Contents

```
clc
clear all
close all
```

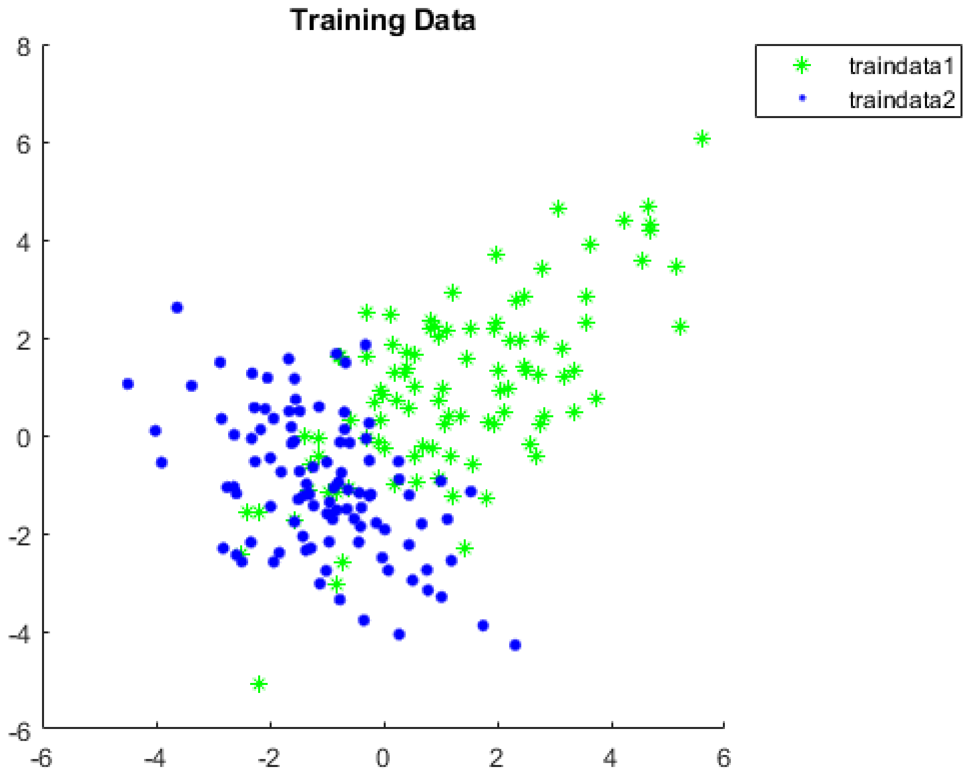
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                           Data Generation
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Data Generation

```
mu1 = [1 1];
sigma1 = [3 2;2 3];
mu2 = [-1 -1];
sigma2 = [2 -1;-1 2];
x1 = mvnrnd(mu1,sigma1,200);
y1 = ones(max(size(x1)),1);
x2 = mvnrnd(mu2,sigma2,200);
y2 = -1*ones(max(size(x2)),1);
traindata1 = x1(1:100,:); testdata1 = x1(101:end,:);
traindata2 = x2(1:100,:); testdata2 = x2(101:end,:);
```

