Задание 1. Построение уравнений регрессии методом наименьших квадратов и главных компонент

1) Сгенерировать набор из N = 500 точек, лежащих на прямой y = a + bx, для a = 1,0; b = 0,8; в диапазоне значений x = [0, 10];

```
x = round(rand([1 500])*10,2);
y = 1 + 0.8*x;
```

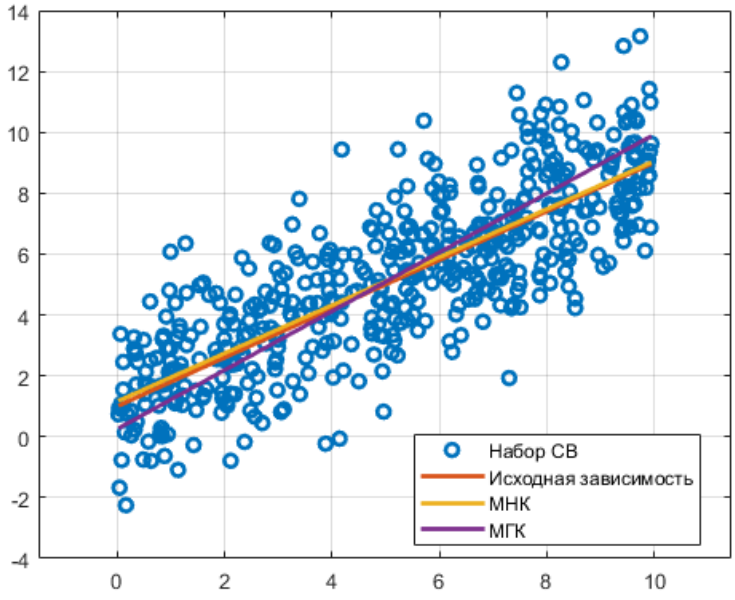
2) Сгенерировать вектор погрешностей из N значений случайной величины, распределенной по нормальному закону с нулевым средним значением и СКО, равным 1,7.

3) Прибавить полученный случайный вектор к координате «y» полученных на первом этапе пар значений.

```
y1 = normrnd(0, 1.7, [1 500])+y;
```

4) Построить уравнение регрессии по полученным зашумленным парам точек с использованием МНК и МГК. Сравнить с исходным уравнением.

```
C = cov(x,y1);
b = C(1,2)/C(1,1);
a = mean(y1)-b*mean(x);
y2 = a + b*x; %МНК
xm = x - mean(x);
ym = y1 - mean(y1);
s1 = cov(xm, ym);
ev = pcacov(s1);
b = ev(2,1)/ev(1,1);
a1 = mean(y1) - b*mean(x);
y3 = a1 + b*x; %МГК
figure
plot(x, y1, 'o', "LineWidth", 2)
hold on
plot(x, [y; y2; y3], "LineWidth", 2)
xlim([min(x)-0.15*range(x) max(x)+0.15*range(x)])
grid on
legend('Набор СВ', 'Исходная зависимость', 'МНК', 'МГК', "Location","best")
```

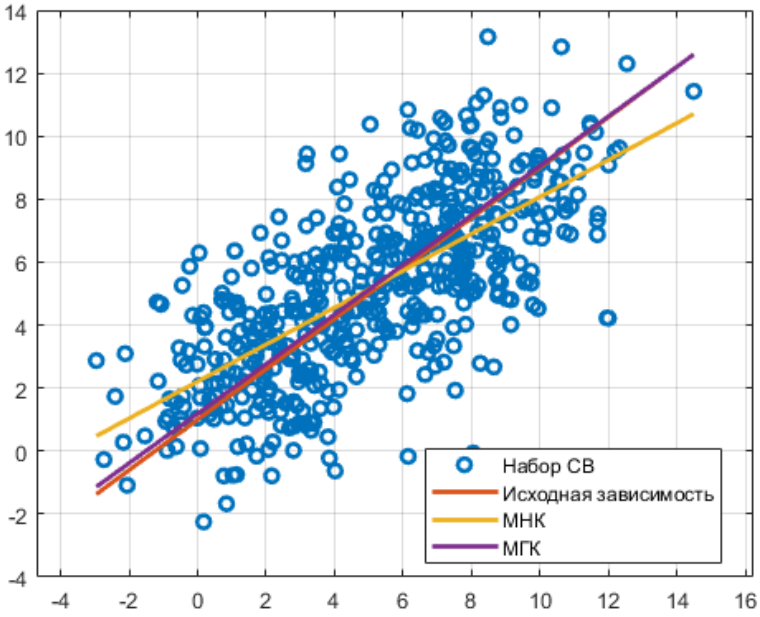


5) Сгенерировать независимо два вектора погрешностей по N значений случайной величины, распределенной по нормальному закону с нулевым средним значением и СКО, равным 1,7. Прибавить их к координатам соответственно y и x исходных не зашумленных данных.

```
x1 = normrnd(0, 1.7, [1 500])+x;
yn = 1 + 0.8*x1;
```

