Задачи работы:

1) Собрать двумерную выборку для двух разделимых классов;
2) Проверить собранные выборки на нормальность;
3) Выбрать параметр, лучше всего показывающий различие двух классов между собой.

Используемые ресурсы и ПО:

- база данных “CelebA”,
- открытая документация с сайта Mathworks,
- среда Matlab версии 2020a.

1 Сбор данных для формирования выборки

Для выполнения этого пункта работы, с учетом специфики научно-исследовательской работы был найден алгоритм простой сегментации лица и переработан для итеративной работы с изображениями для поиска интересующих параметров двух классов. Пример работы исходного алгоритма (рисунок 1):

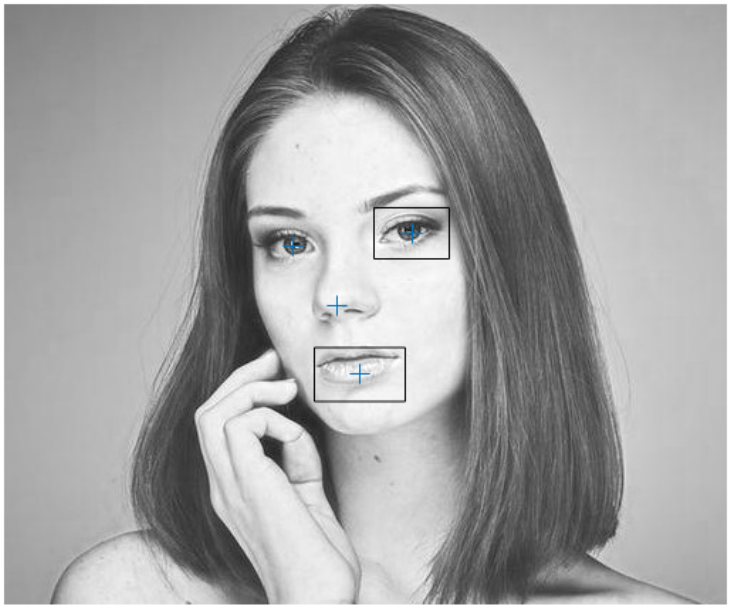

Рисунок 1 – Пример работы алгоритма сегментации

В качестве двух классов для разделения были выбраны глаз и рот, а в качестве признаков подсчитывалась информация о площади прямоугольных областей, в которой заключались объекты первого и второго класса расстояния от центров интересующих областей до условного центра лица (носа). Затем эти данные нормировались при помощи расстояния между центрами областей глаз (с допущением о схожести пропорций) и заносились в вектор данных вместе с нормировочным параметром.

Сбор данных осуществлялся на основе открытой базы данных «CelebA», содержащую в себе 1000 выровненных и обрезанных изображений лиц (рисунок 2).

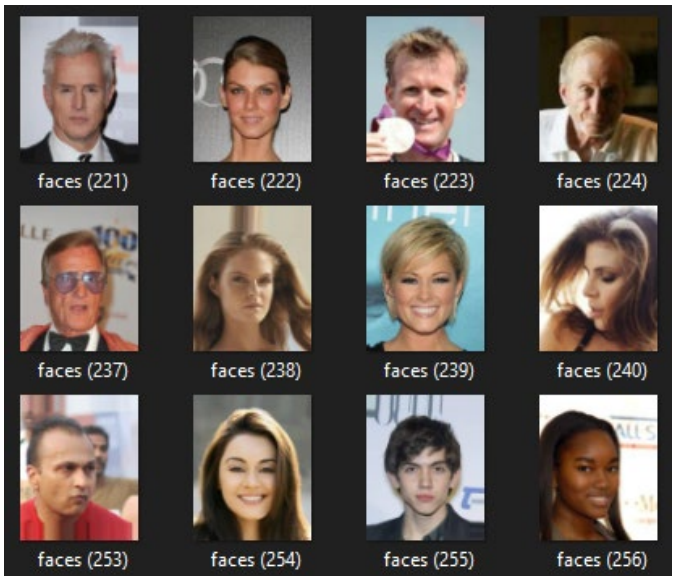

Рисунок 2 – Экземпляры фото из базы данных «CelebA»

Все изображения были проанализированы алгоритмом, после чего на выходе получился массив данных 321x5. Несоответствие ожидаемого количества рядов полученному результату объясняется тем, что на лицах некоторых людей из базы есть аксессуары типа очков, головных уборов и тд, которые перекрывают области лица, из-за чего алгоритм не мог произвести сегментацию и переходил к следующему изображению.

