```
mu1 = [1, 1];
Sigma1 = [2, 0; 0, 1];
mu2 = [3, 2];
Sigma2 = [2, 0; 0, 1];

N1 = 35000;
N2 = 35000;
dataI1 = randperm(N1);
dataI2 = randperm(N2);

R1 = zeros(N1, 3);
R2 =  ones(N2, 3);
R1(:, 1:2) = mvnrnd(mu1, Sigma1, N1);
R2(:, 1:2) = mvnrnd(mu2, Sigma2, N2);
```

ابتدا میانگین و سیگمای دسته اعداد ۱ و ۲ مشخص شده ، پس‌ازآن تعداد هر دسته نیز مشخص گردیده است. گفتنی است که تعداد اعداد رندم تولیدشده می‌تواند متفاوت در نظر گرفته شود.

با استفاده از تابع randperm تعدادی عدد از ۱ تا بیشترین مقدار هر دسته عدد تولید خواهد شد که این اعداد تکراری نبوده و نمایانگر اندیس داده‌های انتخابی جهت آموزش یا آزمون می‌باشند. در موقع انتخاب هر عضو R1 یا R2 از مقادیر dataI1 و dataI2 جهت دسترسی به اندیس موردنظر استفاده خواهد شد این موضوع به انتخاب داده‌های آموزشی یا آزمون به‌صورت کاملاً رندم از هرکجای لیست عددی منجر می‌گردد.

```
trainP = 0.75;

[eSigma1, em1] = Covariance(R1(dataI1(1:trainP*N1), 1),
R1(dataI1(1:trainP*N1), 2));
[eSigma2, em2] = Covariance(R2(dataI2(1:trainP*N2), 1),
R2(dataI2(1:trainP*N2), 2));

plot(R1(dataI1(trainP*N1+1:N1), 1), R1(dataI1(trainP*N1+1:N1), 2), '+m')
plot(R2(dataI2(trainP*N2+1:N2), 1), R2(dataI2(trainP*N2+1:N2), 2), '+b')
```

درصد داده‌های آموزشی مشخص شده و با استفاده از تابع کوواریانس مقادیر سیگما۱، میانگین۱، سیگما۲ و میانگین۲ تخمین زده می‌شود. گفتنی است که میانگین نیز در داخل تابع کوواریانس محاسبه می‌شود و به برنامه بازگشت می‌شود. در ادامه داده‌های آزمون با استفاده از تابع plot در دو دسته ترسیم می‌گردند.

```
MinXY = min(min(min(R1(:, 1), R1(:, 2)), min(min(R2(:, 1), R2(:, 2)))));
MaxXY = max(max(max(R1(:, 1), R1(:, 2)), max(max(R2(:, 1), R2(:, 2)))));
X = MinXY:0.1:MaxXY;
Y = MinXY:0.1:MaxXY;

f1 = zeros(size(X, 1), size(X, 1));
f2 = zeros(size(X, 1), size(X, 1));
p1 = zeros(size(X, 1), size(X, 1));
p2 = zeros(size(X, 1), size(X, 1));
```

