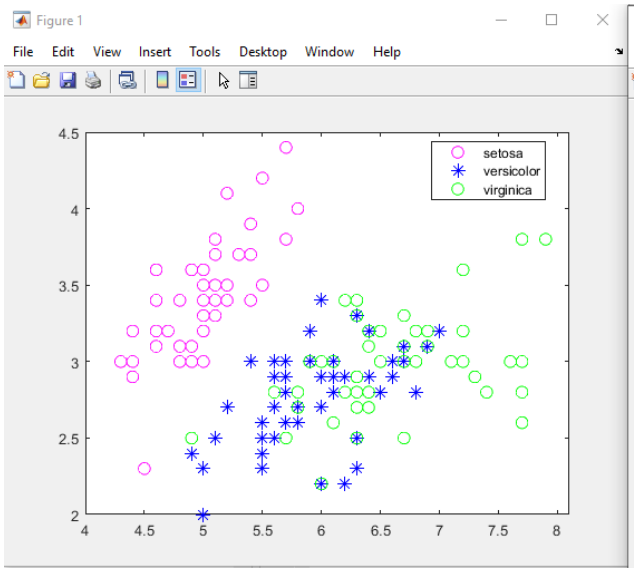
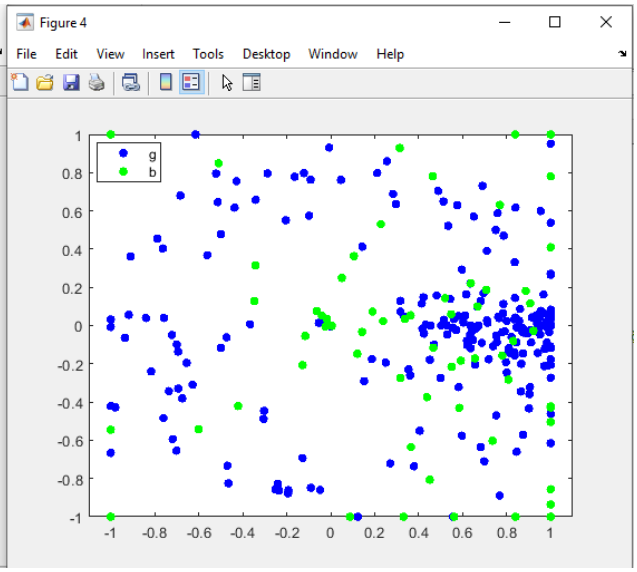
# EXPERIMENT 1

