Discriminant Analysis

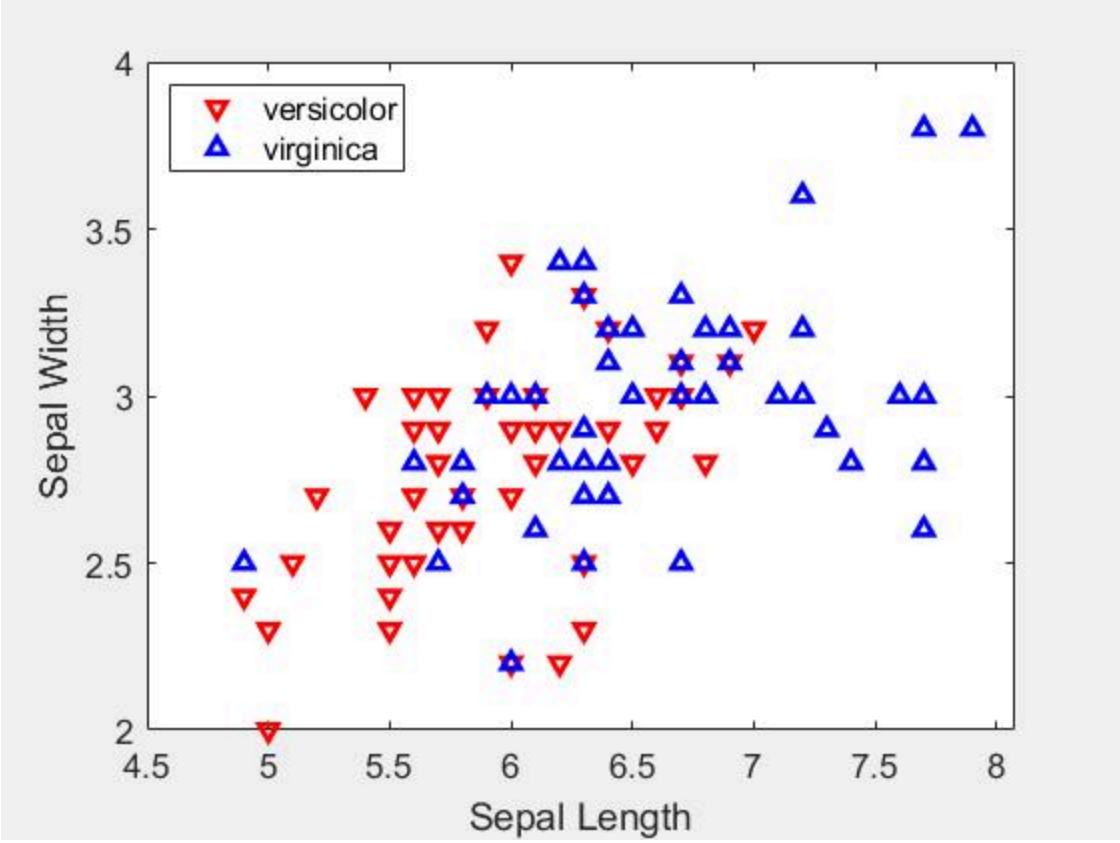
Table of Contents

Demonstrates matlab's discriminant analysis functions. by Ahmet Sacan.

Load Fisher's Iris Data

Fisher's iris data consists of measurements on the sepal length, sepal width, petal length, and petal width for 150 iris flower specimens. There are 50 specimens from each of three species: setosa, versicolor, and virginica. See more information at: https://en.wikipedia.org/wiki/Iris_flower_data_set

```
load fisheriris


% We'll skip rows 1..50 which are the 'setosa' species. and use the
% remaining rows, which are versicolor and virginica species.
sepallength = meas(51:end,1);
sepalwidth = meas(51:end,2);
group = species(51:end);

% Let's see how the how the sepal measurements (length and width)
 differ
% between species.
h = gscatter(sepallength,sepalwidth,group,'rb','v^',[],'off');
 set(h,'LineWidth',2)
xlabel('Sepal Length'); ylabel('Sepal Width');
 legend('versicolor','virginica','Location','NW');

%we'll repeat the visualization of the dataset over and over, so let's
 make
%a function for it.
visualize2classdata=@()
{set(gscatter(sepallength,sepalwidth,group,'rb','v^',
[],'off'),'LineWidth',2) xlabel('Sepal Length') ylabel('Sepal Width')
 legend('Fisher versicolor','Fisher virginica','Location','NW')};
```