6) Построить уравнение регрессии по полученным зашумленным парам точек с использованием МНК и МГК. Сравнить с исходным уравнением.

```
s1 = cov(x1,y1);
bn = s1(1,2)/s1(1,1);
an = mean(y1)-bn*mean(x1);
yn2 = an + bn*x1; %МНК
xm1 = x1 - mean(x1);
s1 = cov(xm1, ym);
evn = pcacov(s1);
bn1 = evn(2,1)/evn(1,1);
an1 = mean(y1) - bn1*mean(x1);
yn3 = an1 + bn1*x1; %МГК
figure
plot(x1, y1, 'o', "LineWidth",2)
hold on
plot(x1, [yn; yn2; yn3], "LineWidth", 2)
xlim([min(x1)-0.1*range(x1) max(x1)+0.1*range(x1)])
grid on
legend('Набор СВ', 'Исходная зависимость', 'МНК', 'МГК', "Location","best")
```



7) Сделать выводы о том, когда для построения уравнения регрессии необходимо использовать МНК, а когда – МГК.

Вывод

Из приведенных результатов видно, что МНК лучше работает, когда данные располагаются в коридоре с явно видимым трендом, в то время как МГА лучше справляется с облаками, данные в которых имеют многомерное нормальное распределение.

Задание 2*. Снижение размерности данных методом главных компонент с использованием свойств сингулярного разложения матриц

1) Сгенерировать 3 реализации 10-мерного нормального распределения с нулевым вектором средних и единичными дисперсиями при условии равенства 0,2 всех парных коэффициентов корреляции.

Рассчитать собственные вектора и собственные числа двумя способами, увеличив размер выборки до 100, а размерность вектора данных – до 1000. Сравнить время расчета. Сделать выводы.

```
N = 100;
m_test(1,:) = round(rand(1, 1000),3)*10;
m_test(2,:) = round(rand(1, 1000),1)*15;
m_test(3,:) = round(rand(1, 1000),1)*10;
s = ones([1000 1000])*0.2 + eye(1000)*0.8;
for intv = 1:3
```

2) Рассчитать собственные вектора и собственные числа, используя МГК

```
    test = mvnrnd(m_test(intv,:), s, N);
    c_test = cov(test);
    disp(' ')
    disp('PCA')
    tic
    [ev, val] = pcacov(c_test);
    toc
```

3) Рассчитать собственные вектора и собственные числа, используя свойства сингулярного разложения матрицы наблюдений.

```
    disp('SVD')
    tic
    [U,S] = svd(c_test);
    toc
```

4) Сравнить полученные двумя способами результаты.

```
    disp('Сравнение результатов расчета двумя методами')
    sum(round(abs(ev)) - round(abs(U)),"all")
end
```

```
PCA
Elapsed time is 0.301693 seconds.
SVD
Elapsed time is 0.290840 seconds.
Сравнение результатов расчета двумя методами
ans = 0

PCA
Elapsed time is 0.336324 seconds.
SVD
Elapsed time is 0.321139 seconds.
Сравнение результатов расчета двумя методами
ans = 0

PCA
Elapsed time is 0.347631 seconds.
SVD
Elapsed time is 0.271978 seconds.
Сравнение результатов расчета двумя методами
ans = 0
```

Вывод

Эксперимент показал, что во всех реализациях 10-мерного нормального распределения быстрее с задачей поиска СВ и СЗ справляетя метод сингулярного разложения матрицы, по результатам, при этом, практически не отличаясь от МГК.

Задание 3. Построение разделяющей прямой методом ЛДА

1) Сгенерировать два набора данных (по 200 точек в каждом) как реализации двумерного нормального распределения. Первый набор данных: среднее значение равно [1; 3], СКО = [0,7; 0,8], коэффициент корреляции = 0,2. Второй набор данных: среднее значение равно [3;2], СКО = [0,3; 1,8], коэффициент корреляции = 0,7

```
N = 200;
mu1 = [1 3];
mu2 = [3 2];
s1 = [0.7^2 0.8*0.7*0.2; 0.8*0.7*0.2 0.8^2];
s2 = [0.3^2 0.3*1.8*0.7; 0.3*1.8*0.7 1.8^2];
x1 = mvnrnd(mu1, s1, N)';
x2 = mvnrnd(mu2, s2, N)';
corr(x2(1,:)', x2(2,:)')
```

```
ans = 0.6836
```

```
corr(x1(1,:)', x1(2,:)')
```

