

- [Part A](#)
- [part B](#)
- [Part C](#)

```
close all
clear all

warning('off', 'all')

%Data
id = 1:1:20;

hrs = [0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 1.75, 2.00, 2.25, 2.50, 2.75, 3.00, ...
       3.25, 3.50, 4.00, 4.25, 4.50, 4.75, 5.00, 5.50];
pss = [0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1];

x = hrs;
y = pss;
```

## Part A

```
%minimize J
% -1/20 * sum(pss[i] * ln(1/(1 + exp(-(w0 + w1*hrs[i])))) + (1- pss[i]) * ln(1 - 1/(1 + exp(-(w0 +
w1*hrs[i])))))
w0 = -4.077;
w1 = 1.5046;

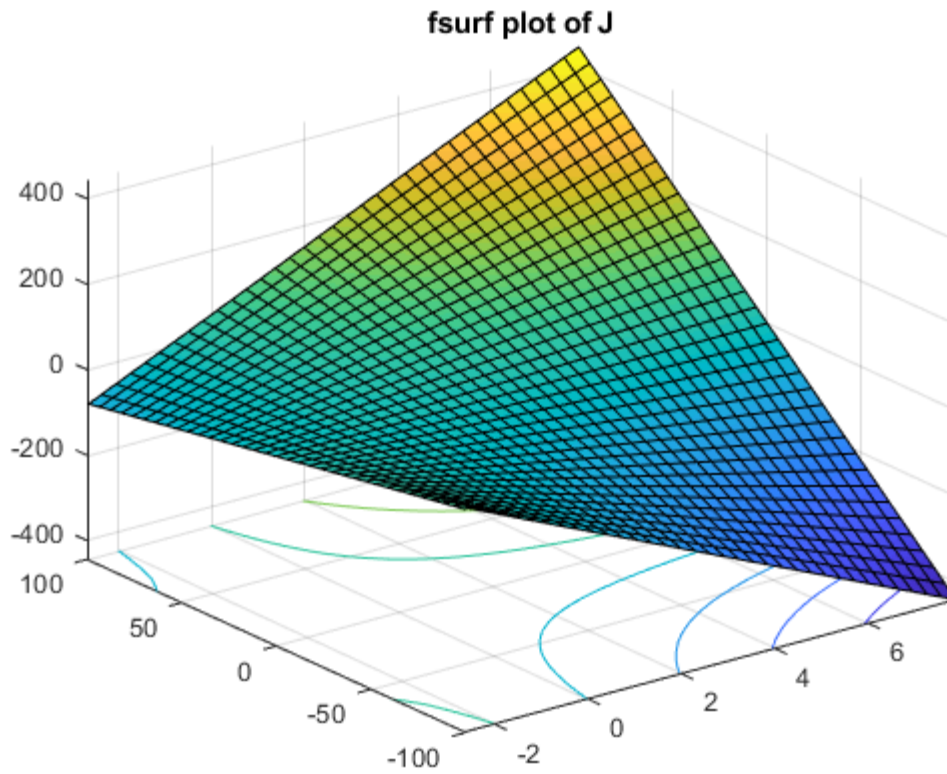
for i = 1:length(hrs)
    Jtemp = -1/20 * sum(pss(i) * log(1/(1 + exp(-(w0 + w1*hrs(i)))))) + (1- pss(i)) * log(1 - 1/(1 +
exp(-(w0 + w1*hrs(i))))));
    J(i) = Jtemp;
end
for ii = 1:20
    t = -10;
    g = -10;
    fsurf(@(x,y) -1/20 * sum(y * log(1/(1 + exp(-(t + g*x)))))) + (1- y) * log(1 - 1/(1 + exp(-(t +
g*x))))), [-100 100 -100 100], 'ShowContours','on')
    hold on
    t = t+1;
    g = g+1;
end
title('fsurf plot of J')

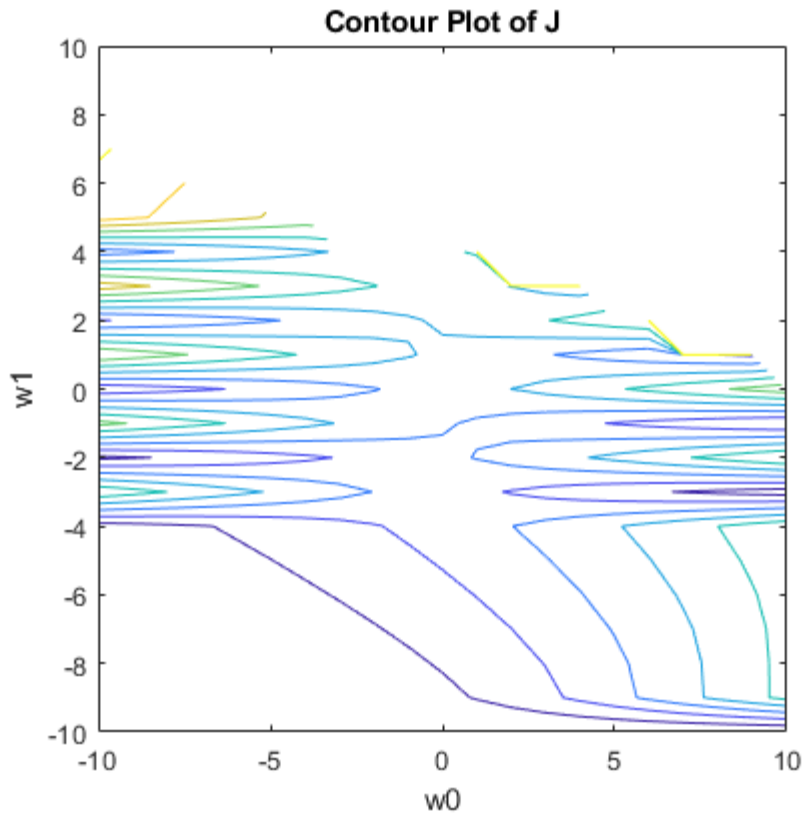
w_m = -10:10;
w2_m = -10:10;

[X1, X2] = meshgrid(w_m, w2_m);

for i = 1:length(hrs)
    Jtemp2 = -1/20 * sum(pss(i) * log(1./(1 + exp(-(X1 + X2.*hrs(i)))))) + (1- pss(i)) * log(1 -
1./(1 + exp(-(X1 + X2.*hrs(i))))));
    Jtemp3(i+1,:) = Jtemp2;
end
```

```
figure;  
contour(X1,X2,Jtemp3)  
axis([-10 10 -10 10])  
axis square  
xlabel('w0')  
ylabel('w1')  
title('Contour Plot of J')
```