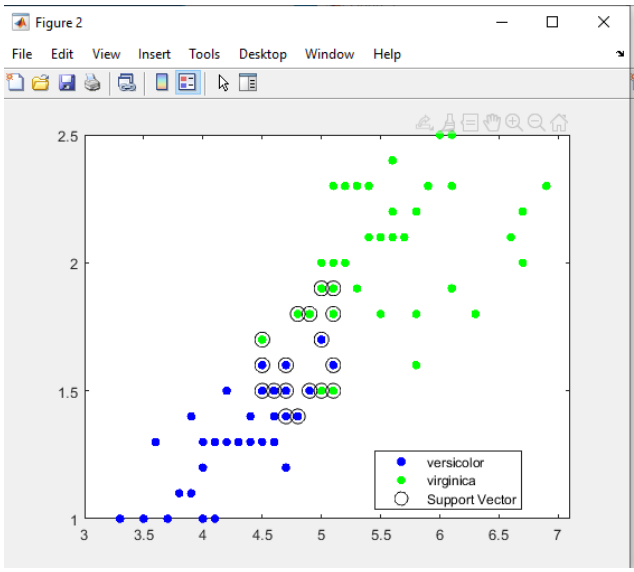
## OUTPUTS

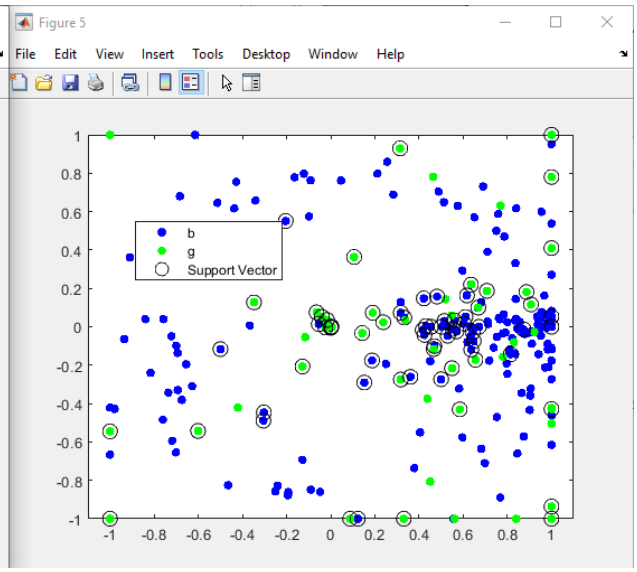
### Part 1: Visualize data



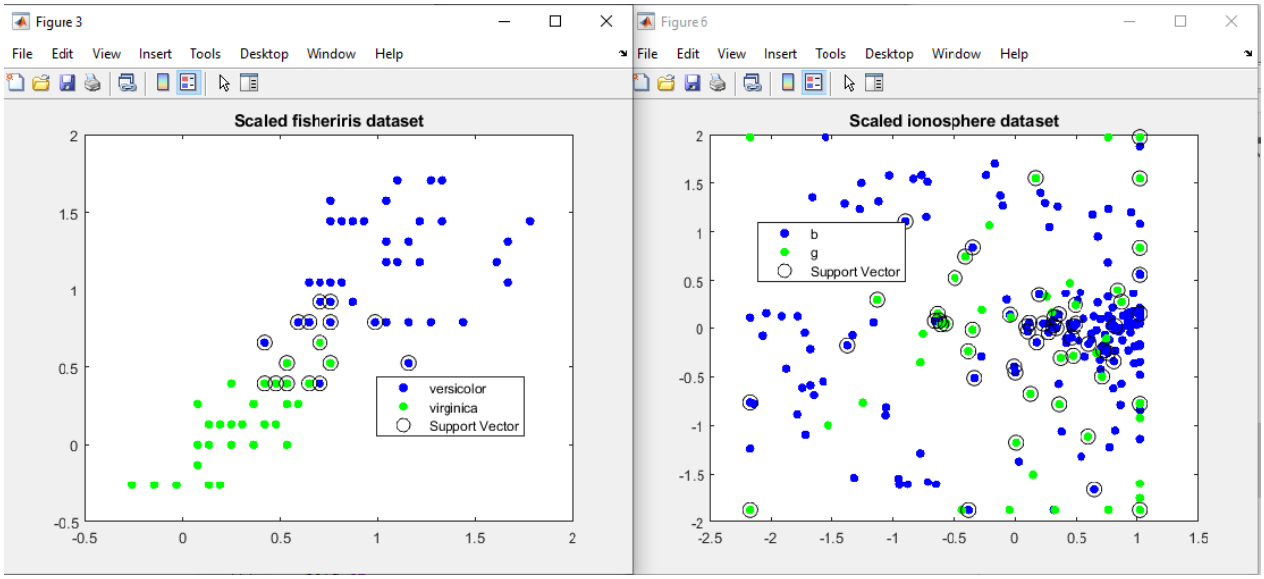




### Part 2:



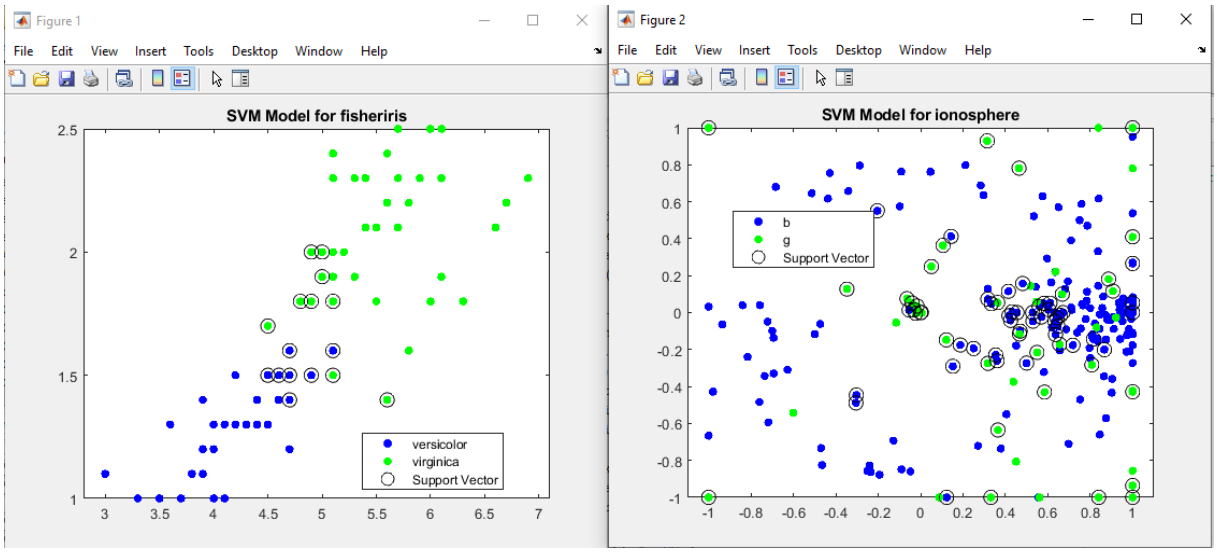



**Part 3:**

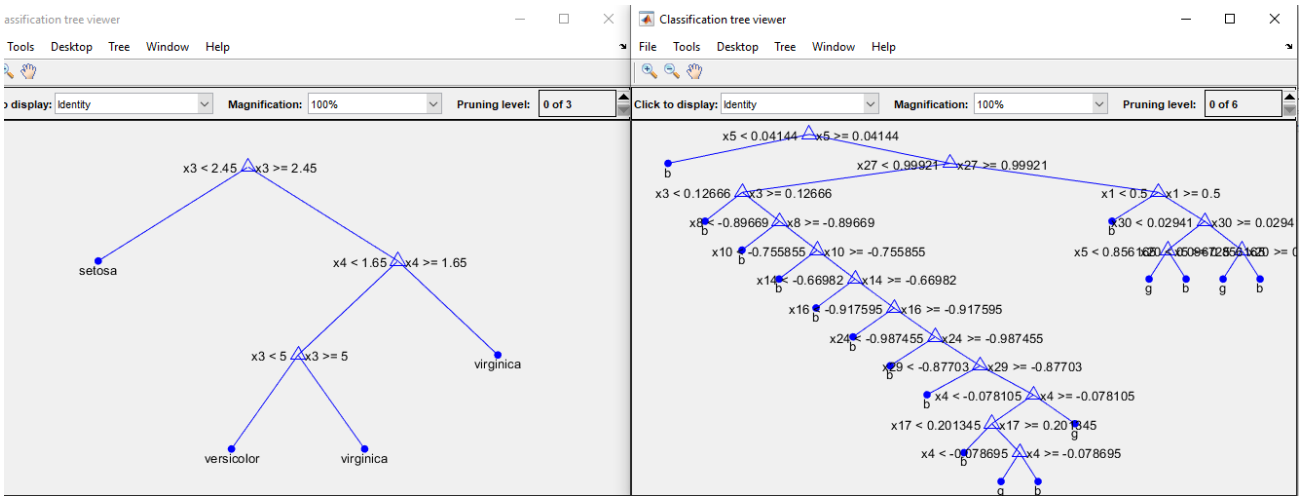



# EXPERIMENT 2

**Part 1: SVM Model**

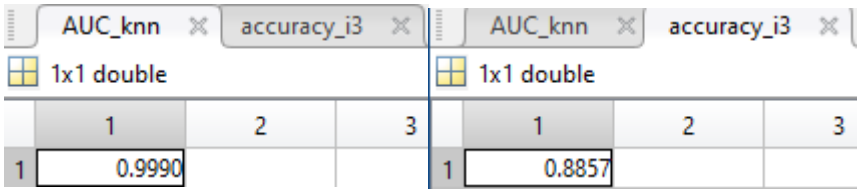


## Part 2: Decision Tree Model




## Part 3: KNN Methods

KNN does not create a model so we cannot observe. we can look at the success.



# EXPERIMENT 3

**Precision and recall values for fisheriris dataset:**

```
precision =
    0.9000

recall =
     1

precision =
    0.8750

recall =
     1

precision =
     1
```

```
recall =
     1

precision =
    0.8571

recall =
    0.8571

precision =
    0.8000

recall =
     1

precision =
     1
```

```
recall =
     1
```

**Precision and recall values for ionosphere dataset:**

```
precision_i =
    0.8800

recall_i =
    0.9565

precision_i =
    0.9048

recall_i =
    0.9048

precision_i =
    0.6111

recall_i =
     1
```

```
precision_i =
    0.4667

recall_i =
    0.7778

precision_i =
    0.5000

recall_i =
    0.8750

precision_i =
    0.5833

recall_i =
    0.8750
```

```
precision_dct =
    0.8846

recall_dct =
    0.7931

precision_dct =
    0.8095

recall_dct =
    0.7391

precision_dct =
    0.9444

recall_dct =
    0.6800
```

```
precision_dct =
    0.8125

recall_dct =
    0.8125

precision_dct =
    0.9286

recall_dct =
    0.7647

precision_dct =
     1

recall_dct =
    0.8571
```

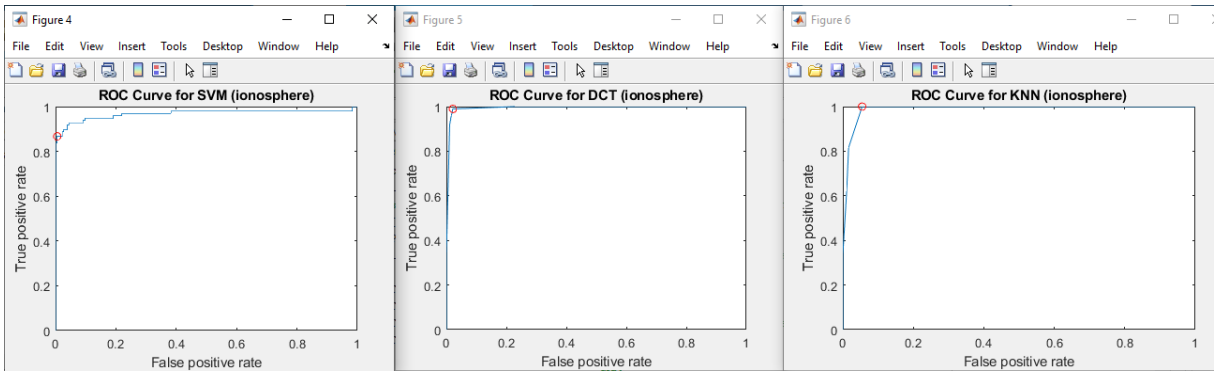
```
precision_knn =
    0.7600

recall_knn =
    0.9048

precision_knn =
    0.8571

recall_knn =
    0.9474

precision_knn =
    0.8889

recall_knn =
     1
```

```
precision_knn =
    0.8000

recall_knn =
    0.8000

precision_knn =
    0.7857

recall_knn =
    0.9167

precision_knn =
    0.5833

recall_knn =
    0.8750
```