Apply Linear LDA

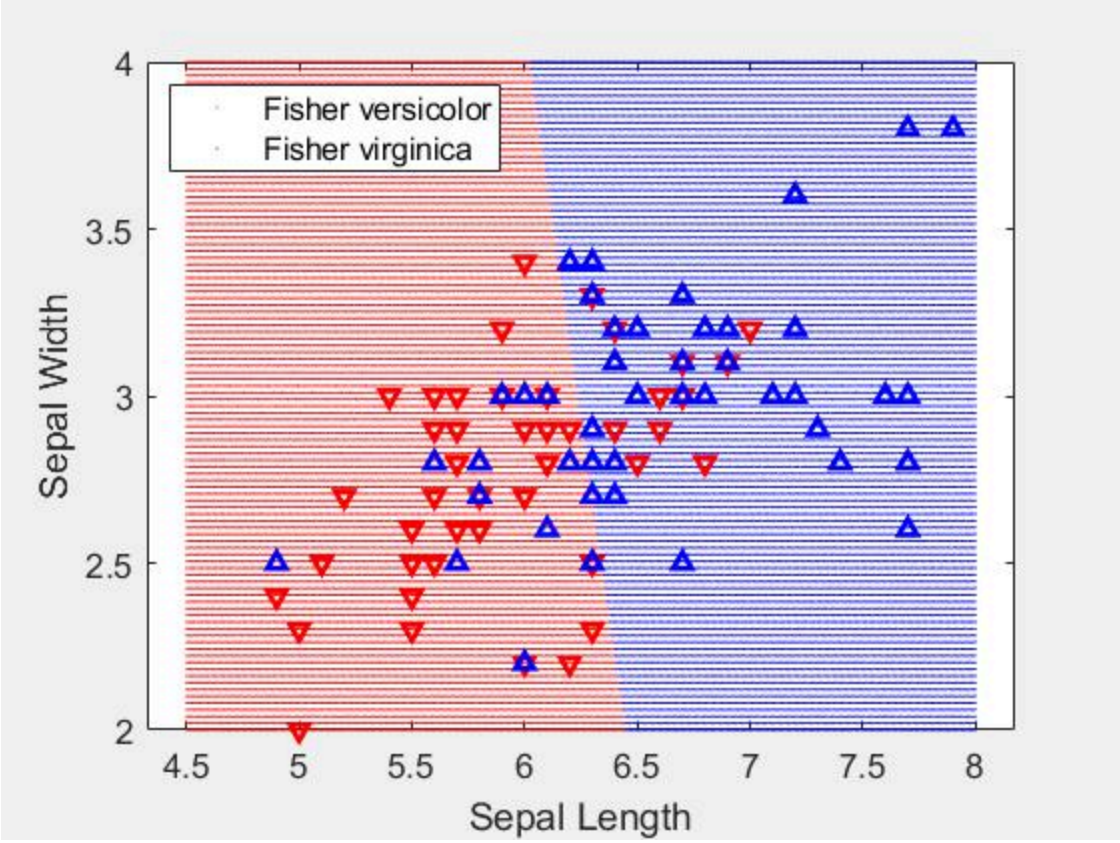
Build/Train the model from the data. We can later apply the model to test cases.

```
mdl = fitcdiscr([sepallength
 sepalwidth],group,'discrimtype','linear');
```

Visualize the classification

We'll classify a meshgrid of sepallength and sepalwidth values into groups and visualize their classification. This is not an ideal way of showing which regions belong to which class, but it works. [L W] are the [spallength sepalwidth] of our "test" cases.

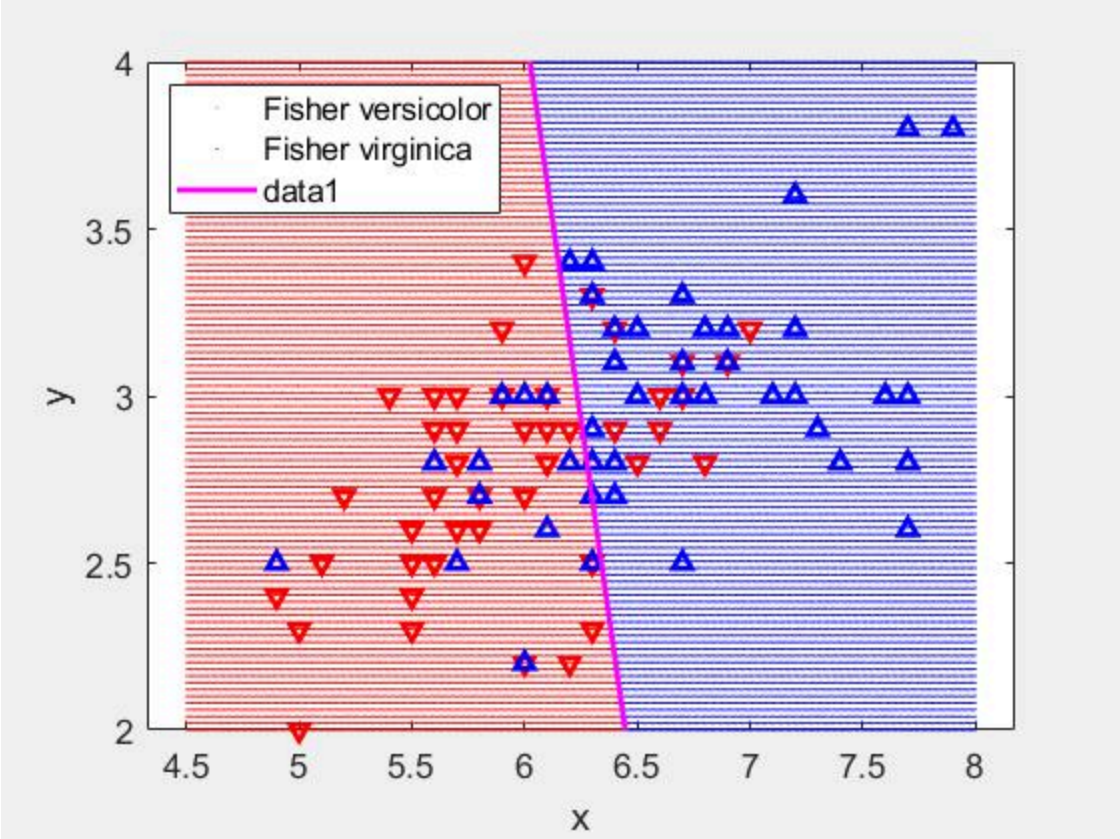
```
[L,W] = meshgrid(linspace(4.5,8,500),linspace(2,4,100)); L=L(:);
 W=W(:);
pred = mdl.predict([L W]);
h=gscatter(L,W,pred,'rb','.',1,'off');
 set(h,'LineWidth',2,'MarkerSize',2)
hold on; visualize2classdata(); hold off;
```



