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

% mu1 = [1, 1];
% Sigma1 = [2, 0.5; 0.5, 1];
% mu2 = [3, 2];
% Sigma2 = [1, 0.5; 0.5, 2];

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، پس از آن تعداد هر دسته نیز مشخص گردیده است. گفتنی است که تعداد اعداد رندم تولید شده 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 datal1 و datal2 جهت دسترسی به اندیس موردنظر استفاده خواهد شد. این موضوع به انتخاب داده‌های آموزشی یا آزمون به صورت 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 سیگما ۱، میانگین ۱، سیگما ۲ و میانگین ۲ 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.

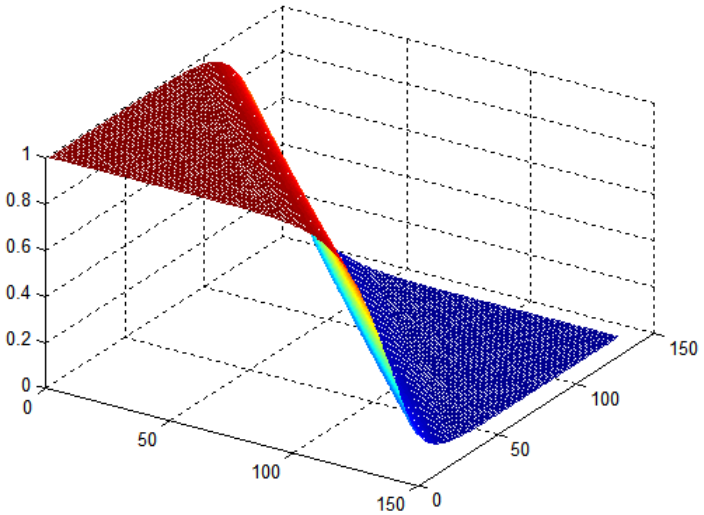
```
MinXY = min(min(min(R1(:, 1), R1(:, 2)), min(min(R2(:, 1), R2(:, 2)))));
MaxXY = max(max(max(R1(:, 1), R1(:, 2)), max(max(R2(:, 1), R2(:, 2)))));
X = MinXY:0.1:MaxXY;
Y = MinXY:0.1:MaxXY;

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));
```

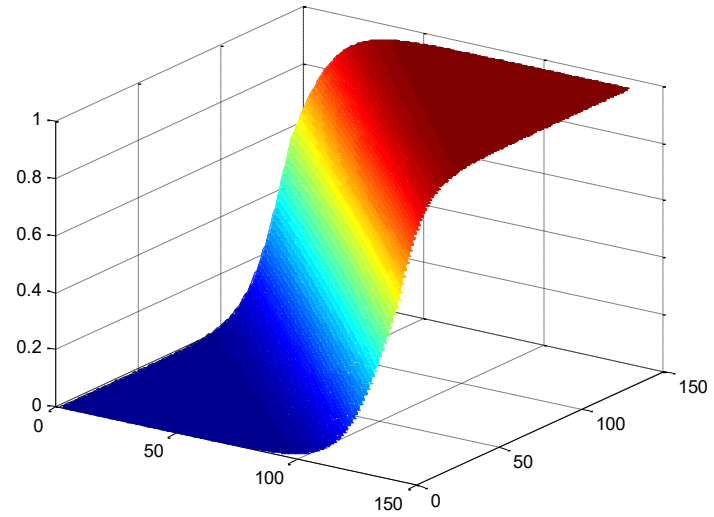
کمینه و بیشینه مقادیر ستون اول و دوم بردارهای R1 و R2 جهت ترسیم و استفاده در حلقه‌های تکرار به دست آمده و متغیرهای f1، f2، p1 و p2 به صورت ماتریس‌های صفر مربعی تعریف می‌شوند.

```
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
```

در حلقه اصلی برنامه، با استفاده از تابع mvar و بر اساس میانگین و ماتریس کوواریانس هر کلاس، برای هر مختصات x و y مقدار چگالی احتمال دو توزیع گاوسی محاسبه شده و در ماتریس‌های f1 و f2 قرار می‌گیرد. این مقادیر همان سطوحی هستند که در نمایش سه‌بعدی به صورت نواحی فلت دیده می‌شوند و نشان می‌دهند هر نقطه با چه شدت به کلاس مربوطه وابسته است. سپس نسبت‌های p1=f1/(f1+f2) و p2=f2/(f1+f2) محاسبه می‌شوند که بیانگر احتمال پسین هر کلاس هستند. اختلاف این دو مقدار یعنی p1-p2 برای تعیین مرز تصمیم استفاده می‌شود و مرز جداکننده دقیقاً در نقاطی قرار دارد که دو احتمال برابر می‌شوند. بنابراین برای ترسیم مرز تصمیم، کانتور سطح صفر تابع p1-p2 رسم می‌شود و این خط ناحیه تفکیک دو کلاس را مشخص می‌کند.






```
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, [0, 0]);
```

