

Старт

```
clc
close all
clearvars
```

Загрузка данных

```
filename = "ECG_dataset.xlsx";
N = readtable(filename,"Sheet","HP");
A1 = readtable(filename,"Sheet","ЖТ");
A2 = readtable(filename,"Sheet","ФЖ");
```

Предподготовка данных

```
mark01 = [zeros(15,1);ones(30,1)];
mark023 = [zeros(15,1);ones(15,1).*2;ones(15,1).*3];

train = [N(1:15,1:end); A1(1:15,1:end); A2(1:15,1:end)];
train = [table(mark01), table(mark023), train];
train = train(randperm(height(train)),:);
train = [train{:,1}, train{:,2}, train{:,4:end} ./ train{:,3}];

test = [N(16:30,1:end); A1(16:30,1:end); A2(16:30,1:end)];
test = [table(mark01), table(mark023), test];
test = test(randperm(height(train)),:);
test = [test{:,1}, test{:,2}, test{:,4:end} ./ test{:,3}];
```

Классификация k-ближайших (взвешенный) и расчет статистик

```
disp('КЛАССИФИКАЦИЯ К-БЛИЖАЙШИХ')
disp('ВЗВЕШЕННЫЕ ДИСТАНЦИИ')
acc_data = zeros(6,2);
for i = 3:9
    model01 = fitcknn(train(:,3:end),train(:,1),'NumNeighbors',i,'DistanceWeight','squaredinverse');
    label01 = predict(model01,test(:,3:end));
    predictid = [label01, test];

    model23 = fitcknn(train(train(:,2) == 2 | train(:,2) == 3, 3:end), ...
        train(train(:,2) == 2 | train(:,2) == 3, 2),'NumNeighbors',i,'DistanceWeight','squaredinverse');
    label23 = predict(model23, predictid(predictid(:,1) == 1, 4:end));
    predictid(predictid(:,1) == 1) = label23;

    metrics = zeros(3,3);
    for j = [0,2,3]
        temp = zeros(3,1);
        temp(1,1) = length(find(predictid(:,1) == j & ...
            predictid(:,3) == 0));
        temp(2,1) = length(find(predictid(:,1) == j & ...
```

```

        predictid(:,3) == 2));
temp(3,1) = length(find(predictid(:,1) == j & ...
    predictid(:,3) == 3));
if j == 0
    metrics(:,1) = temp;
elseif j == 2
    metrics(:,2) = temp;
elseif j == 3
    metrics(:,3) = temp;
end
end
fprintf('Число соседей: %.0f\n',i);
disp(table({'1';'2';'3'},metrics,'VariableNames',{'Истинный класс','Результат распознавания'}));
acc = metrics(1,1)+metrics(2,2)+metrics(3,3);
acc = acc/45;
acc_data(i-2,1) = acc;
acc_data(i-2,2) = i;
fprintf('Общая точность: %.2f\n\n',acc);
end
figure()
plot(acc_data(:,2),acc_data(:,1));
ylim([0.6 1]);
ylabel('Общая точность');
xlabel('Число соседей');
title('Взвешенный k-nearest');

```

Классификация k-ближайших (равные веса дистанций)

```

disp('РАВНЫЕ ДИСТАНЦИИ')
acc_data = zeros(6,2);
for i = 3:9
    model01 = fitcknn(train(:,3:end),train(:,1),'NumNeighbors',i,'DistanceWeight','equal');
    label01 = predict(model01,test(:,3:end));
    predictid = [label01, test];

    model23 = fitcknn(train(train(:,2) == 2 | train(:,2) == 3, 3:end), ...
        train(train(:,2) == 2 | train(:,2) == 3, 2),'NumNeighbors',i,'DistanceWeight','equal');
    label23 = predict(model23, predictid(predictid(:,1) == 1, 4:end));
    predictid(predictid(:,1) == 1) = label23;

    metrics = zeros(3,3);
    for j = [0,2,3]
        temp = zeros(3,1);
        temp(1,1) = length(find(predictid(:,1) == j & ...
            predictid(:,3) == 0));
        temp(2,1) = length(find(predictid(:,1) == j & ...
            predictid(:,3) == 2));
        temp(3,1) = length(find(predictid(:,1) == j & ...
            predictid(:,3) == 3));
    if j == 0