Visualize the decision boundary

We can actually do better than above. The coefficients we get from the classify function tell us exactly where the classes are separated. We'll use ezplot(), which is just an alternative to managing inputs to plot() ourselves.

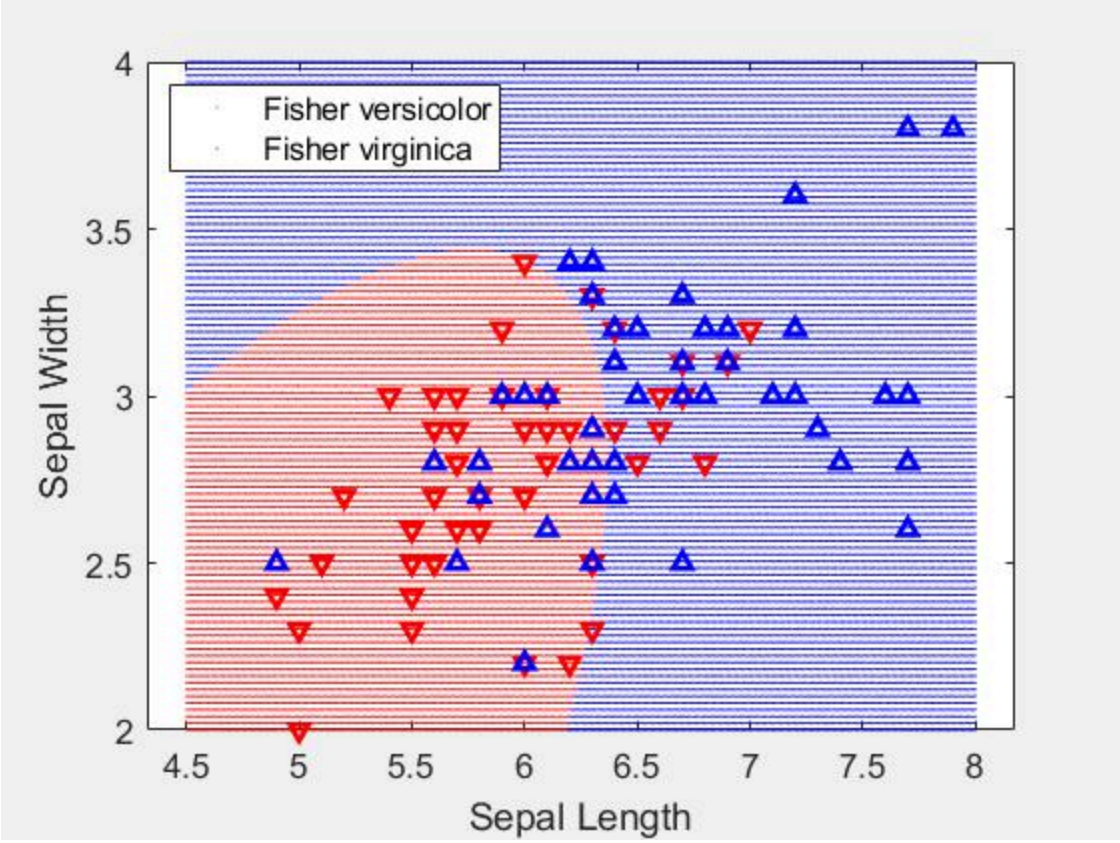
```
hold on;
A = mdl.Coeffs(1,2).Linear;
B = mdl.Coeffs(1,2).Const;
h = ezplot(@(x,y) [x y]*A + B, [4.5 8 2 4]); title('');
set(h,'Color','m','LineWidth',2);
hold off;
```



