr2 = [ones(T-n,1) X2]
Y_cap_test = Xr2*beta;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
%As we can see that R2 of the second observation variable for test data set is
%very poor so it is expected that there is a presence of 2 clusters whose trend
%can be astimeted by two diffrent sets of regression coefficients.
```

2. PLS

```
clc;
 clear all
 close all;
 %loading data

 %Responce variables
 Y_actual = xlsread('Y.xlsx');
 %Regression variables
 X_actual = xlsread('X.xlsx')';
 V1 = cond(X_actual);
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 T =108;
 n= 76;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :1050
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);

 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%% PLS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[XL,YL,XS,YS,BETA,PCTVAR,MSE,stats] = plsregress(X_Nt,Y_Nt,15);
Y_cap_training = [ones(n,1)  X_Nt]*BETA;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)))
R2_TRAINING1 =1- ((E_training(:,1)'*E_training(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)))
R2_TRAINING2 = 1-((E_training(:,2)'*E_training(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining observations%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xt = [ones((T-n),1) X_Nr];
```

```
Y_cap_test = Xt*BETA;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
```

3. Clustering

3.1 Hierarchical

```
 clc;

 clear all
 close all;
 %loading data

 %Responce variables
 Y_actual = xlsread('Y.xlsx');
 %Regression variables
 X_actual = xlsread('X.xlsx')';
 V1 = cond(X_actual);
 %Normalizing the data
X_N= normalize(X_actual);
Y_N= normalize(Y_actual);
V2 = cond(X_actual);
%PCA of regressors
[coeff,score,latent,tsquared,explained,mu] = pca(X_N);
%%%%%%%Single Linkage%%%%%%%%%%
Z1 = linkage(X_actual,'single');
figure(1)
dendrogram(Z1,0)
%%%%%%%Complete Linkage%%%%%%%%%%
Z2 = linkage(X_actual,'complete');
figure(2)
dendrogram(Z2,0)
%%%%%%%Average Linkage%%%%%%%%%%
Z3 = linkage(X_actual,'average');
figure(3)
dendrogram(Z3,0)
```

3.2 Nonhierarchical

```
 clc;

 clear all;
 close all;
 %loading data
 %Responce variables
 Y_actual = xlsread('Y.xlsx');
 %Regression variables
 X_actual = xlsread('X.xlsx')';
 V1 = cond(X_actual);

%Clustering
idx = kmeans(X_actual,2);
[coeff,score,latent,tsquared,explained,mu] = pca(X_actual);
[a b]=size(score);
[r c] = size(X_actual);
j=1;
k=1;
for i = 1:r
    if idx(i) == 1
       classS1(j,:) = score(i,:);
       classX1(j,:) = X_actual(i,:);
       classY1(j,:) = Y_actual(i,:);
        j=j+1;
    else
        classS2(k,:) = score(i,:);
        classX2(k,:) = X_actual(i,:);
        classY2(k,:) = Y_actual(i,:);
        k=k+1;
    end
end
PCS = [classS1 ; classS2];
disp(size(classX1))
disp(size(classX2))
subplot(2,3,1)
    plot(PCS(1:61,1),PCS(1:61,2),'or',PCS(65:108,1),PCS(65:108,2),'*')
    xlabel('Scores along PC1')
    ylabel('Scores along PC2')

    legend('Class 1','Class 2')
subplot(2,3,2)
    plot(PCS(1:61,1),PCS(1:61,3),'or',PCS(65:108,1),PCS(65:108,3),'*')
    xlabel('Scores along PC1')
    ylabel('Scores along PC3')

    legend('Class 1','Class 3')
subplot(2,3,3)
    plot(PCS(1:61,1),PCS(1:61,4),'or',PCS(65:108,4),PCS(65:108,4),'*')
    xlabel('Scores along PC1')
    ylabel('Scores along PC4')

    legend('Class 1','Class 4')
subplot(2,3,4)
    plot(PCS(1:61,3),PCS(1:61,4),'or',PCS(65:108,3),PCS(65:108,4),'*')
    xlabel('Scores along PC3')
    ylabel('Scores along PC4')

    legend('Class 3','Class 4')
subplot(2,3,5)
    plot(PCS(1:61,2),PCS(1:61,3),'or',PCS(65:108,2),PCS(65:108,3),'*')
    xlabel('Scores along PC2')
    ylabel('Scores along PC3')

    legend('Class 2','Class 3')
subplot(2,3,6)
    plot(PCS(1:61,2),PCS(1:61,4),'or',PCS(65:108,2),PCS(65:108,4),'*')
    xlabel('Scores along PC2')
    ylabel('Scores along PC4')

    legend('Class 2','Class 4')
```

4. Clustering PCR

4.1 Cluster – i

```
clc;

clear all;
close all;
load ClassX1.mat;
load ClassY1.mat;
 %loading data
 %Responce variables
 Y_actual = classY1;
 %Regression variables
 X_actual = classX1;
 V1 = cond(X_actual);