```

```

        metrics(:,1) = temp;
    elseif j == 2
        metrics(:,2) = temp;
    elseif j == 3
        metrics(:,3) = temp;
    end
end
fprintf('Число соседей: %.0f\n',i);
disp(table({'1';'2';'3'},metrics,'VariableNames',{'Истинный класс','Результат распознавания'}));
acc = metrics(1,1)+metrics(2,2)+metrics(3,3);
acc = acc/45;
acc_data(i-2,1) = acc;
acc_data(i-2,2) = i;
fprintf('Общая точность: %.2f\n\n',acc);
end
figure()
plot(acc_data(:,2),acc_data(:,1));
ylim([0.6 1]);
ylabel('Общая точность');
xlabel('Число соседей');
title('k-nearest равные веса дистанций');

```

Метод главных компонент

```

[~,score,~,~,explained,~] = pca([N{:,1:end};A1{:,1:end};A2{:,1:end}]);
figure()
gscatter(score(:,1), score(:,2),[ones(30,1);ones(30,1).*2;ones(30,1).*3]);
title('Диаграмма рассеяния');
xlabel('Первая ГК');
ylabel('Вторая ГК');
legend('НР','ЖТ','ФЖ','FontSize',10,'Location','northeast')
disp('Дисперсия в % первой ГК')
disp(explained(1))
disp('Дисперсия в % второй ГК')
disp(explained(2))

```

Случай независимых признаков (по минимуму расстояний)

```

disp('СЛУЧАЙ НЕЗАВИСИМЫХ ПРИЗНАКОВ')
% НР + ЖТ - это первый класс, ФЖ - это второй класс
vNA1 = [N{:,1:end}; A1{:,1:end}];
vA2 = A2{:,1:end};
mvNA1 = mean(vNA1)';
mvA2 = mean(vA2)';
W1 = mvA2 - mvNA1;
a1 = -415;
pNA1 = vNA1*W1/norm(W1);
pA2 = vA2*W1/norm(W1);

disp(table(W1/norm(W1),'VariableNames',{'Весовой вектор W'}))

```

```

disp(table(a1, 'VariableNames', {'a1'}))
disp(table(mean(pNA1), var(pNA1), mean(pA2), var(pA2), 'VariableNames', {'Среднее класса НР+ЖТ', 'Дисперсия класса НР+ЖТ'}))

% НР - это первый класс, ЖТ - это второй класс
vN = N(:, 1:end);
vA1 = A1(:, 1:end);
mvN = mean(vN)';
mvA1 = mean(vA1)';
W2 = mvN - mvA1;
a2 = -470;
pN = vN*W2/norm(W2);
pA1 = vA1*W2/norm(W2);

disp(table(W2/norm(W2), 'VariableNames', {'Весовой вектор W'}))
disp(table(a2, 'VariableNames', {'a1'}))
disp(table(mean(pN), var(pN), mean(pA1), var(pA1), 'VariableNames', {'Среднее класса НР', 'Дисперсия класса НР'}))

% Гистограммы и гауссовское распределение
% Величины x01, x02 используются для расчета порога
% a1, a2 - пороговые значения, могут высчитываться через x0 или задаваться
% эмпирически

figure()
y = -1200:1:200;
mu = mean(pNA1);
sigma = std(pNA1);
g11 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
histogram(pNA1, 'BinWidth', 100, 'Normalization', 'pdf')
hold on
plot(y, g11, 'LineWidth', 2);
hold on;
mu = mean(pA2);
sigma = std(pA2);
g12 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
histogram(pA2, 'BinWidth', 100, 'Normalization', 'pdf')
hold on
plot(y, g12, 'LineWidth', 2);
hold on
xline(a1, 'LineWidth', 3)
legend('НР+ЖТ', 'Огибающая НР+ЖТ', 'ФЖ', 'Огибающая ФЖ', 'Порог', 'FontSize', 10, 'Location', 'best')
title('Гистограммы НР+ЖТ/ФЖ', 'FontSize', 10)
subtitle('Случай независимых признаков')
xlim([-1200, 200])

figure()
y = -1200:1:200;
mu = mean(pN);
sigma = std(pN);
g13 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
histogram(pN, 'BinWidth', 100, 'Normalization', 'pdf')