## part B

```

w0 = 0;
w1 = -4;
alpha = 2;

for i = 1:length(hrs)
    dw0 = 1/length(hrs)*sum((-1 + y(i) + y(i) + y(i) * exp(-w0 - w1*x(i)))/(exp(-w0-w1*x(i)) + 1));

    dw1 = 1/length(hrs)*sum((-x(i) * (-y(i) + 1) + y(i)*x(i)*exp(-w0-w1*(x(i))))/ (exp(-w0-w1*x(i))
+ 1));

    w0new = w0 - alpha*dw0;
    w1new = w1 - alpha*dw1;
end

w0tot = [w0 w0new];
w1tot = [w1 w1new];

for kk = 1:19
    for i = 1:length(hrs)
        dw0 = 1/length(hrs)*sum((-1 + y(i) + y(i) + y(i) * exp(-w0new - w1new*x(i)))/(exp(-
w0new-w1new*x(i)) + 1));

        dw1 = 1/length(hrs)*sum((-x(i) * (-y(i) + 1) + y(i)*x(i)*exp(-w0new-w1new*(x(i))))/
(exp(-w0new-w1new*x(i)) + 1));

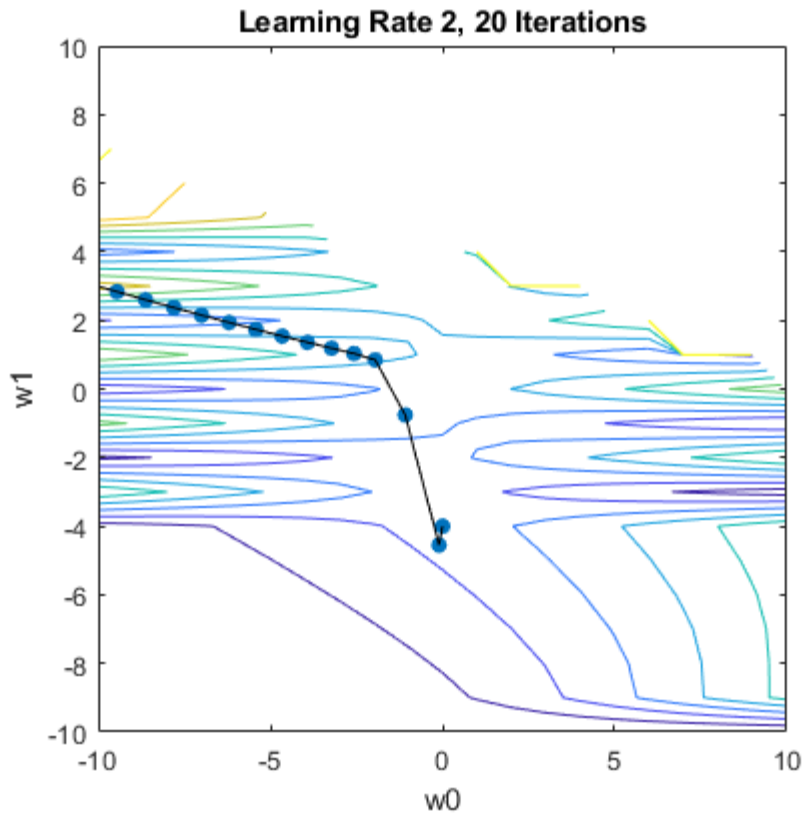
        w0new = w0new - alpha*dw0;
        w1new = w1new - alpha*dw1;
    end
    w0tot = [w0tot w0new];
    w1tot = [w1tot w1new];
end