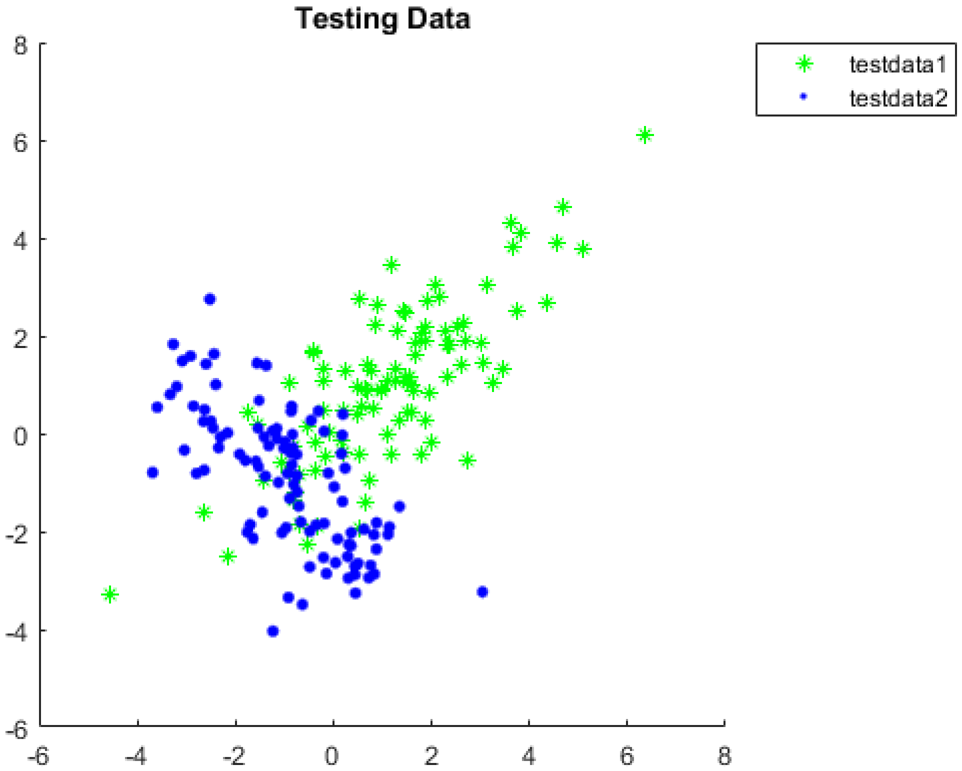
Training Data

```
figure('Name','Training Data')
scatter(traindata1(:,1),traindata1(:,2),'*','green')
hold on
scatter(traindata2(:,1),traindata2(:,2),200,'.','blue')
legend('traindata1','traindata2','Location','northeastoutside')
title('Training Data')
snapnow
```



Testing Data

```
figure('Name','Testing Data')
scatter(testdata1(:,1),testdata1(:,2),'*','green')
hold on
scatter(testdata2(:,1),testdata2(:,2),200,'.','blue')
legend('testdata1','testdata2','Location','northeastoutside')
title('Testing Data')
snapnow
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                     Linear Discriminant Analysis(LDA)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Estimated mean and covariance

```
Xtrain = [traindata1; traindata2];
Ytrain = [y1(1:100); y2(1:100)];
mu1est = mean(Xtrain(1:100,:));
mu2est = mean(Xtrain(101:200,:));
muest = [repmat(mu1est,100,1); repmat(mu2est,100,1)];
sigmaest = (1/(200-1))*(Xtrain - muest)'*(Xtrain - muest);
fprintf('Estimated covariance matrix (LDA) is:');
sigmaest
```

```
Estimated covariance matrix (LDA) is:
sigmaest =

    2.4633    0.6908
    0.6908    2.7316
```

Calculating weight and bias

```
fprintf('Weights using LDA are:');
w = (mu1 - mu2)*inv(sigmaest)
```

```
fprintf('Bias using LDA is:');
b = 0.5*(mu2*inv(sigmaest)*mu2') - 0.5*(mu1*inv(sigmaest)*mu1') + ...
    log((100/200)/(100/200))
```