```

```

hold on
plot(y,g13,'LineWidth',2);
hold on
histogram(pA1,'BinWidth',100,'Normalization','pdf')
hold on
mu = mean(pA1);
sigma = std(pA1);
g14 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y,g14,'LineWidth',2);
hold on
xline(a2,'LineWidth',3)
legend('НР', 'Огибающая НР', 'ЖТ', 'Огибающая ЖТ', 'Порог', 'FontSize',10, 'Location', 'northeast')
title('Гистограммы НР/ЖТ', 'FontSize',10)
subtitle('Случай независимых признаков')
xlim([-1200, 200])

% Метрики точности и Area under the curve (AUC)
xline1 = -1200:1:200;
xline2 = -1200:1:200;
nNA1 = normpdf(xline1,mean(pNA1),std(pNA1));
nA2 = normpdf(xline1,mean(pA2),std(pA2));
nN = normpdf(xline2,mean(pN),std(pN));
nA1 = normpdf(xline2,mean(pA1),std(pA1));

p = [pA2;pNA1];
predictClass = zeros(90,1);
for i=1:length(p)
    if p(i) > a1
        predictClass(i) = 1;
    else
        predictClass(i) = 2;
    end
end

[acc, sens, spec, TP, TN, FP, FN] = accuracy(predictClass,[ones(30,1);ones(60,1).*2]);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{'Точность','Чувствительность','Специфичность'}))

p = [pN;pA1];
predictClass = zeros(60,1);
for i=1:length(p)
    if p(i) > a2
        predictClass(i) = 1;
    else
        predictClass(i) = 2;
    end
end

[acc, sens, spec, TP, TN, FP, FN] = accuracy(predictClass,[ones(30,1);ones(30,1).*2]);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{'Точность','Чувствительность','Специфичность'}))

```

```

[acc, sens, spec, TP, TN, FP, FN] = AUC(nA2, nNA1, xline1, a1);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{ 'Тоность', 'Чувствительность', 'Специфичность'}));

[acc, sens, spec, TP, TN, FP, FN] = AUC(nN, nA1, xline2, a2);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{ 'Тоность', 'Чувствительность', 'Специфичность'}));

% ROC-кривые
ROC11 = ROC([pA2;pNA1], [ones(30,1);ones(60,1).*2], xline1);
ROC12 = ROC([pN;pA1], [ones(30,1);ones(30,1).*2], xline2);
GROC11 = GROC(nA2, nNA1, xline1);
GROC12 = GROC(nN, nA1, xline2);

figure()
subplot(1,2,1)
plot(GROC11(1,:),GROC11(2,:), LineWidth=2)
hold on
plot(ROC11(1,:),ROC11(2,:), 'LineWidth',2, 'Color', '#A2142F')
xlabel('Специфичность')
ylabel('Чувствительность')
title('ROC кривая НР+ЖТ/ФЖ')
subtitle('Случай независимых признаков')
legend('По Гауссу', 'По выборке', 'FontSize',10, 'Location', 'northeast')
ylim([0,1])
xlim([0,1])

subplot(1,2,2)
plot(GROC12(1,:),GROC12(2,:), 'LineWidth',2, 'Color', '#0072BD')
hold on
plot(ROC12(1,:),ROC12(2,:), 'LineWidth',2, 'Color', '#A2142F')
xlabel('Специфичность')
ylabel('Чувствительность')
title('ROC кривая НР/ЖТ')
subtitle('Случай независимых признаков')
legend('По Гауссу', 'По выборке', 'FontSize',10, 'Location', 'northeast')
ylim([0,1])
xlim([0,1])