 %prepration of training data set
 X_train = [X_actual(1:20,:) ; X_actual(28:47,:) ; X_actual(61:74,:);
X_actual(21:27,:) ; X_actual(48:59,:)];
 Y_train = [Y_actual(1:20,:) ; Y_actual(28:47,:) ; Y_actual(61:74,:);
Y_actual(21:27,:) ; Y_actual(48:59,:)];

 %prepration of test set
 %X_test = [X_actual(21:27,:) ; X_actual(48:54,:)];
 %Y_test = [Y_actual(21:27,:) ; Y_actual(48:54,:)];
 X_actual = X_train;
 Y_actual = Y_train;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 [T q] = size(X_actual);
 n= 53;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :q
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);

 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PCA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EXTRA CALCUALTION FOR VALIDATION OF THE DATA FROM PCA FUNCTION
%cov_X= cov(X_N);
%[eg ev] = eig(cov_X);
%temp = diag(ev);
%sortv = sort(temp);
%PCA of regressors
[coeff,score,latent,tsquared,explained,mu] = pca(X_Nt);
%scatter(score(:,1), score(:,2),score(:,3))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Scree Plot%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% sum = 0;
% for i = 1:43
%   sum = sum + explained(i,1);
%   scree(i) = sum;
% end
%plot(scree,'+','LineWidth',2)
%First 4 principle components explains almost 99% variance of regressors
%data set
%So new set of regressors is the first four column of score matrix
X = score(:,1:4);
V3 = cond(X);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PCR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Here,the first 72 observations are used to train the model
Xr1=[ones(n,1) X]
[beta,Sigma,E,CovB,logL] = mvregress(Xr1,Y_Nt);
Y_cap_training = Xr1*beta;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
```

```
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)))
R2_TRAINING1 =1- ((E(:,1)'*E(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)))
R2_TRAINING2 = 1-((E(:,2)'*E(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining 36 observations%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%determining the prinnciple score of test regressors from coefficients of
%training data
X2 = X_Nr*coeff(:,1:4);
Xr2 = [ones(T-n,1) X2]
Y_cap_test = Xr2*beta;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
```

5. 2 Class – ii

```
_clc;

clear all;
close all;
load ClassX2.mat;
load ClassY2.mat;
 %loading data
 %Responce variables
 Y_actual = classY2;
 %Regression variables
 X_actual = classX2;
 V1 = cond(X_actual);



  %prepration of training data set
 X_train = [X_actual(1:5,:) ; X_actual(15:20,:) ; X_actual(26:28,:);
X_actual(6:14,:) ; X_actual(21:25,:)];
 Y_train = [Y_actual(1:5,:) ; Y_actual(15:20,:) ; Y_actual(26:28,:);
Y_actual(6:14,:) ; Y_actual(21:25,:)];
 X_actual = X_train;
 Y_actual = Y_train;

 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 [T q] = size(X_actual);
 n= 20;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :q
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);

 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PCA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%EXTRA CALCUALTION FOR VALIDATION OF THE DATA FROM PCA FUNCTION
%cov_X= cov(X_N);
%[eg ev] = eig(cov_X);
%temp = diag(ev);
%sortv = sort(temp);
%PCA of regressors
[coeff,score,latent,tsquared,explained,mu] = pca(X_Nt);
%scatter(score(:,1), score(:,2),score(:,3))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Scree Plot%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sum = 0;
for i = 1:19
  sum = sum + explained(i,1);
  scree(i) = sum;
end
plot(scree,'+','LineWidth',2)
%First 4 principle components explains almost 99% variance of regressors
%data set
%So new set of regressors is the first four column of score matrix
X = score(:,1:4);
V3 = cond(X);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%PCR%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Here,the first 72 observations are used to train the model
Xr1=[ones(n,1) X]
[beta,Sigma,E,CovB,logL] = mvregress(Xr1,Y_Nt);
Y_cap_training = Xr1*beta;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
```

```
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)))
R2_TRAINING1 =1- ((E(:,1)'*E(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)))
R2_TRAINING2 = 1-((E(:,2)'*E(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining 36 observations%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%determining the prinnciple score of test regressors from coefficients of
%training data
X2 = X_Nr*coeff(:,1:4);
Xr2 = [ones(T-n,1) X2]
Y_cap_test = Xr2*beta;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
```

6. Clustering PLS

6.1 Class- i

