

实 验 报 告

学号：2111078 姓名：李岚琦

学院：人工智能学院 专业：智能科学与技术

实验二：K近邻模型

1. 实验目的

学习K近邻模型进行分类预测

1. 实验原理

对每个测试样本，在训练集中找出与其距离最近的K个点，根据这K个点的类别判断测试样本的类别

1. 实验步骤与代码结果
2. 生成并绘制样本数据点

代码：%%%%%%%%%%%%%%%%%%% 数据生成 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 2000; % 样本量大小

X = rand(n,2)\*10; % 2n \* 2的数据矩阵，每一行表示一个数据点，第一列表示x轴坐标，第二列表示y轴坐标

Y = zeros(n,1); % 类别标签

for i=1:n

if 0<X(i,1) && X(i,1)<3 && 0<X(i,2) && X(i,2)<3 % 根据x和y轴坐标确定分类

Y(i) = 1;

end

if 0<X(i,1) && X(i,1)<3 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 2;

end

if 0<X(i,1) && X(i,1)<3 && 7<X(i,2) && X(i,2)<10

Y(i) = 3;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 0<X(i,2) && X(i,2)<3

Y(i) = 4;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 5;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 7<X(i,2) && X(i,2)<10

Y(i) = 6;

end

if 7<X(i,1) && X(i,1)<10 && 0<X(i,2) && X(i,2)<3

Y(i) = 7;

end

if 7<X(i,1) && X(i,1)<10 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 8;

end;

if 7<X(i,1) && X(i,1)<10 && 7<X(i,2) && X(i,2)<10

Y(i) = 9;

end;

end

X = X(Y>0,:); % 注意X是在[0,10]\*[0,10]范围内均匀生成的，而我们只标出了一部分X，类别之间的白色间隔中的点没有标，因此需要将这些点去掉

Y = Y(Y>0,:); % X(Y>0,:)表示只取X中对应的Y大于0的行，这是因为白色间隔中的点的Y都为0

n = length(Y); % 去除掉白色间隔剩下的点

figure(1)

set (gcf,'Position',[1,1,700,600], 'color','w')

set(gca,'Fontsize',18)

plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10); % 画第一类数据点 X(Y==1,1)表示类别为1（Y==1）的点的第一维度坐标，X(Y==1,2)表示类别为1的点的第二维度坐标

hold on;

plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10); % 画第二类数据点

hold on;

plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10); % 画第三类数据点

hold on;

plot(X(Y==4,1),X(Y==4,2),'g\*','LineWidth',1,'MarkerSize',10); % 画第四类数据点

hold on;

plot(X(Y==5,1),X(Y==5,2),'m\*','LineWidth',1,'MarkerSize',10); % 画第五类数据点

hold on;

plot(X(Y==6,1),X(Y==6,2),'c\*','LineWidth',1,'MarkerSize',10); % 画第六类数据点

hold on;

plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10); % 画第七类数据点

hold on;

plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10); % 画第八类数据点

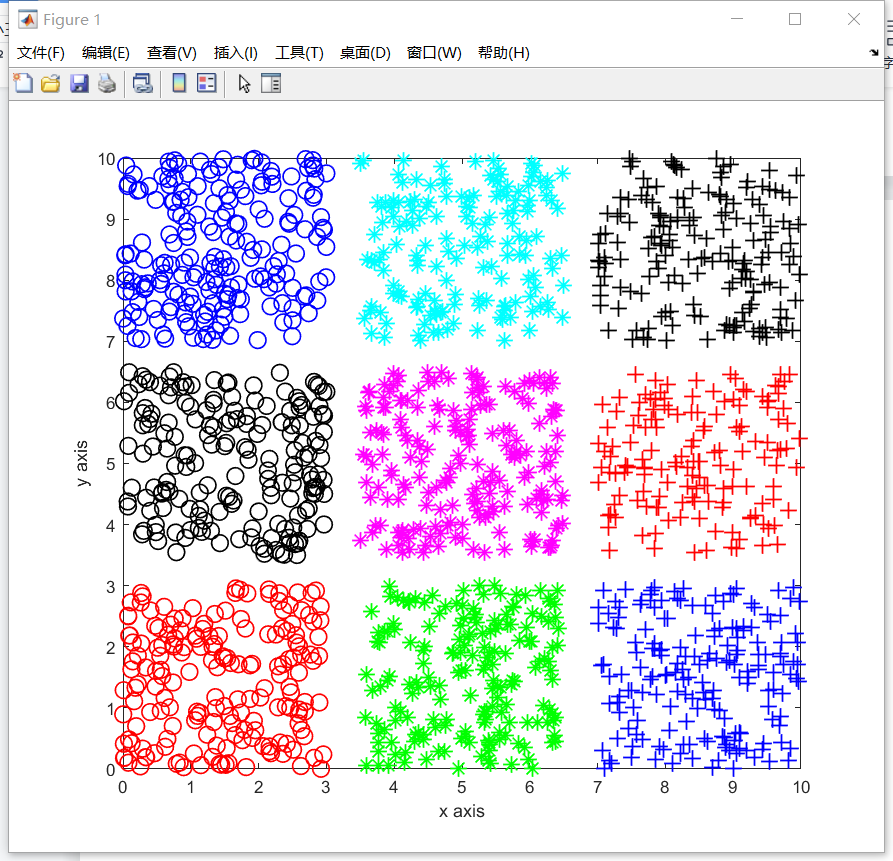
hold on;

plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10); % 画第九类数据点

hold on;

xlabel('x axis');

ylabel('y axis');