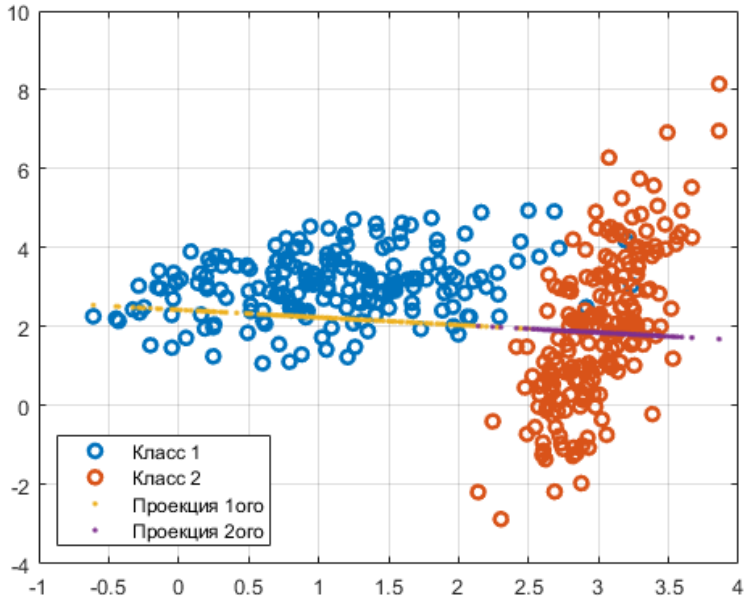
```
ans = 0.2751
```

```
figure
plot(x1(1,:), x1(2,:), 'o', "LineWidth",2)
hold on
plot(x2(1,:), x2(2,:), 'o', "LineWidth",2)
grid on
```

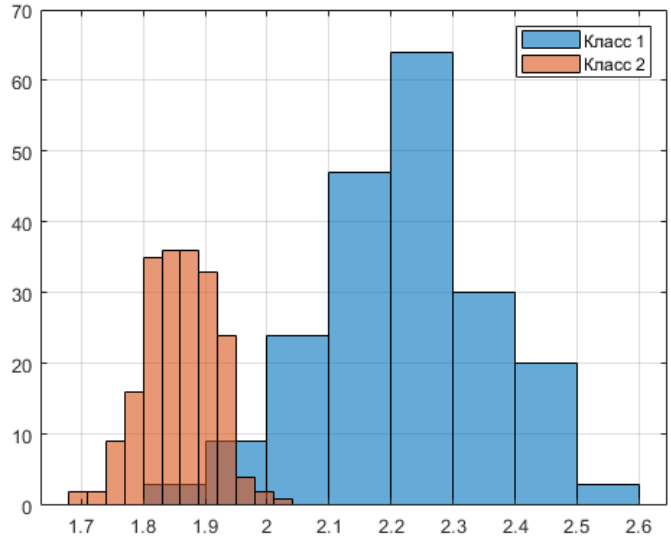
2) Выполнить проекцию на одно направление методом ЛДА. Построить гистограммы значений полученного признака для обоих классов.

```
sw1 = (N - 1)*cov(x1');
sw2 = (N - 1)*cov(x2');
sw = sw1+sw2;
w = sw\(mu1-mu2)';
y1 = w'*x1;
y2 = w'*x2;
b = w(2)/w(1);
x = [x1(1,:), x2(1,:)];
y = [x1(1,:), x2(1,:)];
a = mean(y) - b*mean(x);
y_new1 = a + b*x1(1,:);
y_new2 = a + b*x2(1,:);
plot(x1(1,:), y_new1, '.', "LineWidth", 4)
plot(x2(1,:), y_new2, '.', "LineWidth", 4)
legend('Класс 1', 'Класс 2', 'Проекция 1ого', 'Проекция 2ого', "Location",
'southwest')
```

```
hold off
```




```
figure
histogram(y_new1)
hold on
histogram(y_new2)
grid on
legend('Класс 1', 'Класс 2')
```



Вывод

В результате выполнения задания были получены новые значения данных с уменьшенной размерностью, с учетом критерия о наилучшем разделении двух классов, при помощи ЛДА.

Задание 1. Определение байесовской ошибки классификации

1) Для двух классов выполнить расчет координат «идеального» проекционного вектора методом ЛДА. Условные вероятности данных в каждом из классов подчиняются двумерному нормальному распределению с параметрами:

- первый класс: среднее значение = [1; 3], СКО = [0,7; 0,8], коэффициент корреляции = 0,2.

- второй класс: среднее значение = [3;2], СКО = [0,3; 1,8], коэффициент корреляции = 0,7.

При расчете принять априорные вероятности классов одинаковыми.

```
mu1 = [1 3];
mu2 = [3 2];
s1 = [0.7^2 0.2*0.7*0.8; 0.2*0.7*0.8 0.8^2];
s2 = [0.3^2 0.7*0.3*1.8; 0.7*0.3*1.8 1.8^2];
p1 = 0.5; % априорные вероятности классов
p2 = 0.5;
sw = p1*s1+p2*s2;
```

2) Рассчитать байесовскую ошибку классификации. Для этого рассчитать параметры условных вероятностных распределений после проекции данных на «идеальный» проекционный вектор, найти оптимальное пороговое значение для классификации и вероятности ошибочной классификации в одну и другую стороны.