Полученный массив данных представлен в таблице 1.

Таблица 1 – Собранный массив данных

№\Параметр	Нормировочный размер	Область глаза (Класс 1)		Область рта (Класс 2)	
		Площадь	Расстояние до центра	Площадь	Расстояние до центра
1	84,04	0,2702	0,88	0,3443	0,6021
2	44,05	0,4031	0,86	0,4948	0,485
3	40,50	0,5792	0,82	0,6401	0,531
4	44,57	0,4349	0,77	0,5627	0,4956
5	41,00	0,4449	0,84	0,3925	0,4417
6	45,00	0,4148	0,72	0,4429	0,5005
7	41,01	0,3163	0,79	0,4839	0,5126
8	39,53	0,4506	0,89	0,5209	0,4554
9	44,00	0,4462	0,68	0,5422	0,5805
…	…	…	…	…	…

Алгоритм сбора данных по базе фото представлен в приложении А.

2 Проверка данных на нормальность

Прежде чем проверять данные на нормальность стоить отобразить классы на признаковом пространстве (рисунок 3):

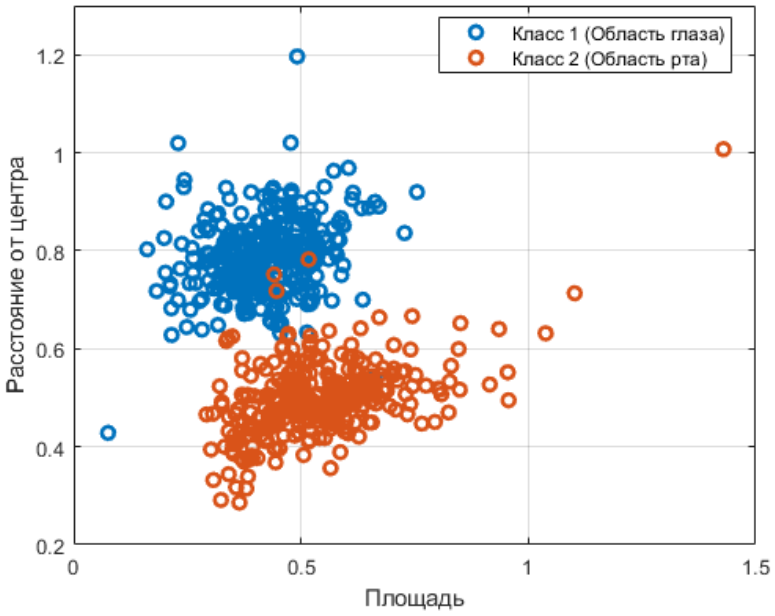

Рисунок 3 – Отображение «сырых» данных на признаковом пространстве

Невооруженным взглядом видно, что есть значения, лежащие сильно далеко от центров облаков данных, эти выбросы можно списать на погрешности работы выбранного алгоритма сегментации с некоторыми изображениями из БД, поскольку она включает в себя и фото, сделанные не только в анфас. Поэтому на некоторых фото сегменты лица могли быть определены неправильно. Такие экземпляры можно считать шумом, от которого стоит избавиться перед проверкой гипотез о нормальности распределения. Очищенные данные принимают следующий вид (рисунок 4):