运行结果：

1. 生成测试数据并绘制

代码：%%%%%%%%%%%%%%%%%%% 数据生成 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

n = 2000; % 样本量大小

X = rand(n,2)\*10; % 2n \* 2的数据矩阵，每一行表示一个数据点，第一列表示x轴坐标，第二列表示y轴坐标

Y = zeros(n,1); % 类别标签

for i=1:n

if 0<X(i,1) && X(i,1)<3 && 0<X(i,2) && X(i,2)<3 % 根据x和y轴坐标确定分类

Y(i) = 1;

end

if 0<X(i,1) && X(i,1)<3 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 2;

end

if 0<X(i,1) && X(i,1)<3 && 7<X(i,2) && X(i,2)<10

Y(i) = 3;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 0<X(i,2) && X(i,2)<3

Y(i) = 4;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 5;

end

if 3.5<X(i,1) && X(i,1)<6.5 && 7<X(i,2) && X(i,2)<10

Y(i) = 6;

end

if 7<X(i,1) && X(i,1)<10 && 0<X(i,2) && X(i,2)<3

Y(i) = 7;

end

if 7<X(i,1) && X(i,1)<10 && 3.5<X(i,2) && X(i,2)<6.5

Y(i) = 8;

end;

if 7<X(i,1) && X(i,1)<10 && 7<X(i,2) && X(i,2)<10

Y(i) = 9;

end;

end

X = X(Y>0,:); % 注意X是在[0,10]\*[0,10]范围内均匀生成的，而我们只标出了一部分X，类别之间的白色间隔中的点没有标，因此需要将这些点去掉

Y = Y(Y>0,:); % X(Y>0,:)表示只取X中对应的Y大于0的行，这是因为白色间隔中的点的Y都为0

n = length(Y); % 去除掉白色间隔剩下的点

figure(1)

set (gcf,'Position',[1,1,700,600], 'color','w')

set(gca,'Fontsize',18)

plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10); % 画第一类数据点 X(Y==1,1)表示类别为1（Y==1）的点的第一维度坐标，X(Y==1,2)表示类别为1的点的第二维度坐标

hold on;

plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10); % 画第二类数据点

hold on;

plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10); % 画第三类数据点

hold on;

plot(X(Y==4,1),X(Y==4,2),'g\*','LineWidth',1,'MarkerSize',10); % 画第四类数据点

hold on;

plot(X(Y==5,1),X(Y==5,2),'m\*','LineWidth',1,'MarkerSize',10); % 画第五类数据点

hold on;

plot(X(Y==6,1),X(Y==6,2),'c\*','LineWidth',1,'MarkerSize',10); % 画第六类数据点

hold on;

plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10); % 画第七类数据点

hold on;

plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10); % 画第八类数据点

hold on;

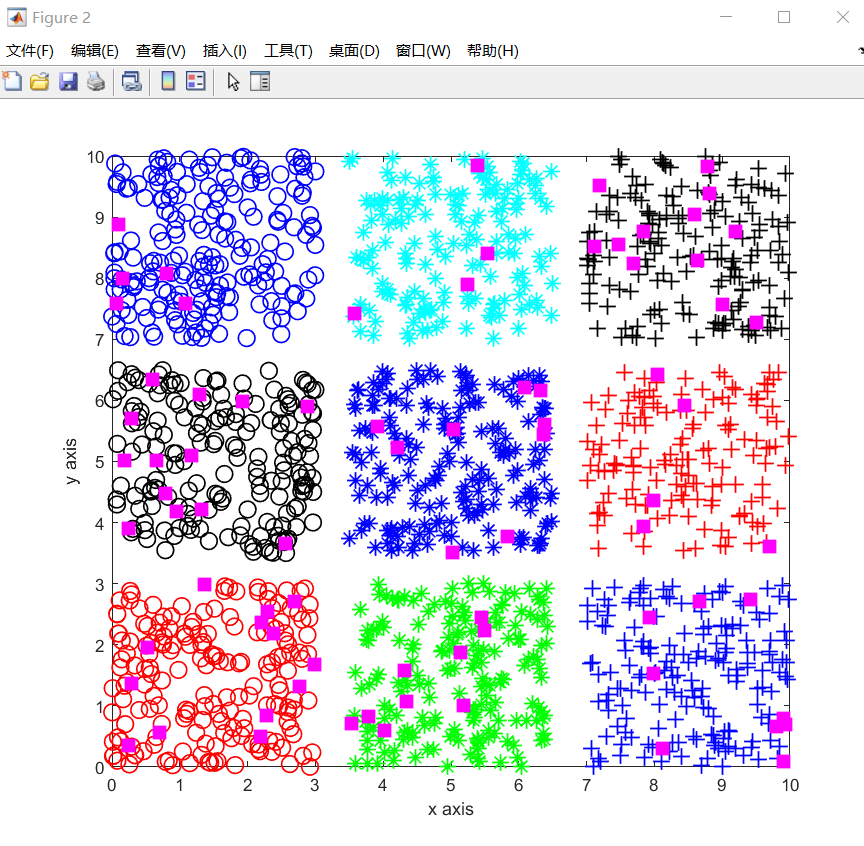
plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10); % 画第九类数据点

hold on;

xlabel('x axis');

ylabel('y axis');

运行结果：



1. 进行K近邻模型的预测编程

%%%%%%%%%%%%%%%%%% K-近邻算法：学生实现 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

K=11;%设定K的值

Xtdistance=zeros(m,n); %用来储存每个测试点距离n个样本点的距离的矩阵

tmpclass=zeros(m,n); %用来对应上述各个测试点的类型

Ytclass=zeros(m,1);%用来记录最终预测出的第m个测试点的类型

Xt11=zeros(m,n);%记录排序后的距离

for i=1:1:m

for j=1:1:n

Xtdistance(i,j)=(Xt(i,1)-X(j,1))^2+(Xt(i,2)-X(j,2))^2;

tmpclass(i,j)=Y(j,1);

end

end

tmp1class=tmpclass;

%对Xtdistance的每一行进行升序排列,同时将每个距离对应的样本点的类型也进行同步排序

for i=1:1:m

Xtmp=Xtdistance(i,:);

[Xtmp,I]=sort(Xtmp);