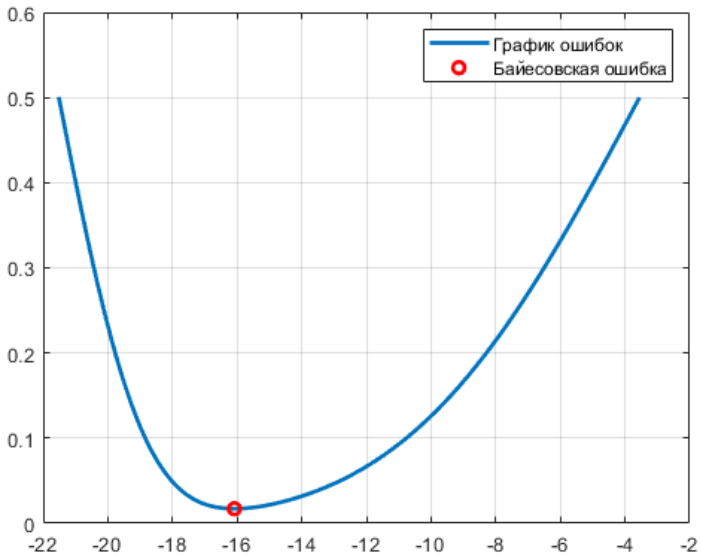
```
w = sw\(mu1-mu2)'; % идеальный проекционный вектор
mu1_new = w'*mu1';
mu2_new = w'*mu2';
s1_new = w'*s1*w;
s2_new = w'*s2*w;
func = @(y)(p1*exp(-((y-mu1_new)^2)/(2*s1_new))/sqrt(2*pi*s1_new)-p2*exp(-((y-
mu2_new)^2)/(2*s2_new))/sqrt(2*pi*s2_new));
z1 = min(mu1_new,mu2_new);
z2 = max(mu1_new,mu2_new);
intv = [z1 z2];
zer_B = fzero(func,intv);
zb1 = (zer_B - mu1_new)/sqrt(s1_new); % вероятность классификации граничного объекта
в 1 класс
zb2 = (zer_B - mu2_new)/sqrt(s2_new); % вероятность классификации граничного объекта
во 2 класс
```

3) Построить зависимость вероятности ошибочной классификации от порогового значения. Убедиться, что байесовская ошибка является наименьшей возможной вероятностью ошибочной классификации.

```
Err_B = 1 - normcdf(zb2,0,1) + normcdf(zb1,0,1) % байесовская ошибка классификации
```

```
Err_B = 0.0357
```

```
n1 = 200;
curr = linspace(z1, z2, n1);
Error = zeros([1 n1]);
for i = 1:n1
    Z1 = (curr(i)-mu1_new)/sqrt(s1_new);
    Z2 = (curr(i)-mu2_new)/sqrt(s2_new);
    Error(i) = 1-normcdf(Z2,0,1)+normcdf(Z1,0,1);
end
figure
plot(curr,Error, "LineWidth", 2);
hold on
plot(zer_B,Err_B,'ro', "LineWidth", 2);
legend('График ошибок', 'Байесовская ошибка')
grid on
```



Задание 2. Построение классификатора методом ЛДА

1) Для каждого из описанных в задании 1 классов сгенерировать по два набора данных: группу обучения и группу контроля. Общее число примеров обоих классов в группе обучения – 6000, в группе контроля – 2000. Принять оба класса равновероятными (для MATLAB функция mvnrnd(mu, sigma, N) – многомерное нормальное распределение, известны средние значения, ковариационная матрица и число элементов в наборах данных).

6) Повторить пп.1-5 для размера обучающей выборки 200. Сделать выводы.

```
NS = [6000 200];
NC = 2000;
for j = 1:2
    xs1 = mvnrnd(mu1, s1, NS(j))';
```

```
    xc1 = mvnrnd(mu1, s1, NC)';
    xs2 = mvnrnd(mu2, s2, NS(j))';
    xc2 = mvnrnd(mu2, s2, NC)';
    figure
    plot(xs1(1,:), xs1(2,:), '.')
    hold on
    plot(xs2(1,:), xs2(2,:), '.')
    plot(xc1(1,:), xc1(2,:), '.')
    plot(xc2(1,:), xc2(2,:), '.')
    legend('Класс 1 (обуч.)', 'Класс 2 (обуч.)', 'Класс 1 (контр.)', 'Класс 2
(контр.)', 'Location', 'southwest')
    grid on
    hold off
```

2) По группе обучения выполнить расчет проекционного вектора методом ЛДА.

```
    sw1 = (NS(j) - 1)*cov(xs1');
    sw2 = (NS(j) - 1)*cov(xs2');
    sw = sw1+sw2;
    w = sw\(mu1-mu2)';
    y1 = w'*xs1;
    y2 = w'*xs2;
    b = w(2)/w(1);
    x = [xs1(1,:), xs2(1,:)];
    y = [xs1(2,:), xs2(2,:)];
    a = mean(y) - b*mean(x);
```

3) Спроецировать данные из группы обучения на построенный вектор и определить пороговое значение, для которого суммарная частота ошибочной классификации будет минимальна.

```
    y_new1 = a + b*xs1(1,:);
    y_new2 = a + b*xs2(1,:);
    figure
    plot(xs1(1,:), xs1(2,:), '.')
    hold on
    plot(xs2(1,:), xs2(2,:), '.')
    plot(xs1(1,:), y_new1, '.', "LineWidth", 2)
    plot(xs2(1,:), y_new2, '.', "LineWidth", 2)
    legend('Класс 1', 'Класс 2', 'Проекция 1ого', 'Проекция 2ого', "Location",
"southwest")
    grid on
    hold off
```