```

```

figure;
contour(X1,X2,Jtemp3)
axis([-10 10 -10 10])
axis square
hold on
scatter(w0tot,w1tot,'filled')
hold on
plot(w0tot,w1tot, '-k')
title('Learning Rate 2, 20 Iterations')
xlabel('w0')
ylabel('w1')

```



## Part C

```

w0C = 0;
w1C = -4;
alpha = 0.12;

for i = 1:length(hrs)
    dw0 = 1/length(hrs)*sum((-1 + y(i) + y(i) + y(i) * exp(-w0 - w1*x(i)))/(exp(-w0-w1*x(i)) + 1));

    dw1 = 1/length(hrs)*sum((-x(i) * (-y(i) + 1) + y(i)*x(i)*exp(-w0-w1*(x(i))))/ (exp(-w0-w1*x(i))
+ 1));

    w0newC = w0 - alpha*dw0;
    w1newC = w1 - alpha*dw1;
end

w0totC = [w0C w0newC];
w1totC = [w1C w1newC];

```

```

for kk = 1:100
    for i = 1:length(hrs)
        dw0 = 1/length(hrs)*sum((-1 + y(i) + y(i) + y(i) * exp(-w0newC - w1newC*x(i)))/(exp(-w0newC-w1newC*x(i)) + 1));

        dw1 = 1/length(hrs)*sum(-(-x(i) * (-y(i) + 1) + y(i)*x(i)*exp(-w0newC-w1newC*(x(i))))/(exp(-w0newC-w1newC*x(i)) + 1));

        w0newC = w0newC - alpha*dw0;
        w1newC = w1newC - alpha*dw1;
    end
    w0totC = [w0totC w0newC];
    w1totC = [w1totC w1newC];
end

figure;
contour(X1,X2,Jtemp3)
axis([-10 10 -10 10])
axis square
hold on
scatter(w0totC,w1totC,'filled')
hold on
plot(w0totC,w1totC, '-k')

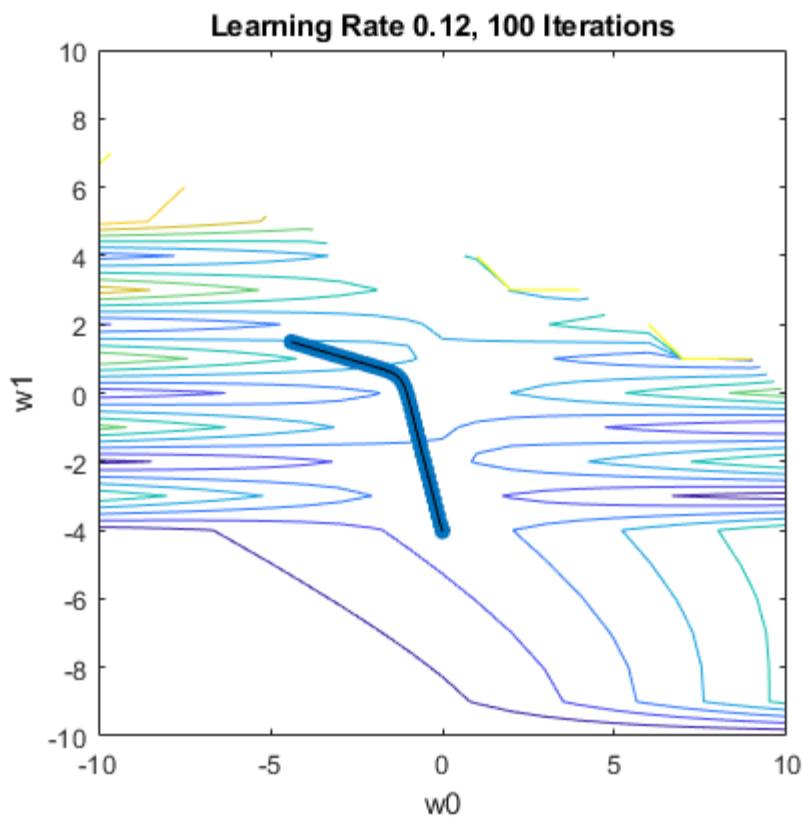
title('Learning Rate 0.12, 100 Iterations')
xlabel('w0')
ylabel('w1')

disp('I think there is something wrong with my equation, but even so I got close to the optimal values')
disp(w0newC)
disp(w1newC)

```

I think there is something wrong with my equation, but even so I got close to the optimal values  
-4.4030

1.4920



## Contents      Problem 2

- [prob2](#)
- [Part A](#)
- [Part B](#)

### prob2

```
clear all
close all