Xt11(i,:)=Xtmp;

io=tmpclass(i,:);

tmp1class(i,:)=io(I);

end

%统计前K个中哪个类型出现的次数最多；

for i=1:1:m

tmp=tmp1class(i,1:K);

Ytclass(i,1)=mode(tmp);

end

%%%%%%%%%%%%%%%%%% 给出模型的预测输出，并与测试数据的真实输出比较，计算错误率 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Ym = zeros(m,1); % 将测试样本的预测类别存放在Ym中

Ym=Ytclass;

1. 画出K近邻对测试点的预测结果

代码：

%%%%%%%%%%%%%%%%%% 画出预测结果 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

figure(3)

set (gcf,'Position',[1,1,700,600], 'color','w')

set(gca,'Fontsize',18)

plot(X(Y==1,1),X(Y==1,2),'ro','LineWidth',1,'MarkerSize',10); % 画第一类数据点

hold on;

plot(X(Y==2,1),X(Y==2,2),'ko','LineWidth',1,'MarkerSize',10); % 画第二类数据点

hold on;

plot(X(Y==3,1),X(Y==3,2),'bo','LineWidth',1,'MarkerSize',10); % 画第三类数据点

hold on;

plot(X(Y==4,1),X(Y==4,2),'g\*','LineWidth',1,'MarkerSize',10); % 画第四类数据点

hold on;

plot(X(Y==5,1),X(Y==5,2),'b\*','LineWidth',1,'MarkerSize',10); % 画第五类数据点

hold on;

plot(X(Y==6,1),X(Y==6,2),'c\*','LineWidth',1,'MarkerSize',10); % 画第六类数据点

hold on;

plot(X(Y==7,1),X(Y==7,2),'b+','LineWidth',1,'MarkerSize',10); % 画第七类数据点

hold on;

plot(X(Y==8,1),X(Y==8,2),'r+','LineWidth',1,'MarkerSize',10); % 画第八类数据点

hold on;

plot(X(Y==9,1),X(Y==9,2),'k+','LineWidth',1,'MarkerSize',10); % 画第九类数据点

hold on;

plot(Xt(Ym==1,1),Xt(Ym==1,2),'ro','MarkerFaceColor','r','LineWidth',1,'MarkerSize',10); % 画第一类数据点

hold on;

plot(Xt(Ym==2,1),Xt(Ym==2,2),'ko','MarkerFaceColor','k','LineWidth',1,'MarkerSize',10); % 画第二类数据点

hold on;

plot(Xt(Ym==3,1),Xt(Ym==3,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10); % 画第三类数据点

hold on;

plot(Xt(Ym==4,1),Xt(Ym==4,2),'go','MarkerFaceColor','g','LineWidth',1,'MarkerSize',10); % 画第四类数据点

hold on;

plot(Xt(Ym==5,1),Xt(Ym==5,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10); % 画第五类数据点

hold on;

plot(Xt(Ym==6,1),Xt(Ym==6,2),'co','MarkerFaceColor','c','LineWidth',1,'MarkerSize',10); % 画第六类数据点

hold on;

plot(Xt(Ym==7,1),Xt(Ym==7,2),'bo','MarkerFaceColor','b','LineWidth',1,'MarkerSize',10); % 画第七类数据点

hold on;

plot(Xt(Ym==8,1),Xt(Ym==8,2),'ro','MarkerFaceColor','r','LineWidth',1,'MarkerSize',10); % 画第八类数据点

hold on;

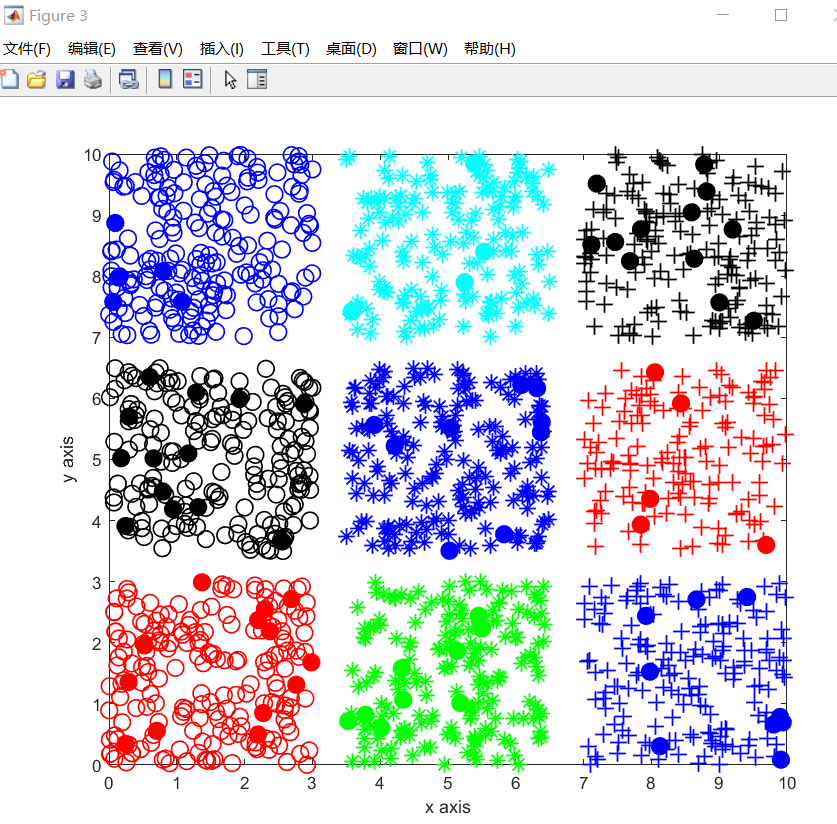
plot(Xt(Ym==9,1),Xt(Ym==9,2),'ko','MarkerFaceColor','k','LineWidth',1,'MarkerSize',10); % 画第九类数据点

hold on;

xlabel('x axis');

ylabel('y axis');