```

ЛДФ (ДВУХКЛАССОВАЯ ЗАДАЧА)

```

disp('ПО КРИТЕРИЮ ФИШЕРА (ДВУХКЛАССОВАЯ ЗАДАЧА)')
% НР + ЖТ - это первый класс, ФЖ - это второй класс
E1 = cov(vNA1);
E2 = cov(vA2);
E = E1+E2;
mvNA1 = mean(vNA1)';
mvA2 = mean(vA2)';
W1 = E\ (mvNA1-mvA2);
pNA1 = vNA1*W1/norm(W1);
pA2 = vA2*W1/norm(W1);
a1 = 1.1;

```

```

disp(table(W1,'VariableNames',{'Весовой вектор W'}))
disp(table(a1,'VariableNames',{'a1'}))
disp(table(mean(pNA1),var(pNA1),mean(pA2),var(pA2),'VariableNames',{'Среднее класса НР+ЖТ','Дисперсия

% НР - это первый класс, ЖТ - это второй класс
E1 = cov(vN);
E2 = cov(vA1);
E = E1+E2;
mvN = mean(vN)';
mvA1 = mean(vA1)';
W2 = E\ (mvN-mvA1);
pN = vN*W2/norm(W2);
pA1 = vA1*W2/norm(W2);
a2 = 1.5;

disp(table(W2,'VariableNames',{'Весовой вектор W'}))
disp(table(a2,'VariableNames',{'a1'}))
disp(table(mean(pN),var(pN),mean(pA1),var(pA1),'VariableNames',{'Среднее класса НР','Дисперсия

% Гистограммы и гаусовское распределение
% a1, a2 - порог, который задается эмпирически, глядя на картинку
% x01, x02 - порог, который высчитывается формульно

figure()
y = -30:0.1:30;
histogram(pNA1,'BinWidth',5,'Normalization','pdf')
hold on
mu = mean(pNA1);
sigma = std(pNA1);
g21 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y, g21,'LineWidth',2);
hold on
histogram(pA2,'BinWidth',5,'Normalization','pdf')
hold on
mu = mean(pA2);
sigma = std(pA2);
g22 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y, g22,'LineWidth',2);
hold on
xline(a1,'LineWidth',3)
legend('НР+ЖТ', 'Огибающая НР+ЖТ', 'ФЖ', 'Огибающая ФЖ', 'Порог', 'FontSize', 10, 'Location', '
title('Гистограммы НР+ЖТ/ФЖ', 'FontSize', 10)
subtitle('По критерию Фишера')
xlim([-30, 30])

figure()
histogram(pN,'BinWidth',5,'Normalization','pdf')
hold on
mu = mean(pN);