```
Weights using LDA are:
w =

    0.6529    0.5671

Bias using LDA is:
b =

     0
```

Classification accuracy

```
Xtest = [testdata1; testdata2];
Ytest = [y1(101:end); y2(101:end)];
Ylearned = Xtest*w' + b;
for i = 1:max(size(Ylearned))
    if Ylearned(i) > 0
        Ylearned(i) = 1;
    else
        Ylearned(i) = -1;
    end
end
correcty = (Ylearned == Ytest);
fprintf('Accuracy of classification for LDA in percentage:');
accuracy = (sum(correcty)/max(size(Ytest)))*100
```

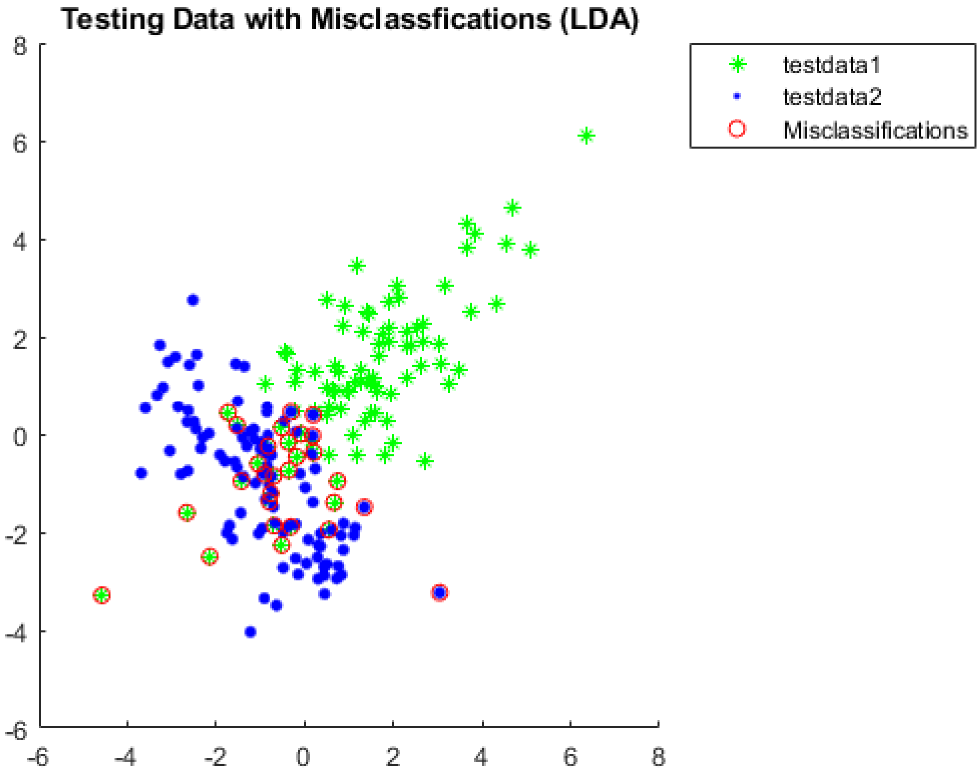
```
Accuracy of classification for LDA in percentage:
accuracy =

   85.5000
```

Misclassifications

```
figure('Name','Testing Data with Misclassfications (LDA)')
scatter(testdata1(:,1),testdata1(:,2),'*','green')
hold on
scatter(testdata2(:,1),testdata2(:,2),200,'.','blue')
title('Testing Data with Misclassfications (LDA)')
hold on
Misclass = [0 0];
for i=1:200
    if correcty(i) == 0
        Misclass = [Misclass; Xtest(i,:)];
    end
end
Misclass = Misclass(2:end,:);
scatter(Misclass(:,1),Misclass(:,2),'o','red')
legend('testdata1','testdata2','Misclassifications','Location',...
    'northeastoutside')
```

```
snapnow
hold off
wlda = w;
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                          Logistic Regression(LR)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

IRLS

```
Ytrainlr = Ytrain;
for i = 1:max(size(Ytrain))
    if Ytrainlr(i)==-1
        Ytrainlr(i) = 0;
    end
end
w = [0 0 0];
Xtrainlr = [ones(max(size(Xtrain)),1) Xtrain];
m = max(size(Xtrain));
a = [1 1 1];
j=1;
while sum(abs(a-w)) > 0.000000000000001
    a = w;
    for i = 1:m
        p(i,1) = exp(Xtrainlr(i,:)*w')/(1 + exp(Xtrainlr(i,:)*w'));
        s(i,1) = p(i)*(1-p(i));
        z(i,1) = (Xtrainlr(i,:)*w') + ((Ytrainlr(i) - p(i))./s(i));
    end
    S = diag(s);
    w = inv(Xtrainlr'*S*Xtrainlr)*Xtrainlr'*S*z;
    w = w';
end
fprintf('Weights using LR are:');
w
```

```
Weights using LR are:
w =

    0.3201    0.9710    0.6497
```

Classification accuracy

```
Ytestlr = Ytest;
Xtestlr = [ones(max(size(Xtest)),1) Xtest];
Ylearnedlr = Xtestlr*w' + b;
for i = 1:max(size(Ylearnedlr))
    if Ylearnedlr(i) > 0
        Ylearnedlr(i) = 1;
    else
        Ylearnedlr(i) = -1;
    end
end
correcty = (Ylearnedlr == Ytestlr);
fprintf('Accuracy of classification for LR in percentage:');
accuracy = (sum(correcty)/max(size(Ytest)))*100
```