运行结果：



1. 报告使用K近邻预测的错误率

代码：

%%%统计错误率%%%

ero=0;

for i=1:1:m

if Ym(i,1)~=Yt(i,1)

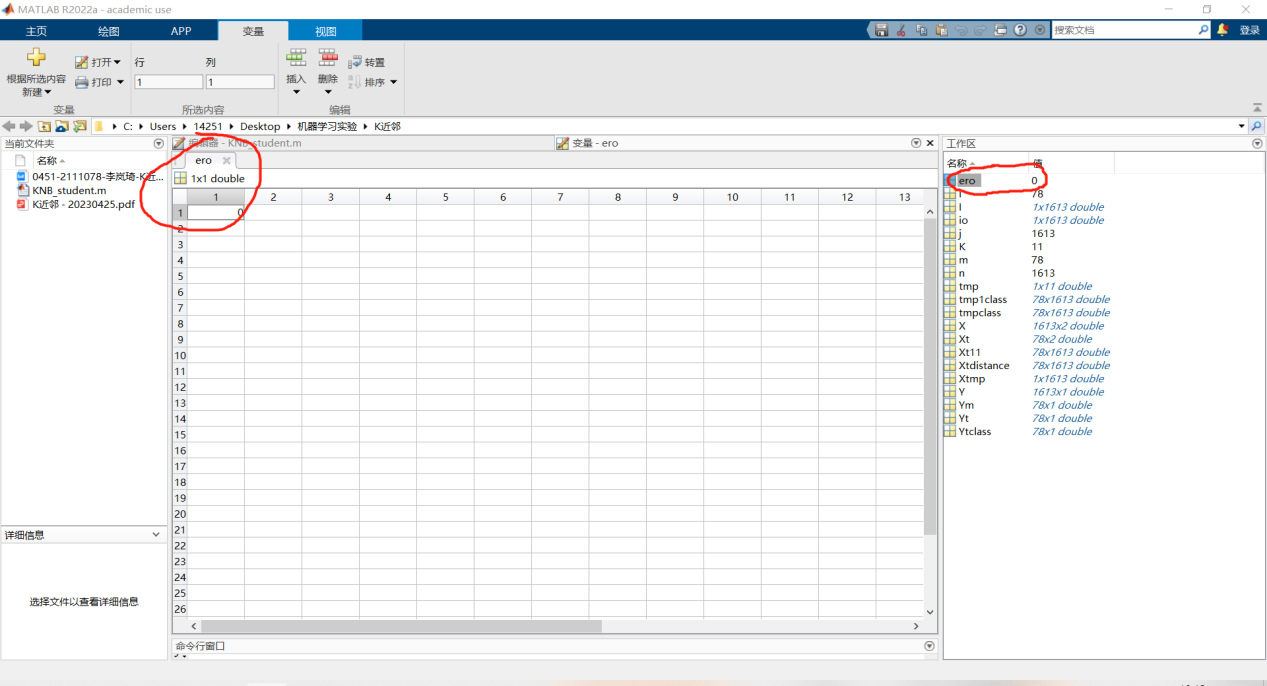
ero=ero+1;

end

end

ero=ero/m;