```

```

sigma = std(pN);
g23 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y,g23,"LineWidth",2)
histogram(pA1,'BinWidth',5,'Normalization','pdf')
hold on
mu = mean(pA1);
sigma = std(pA1);
g24 = exp(-(y-mu).^2./(2*sigma^2))./(sigma*sqrt(2*pi));
plot(y,g24,"LineWidth",2)
hold on
xline(a2,'LineWidth',3)
legend('HP', 'Огибающая HP', 'ЖТ', 'Огибающая ЖТ', 'Порог','FontSize',10,'Location','northeast')
title('Гистограммы HP/ЖТ','FontSize',10)
subtitle('По критерию Фишера')
xlim([-30, 30])

xlinespape = -30:0.1:30;
nNA1 = normpdf(xlinespape,mean(pNA1),std(pNA1));
nA2 = normpdf(xlinespape,mean(pA2),std(pA2));
nN = normpdf(xlinespape,mean(pN),std(pN));
nA1 = normpdf(xlinespape,mean(pA1),std(pA1));

% Метрики точности и Area under the curve (AUC)
p = [pNA1;pA2];
predictClass = zeros(90,1);
for i=1:length(p)
    if p(i) > a1
        predictClass(i) = 1;
    else
        predictClass(i) = 2;
    end
end

[acc, sens, spec, TP, TN, FP, FN] = accuracy(predictClass,[ones(60,1);ones(30,1).*2]);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{'Точность','Чувствительность','Специфичность'}))

p = [pN;pA1];
predictClass = zeros(60,1);
for i=1:length(p)
    if p(i) > a2
        predictClass(i) = 1;
    else
        predictClass(i) = 2;
    end
end

[acc, sens, spec, TP, TN, FP, FN] = accuracy(predictClass,[ones(30,1);ones(30,1).*2]);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{'Точность','Чувствительность','Специфичность'}))

[acc, sens, spec, TP, TN, FP, FN] = AUC(nNA1, nA2, xlinespape, a1);

```



```

disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{ 'Тоность', 'Чувствительность', 'Специфичность'}));

[acc, sens, spec, TP, TN, FP, FN] = AUC(nN, nA1, xlinesapce, a2);
disp(table(acc,sens,spec,TP,TN,FP,FN,'VariableNames',{ 'Тоность', 'Чувствительность', 'Специфичность'}));

% ROC-кривые
ROC31 = ROC([pNA1;pA2], [ones(60,1);ones(30,1).*2], xlinesapce);
ROC32 = ROC([pN;pA1], [ones(30,1);ones(30,1).*2], xlinesapce);
GROC31 = GROC(nNA1, nA2, xlinesapce);
GROC32 = GROC(nN, nA1, xlinesapce);

figure()
subplot(1,2,1)
plot(GROC31(1,:),GROC31(2:3,:), 'LineWidth',2, 'Color', '#0072BD')
hold on
plot(ROC31(1,:),ROC31(2:3,:), 'LineWidth',2, 'Color', '#A2142F')
xlabel('Специфичность')
ylabel('Чувствительность')
title('ROC кривая НР+ЖТ/ФЖ')
subtitle('По критерию Фишера')
legend('По Гауссу', 'По выборке', 'FontSize',10, 'Location', 'northeast')
xlim([0,1])

subplot(1,2,2)
plot(GROC32(1,:),GROC32(2:3,:), 'LineWidth',2, 'Color', '#0072BD')
hold on
plot(ROC32(1,:),ROC32(2:3,:), 'LineWidth',2, 'Color', '#A2142F')
xlabel('Специфичность')
ylabel('Чувствительность')
title('ROC кривая НР/ЖТ')
subtitle('По критерию Фишера')
legend('По Гауссу', 'По выборке', 'FontSize',10, 'Location', 'northeast')
xlim([0,1])

```

ПО КРИТЕРИЮ ФИШЕРА (МНОЖЕСТВЕННЫЙ АНАЛИЗ)

```

disp('ПО КРИТЕРИЮ ФИШЕРА (МНОЖЕСТВЕННЫЙ АНАЛИЗ)')
inputData = [N{:,1:end};A1{:,1:end};A2{:,1:end}];
M = mean(inputData);

Sb=0;
Sw=0;
Pi = 1/3;
for c = 0:2
    Xi = inputData((1+30*c):30*(c+1),:);
    Mi = mean(Xi);
    Sb = Sb + Pi*(Mi-M)'*(Mi-M);
    Sw = Sw + Pi*cov(Xi);
end