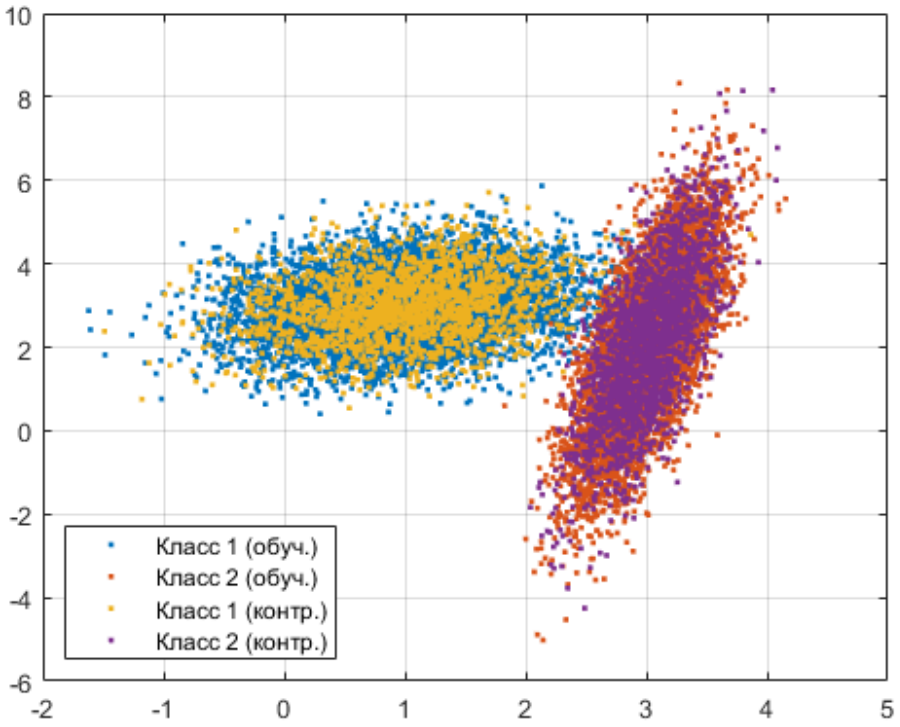
4) Рассчитать относительную частоту ошибочной классификации при применении построенного классификатора к группе контроля.

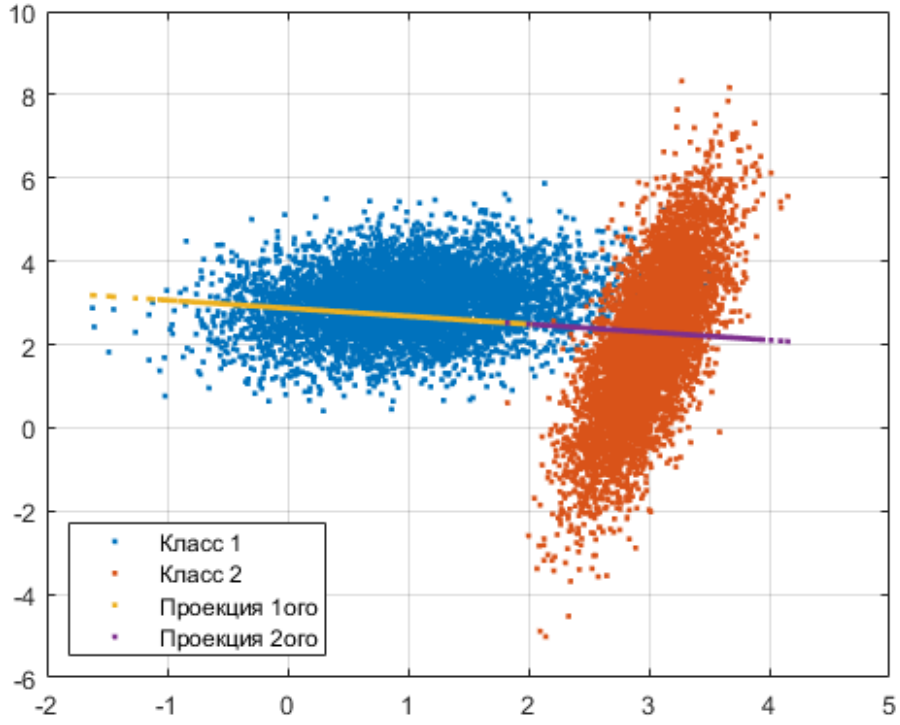
```
    m1=mean(y_new1);
    m2=mean(y_new2);
    n1 = 100;
    err_step = linspace(m1, m2, n1);
    er1=zeros([1,n1]);
    for i=1:n1
        N1 = sum(y_new1<err_step(i));
        N2 = sum(y_new2>err_step(i));
        er1(i)=(N1+N2)/(2*NS(j));
    end
    k=find(er1==min(er1));
    figure
    plot(err_step,er1, "LineWidth", 2);
    hold on
    plot(err_step(k(1)),er1(k(1)),'o', "LineWidth", 2);
    legend('График ошибок', 'Байесовская ошибка')
    grid on
    hold off

    yy11=w'*xc1;
    yy22=w'*xc2;
    xx1=[xc1(1,:), xc2(1,:)];
    yy1=[xc1(2,:), xc2(2,:)];
    a1=mean(yy1)-b*mean(xx1);
    yyy11=a1+b*xc1(1,:);
    yyy22=a1+b*xc2(1,:);
    figure
    plot(xc1(1,:), xc1(2,:), '.')
    hold on
    plot(xc2(1,:), xc2(2,:), '.')
    plot (xc1(1,:),yyy11, "LineWidth", 3)
    plot (xc2(1,:),yyy22, "LineWidth", 3)
    legend('Класс 1', 'Класс 2', 'Проекция 1ого', 'Проекция 2ого', "Location",
"southwest")
    grid on
    hold off
    N11 = sum(yyy11<err_step(k(1)));
    N22 = sum(yyy22>err_step(k(1)));
    Err2 =(N11+N22)/(2*NC)
```