运行结果：



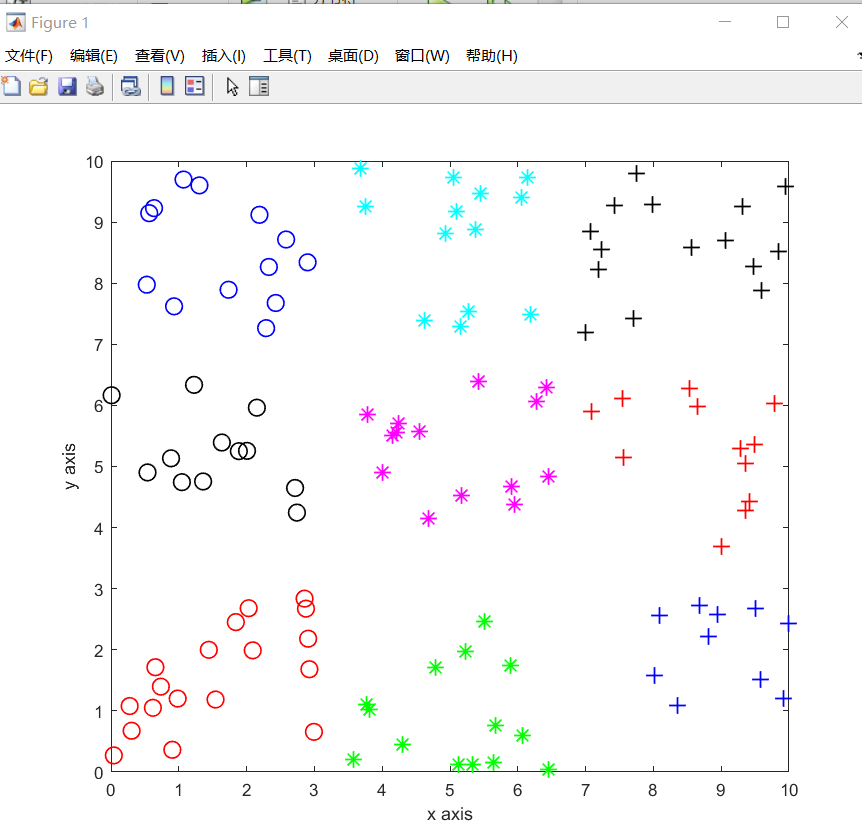
得知，在此次模型预测中，错误率为0

1. 附加题

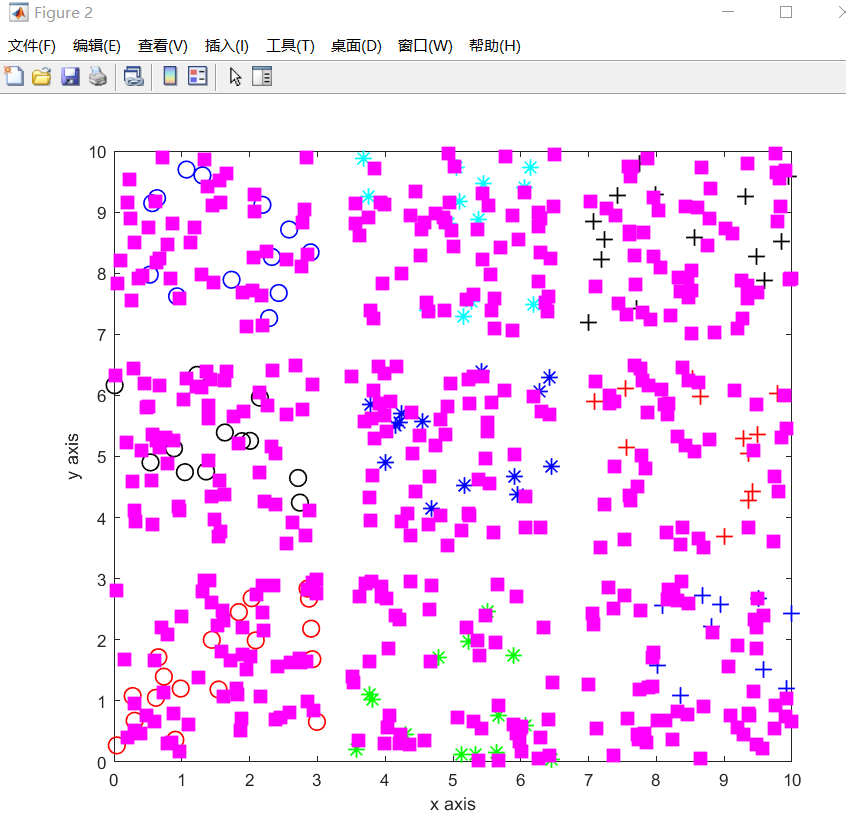
附加题1 : 减少训练样本，增加测试样本，报告错误率

将n改为150，m改为600，运行有：

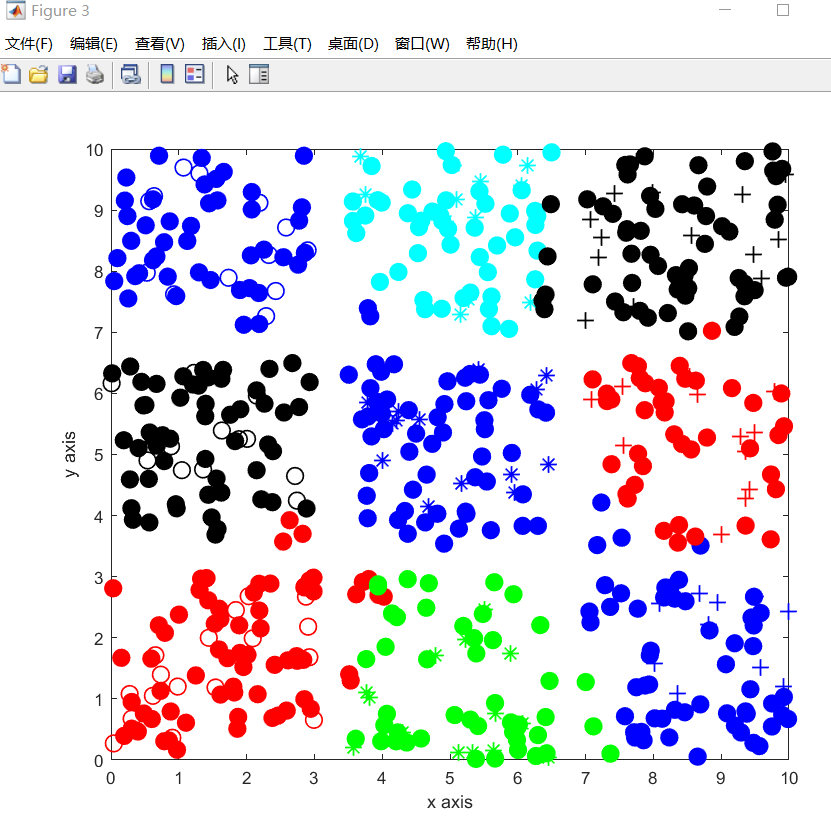
样本点：



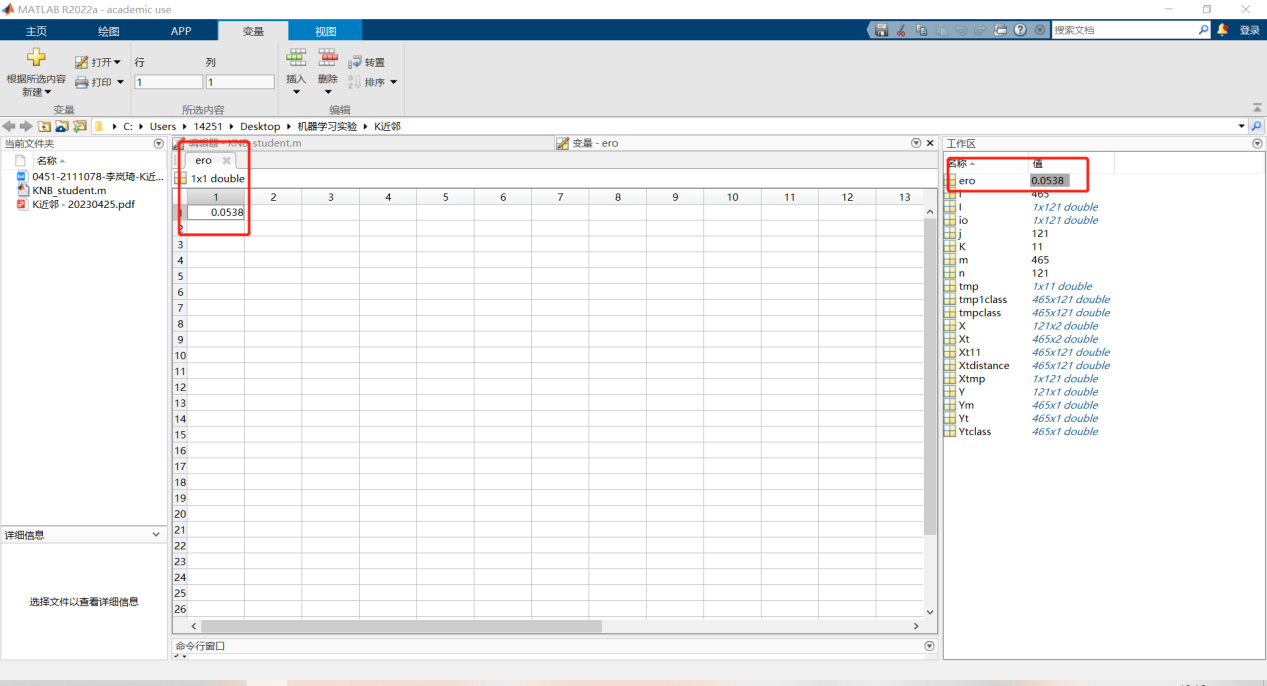