```

```

[V, D] = eig(Sw\Sb);

W1 = V(:,1);
W2 = V(:,2);
disp('Собственный вектор W1')
disp(W1)
disp('Собственный вектор W2')
disp(W2)

a = [real(D(1,1)), real(D(2,2))];
a(1) = a(1)-4.5;
a(2) = -a(2);
disp('Порог классификации')
disp(a)

pN1=vN*W1/norm(W1);
pA11=vA1*W1/norm(W1);
pA21=vA2*W1/norm(W1);

pN2=vN*W2/norm(W2);
pA12=vA1*W2/norm(W2);
pA22=vA2*W2/norm(W2);

disp(table(pN1,pA11,pA21,pN2,pA12,pA22,'VariableNames',{ 'Проекция HP на W1', 'Проекция ЖТ на W1', 'Проекция ФЖ на W1'}))
disp(table(mean(pN1),mean(pA11),mean(pA21),mean(pN2),mean(pA12),mean(pA22),'VariableNames',{ 'Проекция HP на W1', 'Проекция ЖТ на W1', 'Проекция ФЖ на W1'}))
disp(table(var(pN1),var(pA11),var(pA21),var(pN2),var(pA12),var(pA22),'VariableNames',{ 'Проекция HP на W1', 'Проекция ЖТ на W1', 'Проекция ФЖ на W1'}))

figure()
scatter(pN1,pN2,'filled')
hold on
scatter(pA11,pA12,'filled')
hold on
scatter(pA21,pA22,'filled')
xline(a(1),'LineWidth',2,'Color','r')
yline(a(2),'LineWidth',2,'Color','r')
xlabel('W1')
ylabel('W2')
legend('HP','ЖТ','ФЖ','FontSize',10,'Location','northeast')
title('Проекция объектов на плоскость')

```

Сравнительные ROC кривые для всех трех методов

```

figure()
subplot(1,2,1)
plot(ROC11(1,:), ROC11(2,:), 'LineWidth',2,'Color','#0072BD')
hold on
plot(ROC31(1,:), ROC31(2,:), 'LineWidth',2,'Color','#A2142F')
title('HP+ЖТ/ФЖ')
subtitle('ROC кривые по гистограммам')
legend('По минимуму расстояний','По критерию Фишера','FontSize',10,'Location','northeast')

```

```

xlabel('Специфичность')
ylabel('Чувствительность')
xlim([0, 1])
ylim([0, 1])

subplot(1,2,2)
plot(GROC11(1,:), GROC11(2,:), 'LineWidth',2)
hold on
plot(GROC31(1,:), GROC31(2,:), 'LineWidth',2, 'Color', '#A2142F')
title('НР+ЖТ/ФЖ')
subtitle('ROC кривые по оценке Гаусса')
legend('По минимуму расстояний', 'По критерию Фишера', 'FontSize',10, 'Location', 'northeast')
xlabel('Специфичность')
ylabel('Чувствительность')
xlim([0, 1])
ylim([0, 1])

figure()
subplot(1,2,1)
plot(ROC12(1,:), ROC12(2,:), 'LineWidth',2, 'Color', '#0072BD')
hold on
plot(ROC32(1,:), ROC32(2,:), 'LineWidth',2, 'Color', '#A2142F')
title('НР/ЖТ')
subtitle('ROC кривые по гистограммам')
legend('По минимуму расстояний', 'По критерию Фишера', 'FontSize',10, 'Location', 'northeast')
xlabel('Специфичность')
ylabel('Чувствительность')
xlim([0, 1])
ylim([0, 1])

subplot(1,2,2)
plot(GROC12(1,:), GROC12(2,:), 'LineWidth',2, 'Color', '#0072BD')
hold on
plot(GROC32(1,:), GROC32(2,:), 'LineWidth',2, 'Color', '#A2142F')
title('НР/ЖТ')
subtitle('ROC кривые по оценке Гаусса')
legend('По минимуму расстояний', 'По критерию Фишера', 'FontSize',10, 'Location', 'northeast')
xlabel('Специфичность')
ylabel('Чувствительность')
xlim([0, 1])
ylim([0, 1])

```