Apply Quadratic DA

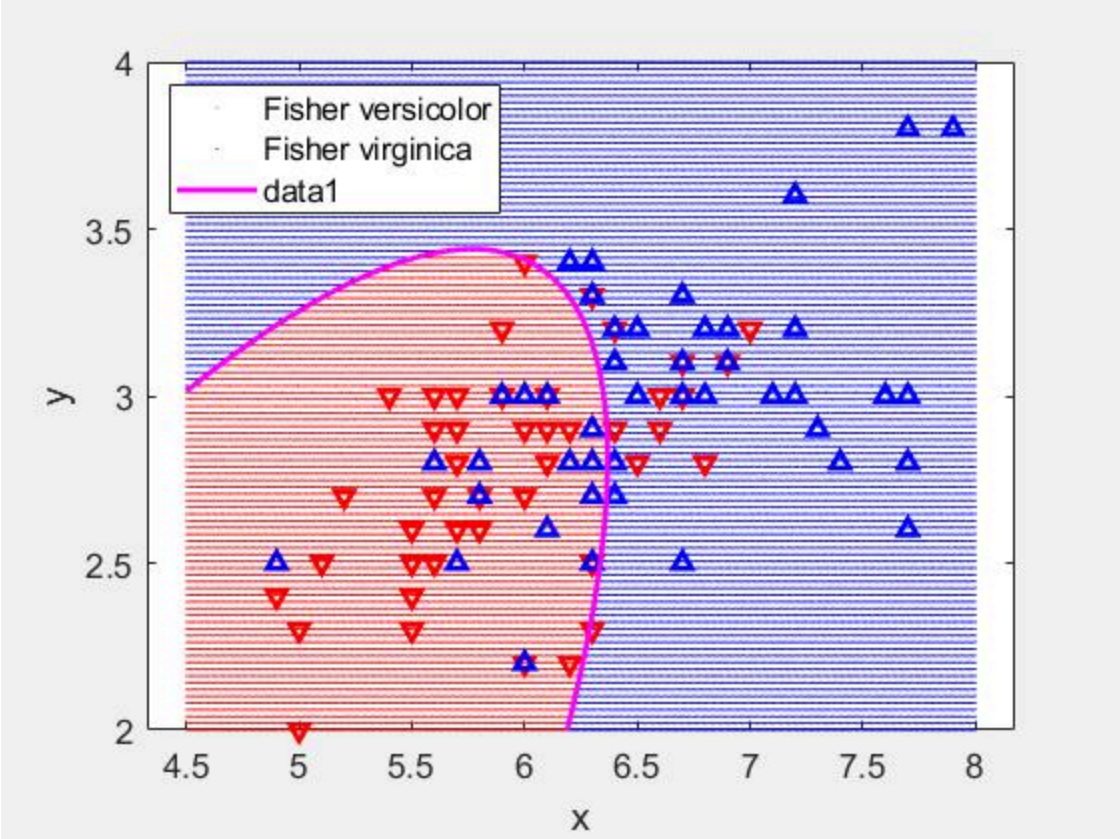
Same as above, but with 'quadratic' option for classify()

```
mdl = fitcdiscr([sepallength
 sepalwidth],group,'discrimtype','quadratic');
pred = mdl.predict([L W]);
h=gscatter(L,W,pred,'rb','.',1,'off');
 set(h,'LineWidth',2,'MarkerSize',2)
hold on; visualize2classdata(); hold off;
```



Visualize the decision boundary

```
hold on;
A = mdl.Coeffs(1,2).Linear;
B = mdl.Coeffs(1,2).Const;
Q = mdl.Coeffs(1,2).Quadratic;
h = ezplot(@(x,y) sum(([x y]*Q).*[x y],2) + [x y]*A + B, [4.5 8 2 4]);
  title('');
set(h,'Color','m','LineWidth',2);
hold off
```