测试点：



预测分类结果：



统计错误率：



可以看出，错误率由原来的0改变为了0.0538，有了提升

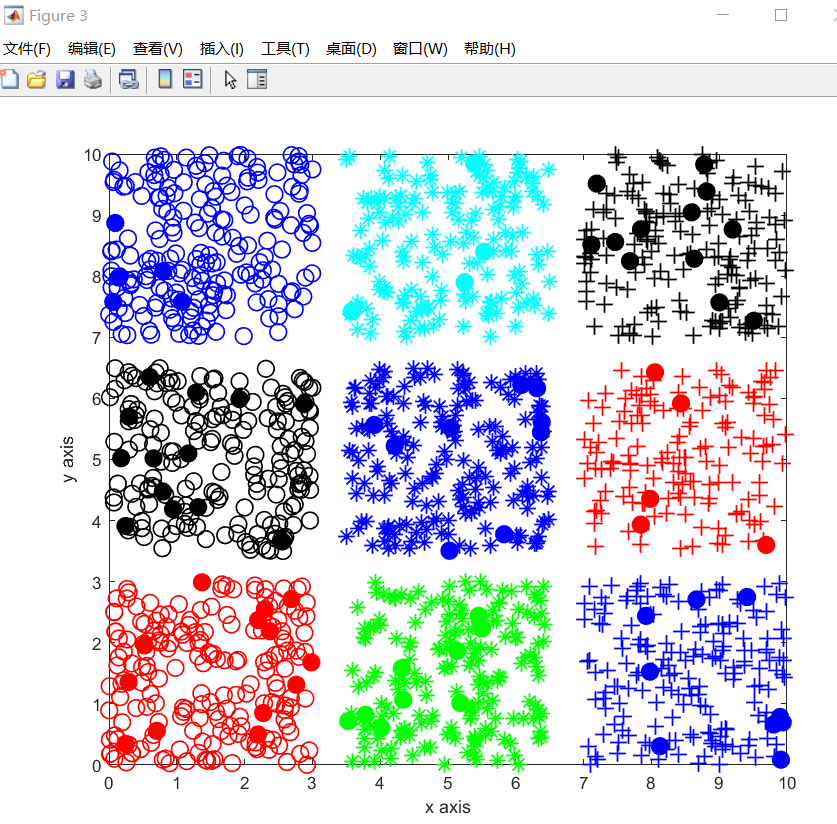
附加题2: 使用不同的度量函数，观测其对分类效果的影响

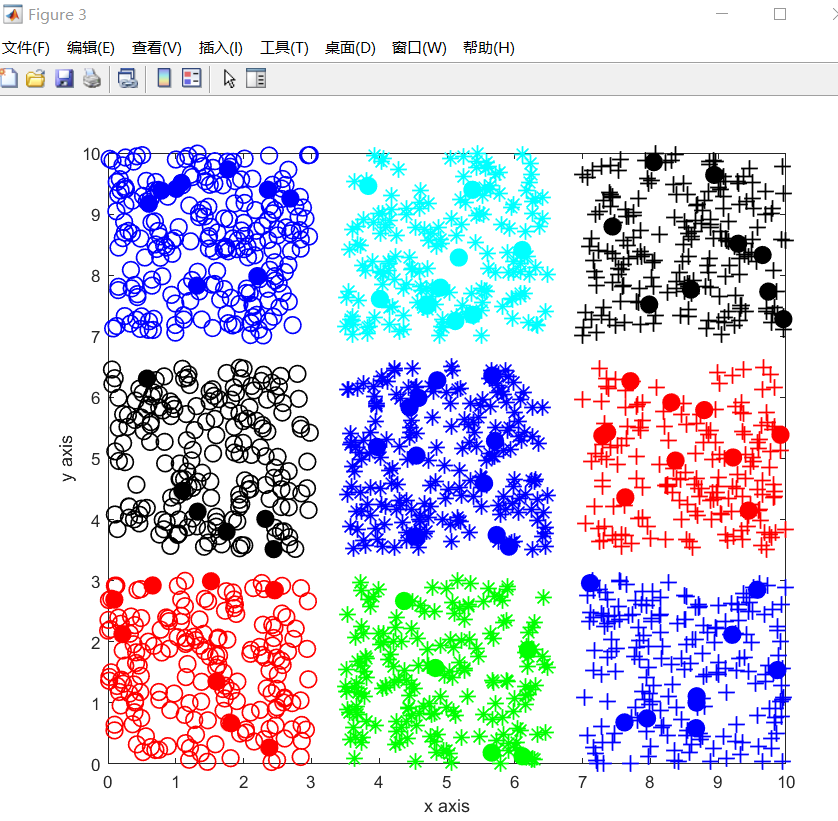
原代码中的距离为欧式几何距离，现将其改变为曼哈顿距离

原代码：Xtdistance(i,j)=(Xt(i,1)-X(j,1))^2+(Xt(i,2)-X(j,2))^2;