```

### Part A

```
opts = detectImportOptions('biopsy_data_missing_values.csv', 'NumHeaderLines', 1);
preview('biopsy_data_missing_values.csv', opts)

A = readtable('biopsy_data_missing_values.csv', 'HeaderLines', 1);

for i = 1:height(A)
    if isempty(A.Var2{i})
        A.Var2{i} = 'Irregular';
    end
end
```

```
nanind = find(isnan(A.Var1));
var1tmp = A.Var1;
var1tmp(nanind(1)) = 6;
var1tmp(nanind(2)) = 10;
A.Var1 = var1tmp;
```

A

ans =

8×7 table

Var1	Var2	Var3	Var4	Var5	Var6	Var7
1	'Circle'	'Large'	'Convex'	'Smooth'	'Dark'	'Malignant'
2	'Circle'	'Large'	'Concave'	'Smooth'	'Dark'	'Malignant'
3	'Circle'	'Large'	'Flat'	'Smooth'	'Red'	'Malignant'
4	' '	'Small'	'Concave'	'Rough'	'Dark'	'Malignant'
5	'Circle'	'Large'	'Flat'	'Rough'	'Neutral'	'Malignant'
NaN	'Circle'	'Large'	'Concave'	'Rough'	'Dark'	'Malignant'
7	'Circle'	'Large'	'Convex'	'Smooth'	'Neutral'	'Benign'
8	' '	'Large'	'Concave'	'Smooth'	'Red'	'Benign'

A =

12×7 table

Var1	Var2	Var3	Var4	Var5	Var6	Var7
1	'Circle'	'Large'	'Convex'	'Smooth'	'Dark'	'Malignant'
2	'Circle'	'Large'	'Concave'	'Smooth'	'Dark'	'Malignant'
3	'Circle'	'Large'	'Flat'	'Smooth'	'Red'	'Malignant'
4	'Irregular'	'Small'	'Concave'	'Rough'	'Dark'	'Malignant'
5	'Circle'	'Large'	'Flat'	'Rough'	'Neutral'	'Malignant'
6	'Circle'	'Large'	'Concave'	'Rough'	'Dark'	'Malignant'
7	'Circle'	'Large'	'Convex'	'Smooth'	'Neutral'	'Benign'
8	'Irregular'	'Large'	'Concave'	'Smooth'	'Red'	'Benign'
9	'Triangle'	'Small'	'Convex'	'Rough'	'Dark'	'Benign'
10	'Circle'	'Large'	'Flat'	'Smooth'	'Neutral'	'Benign'
11	'Irregular'	'Large'	'Concave'	'Smooth'	'Dark'	'Benign'
12	'Irregular'	'Large'	'Concave'	'Smooth'	'Red'	'Benign'