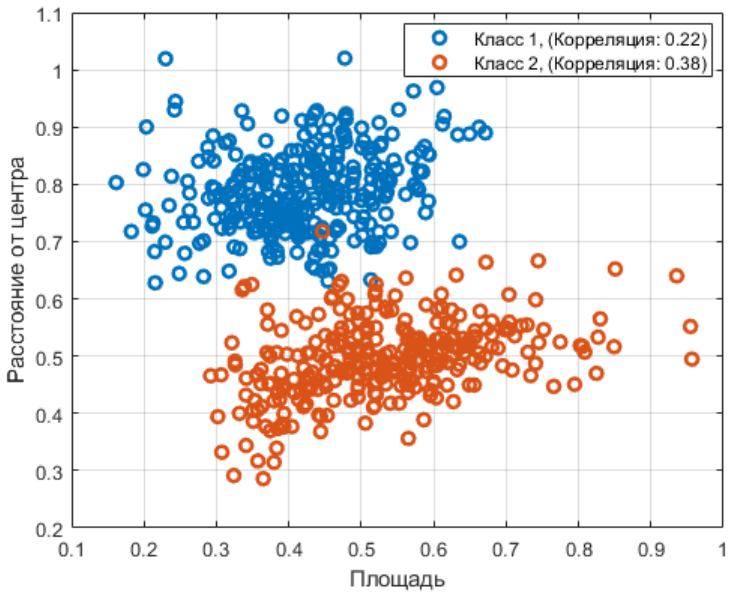

Рисунок 4 – Очищенные данные

Для того, чтобы визуально оценить, насколько близки имеющиеся данные к нормальному распределению, необходимо воспользоваться Q-Q графиком (рисунок 5):

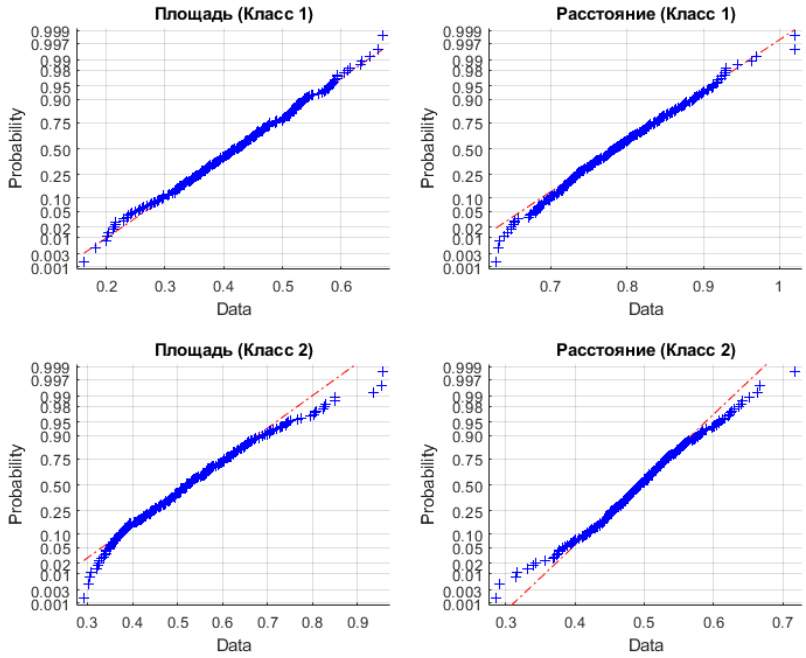

Рисунок 5 – Q-Q график для выборочных данных

По этим графикам видно, что полученные данные очень близки к прямой нормального распределения.

Чтобы удостовериться в характерах эмпирических данных стоит воспользоваться критериями согласия для проверки нулевой гипотезы о нормальном распределении (таблица 2).

Таблица 2 – Проверка нулевой гипотезы

Нулевая гипотеза: Данные распределены по нормальному закону (α = 0.05)				
Параметр\Критерий		**Хи-квадрат**	**Колмогоров-Смирнов**	**Андерсон-Дарлинг**
Класс 1	**Площадь**	0; 0.317	0; 0.977	0; 0.899
	Расстояние	0; 0.458	0; 0.856	0; 0.201
Класс 2	**Площадь**	0; 0.584	0; 0.373	1; 0.007
	Расстояние	0; 0.214	0; 0.626	0; 0.052
Пояснение: (0/1 – нулевая гипотеза принимается/отвергается; рассчитанный уровень значимости)				

Для проверки поставленной нулевой гипотезы с уровнем значимости 0.05 было выбрано три общих критерия согласия: критерий хи-квадрат, критерий Колмогорова-Смирнова и критерий Андерсона-Дарлинга.