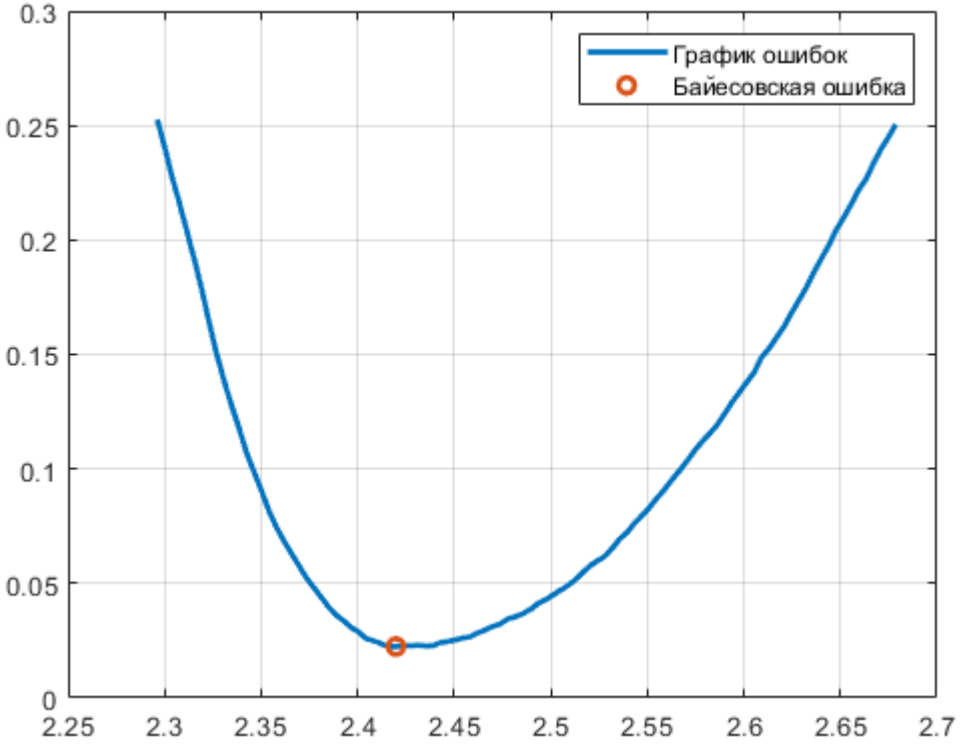
5) Рассчитать относительную частоту ошибочной классификации примеров группы контроля при использовании проекционного вектора и порогового значения, полученных ранее в задании 1. Сравнить полученные значения относительных частот ошибочной классификации.