## Part B

```

ns1 = {'Irregular', 'Large', 'Convex', 'Rough', 'Neutral'};
ns2 = {'Irregular', 'Small', 'Flat', 'Rough', 'Red'};
ns3 = {'Circle', 'Large', 'Concave', 'Smooth', 'Neutral'};
ns4 = {'Circle', 'Large', 'Convex', 'Smooth', 'Dark'};
ns5 = {'Triangle', 'Large', 'Concave', 'Smooth', 'Neutral'};

nstot = {ns1 ns2 ns3 ns4 ns5};

%for ns1 ex

A1 = A(1:6,:); %malignant table (pos)
A2 = A(7:12,:); %benign table (neg)
pc1 = 0.5;
pc2 = 0.5;

for i = 1:length(nstot)
    pshape1 = length(find(strcmp(A1.Var2, nstot{i}{1}))) / height(A1);
    prad1 = length(find(strcmp(A1.Var3, nstot{i}{2}))) / height(A1);
    pconcav1 = length(find(strcmp(A1.Var4, nstot{i}{3}))) / height(A1);
    ptext1 = length(find(strcmp(A1.Var5, nstot{i}{4}))) / height(A1);
    pcoll1 = length(find(strcmp(A1.Var6, nstot{i}{5}))) / height(A1);

    posprob(i) = log(pshape1 * prad1 * pconcav1 * ptext1 * pcoll1 * pc1);

    nshape1 = length(find(strcmp(A2.Var2, nstot{i}{1}))) / height(A2);
    nrad1 = length(find(strcmp(A2.Var3, nstot{i}{2}))) / height(A2);
    nconcav1 = length(find(strcmp(A2.Var4, nstot{i}{3}))) / height(A2);
    ntext1 = length(find(strcmp(A2.Var5, nstot{i}{4}))) / height(A2);
    ncoll1 = length(find(strcmp(A2.Var6, nstot{i}{5}))) / height(A2);

    negprob(i) = log(nshape1 * nrad1 * nconcav1 * ntext1 * ncoll1 * pc2);

end

for ii = 1:5
    if negprob(ii) > posprob(ii)

```



```

        class{ii} = 'benign';
    else
        class{ii} = 'malignant';
        %err on this side, to prompt pt to another test
    end
end
end

samplenames = 1:5;
LogNegProb = negprob;
LogPosProb = posprob;
varnames = {'Sample', 'LogNegProb', 'LogPosProb', 'Classification'};
ResultTable = table(samplenames', LogNegProb', LogPosProb', class', 'VariableNames', varnames)

disp('Sample 5 had was one of two triangle shapes, which was not contained in the Malignant group.')
```

ResultTable =

5×4 table

Sample	LogNegProb	LogPosProb	Classification
1	-5.5576	-6.9439	'benign'
2	-7.8602	-7.8602	'malignant'
3	-3.9482	-4.2358	'benign'
4	-Inf	-Inf	'malignant'
5	-4.6413	-Inf	'benign'

Sample 5 had was one of two triangle shapes, which was not contained in the Malignant group.

- [prob 3](#)
- [Part B](#)
- [Part C](#)
- [Echoing Values](#)

## prob 3

```
clear all
close all

opts = detectImportOptions('iris_dataset.csv', 'NumHeaderLines', 1);
preview('iris_dataset.csv', opts)

A = readtable('iris_dataset.csv', 'HeaderLines', 1);
```

ans =

8×5 table

Var1	Var2	Var3	Var4	Var5
5.1	3.5	1.4	0.2	'Iris-setosa'
4.9	3	1.4	0.2	'Iris-setosa'
4.7	3.2	1.3	0.2	'Iris-setosa'
4.6	3.1	1.5	0.2	'Iris-setosa'
5	3.6	1.4	0.2	'Iris-setosa'
5.4	3.9	1.7	0.4	'Iris-setosa'
4.6	3.4	1.4	0.3	'Iris-setosa'
5	3.4	1.5	0.2	'Iris-setosa'