По результатам проверок нулевая гипотеза была отвергнута только в одном случае из 12: при проверке параметра «Площадь» для второго класса (область рта) при помощи критерия Андерсона-Дарлинга, но остальные два критерия приняли нулевую гипотезу, поэтому предположим, что выборка для данного параметра распределена нормально. Для других параметров все критерии приняли нулевую гипотезу.

3 Выбор наиболее статистически значимого параметра

Теперь можно перейти к выбору признака, лучше всего показывающего разделимость рассмотренных классов, для этого построим их гистограммы на каждой из осей имеющегося признакового пространства (рисунок 6):

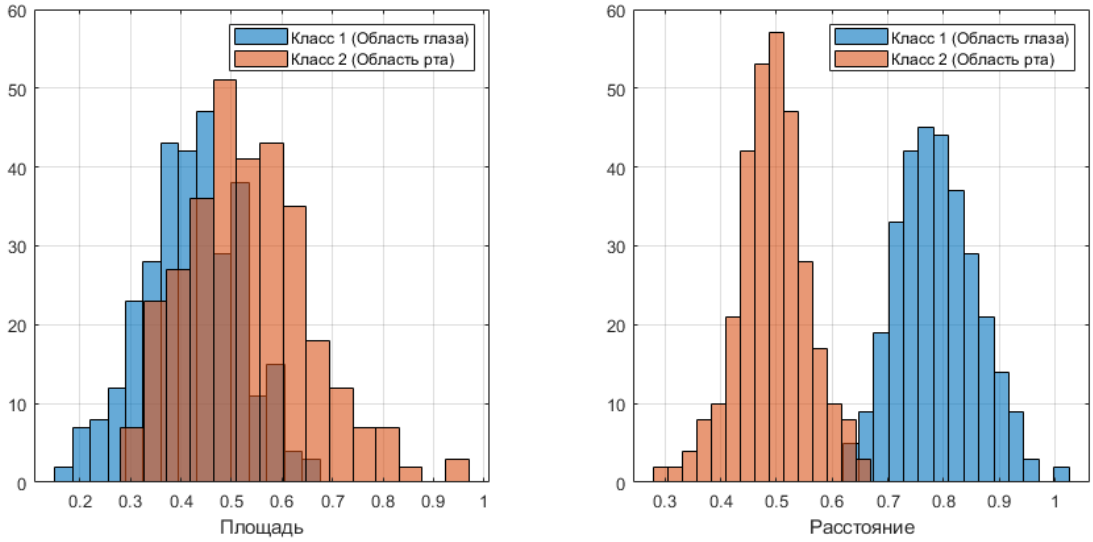

Рисунок 6 – Гистограммы данных на признаковых осях

По приведенным гистограммам видно, что по площади областей глаз и губ отделить один класс от другого очень сложно, поскольку гистограммы очень сильно накладываются друг на друга, что делает суммарную ошибку первого и второго рода очень большой, в то время как по расстояниям от центра лица до центров интересующих областей уже можно говорить о разделимости классов.

Так как по результатам проверок данные считаются нормально распределенными, можно построить теоретические гауссовы кривые со значениями среднего и СКО, найденными по выборкам, для количественной оценки теоретической неустранимой ошибки (рисунок 7).