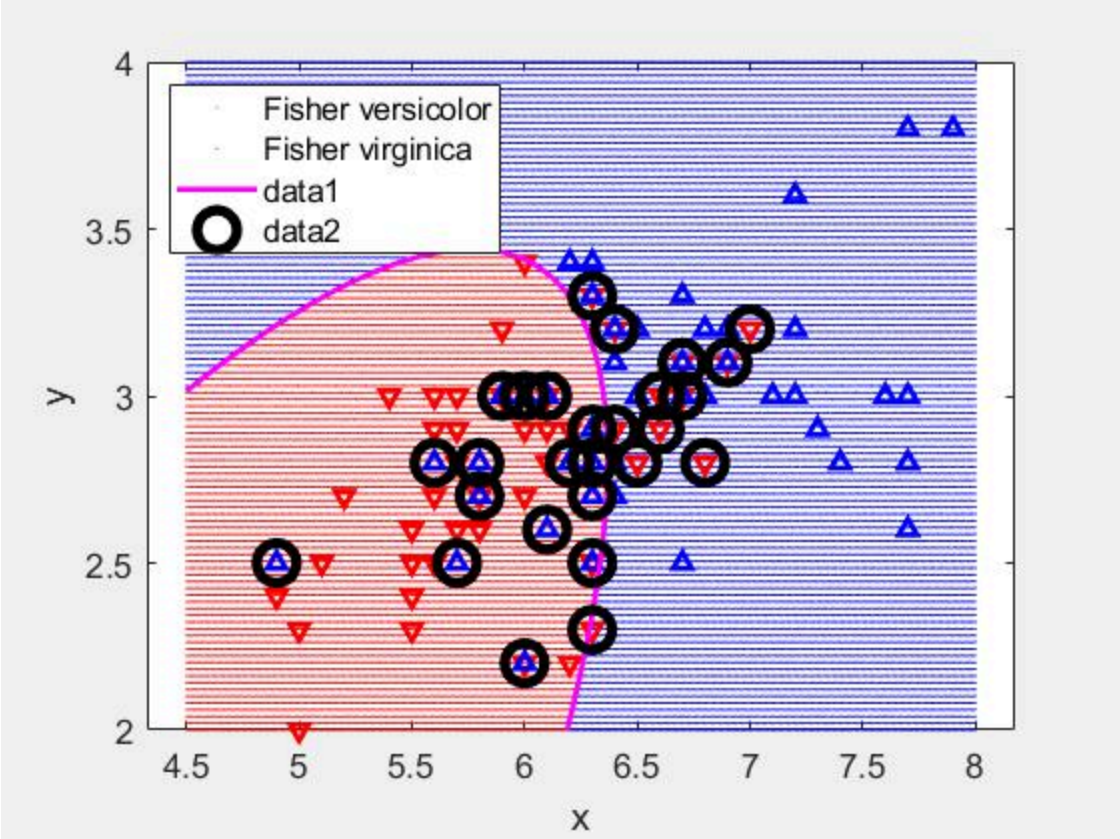
Classification Error

We don't quite know what the correct answers are for [X Y] test cases above. Let's examine how the model performs if we ask it to predict the "training data" whose answers we do know.

```
pred=mdl.predict([sepallength sepalwidth]);
Igood = strcmp(pred,group);
Ibad = ~Igood;
```

show/mark the mis-classified samples on the figure.

```
hold on;
plot(sepallength(Ibad),
 sepalwidth(Ibad), 'ko', 'LineWidth',4, 'MarkerSize',15);
hold off;
```




```
%calculate the misclassification rate (aka error rate)
misclassificationrate = nnz(Ibad) / numel(group)
accuracy = 1-misclassificationrate
```

```
misclassificationrate =

    0.2900


accuracy =

    0.7100
```

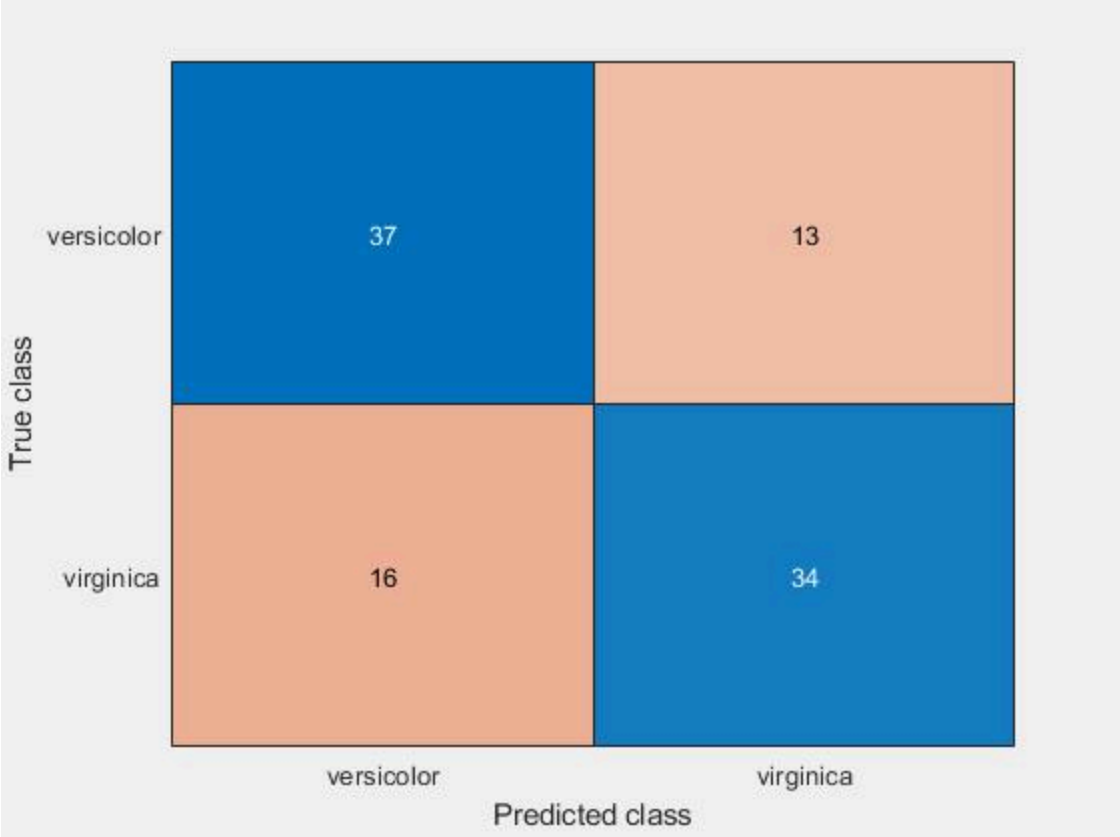
Confusion Matrix

```
[confusionmatrix,matrixlabels] = confusionmat(group,pred)
```