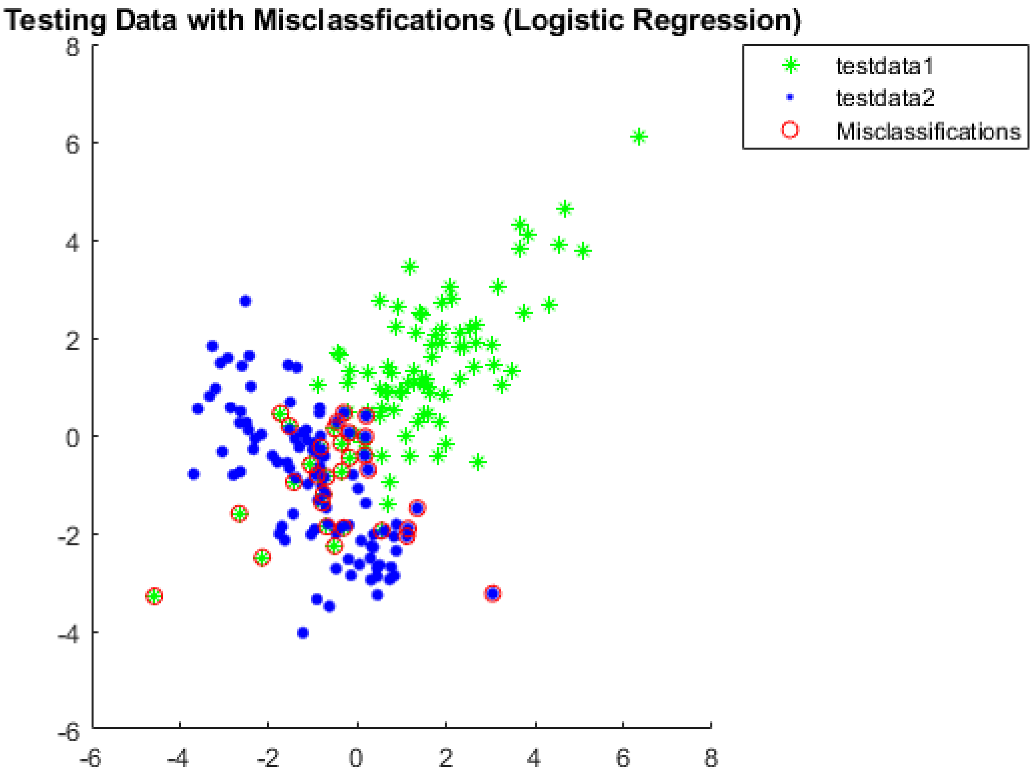
```
Accuracy of classification for LR in percentage:
accuracy =

   84.5000
```

Misclassifications

```
figure('Name','Testing Data with Misclassfications (Logistic Regression)')
scatter(testdata1(:,1),testdata1(:,2),'*','green')
hold on
scatter(testdata2(:,1),testdata2(:,2),200,'.','blue')
title('Testing Data with Misclassfications (Logistic Regression)')
hold on
Misclass = [0 0 0];
for i=1:200
    if correcty(i) == 0
        Misclass = [Misclass; Xtestlr(i,:)];
    end
end
Misclass = Misclass(2:end,:);
scatter(Misclass(:,2),Misclass(:,3),'o','red')
legend('testdata1','testdata2','Misclassifications','Location',...
    'northeastoutside')
snapnow
```

```
hold off
wlr = w;
```



Testing Data with Misclassfications (Logistic Regression)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%                 Quadratic Discriminant Analysis(QDA)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Estimating loggodds

```
mu1est = mean(Xtrain(1:100,:));
mu2est = mean(Xtrain(101:end,:));
sigma1est = (1/(100-1))*(Xtrain(1:100,:) - mu1est)'*(Xtrain(1:100,:) - mu1est);
sigma2est = (1/(100-1))*(Xtrain(101:end,:) - mu2est)'*(Xtrain(101:end,:) - mu2est);
for i = 1:max(size(Xtest))
    lnpr(i) = 0.5*(Xtest(i,:) - mu2)*inv(sigma2est)*(Xtest(i,:) - mu2)'...
    - 0.5*(Xtest(i,:)- mu1)*inv(sigma1est)*(Xtest(i,:) - mu1)' + 0.5*...
    log(det(sigma2est))- 0.5*log(det(sigma1est)) + log((100/200)/...
    (100/200));
end
lnpr = lnpr';
```