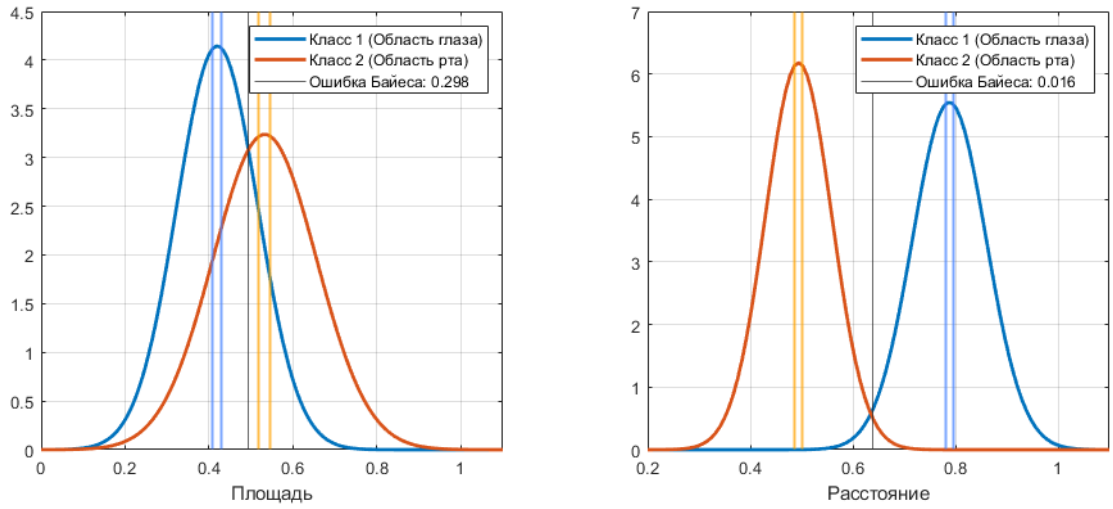

Рисунок 7 – Теоретические кривые распределений данных с доверительными интервалами для средних значений

Доверительные интервалы для средних значений распределений с уровнем значимости 0.05 представлены в таблице 3:

Таблица 3 – Доверительные интервалы

Параметр\Класс	**Класс 1**	**Класс 2**
Площадь	$0.4089 \leq \mu_x \leq 0.4303$	$0.5189 \leq \mu_x \leq 0.5463$
Расстояние	$0.7802 \leq \mu_x \leq 0.7961$	$0.4871 \leq \mu_x \leq 0.5014$

Пусть доверительные интервалы средних значений двух классов на обоих графиках и не пересекаются (малый размер доверительного интервала связан с достаточно большим размером выборки), тем не менее для графиков параметра «площадь» они расположены слишком близко друг к другу, что сильно усложняет процесс классификации и увеличивает значение неустранимой ошибки. Со вторым параметром ситуация гораздо лучше: доверительные

интервалы находятся далеко друг от друга и ошибка Байеса составляет всего 1.6 процента.

Вывод: таким образом, можно сделать вывод, что второй признак (расстояния от центра области сегмента до центра лица) лучше демонстрирует разделимость двух рассмотренных классов, чем первый.

Алгоритм анализа данных представлен в приложении Б.

ПРИЛОЖЕНИЕ А
(справочное)