## Part B

```
%split A into 3
indis = find(strcmp(A.Var5, 'Iris-setosa'));
indver = find(strcmp(A.Var5, 'Iris-versicolor'));
indvir = find(strcmp(A.Var5, 'Iris-virginica'));

A1 = A(1:50,:);
A2 = A(51:100,:);
A3 = A(101:150,:);

scatter(A1.Var1, A1.Var4, 'r')
hold on
scatter(A2.Var1, A2.Var4, 'g')
hold on
scatter(A3.Var1, A3.Var4, 'b')
axis([0 10 0 10])
hold on
```

```

xlabel('Sepal Length')
ylabel('Petal Width')

x1 = 0:0.05:10;
x4 = 0:0.05:10;
[X1, X2] = meshgrid(x1,x4);

sigma1 = 0.2;
sigma4 = sigma1;
amesh1 = 0;
amesh1_x4 = 0;
m = 50;
pc1 = 50/150;
pc2 = 50/150;
pc3 = 50/150;

for i = 1:50
    am1 = exp(-(X1 - A1.Var1(i)).^2./(2*sigma1^2));
    amesh1 = amesh1+am1;
end
for ii = 1:50
    am4 = exp(-(X2 - A1.Var4(ii)).^2./(2*sigma4^2));
    amesh1_x4 = amesh1+am4;
end

meshtotx1a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1;
meshtotx4a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1_x4;
totmesha1 = meshtotx1a1*meshtotx4a1 *pc1;

contour(X1, X2, totmesha1, 0:0.03:0.15)
%abc.levels = [0:0.03:0.15]

%repeat for other groups

for i = 1:50
    am1 = exp(-(X1 - A2.Var1(i)).^2./(2*sigma1^2));
    amesh1 = amesh1+am1;
end
for ii = 1:50
    am4 = exp(-(X2 - A2.Var4(ii)).^2./(2*sigma4^2));
    amesh1_x4 = amesh1+am4;
end
meshtotx1a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1;
meshtotx4a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1_x4;
totmesha1 = meshtotx1a1*meshtotx4a1 *pc1;
contour(X1, X2, totmesha1, 0:0.03:0.15)

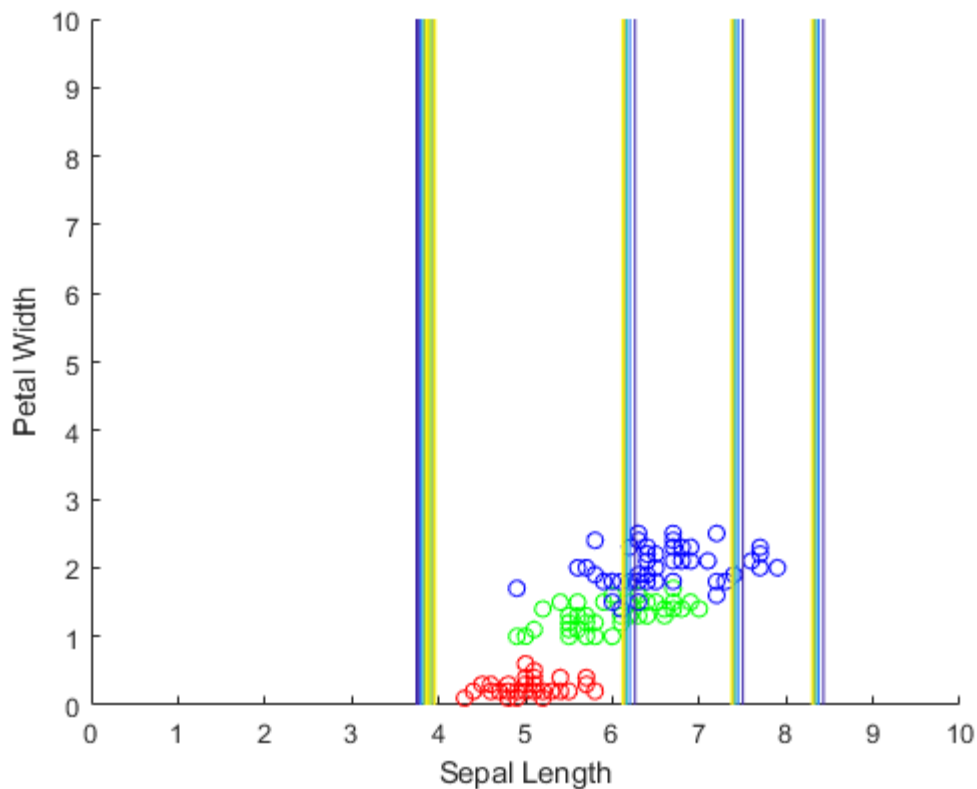
for i = 1:50
    am1 = exp(-(X1 - A3.Var1(i)).^2./(2*sigma1^2));
    amesh1 = amesh1+am1;
end
for ii = 1:50
    am4 = exp(-(X2 - A3.Var4(ii)).^2./(2*sigma4^2));
    amesh1_x4 = amesh1+am4;
end
meshtotx1a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1;