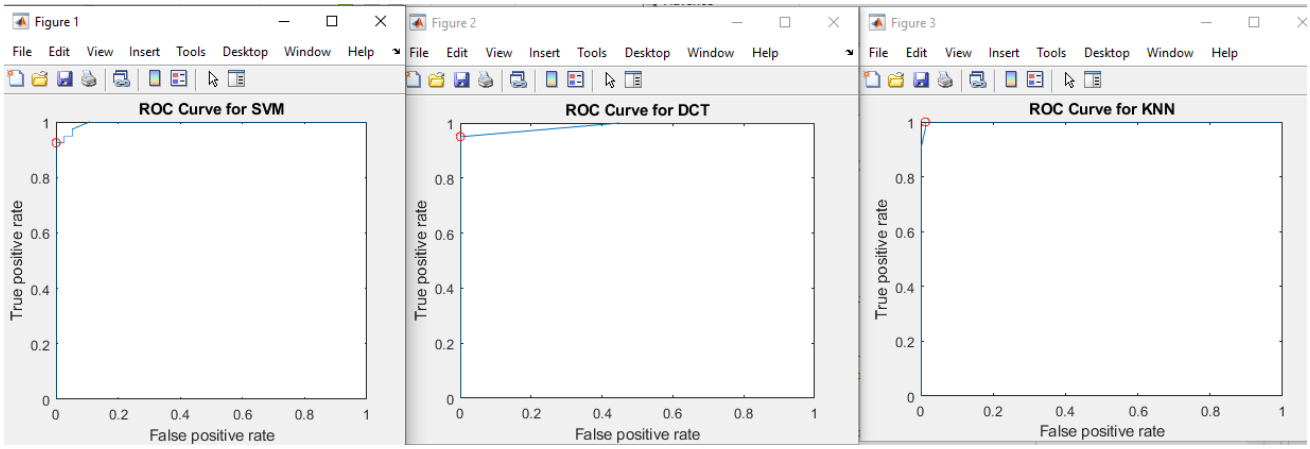
**Note:** I found the TP values with the confusion matrix. Then I found the precision and recall values. Since the confusion matrix is 3x3 in the fisheriris dataset, it had to be found in a different way so KNN and DCT values are missing.

# EXPERIMENT 4

**ROC Curve for ionosphere dataset:**



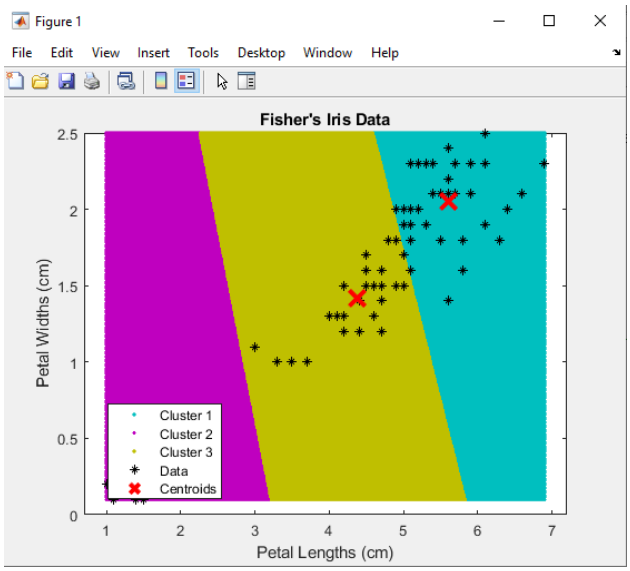

**ROC Curve for fisheriris dataset:**



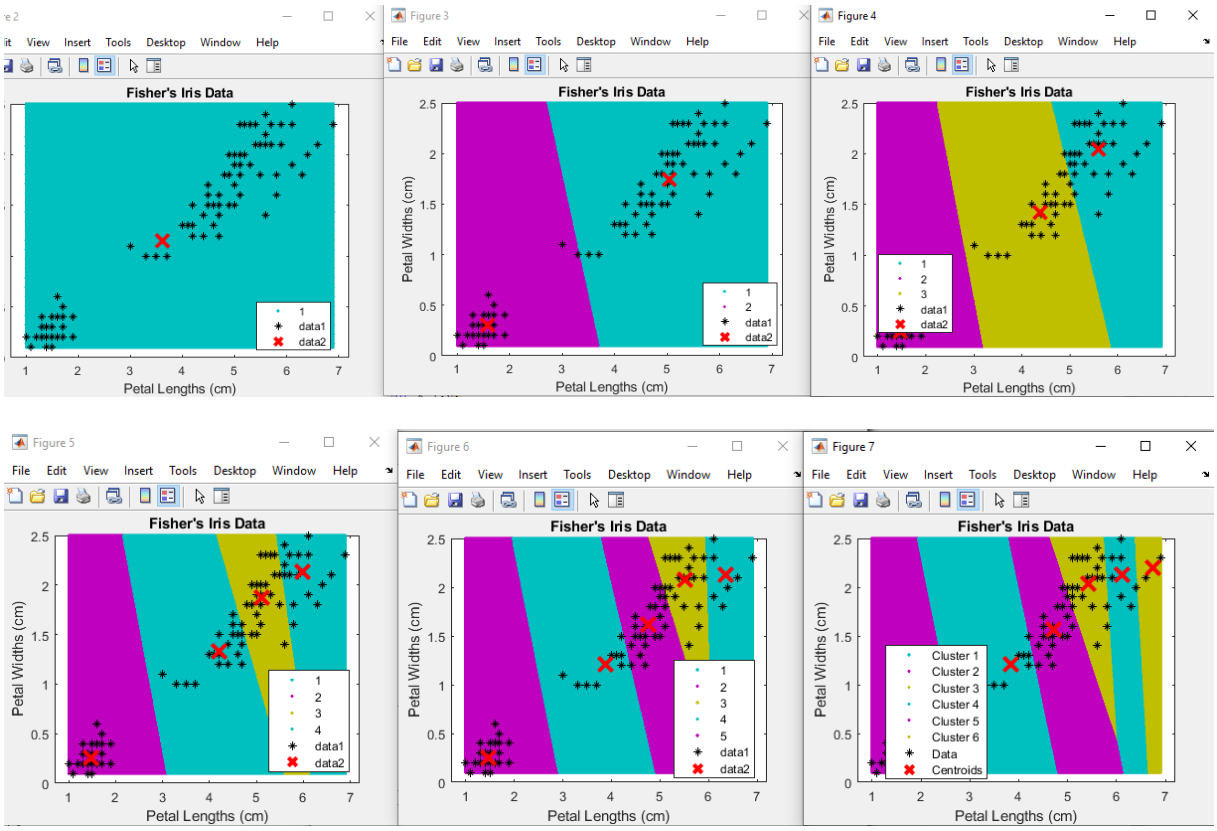

**Experiment Result:** It is desirable that the ROC curve be close to 1 in order to have less system error. When we compare the methods, KNN gives the best results.