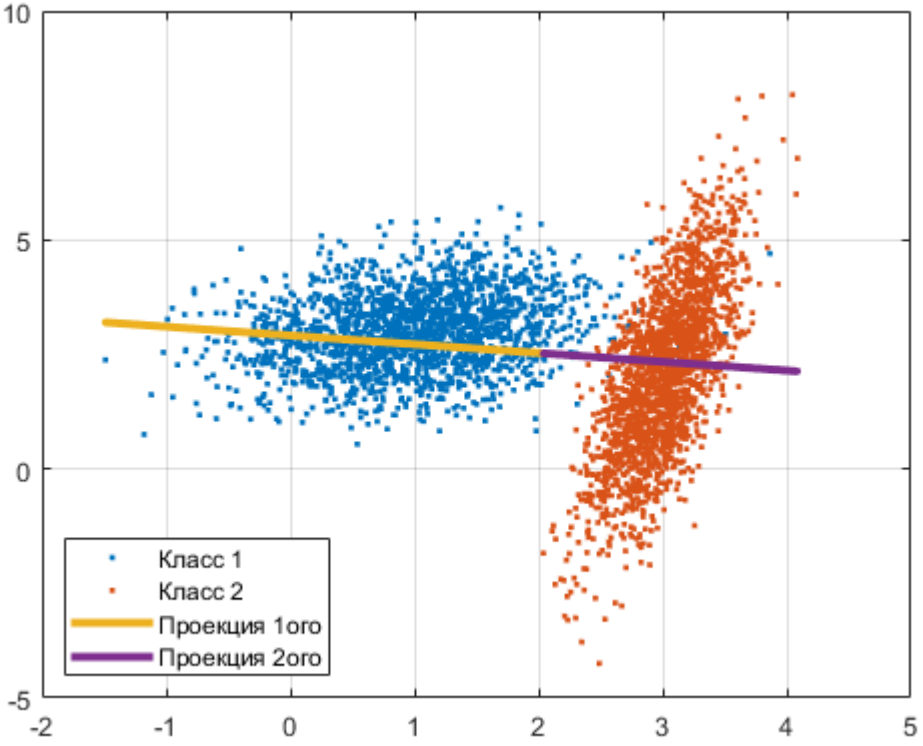
```
    N11 = sum(yyy11<zer_B);
```

```
    N22 = sum(yyy22>zer_B);
    Err3 = (N11+N22)/(2*NC)
end
```