Алгоритм сбора данных по базе фотографий

```
clear variables;
Amnt = 1000;
Norm_L = zeros(1,Amnt);
Sq_Eye = zeros(1,Amnt);
L_Eye = zeros(1,Amnt);
Sq_Mouth = zeros(1,Amnt);
L_Mouth = zeros(1,Amnt);
for i = 1:Amnt
     try
        k= imread("ph1.jpg");
        I=k(:,:,1);
        faceDetect = vision.CascadeObjectDetector();
        bbox=step(faceDetect,I);
        face=imcrop(I,bbox);
        centerx=size(face,1)/2+bbox(1);
        centery=size(face,2)/2+bbox(2);
        eyeDetect = vision.CascadeObjectDetector('RightEye');
        eyebox=step(eyeDetect,face);
        n=size(eyebox,1);
        e=[];
        for it=1:n
            for j=1:n
                if (j > it)
                    if        ((abs(eyebox(j,2)-eyebox(it,2))<68)&&
(abs(eyebox(j,1)-eyebox(it,1))>40))
                        e(1,:)=eyebox(it,:);
                        e(2,:)=eyebox(j,:);
                        d=1;break;
                    end
```

```
            end
        end
        if(d == 1)
            break;
        end
    end
    eyebox(1,:)=e(1,:)
    eyebox(2,:)=e(2,:);
    c=eyebox(1,3)/2
    d=eyebox(1,4)/2;
    eyeCenter1x=eyebox(1,1)+c+bbox(1);
    eyeCenter1y=eyebox(1,2)+d+bbox(2);
    e=eyebox(2,3)/2;
    f=eyebox(2,4)/2;
    eyeCenter2x=eyebox(2,1)+e+bbox(1);
    eyeCenter2y=eyebox(2,2)+f+bbox(2);
 ndetect=vision.CascadeObjectDetector('Nose','MergeThreshold',16);
    nosebox=step(ndetect,face);
    noseCenterx=nosebox(1,1)+(nosebox(1,3)/2)+bbox(1);
    noseCentery=nosebox(1,2)+(nosebox(1,4)/2);
    m=[1,noseCentery,size(face,1),((size(face,2))-
noseCentery)];
    mouth=imcrop(face,m);

mdetect=vision.CascadeObjectDetector('Mouth','MergeThreshold' ,20);
    mouthbox=step(mdetect,mouth);
    for it=1:size(mouthbox,1)
        if(mouthbox(it,2)>20)
            mouthbox(1,:)=mouthbox(it,:);
            break;
        end
    end
    mouthbox(1,2)=mouthbox(1,2)+noseCentery;
    noseCentery=noseCentery+bbox(2);
```

```
        mouthCenterx=mouthbox(1,1)+(mouthbox(1,3)/2)+bbox(1);
        mouthCentery=mouthbox(1,2)+(mouthbox(1,4)/2)+bbox(2);
        shape=[noseCenterx noseCentery;eyeCenter1x
eyeCenter1y;eyeCenter2x eyeCenter2y;mouthCenterx mouthCentery];
        Norm_L(i) = sqrt((eyeCenter1x-eyeCenter2x)^2+(eyeCenter1y-
eyeCenter2y)^2);
        Sq_Eye(i) = eyebox(1, 3)*eyebox(1, 4)/Norm_L(i)^2;
        Sq_Mouth(i) = mouthbox(1, 3)*mouthbox(1, 4)/Norm_L(i)^2;
        L_Eye(i) = sqrt((eyeCenter1x -noseCenterx)^2+(eyeCenter1y-
noseCentery)^2)/Norm_L(i);
        L_Mouth(i) = sqrt((mouthCenterx-noseCenterx)^2+
(mouthCentery-noseCentery)^2)/Norm_L(i);
    catch ME
        disp('Следующее фото')
    end
end
faces = table(Norm_L', Sq_Eye', L_Eye', Sq_Mouth', L_Mouth');
faces.Properties.VariableNames = {'Norm', 'SqEye', 'leye',
'SqMouth', 'lmouth'};
faces = faces(faces.Norm ~= 0, :);
writetable(faces, 'Faces.xls');
```

ПРИЛОЖЕНИЕ Б
(справочное)