# EXPERIMENT 5

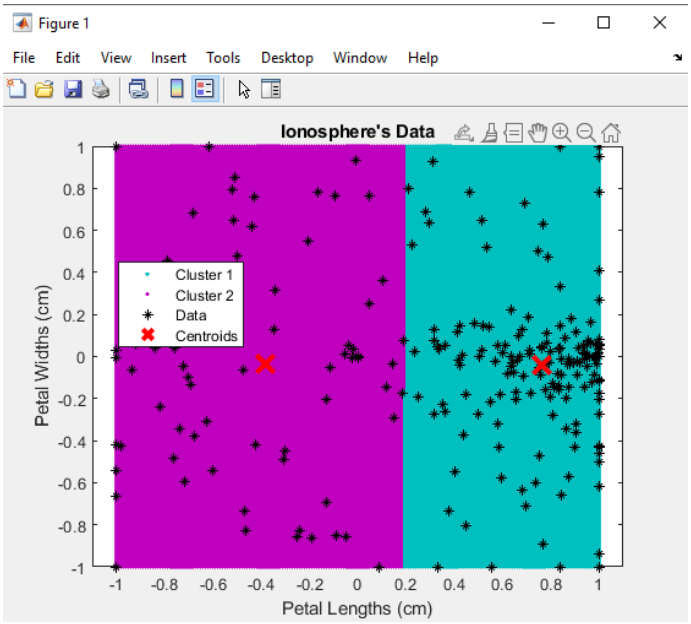
## For k= 3 (fisheriris dataset)