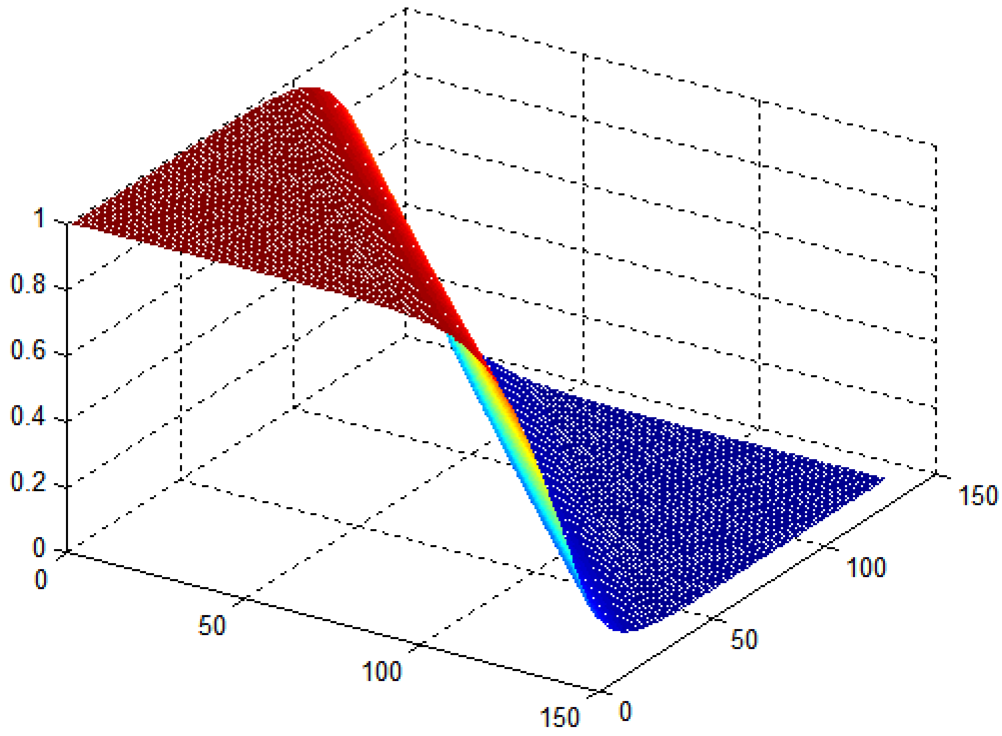
کمینه و بیشینهٔ مقادیر ستون اول و دوم بردارهای R1 و R2 جهت ترسیم و استفاده در حلقه‌های تکرار به دست آمده و متغیرهای f1، f2، p1 و p2 به‌صورت ماتریس‌های صفر مربعی تعریف می‌شوند.

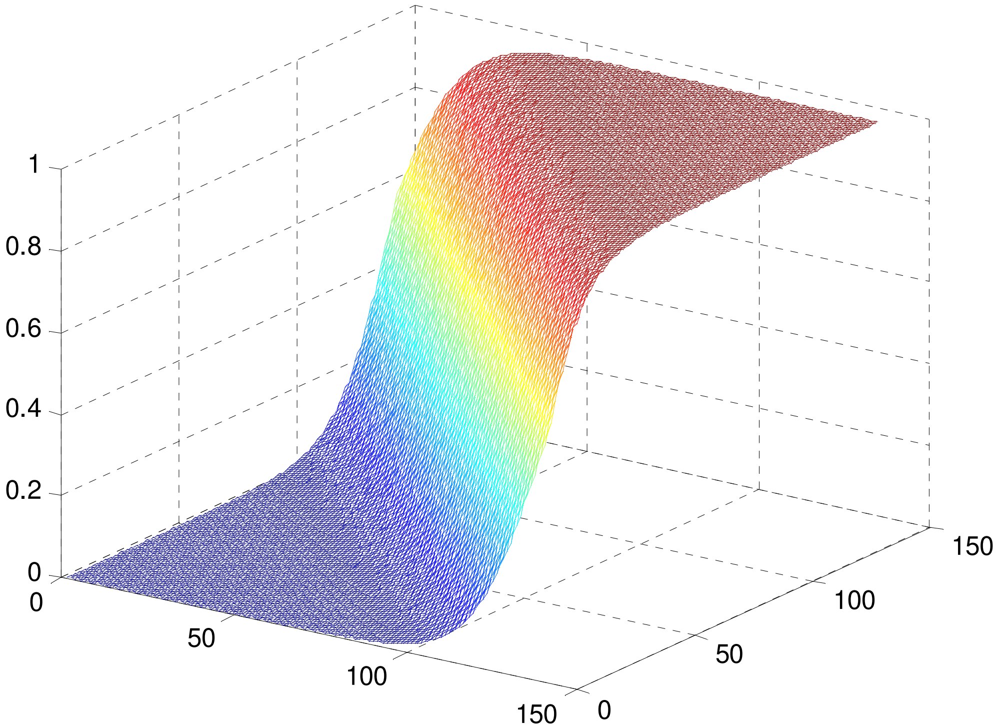
```
i = 1;
for y = MinXY:0.1:MaxXY
    j = 1;
    for x = MinXY:0.1:MaxXY
        f1(i, j) = mvar(x, y, em1(1), em1(2), eSigma1);
        f2(i, j) = mvar(x, y, em2(1), em2(2), eSigma2);
```

```
        p1(i, j) = f1(i, j) / (f1(i, j) + f2(i, j));
        p2(i, j) = f2(i, j) / (f1(i, j) + f2(i, j));
        j = j + 1;
    end;
    i = i + 1;
end;

contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, f1, 20);
contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, f2, 20);
xlabel('x1');
ylabel('x2');
axis([MinXY, MaxXY, MinXY, MaxXY]);
contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, p1-p2, 1);
```