改变后的代码：Xtdistance(i,j)=abs(Xt(i,1)-X(j,1))+abs(Xt(i,2)-X(j,2));

原运行结果：



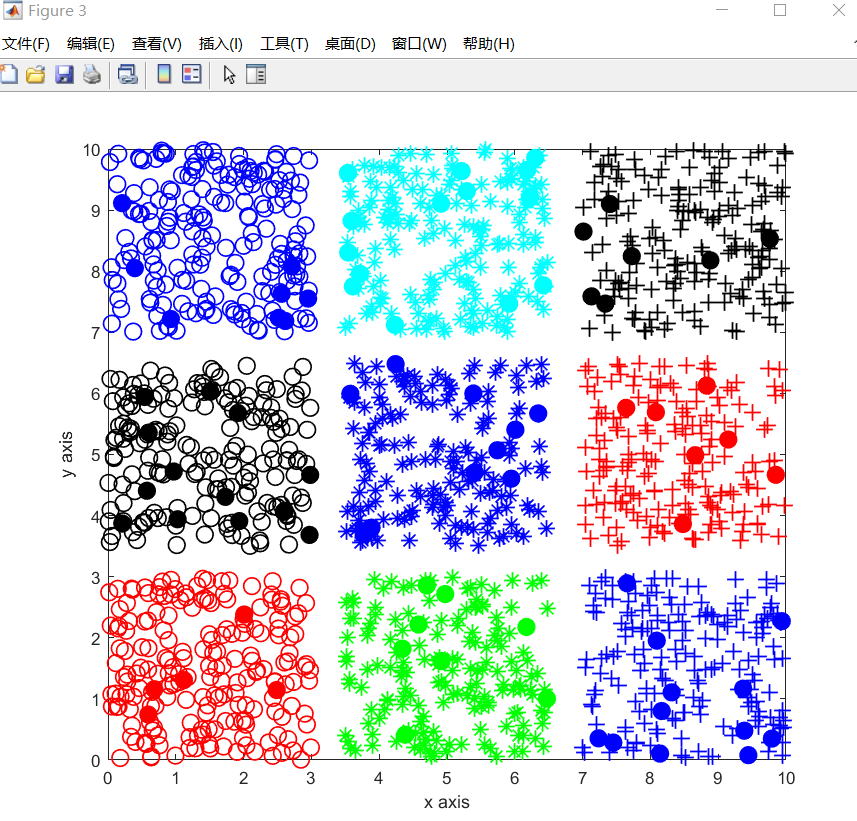
改变后的运行结果：

可以看出，将欧式距离改为曼哈顿距离对于分类结果无显著影响

1. 我们再改变为切比雪夫距离：

代码：Xtdistance(i,j)=max(abs(Xt(i,1)-X(j,1)),abs(Xt(i,2)-X(j,2)));

运行结果：



发现仍无明显区别。推测这是因为本次的样本数据分布较为密集，使得各个距离的度量无明显区分。

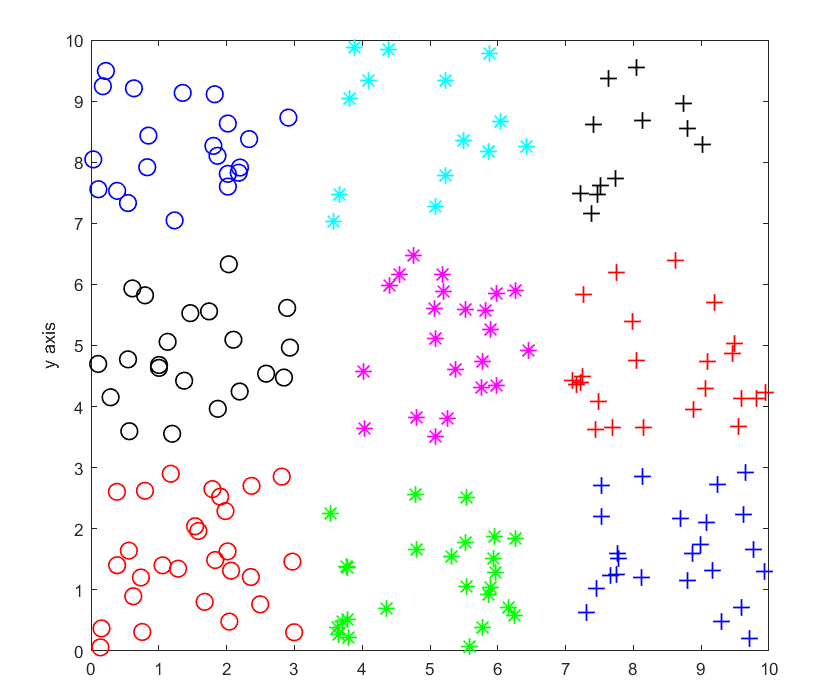
附加题3: 由上一问推测，影响分类效果的可能与样本数据的分布有关，现改变样本数据的分布。

将代码：X = rand(n,2)\*10;