```

```

meshtotx4a1 = 1/m*1/(sigma1*sqrt(2*pi)) .* amesh1_x4;
totmesha1 = meshtotx1a1*meshtotx4a1 *pc1;
contour(X1, X2, totmesha1, 0:0.03:0.15)

```



## Part C

```

x1new = [5.5, 7, 6.5, 6.2];
x4new = [0.5, 1.8, 1.5, 1.7];
bm2 = 50*[];
bm4 = 50*[];
bm5 = 5*[];
bm6 = 5*[];
%c1
for i = 1:length(x1new)
    for kk = 1:50
        bm1 = exp(-(x1new(i) - A1.Var1(kk))^2/(2*sigma1^2));
        bm2(kk) = bm1;

        end
        bm5(i) = sum(bm2);
    end
    for i = 1:length(x1new)
        for kk = 1:50
            bm3 = exp(-(x4new(i) - A1.Var4(kk))^2/(2*sigma1^2));
            bm4(kk) = bm3;

            end
            bm6(i) = sum(bm4);
        end
        %now have the sum of gauss
        %mult k, attributes and P(c1)

class1Q = (1/m*1/(sigma1*sqrt(2*pi)) * bm5) .* (1/m*1/(sigma1*sqrt(2*pi)) * bm6) .* pc1;