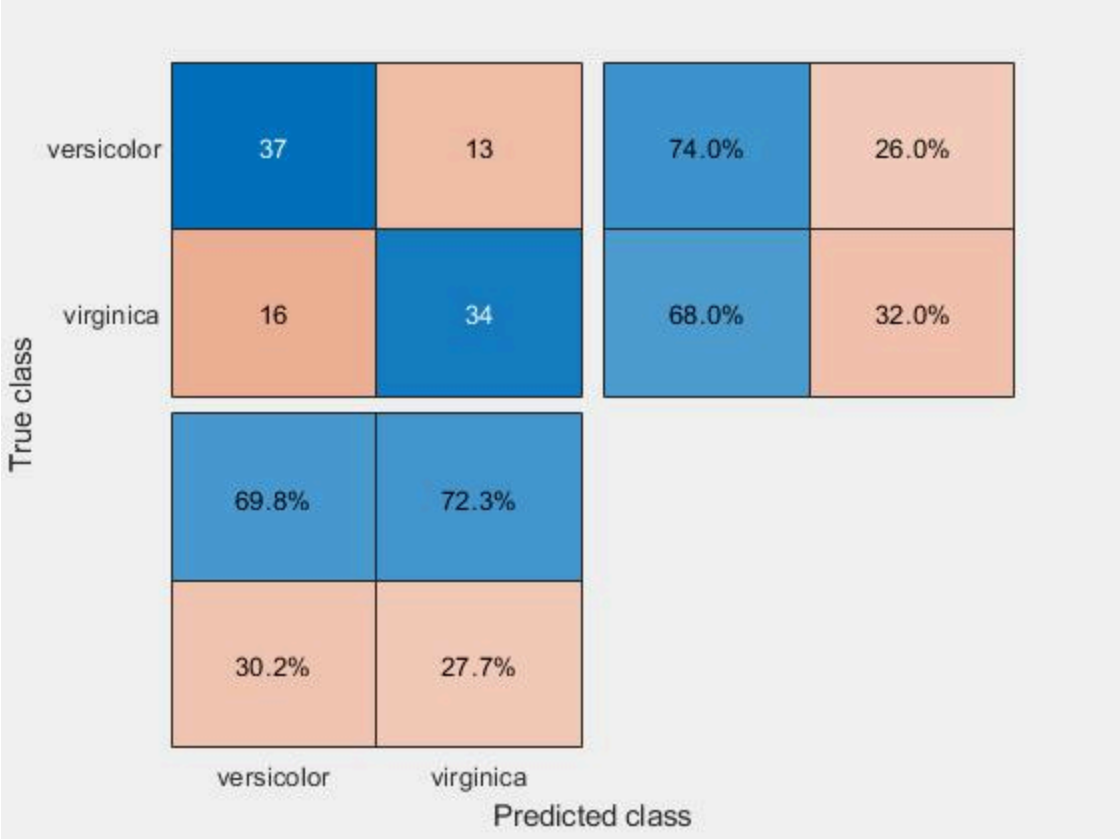
```
confusionmatrix =

    37    13
    16    34
```

```
matrixlabels =

  2×1 cell array

    {'versicolor'}
    {'virginica' }
```

```
cm = confusionchart(group,pred);
```