نمودارهای هم تراز f1 و f2 ترسیم گردیده و اختلاف p1-p2 برای تعیین مرز جداکننده مورد استفاده قرار می گیرد. مقدار p1-p2 اختلاف احتمال پسین دو کلاس را نشان می دهد و مرز تصمیم دقیقاً در نقاطی قرار دارد که این اختلاف برابر صفر است. بنابراین برای ترسیم خط جداکننده، کانتور سطح صفر تابع p1-p2 رسم می شود که ناحیه تفکیک دو کلاس را مشخص می کند.

```
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));

gt1 = zeros(length(Y), length(X));
gt2 = zeros(length(Y), length(X));

i = 1;
for y = MinXY:0.1:MaxXY
    j = 1;
    for x = MinXY:0.1:MaxXY
        gt1(i, j) = gi(mu1, Sigma1, Phat_C1, [x, y]);
        gt2(i, j) = gi(mu2, Sigma2, Phat_C2, [x, y]);
        j = j + 1;
    end
    i = i + 1;
end

contour(MinXY:0.1:MaxXY, MinXY:0.1:MaxXY, gt1-gt2, [0, 0], 'k', 'LineWidth', 2);
```

پس از محاسبه احتمال پیشین کلاس‌ها (Phat)، برای ترسیم مرز تصمیم تئوری، مقادیر gt1 و gt2 با استفاده از تابع ممیز بیز (gi) و پارامترهای واقعی توزیع‌ها محاسبه می‌شوند. رسم کانتور سطح صفر اختلاف این دو، مرز تئوری را به صورت یک خط مشکی ضخیم بر روی نمودار نمایش می‌دهد.

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

```
[testSigma1, testm1] = Covariance(R1(dataI1(trainP*N1+1:N1), 1),
R1(dataI1(trainP*N1+1:N1), 2))
[testSigma2, testm2] = Covariance(R2(dataI2(trainP*N2+1:N2), 1),
R2(dataI2(trainP*N2+1:N2), 2))

test_e11 = abs(Sigma1-testSigma1)
test_e12 = abs(mu1-testm1)
test_e21 = abs(Sigma2-testSigma2)
test_e22 = abs(mu2-testm2)
```

احتمال کلاس ۱ و ۲ محاسبه می شود و پس از آن اختلاف مقادیر واقعی و تخمین زده شده میانگین ها و سیگماها برای داده‌های آموزش محاسبه می شود. همچنین میانگین و ماتریس کوواریانس برای داده‌های آزمون نیز به کمک تابع کوواریانس تخمین زده شده و میزان خطای آن‌ها نسبت به پارامترهای واقعی ارزیابی می‌گردد.

```
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
Pt1 = 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
Pt2 = 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 پس از آن اختلاف g1 و g2 به دست آمده 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.

```
Correct_C1 = sum(g11 > g12);
Wrong_C1 = sum(g11 <= g12);

Correct_C2 = sum(g22 > g21);
Wrong_C2 = sum(g22 <= g21);

ConfusionMatrix = [Correct_C1, Wrong_C1; Wrong_C2, Correct_C2]

Ptotal = (Correct_C1 + Correct_C2) / (Correct_C1 + Wrong_C1 + Correct_C2 +
Wrong_C2)
Perror = 1 - Ptotal
```

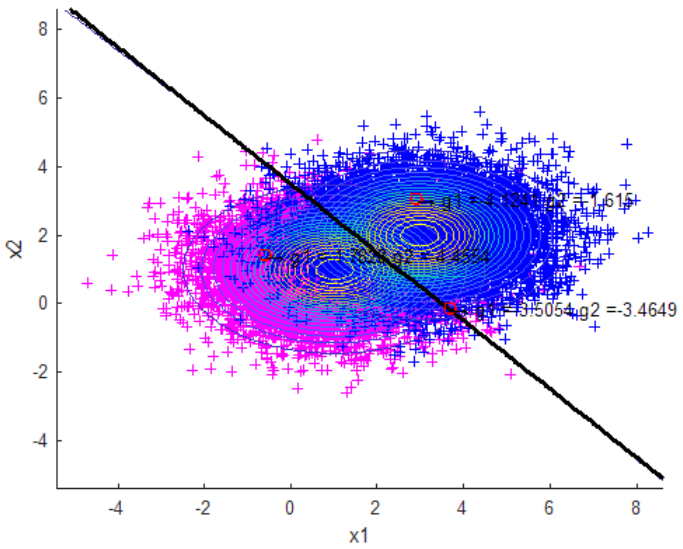
در نهایت برای ارزیابی کلی سیستم روی داده‌های آزمون، ماتریس درهم‌ریختگی (Confusion Matrix) تشکیل شده تا تعداد تخصیص‌های درست و نادرست برای هر کلاس مشخص شود. بر

اساس این ماتریس، درصد دقت کل طبقه‌بندی (Ptotal) و خطای کل سیستم (Perror) محاسبه می‌شود.