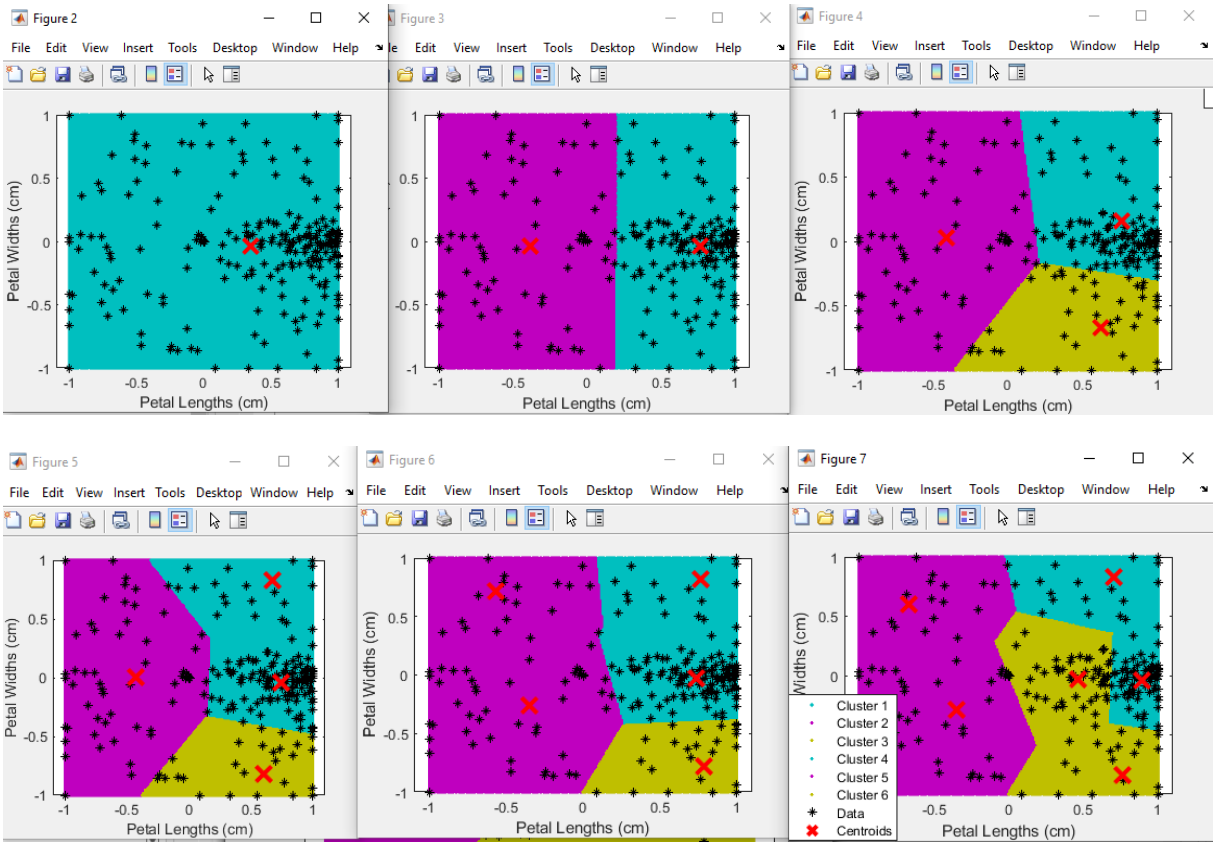

## For k= 1:6



## For k= 2 (ionosphere dataset)




## For k= 1:6



# CODES

## Experiment 1

```
clc,close all,clear all ;
load fisheriris

%Visualize data
figure
gscatter(meas(:,1),meas(:,2),species,'mbg','o*',8)

%Split the datasets randomly(%80 training, %20 testing)
[m,n] = size(meas) ;
P = 0.80 ;
idx = randperm(m);% shuffle the rows, 1 den m ye kadar random satır vektörü
oluşturur.(tekrarlayan öğe yok)

Training_m = meas(idx(1:round(P*m)),:);  % for meas
Testing_m = meas(idx(round(P*m)+1:end),:) ;
Training_s = species(idx(1:round(P*m)),:);  % for species
Testing_s = species(idx(round(P*m)+1:end),:) ;

inds = ~strcmp(Training_s,'setosa'); %removed setosa irises for dual
classification
x = Training_m(inds,3:4);
y = Training_s(inds);
inds1 = ~strcmp(Testing_s,'setosa'); %removed setosa irises for dual
classification
x_test = Testing_m(inds1,3:4);
y_test = Testing_s(inds1);

SVMModel = fitcsvm(x,y) %trainingleri svm ye soktuk
classOrder = SVMModel.ClassNames
sv = SVMModel.SupportVectors;
figure
gscatter(x(:,1),x(:,2),y,'bg')
hold on
plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
legend('versicolor','virginica','Support Vector');
hold off

grouphat = predict(SVMModel,x_test ); group = y_test;
[C,order] =
confusionmat(group,grouphat,'Order',{'versicolor','virginica'});

TP= C(1,1);FP= C(1,2);FN= C(2,1);TN= C(2,2);

accuracy= (TP+TN)/(TP+TN+FP+FN);
precision=TP/(TP+FP);
recall= TP/(TP+FN);
Fscore = 2*TP*(2*TP + FP + FN);
TPR= TP/(TP+FN);%(sensitivity)
FPR= TN /(FP + TN);%(specificity)
```

```
%Part 3:standardization
for j=1:n
CMean = mean(meas(:,j));
CStd  = std( meas(:,j));
for i=1:m
meas(i,j)=(meas(i,j)-CMean)/CStd;
end
end

idx2 = randperm(m);
Training_m2 = meas(idx2(1:round(P*m)),:);  % for meas
Testing_m2 = meas(idx2(round(P*m)+1:end),:) ;
Training_s2 = species(idx2(1:round(P*m)),:);  % for species
Testing_s2 = species(idx2(round(P*m)+1:end),:) ;

inds2 = ~strcmp(Training_s2,'setosa'); %removed setosa irises for dual
classification
x2 = Training_m2(inds2,3:4);
y2 = Training_s2(inds2);
inds3 = ~strcmp(Testing_s2,'setosa'); %removed setosa irises for dual
classification
x_test2 = Testing_m2(inds3,3:4);
y_test2 = Testing_s2(inds3);

SVMModel2 = fitcsvm(x2,y2) %trainingleri svm ye soktuk
classOrder2 = SVMModel2.ClassNames
sv2 = SVMModel2.SupportVectors;
figure
gscatter(x2(:,1),x2(:,2),y2,'bg')
hold on
plot(sv2(:,1),sv2(:,2),'ko','MarkerSize',10);
title('Scaled fisheriris dataset');
legend('versicolor','virginica','Support Vector');
hold off

grouphat2 = predict(SVMModel2,x_test2 ); group2 = y_test2;
[C2,order2] =
confusionmat(group2,grouphat2,'Order',{'versicolor','virginica'})

TP2= C2(1,1);FP2= C2(1,2);FN2= C2(2,1);TN2= C2(2,2);
accuracy2= (TP2+TN2)/(TP2+TN2+FP2+FN2);
precision2=TP2/(TP2+FP2);
recall2= TP2/(TP2+FN2);
Fscore2 = 2*TP2*(2*TP2 + FP2 + FN2);
TPR2= TP2/(TP2+FN2);%(sensitivity);
FPR2= TN2 /(FP2 + TN2);%(specificity);
%-----------------------------for ionosphere---------------------------
load ionosphere

%Visualize data
figure
gscatter(X(:,19),X(:,20),Y,'bg')

%Split the datasets randomly(%80 training, %20 testing)
[c,d] = size(X) ;
P = 0.80 ;
idx_i = randperm(c);% shuffle the rows, 1 den m ye kadar random satır
vektörü oluşturur.(tekrarlayan öğe yok)
```

```
Training_mi = X(idx_i(1:round(P*c)),:);  % for meas
Testing_mi = X(idx_i(round(P*c)+1:end),:) ;
Training_si = Y(idx_i(1:round(P*c)),:);  % for species
Testing_si = Y(idx_i(round(P*c)+1:end),:) ;

SVMModel_i = fitcsvm(Training_mi,Training_si) %trainingleri svm ye soktuk
classOrder_i = SVMModel_i.ClassNames
sv_i = SVMModel_i.SupportVectors;
figure
gscatter(Training_mi(:,19),Training_mi(:,20),Training_si,'bg')
hold on
plot(sv_i(:,19),sv_i(:,20),'ko','MarkerSize',10)
legend('b','g','Support Vector')
hold off

grouphat_i = predict(SVMModel_i,Testing_mi ); group_i = Testing_si;
[C_i,order_i] = confusionmat(group_i,grouphat_i,'Order',{'b','g'});

TP_i= C_i(1,1);FP_i= C_i(1,2);FN_i= C_i(2,1);TN_i= C_i(2,2);

accuracy_i= (TP_i+TN_i)/(TP_i+TN_i+FP_i+FN_i);
precision_i=TP_i/(TP_i+FP_i);
recall_i= TP_i/(TP_i+FN_i);
Fscore_i = 2*TP_i*(2*TP_i + FP_i + FN_i);
TPR_i= TP_i/(TP_i+FN_i);%(sensitivity)
FPR_i= TN_i /(FP_i + TN_i);%(specificity)

%Part 3:standardization
for j=1:d
CMean_i = mean(X(:,j));
CStd_i  = std( X(:,j));
for i=1:c

X(i,j)=(X(i,j)-CMean_i)/CStd_i;
end
end
for i=1:c
    X(i,2)=[-0.0310];  %for NaN error
end

idx2_i = randperm(c);
Training_m2i = X(idx2_i(1:round(P*c)),:);  % for X
Testing_m2i = X(idx2_i(round(P*c)+1:end),:) ;
Training_s2i = Y(idx2_i(1:round(P*c)),:);  % for Y
Testing_s2i = Y(idx2_i(round(P*c)+1:end),:) ;

SVMModel2_i = fitcsvm(Training_m2i,Training_s2i) %trainingleri svm ye
soktuk
classOrder2_i = SVMModel2_i.ClassNames
sv2_i = SVMModel2_i.SupportVectors;
figure
gscatter(Training_m2i(:,19),Training_m2i(:,20),Training_s2i,'bg')
hold on
plot(sv2_i(:,19),sv2_i(:,20),'ko','MarkerSize',10)
title('Scaled ionosphere dataset');
legend('b','g','Support Vector')
hold off

grouphat2_i = predict(SVMModel2_i,Testing_m2i ); group2_i = Testing_s2i;
[C2_i,order2_i] = confusionmat(group2_i,grouphat2_i,'Order',{'b','g'});
```

```
TP2_i= C2_i(1,1);FP2_i= C2_i(1,2);FN2_i= C2_i(2,1);TN2_i= C2_i(2,2);
accuracy2_i= (TP2_i+TN2_i)/(TP2_i+TN2_i+FP2_i+FN2_i);
precision2_i=TP2_i/(TP2_i+FP2_i);
recall2_i= TP2_i/(TP2_i+FN2_i);
Fscore2_i = 2*TP2_i*(2*TP2_i + FP2_i + FN2_i);
TPR2_i= TP2_i/(TP2_i+FN2_i);%(sensitivity)
FPR2_i= TN2_i /(FP2_i + TN2_i);%(specificity)
```