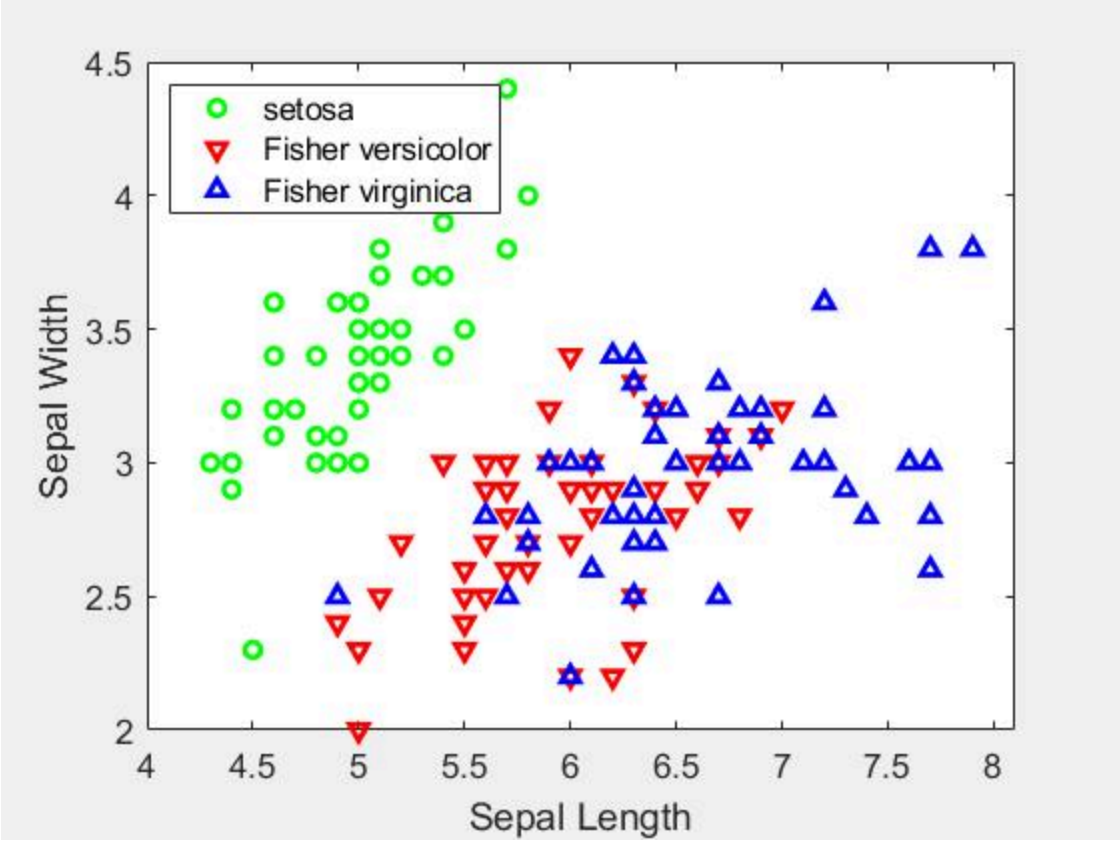



```
cm.RowSummary = 'row-normalized';
cm.ColumnSummary = 'column-normalized';
```



Discriminant Analysis of More than 2 Groups

```
sepallength = meas(:,1);
sepalwidth = meas(:,2);
group = species(:);

h = gscatter(sepallength,sepalwidth,group,'grb','ov^',[],'off');
 set(h,'LineWidth',2)
xlabel('Sepal Length'); ylabel('Sepal Width'); legend('setosa','Fisher
 versicolor','Fisher virginica','Location','NW');
```

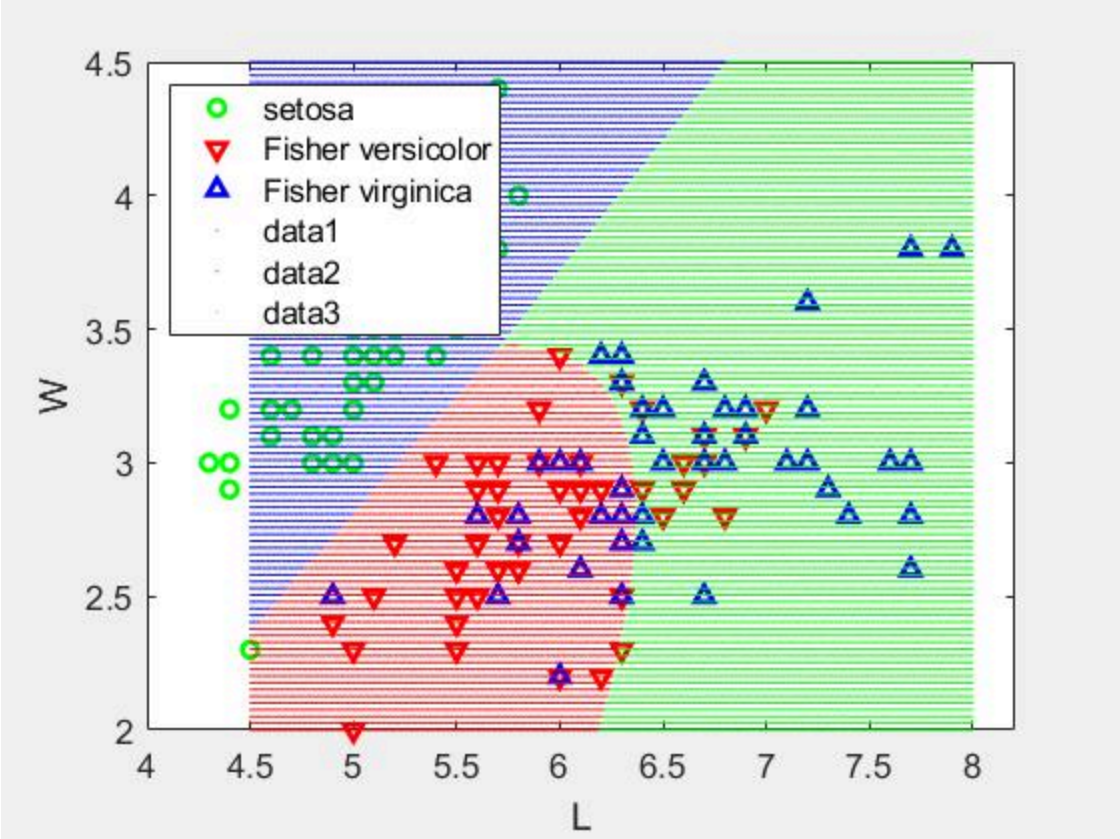



```
mdl = fitcdiscr([sepallength
 sepalwidth],group,'discrimtype','quadratic');
mdl.Coeffs
```

```
ans =

  3×3 struct array with fields:

    DiscrimType
    Const
    Linear
    Quadratic
    Class1
    Class2
```