Алгоритм анализа данных

```
clear variables
tbl = readtable("D:\USR\Рабочий стол\КТвРО\Faces.xls")
figure
plot(tbl.SqEye, tbl.leye, 'o', "LineWidth", 2)
hold on
plot(tbl.SqMouth, tbl.lmouth, 'o', "LineWidth", 2)
grid on
legend('Класс 1 (Область глаза)', 'Класс 2 (Область рта)', "Location","best")
ylim([0.2 1.3])
xlabel('Площадь')
ylabel('Расстояние от центра')
hold off
tbl = rmoutliers(tbl, 'mean');

figure
plot(tbl.SqEye, tbl.leye, 'o', "LineWidth", 2)
hold on
plot(tbl.SqMouth, tbl.lmouth, 'o', "LineWidth", 2)
grid on
legend("Класс 1, (Корреляция: " + num2str(round(corr(tbl.SqEye, tbl.leye), 2)) + ")", "Класс 2, (Корреляция: " + num2str(round(corr(tbl.SqMouth, tbl.lmouth),2)) + ")", "Location","best")
xlabel('Площадь')
ylabel('Расстояние от центра')
hold off
mtrx = [tbl.SqEye tbl.leye tbl.SqMouth tbl.lmouth];
figure
```

```
txt = ["Площадь (Класс 1)", "Расстояние (Класс 1)", "Площадь (Класс
2)", "Расстояние (Класс 2)"];
for i = 1:length(mtrx(1,:))
    disp(' ')
    disp(txt(i))
    [Chi2,chi2_p,                    chi2_stat]                    =
chi2gof(mtrx(:,i),'cdf',{@normcdf,mean(mtrx(:,i)),std(mtrx(:,i))})
    disp(' ')
    x = (mtrx(:,i) - mean(mtrx(:,i)))/std(mtrx(:,i));
    [KS, ks_p, ks_stat] = kstest(x)
    disp(' ')
    [AD, ad_p, st] = adtest(mtrx(:,i))
    disp(' ')
    subplot(2, 2, i)
    normplot(mtrx(:,i))
    title(txt(i))
end

figure
bn = 15;
subplot(1,2,1)
histogram(tbl.SqEye, bn)
hold on
histogram(tbl.SqMouth, bn)
legend('Класс 1 (Область глаза)', 'Класс 2 (Область рта)')
xlabel('Площадь')
grid on
hold off
subplot(1,2,2)
histogram(tbl.leye, bn)
hold on
histogram(tbl.lmouth, bn)
legend('Класс 1 (Область глаза)', 'Класс 2 (Область рта)')
xlabel('Расстояние')
```

```
grid on
hold off

x = linspace(0, 1.1, 130);
y = zeros([length(x) length(mtrx(i,:))]);
m = mean(mtrx);
s = std(mtrx);
for i = 1:length(mtrx(i,:))
    y(:,i) = normpdf(x, m(i) ,s(i))';
end
k = find(y(:,1) <= y(:,3), 1 );
figure
subplot(1,2,1)
plot(x, y(:,1), "LineWidth", 2)
hold on
plot(x, y(:,3), "LineWidth", 2)
xline(x(k))
xlim([0 1.1])
err_sq = (trapz(y(k:end,1)) + trapz(y(1:k,3)))/(trapz(y(:,1))*2);
xlabel('Площадь')
xln = s*1.96/sqrt(length(mtrx(:,1)));
ln1 = xline(m(1) - xln(1), 'color', [69, 130, 255]/255, "LineWidth",
1.5);
ln2 = xline(m(1) + xln(1), 'color', [69, 130, 255]/255, "LineWidth",
1.5);
ln3 = xline(m(3) - xln(3), 'color', [255, 163, 0]/255, "LineWidth",
1.5);
ln4 = xline(m(3) + xln(3), 'color', [255, 163, 0]/255, "LineWidth",
1.5);
legend('Класс 1 (Область глаза)', 'Класс 2 (Область рта)', "Ошибка
Байеса: " + num2str(round(err_sq,3)))
grid on
hold off
k = find(y(:,4) <= y(:,2), 1 );
```

```
subplot(1,2,2)
plot(x, y(:,2), "LineWidth", 2)
hold on
plot(x, y(:,4), "LineWidth", 2)
xline(x(k))
xlim([0.2 1.1])
err_ln = (trapz(y(k:end,4)) + trapz(y(1:k,2)))/(trapz(y(:,2))*2);
xlabel('Расстояние')
ln5 = xline(m(2) - xln(2), 'color', [69, 130, 255]/255, "LineWidth",
1.5);
ln6 = xline(m(2) + xln(2), 'color', [69, 130, 255]/255, "LineWidth",
1.5);
ln7 = xline(m(4) - xln(4), 'color', [255, 163, 0]/255, "LineWidth",
1.5);
ln8 = xline(m(4) + xln(4), 'color', [255, 163, 0]/255, "LineWidth",
1.5);
legend('Класс 1 (Область глаза)', 'Класс 2 (Область рта)', "Ошибка
Байеса: " + num2str(round(err_ln,3)))
grid on
hold off
grid on
hold off
```