```
[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.

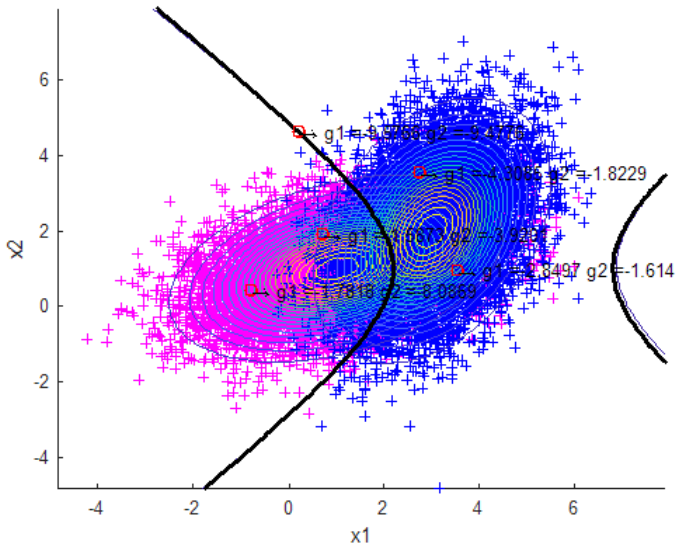
نتایج اجرای قسمت اول:




```
eSigma1 =
    2.0093   -0.0019
   -0.0019    0.9998
em1 =
    1.0025    1.0018
eSigma2 =
    1.9865   -0.0021
   -0.0021    1.0103
em2 =
    2.9920    2.0069
e11 =
    0.0093    0.0019
    0.0019    0.0002
e12 =
    0.0025    0.0018
e21 =
    0.0135    0.0021
    0.0021    0.0103
e22 =
    0.0080    0.0069
testSigma1 =
    1.9993   -0.0178
   -0.0178    0.9768
```

```
testm1 =
    0.9733    0.9937
testSigma2 =
    1.9833    0.0098
    0.0098    0.9626
testm2 =
    3.0099    2.0021
test_e11 =
    0.0007    0.0178
    0.0178    0.0232
test_e12 =
    0.0267    0.0063
test_e21 =
    0.0167    0.0098
    0.0098    0.0374
test_e22 =
    0.0099    0.0021
Pt1 =
    0.8082
Pt2 =
    0.8104
ConfusionMatrix =
        7072        1678
        1659        7091
Ptotal =
    0.8093
Perror =
    0.1907
```

نتایج اجرای قسمت دوم:




```
eSigma1 =
    2.0087    0.5148
    0.5148    1.0097
em1 =
    0.9974    1.0034
eSigma2 =
    1.0154    0.5139
    0.5139    2.0278
em2 =
    2.9973    2.0093
e11 =
```

```
    0.0087    0.0148
    0.0148    0.0097
e12 =
    0.0026    0.0034
e21 =
    0.0154    0.0139
    0.0139    0.0278
e22 =
    0.0027    0.0093
testSigma1 =
    2.0341    0.5080
    0.5080    0.9944
testm1 =
    1.0270    1.0116
testSigma2 =
    0.9702    0.4901
    0.4901    2.0265
testm2 =
    3.0010    2.0025
test_e11 =
    0.0341    0.0080
    0.0080    0.0056
test_e12 =
    0.0270    0.0116
test_e21 =
    0.0298    0.0099
    0.0099    0.0265
test_e22 =
    0.0010    0.0025
Pt1 =
    0.7559
Pt2 =
    0.8555
ConfusionMatrix =
        6614        2136
        1264        7486
Ptotal =
    0.8057
Perror =
    0.1943
```