Classification Accuracy

```
for i = 1:max(size(lnpr))
    if lnpr(i) > 0
        Ylearnedqda(i,1) = 1;
    else
```

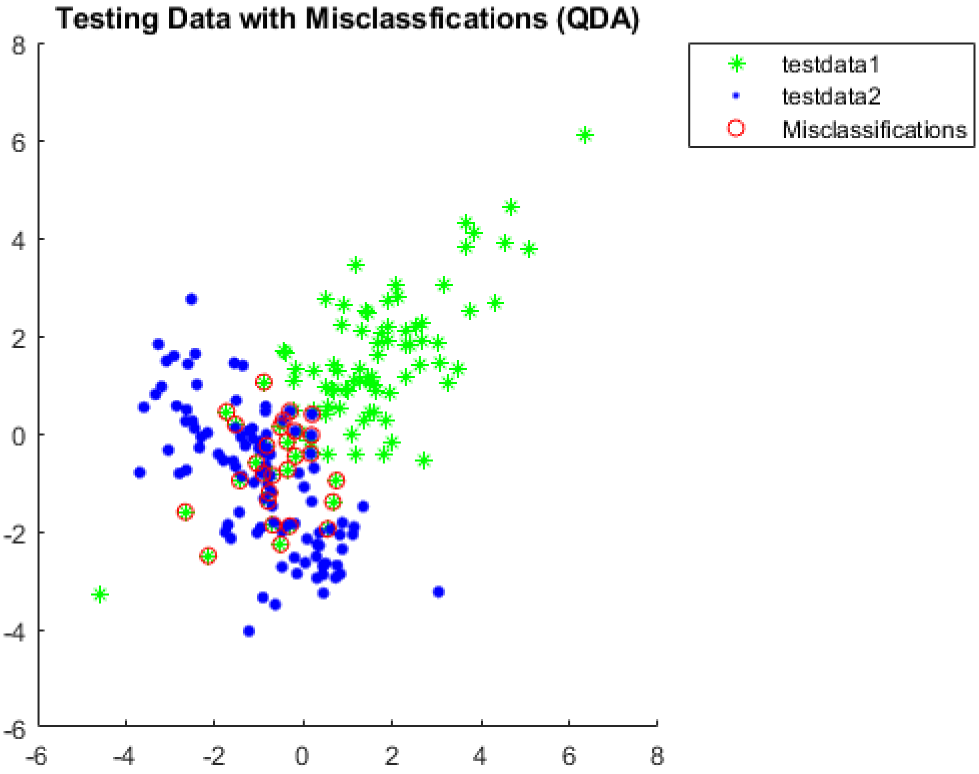
```
        Ylearnedqda(i,1) = -1;
    end
end
correcty = (Ylearnedqda == Ytest);
fprintf('Accuracy of classification for QDA in percentage:');
accuracy = (sum(correcty)/max(size(Ytest)))*100
```

```
Accuracy of classification for QDA in percentage:
accuracy =

    86
```

Misclassifications

```
figure('Name','Testing Data with Misclassfications (QDA)')
scatter(testdata1(:,1),testdata1(:,2),'*','green')
hold on
scatter(testdata2(:,1),testdata2(:,2),200,'.','blue')
title('Testing Data with Misclassfications (QDA)')
hold on
Misclass = [0 0];
for i=1:200
    if correcty(i) == 0
        Misclass = [Misclass; Xtest(i,:)];
    end
end
Misclass = Misclass(2:end,:);
scatter(Misclass(:,1),Misclass(:,2),'o','red')
legend('testdata1','testdata2','Misclassifications','Location',...
    'northeastoutside')
snapnow
hold off
wqda = w;
```



Published with MATLAB® R2018a