Visualize prediction of hypothetical samples

```
[L,W] = meshgrid(linspace(4.5,8,500),linspace(2,4.5,100)); L=L(:);
 W=W(:);
pred = mdl.predict([L W]);
hold on;
gscatter(L,W,pred,'rbg','.',1,'off');
hold off
```



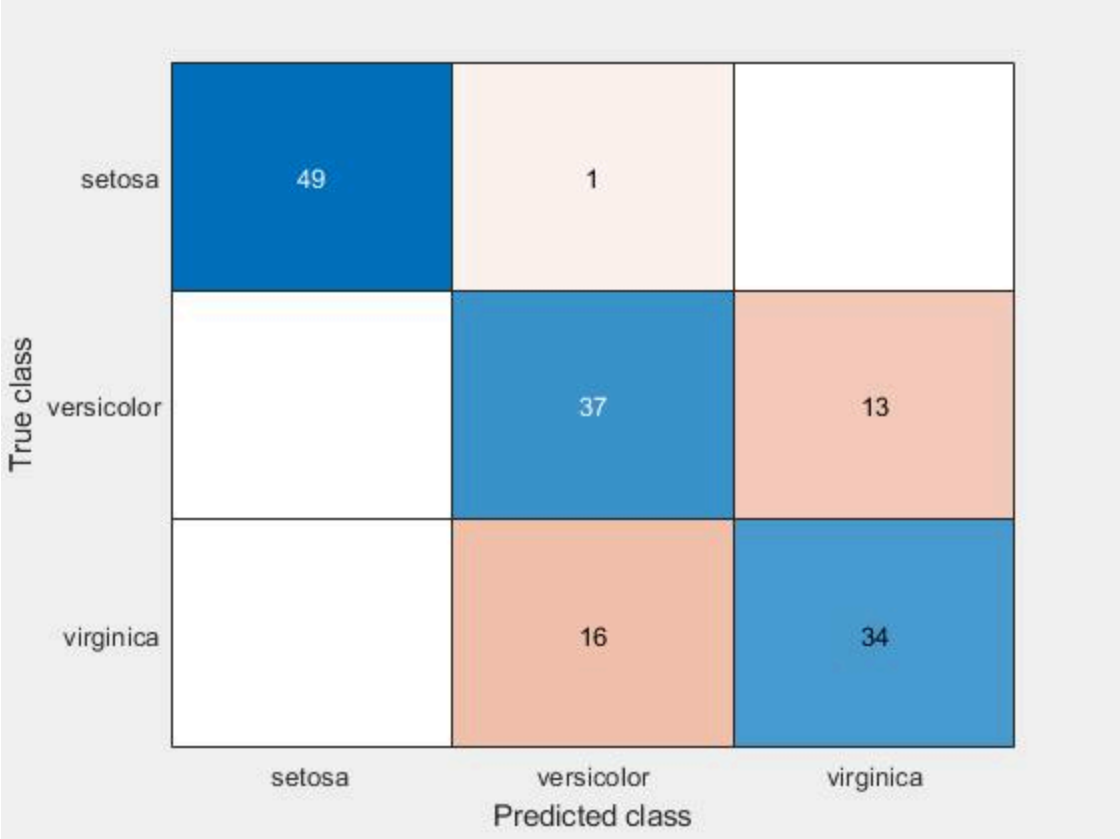
Evaluate prediction of training samples

```
pred=mdl.predict([sepallength, sepalwidth]);
Igood = strcmp(pred,group);
accuracy = nnz(Igood) / numel(group)
```

```
accuracy =

    0.8000
```

```
confusionchart(group,pred)
```

```
ans =

  ConfusionMatrixChart with properties:

    NormalizedValues: [3×3 double]
         ClassLabels: {3×1 cell}

  Use GET to show all properties
```



Published with MATLAB® R2018b