```

```

%c2
for i = 1:length(xlnew)
    for kk = 1:50
        bm1 = exp(-(xlnew(i) - A2.Var1(kk))^2/(2*sigma1^2));
        bm2(kk) = bm1;

    end
    bm5(i) = sum(bm2);
end
for i = 1:length(xlnew)
    for kk = 1:50
        bm3 = exp(-(x4new(i) - A2.Var4(kk))^2/(2*sigma1^2));
        bm4(kk) = bm3;
    end
    bm6(i) = sum(bm4);
end
%now have the sum of gauss

class2Q = (1/m*1/(sigma1*sqrt(2*pi)) * bm5) .* (1/m*1/(sigma1*sqrt(2*pi)) * bm6) .* pc2;

%c3
for i = 1:length(xlnew)
    for kk = 1:50
        bm1 = exp(-(xlnew(i) - A3.Var1(kk))^2/(2*sigma1^2));
        bm2(kk) = bm1;

    end
    bm5(i) = sum(bm2);
end
for i = 1:length(xlnew)
    for kk = 1:50
        bm3 = exp(-(x4new(i) - A3.Var4(kk))^2/(2*sigma1^2));
        bm4(kk) = bm3;
    end
    bm6(i) = sum(bm4);
end
%now have the sum of gauss

class3Q = (1/m*1/(sigma1*sqrt(2*pi)) * bm5) .* (1/m*1/(sigma1*sqrt(2*pi)) * bm6) .* pc3;

for i = 1:length(xlnew)
    if class1Q(i) > class2Q(i) && class3Q(i)
        classification_tot{i} = 'Iris-sertosa';
    elseif class2Q(i) > class3Q(i) && class1Q(i)
        classification_tot{i} = 'Iris-versicolor';
    elseif class3Q(i) > class2Q(i) && class1Q(i)
        classification_tot{i} = 'Iris-virginica';
    end
end

SampleNumber = 1:4;

varNames = {'SampleNumber', 'Class1QProb', 'Class2QProb', 'Class3QProb', 'classification'};
ResultTable = table(SampleNumber', class1Q', class2Q', class3Q', classification_tot',

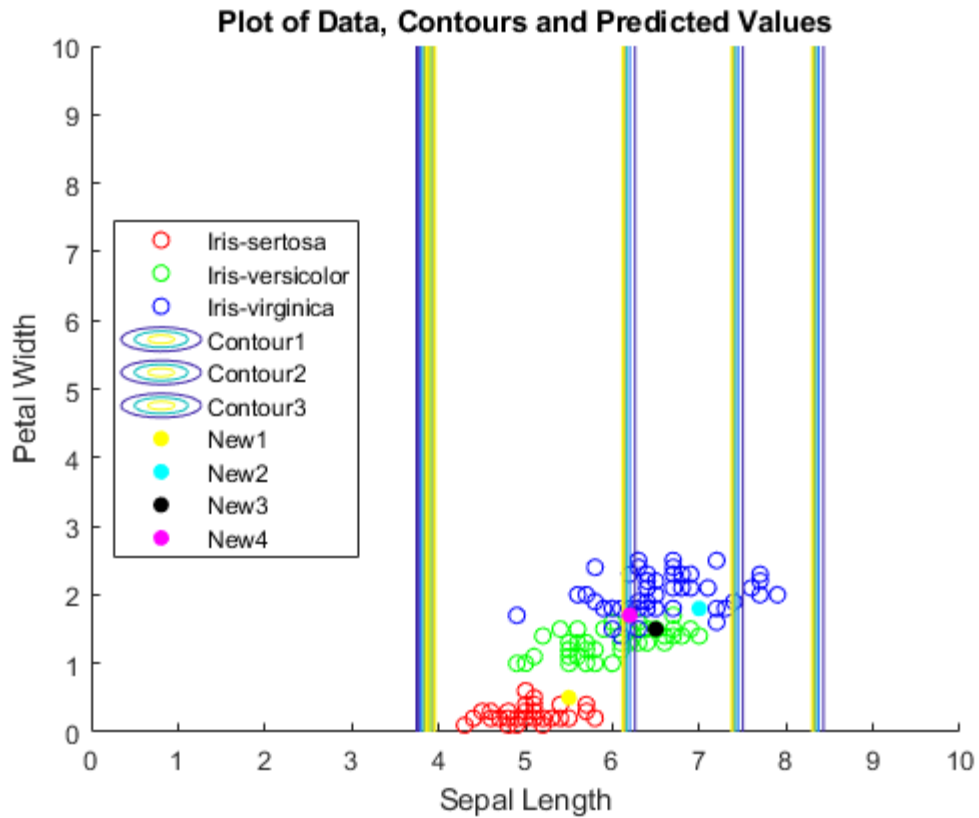
```

```

'VariableNames', varNames);
ae = {'y', 'c', 'k', 'm'};
for i = 1:length(xlnew)
    scatter(xlnew(i), x4new(i), ae{i}, 'filled')
    hold on
end

legend('Iris-sertosa', 'Iris-versicolor', 'Iris-virginica', 'Contour1', 'Contour2', 'Contour3',
'New1', 'New2', 'New3', 'New4', 'Location', 'west')
xlabel('Sepal Length')
ylabel('Petal Width')
title('Plot of Data, Contours and Predicted Values')

```



## Echoing Values

```

diary vjprob3.txt
echo on
ResultTable

disp('I recognize that my contours are incorrect, but I am not sure how to fix it')

echo off

```

ResultTable

ResultTable =

4×5 table

SampleNumber	Class1QProb	Class2QProb	Class3QProb	classification
--------------	-------------	-------------	-------------	----------------

1	0.12985	0.0029045	6.9147e-08	'Iris-sertosa'
2	1.4121e-19	0.017232	0.11379	'Iris-virginica'
3	6.9678e-11	0.16015	0.081672	'Iris-versicolor'
4	3.5286e-11	0.10984	0.15887	'Iris-virginica'

```
disp('I recognize that my contours are incorrect, but I am not sure how to fix it')
I recognize that my contours are incorrect, but I am not sure how to fix it
```

```
echo off
```