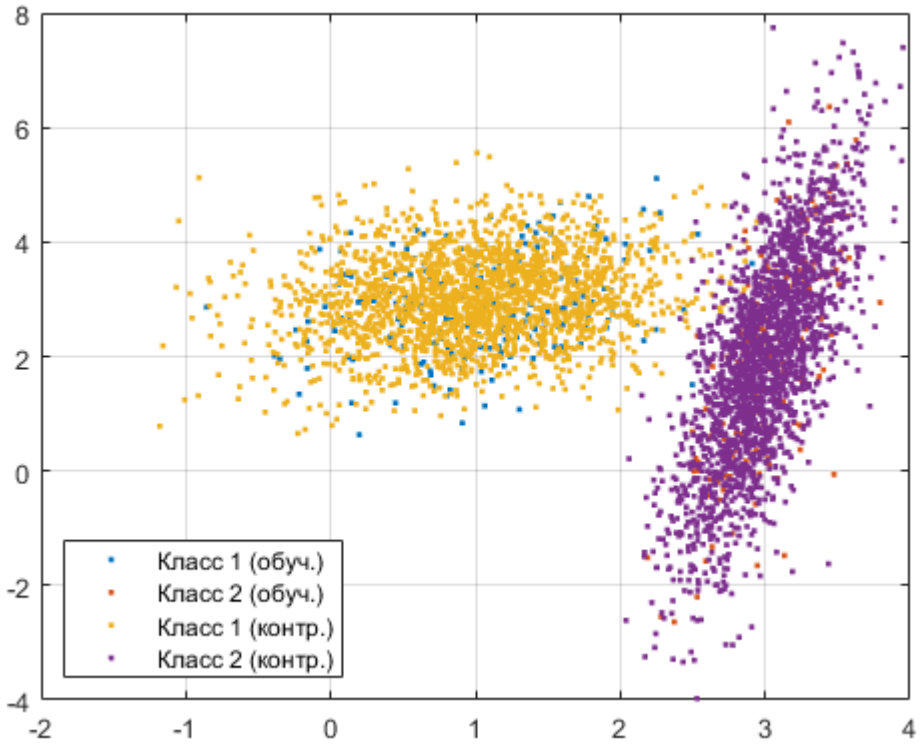


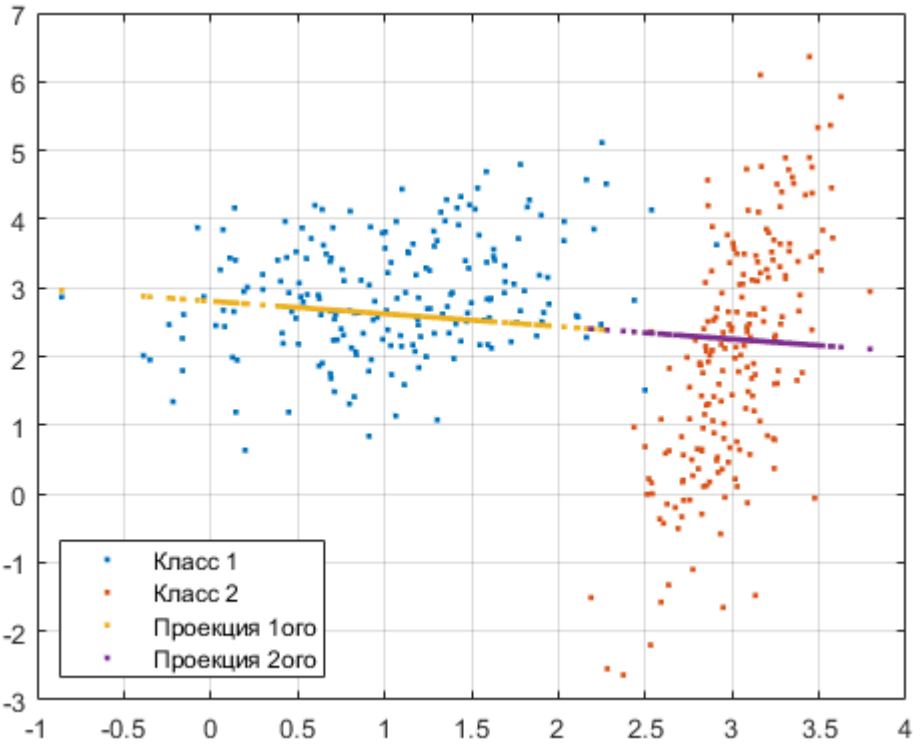


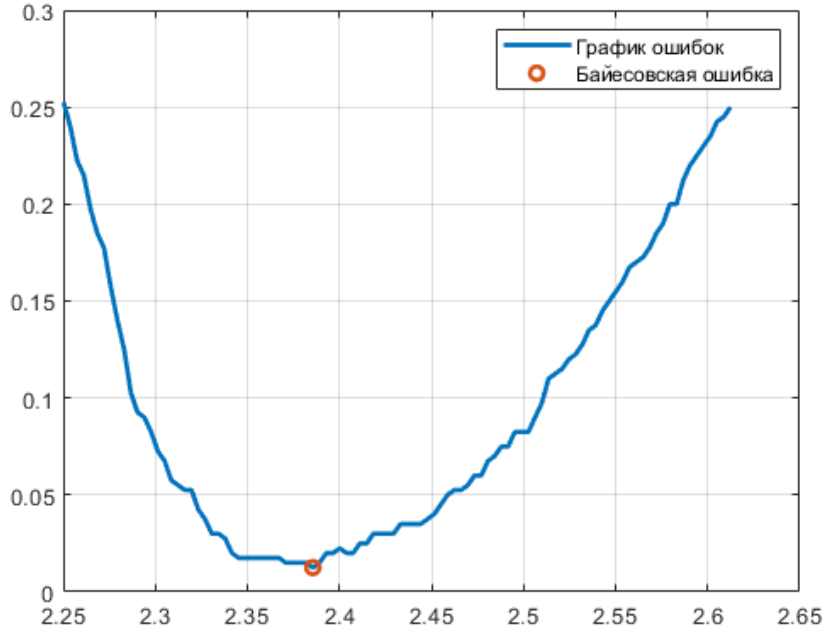


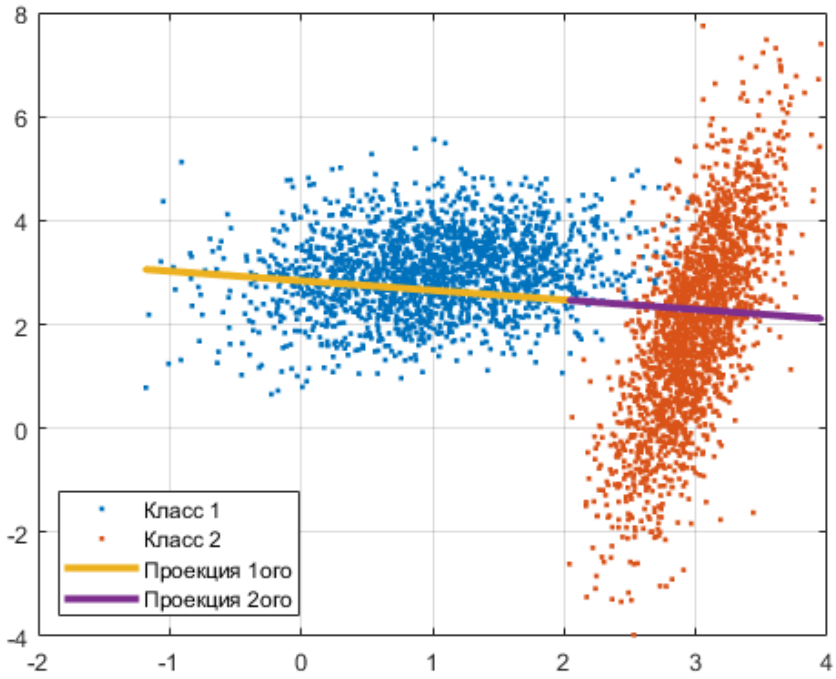



```
Err2 = 0.0275
Err3 = 0.5000
```


Класс 1 (обуч.)
Класс 2 (обуч.)
Класс 1 (контр.)
Класс 2 (контр.)


Класс 1
Класс 2
Проекция 1ого
Проекция 2ого






```
Err2 = 0.0461
Err3 = 0.5000
```

Вывод

Из приведенных результатов видно, что относительная частота ошибочной классификации, подобранная по построенному классификатору гораздо меньше, чем при использовании порогового значения в задании 1.

При уменьшении обучающей выборки с 6000 до 200, качество построенного классификатора уменьшается, но не всегда критично: при выборке в 200 значений частота ошибочной классификации увеличивается, но все еще не так значительно, как при уменьшении до 100.