```
_clc;

clear all;
close all;
load ClassX1.mat;
load ClassY1.mat;
 %loading data

 %Responce variables
 Y_actual = classY1;
 %Regression variables
 X_actual = classX1;
 V1 = cond(X_actual);
 [coeff,score,latent,tsquared,explained,mu] = pca(X_actual);
outlier = isoutlier(score(:,1:4))
 %prepration of training data set
 X_train = [X_actual(1:20,:) ; X_actual(28:47,:) ; X_actual(61:74,:);
X_actual(21:27,:) ; X_actual(48:59,:)];
 Y_train = [Y_actual(1:20,:) ; Y_actual(28:47,:) ; Y_actual(61:74,:);
Y_actual(21:27,:) ; Y_actual(48:59,:)];

 %prepration of test set
 %X_test = [X_actual(21:27,:) ; X_actual(48:54,:)];
 %Y_test = [Y_actual(21:27,:) ; Y_actual(48:54,:)];
 X_actual = X_train;
 Y_actual = Y_train;
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 [T q] = size(X_actual);
 n= 53;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :q
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);

 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%% PLS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[XL,YL,XS,YS,BETA,PCTVAR,MSE,stats] = plsregress(X_Nt,Y_Nt,10);
Y_cap_training = [ones(n,1)  X_Nt]*BETA;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)));
R2_TRAINING1 =1- ((E_training(:,1)'*E_training(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)));
R2_TRAINING2 = 1-((E_training(:,2)'*E_training(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining 36 observations%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xt = [ones((T-n),1) X_Nr];
Y_cap_test = Xt*BETA;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
```

6.2 Class – ii

```
-  clc;

clear all;
close all;
load ClassX2.mat;
load ClassY2.mat;
 %loading data

 %Responce variables
 Y_actual = classY2;
 %Regression variables
 X_actual = classX2;
 V1 = cond(X_actual);
 %prepration of training data set
  X_train = [X_actual(1:7,:) ; X_actual(15:21,:) ; X_actual(28:34,:);
X_actual(8:14,:) ; X_actual(22:27,:)];
  Y_train = [Y_actual(1:7,:) ; Y_actual(15:21,:) ; Y_actual(28:34,:);
Y_actual(8:14,:) ; Y_actual(22:27,:)];

%   %prepration of test set
%   X_test = [X_actual(21:27,:) ; X_actual(48:54,:)];
%   Y_test = [Y_actual(21:27,:) ; Y_actual(48:54,:)];
% %
  X_actual = X_train;
  Y_actual = Y_train;




 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%Normalizing the test and training data%%%%%%%%%%%%%%
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 %training data
 [T q] = size(X_actual);
 n= 24;
 X_Nt = normalize(X_actual(1:n,:));
 Y_Nt = normalize(Y_actual(1:n,:));

 X_mean = mean(X_actual(1:n,:))
 X_std = std(X_actual(1:n,:))

 Y_mean = mean(Y_actual(1:n,:))
 Y_std = std(Y_actual(1:n,:))
 %test data
 l=0;
 for i = 1 :q
 X_Nr(:,i) = (X_actual(n+1:end,i) - X_mean(1,i))/X_std(1,i);
 end
for j = 1:2
 Y_Nr(:,j) = (Y_actual(n+1:end,j) - Y_mean(1,j))/Y_std(1,j);
end
V2 = cond(X_actual);
%%%%%%%%%%%%%%%%%%%%%%%%%%% PLS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[XL,YL,XS,YS,BETA,PCTVAR,MSE,stats] = plsregress(X_Nt,Y_Nt,4);
Y_cap_training = [ones(n,1)  X_Nt]*BETA;
E_training = Y_Nt - Y_cap_training;
%Calcualring R2 for training data
D1=(Y_Nt(:,1)- mean(Y_Nt(:,1)))
R2_TRAINING1 =1- ((E_training(:,1)'*E_training(:,1))/(D1'*D1))
D2=(Y_Nt(:,2)- mean(Y_Nt(:,2)))
R2_TRAINING2 = 1-((E_training(:,2)'*E_training(:,2))/(D2'*D2))
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%Testing the model with the remaining 36 observations%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Xt = [ones((T-n),1) X_Nr];
Y_cap_test = Xt*BETA;
E_test = Y_Nr - Y_cap_test;
%Calcualring R2 for test data
Dt1=(Y_Nr(:,1)- mean(Y_Nr(:,1)));
R2_TEST1 =1-((E_test(:,1)'*E_test(:,1))/(Dt1'*Dt1))
Dt2=(Y_Nr(:,2)- mean(Y_Nr(:,2)));
R2_TEST2 = 1-((E_test(:,2)'*E_test(:,2))/(Dt2'*Dt2))
```