در حلقه اصلی برنامه با استفاده از تابع mvar متناسب با سیگما و میانگین به هر مختصات x و y احتمالی به f1 و f2 نسبت داده می‌شود و نسبت f1/(f1+f2) و f1/(f1+f2) محاسبه می‌شوند. این دو نسبت بیانگر میزان اهمیت عدد رندم تولیدشده با توجه به هر یک از فراخوانی‌های تابع mvar بوده و در حقیقت اگر f1 بیشتر باشد داده در تشکیل نواحی فلت سطح ۱ مربوط به R1 نقش مؤثرتری نسبت تشکیل نواحی فلت سطح ۱ مربوط به R2 دارد. شکل زیر این موضوع را مشخص می‌نماید.






```
contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, f1, 20);
contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, f2, 20);
xlabel('x1');
ylabel('x2');
axis([MinXY, MaxXY, MinXY, MaxXY]);
contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, p1-p2, 1);
```

نمودارهای هم‌تراز f1 و f2 ترسیم گردیده و اختلاف p1-p2 به‌منظور ترسیم خط جداکننده مورد استفاده قرار می‌گیرد. P1-p2 اختلاف احتمالات را محاسبه می‌کند و دادن عدد ۱ به دستور contour باعث می‌شود که کمترین اختلاف بین p1 و p2 به‌عنوان مرز جداکننده ترسیم گردد.

```
Phat_C1 = numel(1:trainP*N1) / (numel(1:trainP*N1) + numel(1:trainP*N2));
Phat_C2 = numel(1:trainP*N2) / (numel(1:trainP*N1) + numel(1:trainP*N2));

e11 = abs(Sigma1-eSigma1);
e12 = abs(mu1-em1);
e21 = abs(Sigma2-eSigma2);
e22 = abs(mu2-em2);
```

احتمال کلاس ۱ و ۲ محاسبه می‌شود و پس‌ازآن اختلاف مقادیر واقعی و تخمین‌زده‌شدهٔ میانگین‌ها و سیگماها محاسبه می‌شود.

```
g11 = zeros(size(trainP*N1+1:N1, 2), 1);
g12 = zeros(size(trainP*N1+1:N1, 2), 1);
i = 1;
for j = trainP*N1+1:N1
    g11(i) = gi(em1, eSigma1, Phat_C1, [R1(dataI1(j), 1), R1(dataI1(j),
2)]);
    g12(i) = gi(em2, eSigma2, Phat_C2, [R1(dataI1(j), 1), R1(dataI1(j),
2)]);
    i = i + 1;
end
```

```
sum((g11 - g12) > 0 | (R1(dataI1(trainP*N1+1:N1),
3)))/(size(trainP*N1+1:N1, 2))

g21 = zeros(size(trainP*N2+1:N2, 2), 1);
g22 = zeros(size(trainP*N2+1:N2, 2), 1);
i = 1;
for j = trainP*N2+1:N2
    g21(i) = gi(em1, eSigma1, Phat_C1, [R2(dataI2(j), 1), R2(dataI2(j),
2)]);
    g22(i) = gi(em2, eSigma2, Phat_C2, [R2(dataI2(j), 1), R2(dataI2(j),
2)]);
    i = i + 1;
end
sum((g22 - g21) > 0 & (R2(dataI2(trainP*N2+1:N2),
3)))/(size(trainP*N2+1:N2, 2))
```