## Experiment 2

```
clc,close all,clear all ;
load fisheriris
%Split the datasets randomly(%80 training, %20 testing)
[m,n] = size(meas) ;
P = 0.80 ;
idx = randperm(m);% shuffle the rows, 1 den m ye kadar random satır vektörü
oluşturur.(tekrarlayan öğe yok)

Training_m = meas(idx(1:round(P*m)),:);  % for meas
Testing_m = meas(idx(round(P*m)+1:end),:) ;
Training_s = species(idx(1:round(P*m)),:);  % for species
Testing_s = species(idx(round(P*m)+1:end),:) ;

%---------------------------SVM-----------------------------------------
inds = ~strcmp(Training_s,'setosa'); %removed setosa irises for dual
classification
x = Training_m(inds,3:4);
y = Training_s(inds);
inds1 = ~strcmp(Testing_s,'setosa'); %removed setosa irises for dual
classification
x_test = Testing_m(inds1,3:4);
y_test = Testing_s(inds1);

SVMModel = fitcsvm(x,y); %trainingleri svm ye soktuk
classOrder = SVMModel.ClassNames;
sv = SVMModel.SupportVectors;
figure
gscatter(x(:,1),x(:,2),y,'bg')
hold on
plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
title('SVM Model for fisheriris');
legend('versicolor','virginica','Support Vector');
hold off

grouphat = predict(SVMModel,x_test ); group = y_test;
[C_SVM,order] =
confusionmat(group,grouphat,'Order',{'versicolor','virginica'})

TP= C_SVM(1,1);FP= C_SVM(1,2);FN= C_SVM(2,1);TN= C_SVM(2,2);

accuracy= (TP+TN)/(TP+TN+FP+FN);
precision=TP/(TP+FP);
recall= TP/(TP+FN);
Fscore = 2*TP*(2*TP + FP + FN);
TPR= TP/(TP+FN);%(sensitivity)
```

```
FPR= FP /(FP + TN);%(specificity)
%---------------------------DCT-----------------------------------------
tc = fitctree(Training_m,Training_s);
view(tc,'mode','graph');

grouphat2 = predict(tc,Testing_m );  group2=Testing_s;
[C_DCT,order2] =
confusionmat(group2,grouphat2,'Order',{'setosa','versicolor','virginica'})%
Konfüzyon matrisi 3x3lüktür,burdanda tp leri bulabiliriz ama svm den farklı
bir yöntem ile

[~,score] = resubPredict(tc);
%İlk sütun setosa'ya, ikincisi versicolor'a,üçüncüsü virginica'ya karşılık
gelir.
%İkili sınıflandırma olmadığı için bu formülden yararlanıcaz-->
score(:,2)?max(score(:,1),score(:,3))
diffscore = score(:,2) - max(score(:,1),score(:,3));
[FPR_dct,TPR_dct,T,AUC_dct] = perfcurve(Training_s,diffscore,'versicolor');
% AUC_dct=accuracy
FPR_dct=mean(FPR_dct);
TPR_dct=mean(TPR_dct);

%---------------------------kNN-----------------------------------------
%KNN model oluşturmaz
KNNMdl=fitcknn(Training_m,Training_s,'NumNeighbors',4);

grouphat3 = predict(KNNMdl,Testing_m );  %group2 ile aynı testing_s
[C_KNN,order3] =
confusionmat(group2,grouphat3,'Order',{'setosa','versicolor','virginica'})

[~,score3] = resubPredict(KNNMdl);
%İkili sınıflandırma olmadığı için bu formülden yararlanıcaz-->
score(:,2)?max(score(:,1),score(:,3))
diffscore3 = score3(:,2) - max(score3(:,1),score3(:,3));
[FPR_knn,TPR_knn,T3,AUC_knn] =
perfcurve(Training_s,diffscore3,'versicolor');
% AUC_knn=accuracy
FPR_knn=mean(FPR_knn);
TPR_knn=mean(TPR_knn);
%-----------------------------for ionosphere--------------------------------
load ionosphere
%Split the datasets randomly(%80 training, %20 testing)
[c,d] = size(X) ;
P = 0.80 ;
idx_i = randperm(c);% shuffle the rows, 1 den m ye kadar random satır
vektörü oluşturur.(tekrarlayan öğe yok)

Training_mi = X(idx_i(1:round(P*c)),:);  % for meas
Testing_mi = X(idx_i(round(P*c)+1:end),:) ;
Training_si = Y(idx_i(1:round(P*c)),:);  % for species
Testing_si = Y(idx_i(round(P*c)+1:end),:) ;

%---------------------------SVM-----------------------

SVMModel_i = fitcsvm(Training_mi,Training_si); %trainingleri svm ye soktuk
classOrder_i= SVMModel_i.ClassNames;
sv_i = SVMModel_i.SupportVectors;
figure
gscatter(Training_mi(:,19),Training_mi(:,20),Training_si,'bg')
hold on
```

```
plot(sv_i(:,19),sv_i(:,20),'ko','MarkerSize',10);
title('SVM Model for ionosphere');
legend('b','g','Support Vector');
hold off

grouphat_i = predict(SVMModel_i,Testing_mi ); group_i = Testing_si;
[Ci_SVM,order_i] = confusionmat(group_i,grouphat_i,'Order',{'b','g'})

TP_i= Ci_SVM(1,1);FP_i= Ci_SVM(1,2);FN_i= Ci_SVM(2,1);TN_i= Ci_SVM(2,2);

accuracy_i= (TP_i+TN_i)/(TP_i+TN_i+FP_i+FN_i);
precision_i=TP_i/(TP_i+FP_i);
recall_i= TP_i/(TP_i+FN_i);
Fscore_i = 2*TP_i*(2*TP_i + FP_i + FN_i);
TPR_i= TP_i/(TP_i+FN_i);%(sensitivity)
FPR_i= FP_i /(FP_i + TN_i);%(specificity)
%----------------------------DCT-----------------------------------------
tc_i = fitctree(Training_mi,Training_si);
view(tc_i,'mode','graph');

grouphat2_i = predict(tc_i,Testing_mi );  group2_i=Testing_si;
[Ci_DCT,order2] =
confusionmat(group2_i,grouphat2_i,'Order',{'b','g'})%Konfüzyon matrisi
2x2liktir.

TP_i2= Ci_DCT(1,1);FP_i2= Ci_DCT(1,2);FN_i2= Ci_DCT(2,1);TN_i2=
Ci_DCT(2,2);

accuracy_i2= (TP_i2+TN_i2)/(TP_i2+TN_i2+FP_i2+FN_i2);
precision_i2=TP_i2/(TP_i2+FP_i2);
recall_i2= TP_i2/(TP_i2+FN_i2);
Fscore_i2 = 2*TP_i2*(2*TP_i2 + FP_i2 + FN_i2);
TPR_i2= TP_i2/(TP_i2+FN_i2);%(sensitivity)
FPR_i2= FP_i2 /(FP_i2 + TN_i2);%(specificity)

%----------------------------kNN-----------------------------------------
%KNN model oluşturmaz
KNNMdl_i=fitcknn(Training_mi,Training_si,'NumNeighbors',4);

grouphat3_i = predict(KNNMdl_i,Testing_mi );  %group2 ile aynı testing_s
[Ci_KNN,order3] = confusionmat(group2_i,grouphat3_i,'Order',{'b','g'})

TP_i3= Ci_KNN(1,1);FP_i3= Ci_KNN(1,2);FN_i3= Ci_KNN(2,1);TN_i3=
Ci_KNN(2,2);

accuracy_i3= (TP_i3+TN_i3)/(TP_i3+TN_i3+FP_i3+FN_i3);
precision_i3=TP_i3/(TP_i3+FP_i3);
recall_i3= TP_i3/(TP_i3+FN_i3);
Fscore_i3 = 2*TP_i3*(2*TP_i3 + FP_i3 + FN_i3);
TPR_i3= TP_i3/(TP_i3+FN_i3);%(sensitivity)
FPR_i3= FP_i3 /(FP_i3 + TN_i3);%(specificity)
```