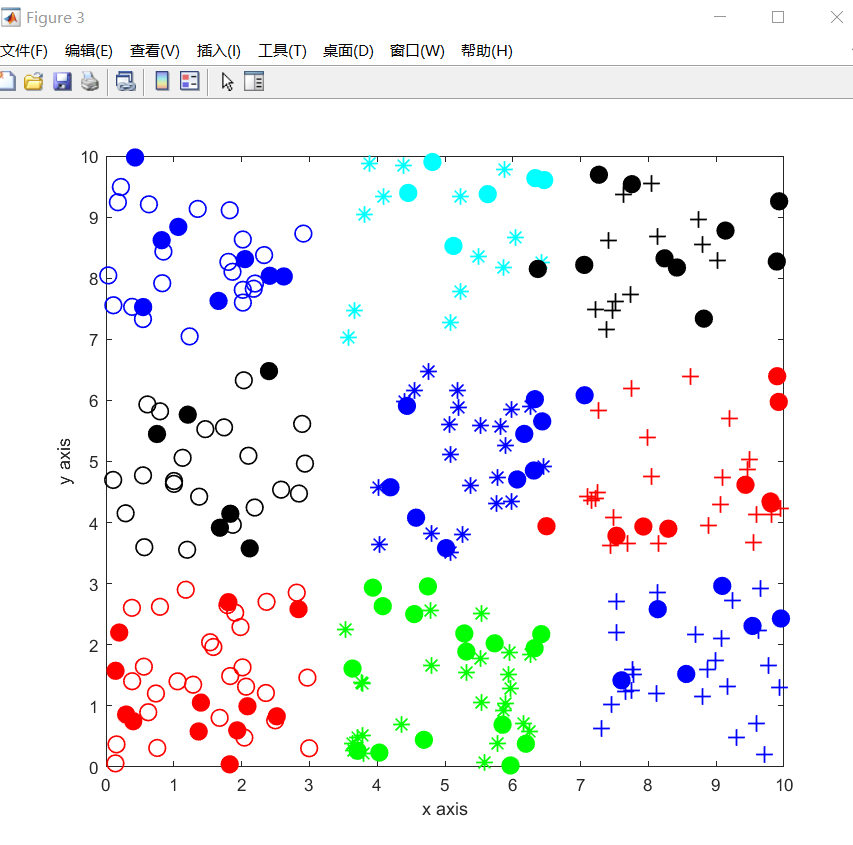
改为：X = randn(n,2)\*10;

运行结果为：

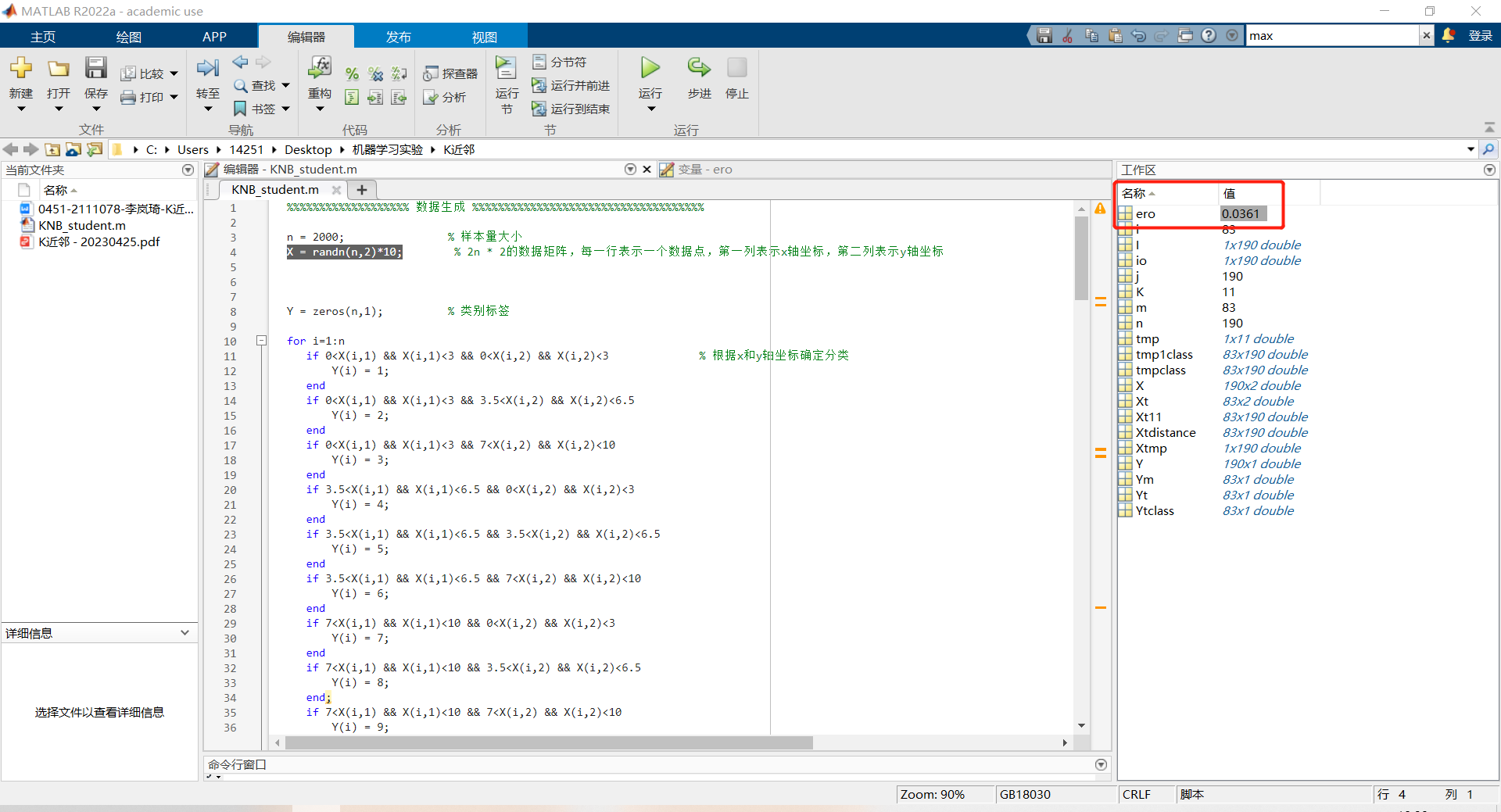
样本分布：



分类预测结果：



错误率为：



错误率变为：0.0361