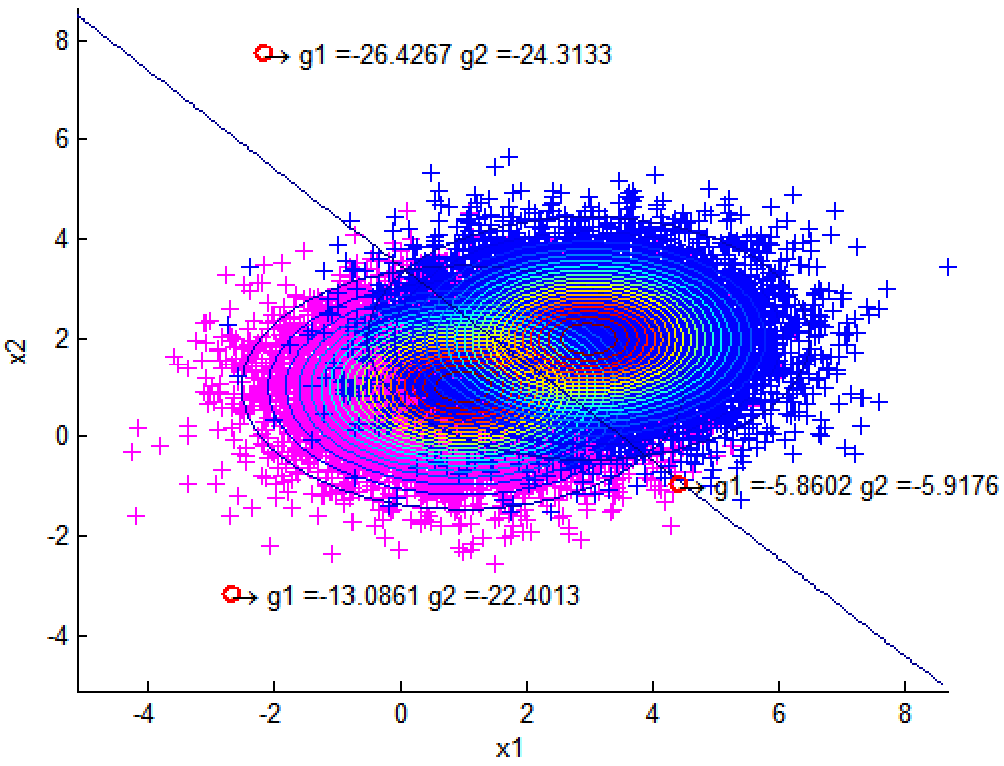
در این قسمت از کد، داده‌های آزمون به توابع g1 و g2 ارسال گردیده و پس‌ازآن اختلاف g1 و g2 به‌دست‌آمده و اگر این مقدار بزرگ‌تر از ۰ بود (متناسب با کلاس محاسبه فرق می‌کند) برای محاسبه درصد صحت پیش‌بینی کلاس داده‌های آزمون مورد استفاده قرار می‌گیرد.

```
[x, y] = ginput;
plot(x, y, 'or', 'LineWidth', 1.5);

for i = 1:size(x)
    g1 = gi(em1, eSigma1, Phat_C1, [x(i), y(i)]);
    g2 = gi(em2, eSigma2, Phat_C2, [x(i), y(i)]);
    text(x(i), y(i), strcat('\rightarrow', ' g1 = ', num2str(g1), ' g2 = ',
num2str(g2)));
end
```

در انتها با استفاده از تابع ginput می‌توانیم هر تعداد داده که خواستیم بر روی تصویر انتخاب نماییم و پس از زدن کلید Enter مقدار تعلق به هر کلاس نمایش داده می‌شود.