## Experiment 3

```
clc,close all,clear all ;
```

```
load fisheriris
%-------------------------SVM-----------------------------------------------
-----
inds = ~strcmp(species,'setosa'); %removed setosa irises for dual
classification
x = meas(inds,3:4);
y = species(inds);
for foldCnt = 5:10
    indices = crossvalind('Kfold',y,foldCnt);
    for i = 1:foldCnt
        test = (indices == i);
        train = ~test;
        SVMModel = fitcsvm(x(train,:),y(train,:));
        sv = SVMModel.SupportVectors;
    end
    figure
    gscatter(x(train,1),x(train,2),y(train,:));
    hold on
    plot(sv(:,1),sv(:,2),'ko','MarkerSize',10);
    hold off
    grouphat = predict(SVMModel,x(test,:) ); group=y(test,:);
    [C_SVM,order] =
confusionmat(group,grouphat,'Order',{'versicolor','virginica'});
    TP= C_SVM(1,1);FP= C_SVM(1,2);FN= C_SVM(2,1);TN= C_SVM(2,2);
    precision=TP/(TP+FP)
    recall= TP/(TP+FN)

end
%-------------DCT-------------------------------------
for foldCnt = 5:10
    indices = crossvalind('Kfold',species,foldCnt);
    for i = 1:foldCnt
        test2 = (indices == i);
        train2 = ~test2;
        DCTMdl = fitctree(meas(train2,:),species(train2,:));
    end
      view(DCTMdl,'mode','graph');
      grouphat2 = predict(DCTMdl,meas(test2,:) );  group2=species(test2,:);
      [C_DCT,order2] =
confusionmat(group2,grouphat2,'Order',{'setosa','versicolor','virginica'});
end
%-------------kNN methods-----------------------------------------
 for foldCnt = 5:10
    indices = crossvalind('Kfold',species,foldCnt);
    for i = 1:foldCnt
        test3 = (indices == i);
        train3 = ~test3;
        KNNMdl=fitcknn(meas(train3,:),species(train3,:),'NumNeighbors',1);
    end
    grouphat3 = predict(KNNMdl,meas(test3,:)); group3=species(test3,:);
   [C_KNN,order3] =
confusionmat(group3,grouphat3,'Order',{'setosa','versicolor','virginica'});
end


%-----------------------------for ionosphere---------------------------
load ionosphere
%-------------------------SVM----------------------
for foldCnt = 5:10
    indices = crossvalind('Kfold',Y,foldCnt);
    for i = 1:foldCnt
```

```
        test_i = (indices == i);
        train_i = ~test_i;
        SVMModel_i = fitcsvm(X(train_i,:),Y(train_i,:));
        sv_i = SVMModel_i.SupportVectors;
    end
    figure
    gscatter(X(train_i,19),X(train_i,20),Y(train_i,:));
    hold on
    plot(sv_i(:,19),sv_i(:,20),'ko','MarkerSize',10);
    hold off

    grouphat_i = predict(SVMModel_i,X(test_i,:) ); group_i=Y(test_i,:);
    [Ci_SVM,orderi] = confusionmat(group_i,grouphat_i,'Order',{'b','g'});
    TP_i= Ci_SVM(1,1);FP_i= Ci_SVM(1,2);FN_i= Ci_SVM(2,1);TN_i=
Ci_SVM(2,2);
    precision_i=TP_i/(TP_i+FP_i)
    recall_i= TP_i/(TP_i+FN_i)
end
%-------------DCT--------------------------------------
for foldCnt = 5:10
    indices = crossvalind('Kfold',Y,foldCnt);
    for i = 1:foldCnt
        test2_i = (indices == i);
        train2_i = ~test2_i;
        DCTMdl_i = fitctree(X(train2_i,:),Y(train2_i,:));
    end
      view(DCTMdl_i,'mode','graph');
      grouphat2_i = predict(DCTMdl_i,X(test2_i,:) );
group2_i=Y(test2_i,:);
      [Ci_DCT,order2i] =
confusionmat(group2_i,grouphat2_i,'Order',{'b','g'});

      TP_dct= Ci_DCT(1,1);FP_dct= Ci_DCT(1,2);FN_dct= Ci_DCT(2,1);TN_dct=
Ci_DCT(2,2);
      precision_dct=TP_dct/(TP_dct+FP_dct)
      recall_dct= TP_dct/(TP_dct+FN_dct)
end
%-------------kNN methods-----------------------------------
 for foldCnt = 5:10
    indices = crossvalind('Kfold',Y,foldCnt);
    for i = 1:foldCnt
        test3_i = (indices == i);
        train3_i = ~test3_i;
        KNNMdl_i=fitcknn(X(train3_i,:),Y(train3_i,:),'NumNeighbors',1);
    end
    grouphat3_i = predict(KNNMdl_i,X(test3_i,:)); group3_i=Y(test3_i,:);
   [Ci_KNN,order3i] = confusionmat(group3_i,grouphat3_i,'Order',{'b','g'});

    TP_knn= Ci_KNN(1,1);FP_knn= Ci_KNN(1,2);FN_knn= Ci_KNN(2,1);TN_knn=
Ci_KNN(2,2);
    precision_knn=TP_knn/(TP_knn+FP_knn)
    recall_knn= TP_knn/(TP_knn+FN_knn)
end
```

## Experiment 4

```
clc,close all,clear all ;
```

```
load fisheriris
%Split the datasets randomly(%80 training, %20 testing)
[m,n] = size(meas) ;
P = 0.80 ;
idx = randperm(m);% shuffle the rows, 1 den m ye kadar random satır vektörü
oluşturur.(tekrarlayan öğe yok)

Training_m = meas(idx(1:round(P*m)),:);  % for meas
Testing_m = meas(idx(round(P*m)+1:end),:) ;
Training_s = species(idx(1:round(P*m)),:);  % for species
Testing_s = species(idx(round(P*m)+1:end),:) ;

%---------------------------SVM---------------------------------------
inds = ~strcmp(Training_s,'setosa'); %removed setosa irises for dual
classification
x = Training_m(inds,3:4);
y = Training_s(inds);
inds1 = ~strcmp(Testing_s,'setosa'); %removed setosa irises for dual
classification
x_test = Testing_m(inds1,3:4);
y_test = Testing_s(inds1);

resp = strcmp(y,'versicolor'); % resp = 1 or 0
pred = x;

mdlSVM = fitcsvm(pred,resp,'Standardize',true);
%Compute the posterior probabilities (scores)
mdlSVM = fitPosterior(mdlSVM);
[~,score_svm] = resubPredict(mdlSVM);
[Xsvm,Ysvm,Tsvm,~,OPTROCPT1] =
perfcurve(resp,score_svm(:,mdlSVM.ClassNames),'true');

figure
plot(Xsvm,Ysvm);
hold on
plot(OPTROCPT1(1),OPTROCPT1(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for SVM');
hold off
%---------------------------DCT---------------------------------------
DCTMdl =
fitctree(Training_m,Training_s,'ClassNames',{'setosa','versicolor','virgini
ca'});
%--------ROC Curves for DCT
%Predict the class labels and scores for the species
[~,score] = resubPredict(DCTMdl);
%İlk sütun setosa'ya, ikincisi versicolor'a,üçüncüsü virginica'ya karşılık
gelir.
%İkili sınıflandırma olmadığı için bu formülden yararlanıcaz-->
score(:,2)?max(score(:,1),score(:,3))
diffscore = score(:,2) - max(score(:,1),score(:,3));
[X,Y,T,~,OPTROCPT] = perfcurve(Training_s,diffscore,'versicolor');
% X=FPR, Y=TPR, OPTROCPT=Optimal operating point of the ROC curve
figure
plot(X,Y)
hold on
plot(OPTROCPT(1),OPTROCPT(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
```

```
title('ROC Curve for DCT');
hold off
%-------------------------kNN---------------------------------------
KNNMdl=fitcknn(Training_m,Training_s,'NumNeighbors',4);
%--------ROC Curves for KNN
[~,score3] = resubPredict(KNNMdl);
%İkili sınıflandırma olmadığı için bu formülden yararlanıcaz-->
score(:,2)?max(score(:,1),score(:,3))
diffscore3 = score3(:,2) - max(score3(:,1),score3(:,3));
[X3,Y3,T3,~,OPTROCPT3] = perfcurve(Training_s,diffscore3,'versicolor');

figure
plot(X3,Y3)
hold on
plot(OPTROCPT3(1),OPTROCPT3(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for KNN')
hold off

%---------------for ionosphere-----------------------------------------------
load ionosphere
%Split the datasets randomly(%80 training, %20 testing)
[c,d] = size(X) ;
P = 0.80 ;
idx_i = randperm(c);% shuffle the rows, 1 den m ye kadar random satır
vektörü oluşturur.(tekrarlayan öğe yok)

Training_mi = X(idx_i(1:round(P*c)),:);  % for meas
Testing_mi = X(idx_i(round(P*c)+1:end),:) ;
Training_si = Y(idx_i(1:round(P*c)),:);  % for species
Testing_si = Y(idx_i(round(P*c)+1:end),:) ;

%---------------------------SVM---------------------------------------
resp_i = strcmp(Training_si,'b'); % resp = 1 or 0
pred_i = Training_mi;

mdlSVM_i = fitcsvm(pred_i,resp_i,'Standardize',true);
%Compute the posterior probabilities (scores)
mdlSVM_i = fitPosterior(mdlSVM_i);
[~,score_svmi] = resubPredict(mdlSVM_i);
[Xsvmi,Ysvmi,Tsvmi,~,OPTROCPT1i] =
perfcurve(resp_i,score_svmi(:,mdlSVM_i.ClassNames),'true');

figure
plot(Xsvmi,Ysvmi);
hold on
plot(OPTROCPT1i(1),OPTROCPT1i(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for SVM (ionosphere)');
hold off

%---------------------------DCT---------------------------------------


DCTMdl_i = fitctree(pred_i,resp_i);

% %--------ROC Curves for DCT
```

```
% DCTMdl_i = fitPosterior(DCTMdl_i);
[~,score_svmi2] = resubPredict(DCTMdl_i);
[Xsvmi2,Ysvmi2,Tsvmi2,~,OPTROCPT1i2] =
perfcurve(resp_i,score_svmi2(:,DCTMdl_i.ClassNames),'true');
figure
plot(Xsvmi2,Ysvmi2);
hold on
plot(OPTROCPT1i2(1),OPTROCPT1i2(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for DCT (ionosphere)');
hold off
%
% %---------------------------kNN-----------------------------------------
KNNMdl_i=fitcknn(pred_i,resp_i,'NumNeighbors',4);
% %--------ROC Curves for KNN
% KNNMdl_i = fitPosterior(KNNMdl_i);
[~,score_svmi3] = resubPredict(KNNMdl_i);
[Xsvmi3,Ysvmi3,Tsvmi3,~,OPTROCPT1i3] =
perfcurve(resp_i,score_svmi3(:,KNNMdl_i.ClassNames),'true');
figure
plot(Xsvmi3,Ysvmi3);
hold on
plot(OPTROCPT1i3(1),OPTROCPT1i3(2),'ro')
xlabel('False positive rate')
ylabel('True positive rate')
title('ROC Curve for KNN (ionosphere)');
hold off
```

## Experiment 5

```
clc,close all,clear all ;
load fisheriris
%Split the datasets randomly(%80 training, %20 testing)
[m,n] = size(meas) ;
P = 0.80 ;
idx_0 = randperm(m);% shuffle the rows, 1 den m ye kadar random satır
vektörü oluşturur.(tekrarlayan öğe yok)

Training_m = meas(idx_0(1:round(P*m)),:);  % for meas
Testing_m = meas(idx_0(round(P*m)+1:end),:) ;
Training_s = species(idx_0(1:round(P*m)),:);  % for species
Testing_s = species(idx_0(round(P*m)+1:end),:) ;

z = Training_m(:,3:4);
%1.Part-----------------------------------------------------
clusterNbr=3;
rng(1);
[idx,C]=kmeans(z,clusterNbr);

x1 = min(z(:,1)):0.01:max(z(:,1));
x2 = min(z(:,2)):0.01:max(z(:,2));
[x1G,x2G] = meshgrid(x1,x2);
XGrid = [x1G(:),x2G(:)]; % Defines a fine grid on the plot
```

```
idx2Region = kmeans(XGrid,3,'MaxIter',1,'Start',C);

figure;
gscatter(XGrid(:,1),XGrid(:,2),idx2Region,[0,0.75,0.75;0.75,0,0.75;0.75,0.7
5,0],'..');
hold on;
plot(z(:,1),z(:,2),'k*','MarkerSize',5);
plot(C(:,1),C(:,2),'rx','MarkerSize',15,'LineWidth',3);
title 'Fisher''s Iris Data';
xlabel 'Petal Lengths (cm)';
ylabel 'Petal Widths (cm)';
legend('Cluster 1','Cluster 2','Cluster 3','Data','Centroids');
hold off;

%2.Part-----------------------------------------------------------------
------------

for k=1:6;
rng(1);
[idx_b,Cb]=kmeans(z,k);

x1_b = min(z(:,1)):0.01:max(z(:,1));
x2_b = min(z(:,2)):0.01:max(z(:,2));
[x1G_b,x2G_b] = meshgrid(x1_b,x2_b);
XGrid_b = [x1G_b(:),x2G_b(:)]; % Defines a fine grid on the plot

idx2Region_b = kmeans(XGrid_b,k,'MaxIter',1,'Start',Cb);

figure;
gscatter(XGrid_b(:,1),XGrid_b(:,2),idx2Region_b,[0,0.75,0.75;0.75,0,0.75;0.
75,0.75,0],'..');
hold on;
plot(z(:,1),z(:,2),'k*','MarkerSize',5);
plot(Cb(1:k,1),Cb(1:k,2),'rx','MarkerSize',15,'LineWidth',3);
title 'Fisher''s Iris Data';
xlabel 'Petal Lengths (cm)';
ylabel 'Petal Widths (cm)';
hold off;
end
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Data','Centroids');

%-----------------------------------------------------------------------
--------------------
load ionosphere
%Split the datasets randomly(%80 training, %20 testing)
[c,d] = size(X) ;
P = 0.80 ;
idx_0i = randperm(c);% shuffle the rows, 1 den m ye kadar random satır
vektörü oluşturur.(tekrarlayan öğe yok)

Training_mi = X(idx_0i(1:round(P*c)),:);  % for meas
Testing_mi = X(idx_0i(round(P*c)+1:end),:) ;
Training_si = Y(idx_0i(1:round(P*c)),:);  % for species
Testing_si = Y(idx_0i(round(P*c)+1:end),:) ;

zi = Training_mi(:,19:20);

clusterNbri=2;
```

```
rng(1);
[idxi,Ci]=kmeans(zi,clusterNbri);

x1i = min(zi(:,1)):0.01:max(zi(:,1));
x2i = min(zi(:,2)):0.01:max(zi(:,2));
[x1Gi,x2Gi] = meshgrid(x1i,x2i);
XGridi = [x1Gi(:),x2Gi(:)]; % Defines a fine grid on the plot

idx2Regioni = kmeans(XGridi,2,'MaxIter',1,'Start',Ci);

figure;
gscatter(XGridi(:,1),XGridi(:,2),idx2Regioni,[0,0.75,0.75;0.75,0,0.75;0.75,
0.75,0],'..');
hold on;
plot(zi(:,1),zi(:,2),'k*','MarkerSize',5);
plot(Ci(:,1),Ci(:,2),'rx','MarkerSize',15,'LineWidth',3);
title ('Ionosphere''s Data');
xlabel ('Petal Lengths (cm)');
ylabel ('Petal Widths (cm)');
legend('Cluster 1','Cluster 2','Data','Centroids');
hold off;
%2.Part-----------------------------------------------------------------
------------

for k=1:6;
rng(1);
[idx_bi,Cbi]=kmeans(zi,k);

x1_bi = min(zi(:,1)):0.01:max(zi(:,1));
x2_bi = min(zi(:,2)):0.01:max(zi(:,2));
[x1G_bi,x2G_bi] = meshgrid(x1_bi,x2_bi);
XGrid_bi = [x1G_bi(:),x2G_bi(:)]; % Defines a fine grid on the plot

idx2Region_bi = kmeans(XGrid_bi,k,'MaxIter',1,'Start',Cbi);

figure;
gscatter(XGrid_bi(:,1),XGrid_bi(:,2),idx2Region_bi,[0,0.75,0.75;0.75,0,0.75
;0.75,0.75,0],'..');
hold on;
plot(zi(:,1),zi(:,2),'k*','MarkerSize',5);
plot(Cbi(1:k,1),Cbi(1:k,2),'rx','MarkerSize',15,'LineWidth',3);
title 'Fisher''s Iris Data';
xlabel 'Petal Lengths (cm)';
ylabel 'Petal Widths (cm)';
hold off;
end
legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster
6','Data','Centroids');
```