نتایج اجرای قسمت اول:



eSigma1 =

    2.0076   0.0004

    0.0004   0.9927

em1 =

    0.9990   0.9999

eSigma2 =

    2.0188   -0.0022

   -0.0022   0.9866

em2 =

    2.9842   1.9903

e11 =

    0.0076   0.0004

    0.0004   0.0073

e12 =

   1.0e-03 *

0.9651   0.0770

e21 =

0.0188   0.0022

0.0022   0.0134

e22 =

0.0158   0.0097

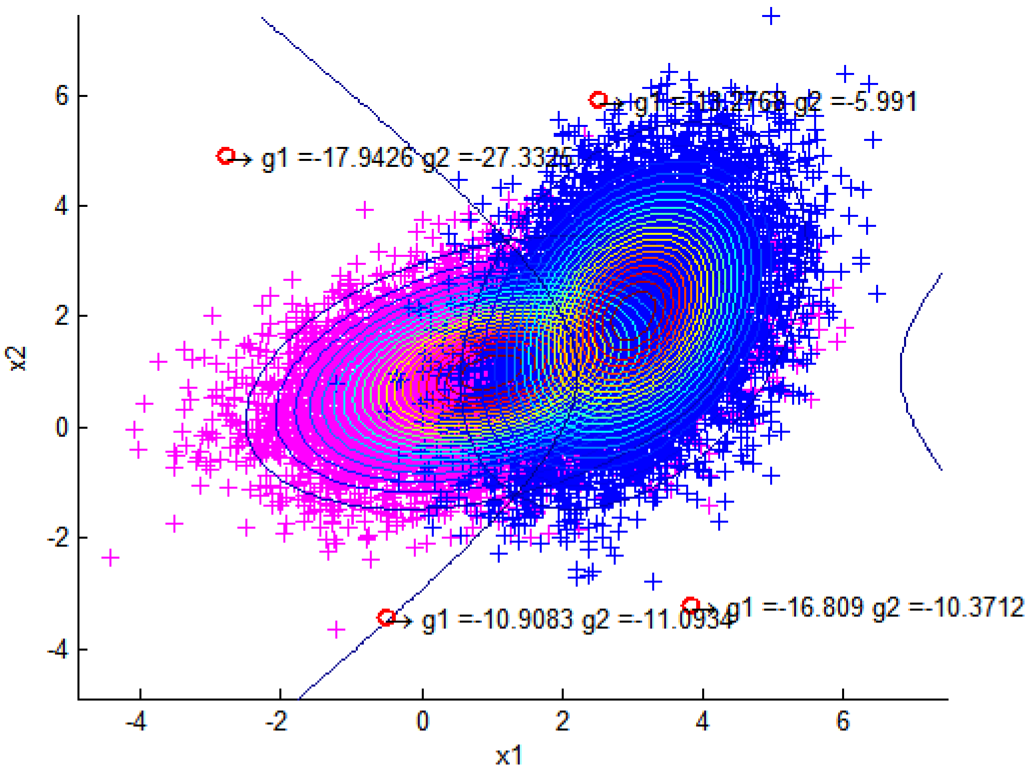
Pt1 =

0.8131

Pt2 =

0.8083

نتایج اجرای قسمت دوم:



eSigma1 =

1.9999   0.5019

0.5019   1.0057

em1 =

```
   0.9882    0.9965

eSigma2 =

   0.9925    0.4937

   0.4937    2.0036

em2 =

   3.0077    2.0156

e11 =

   0.0001    0.0019

   0.0019    0.0057

e12 =

   0.0118    0.0035

e21 =

   0.0075    0.0063

   0.0063    0.0036

e22 =

   0.0077    0.0156

Pt1 =

   0.7662

Pt2 =

   0.8462
```