作业一

关文聪 2016060601008

1. 假设 x=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20),y=(2.94, 4.53, 5.96, 7.88, 9.02, 10.94, 12.14, 13.96, 14.74, 16.68, 17.79, 19.67, 21.20, 22.07, 23.75, 25.22, 27.17, 28.84, 29.84, 31.78).请写出拟合的直线方程，并画图(包括原数据点及拟合的直线)，请打印出来。

解答：使用 matlab 将 x 与 y 输入，并绘制散点图。调用 polyfit 函数，可以对数据进行拟合，设置多项式阶数为 1 即可求出拟合直线的斜率 k 与截距 b，绘图即可。

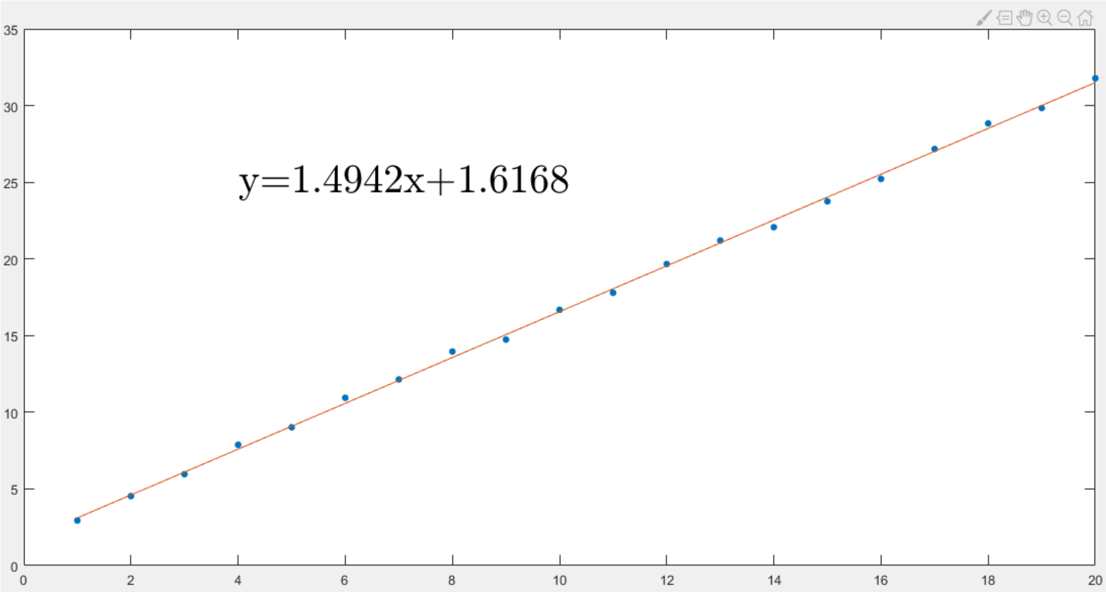
Matlab 代码：

```
x=[1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20];
y=[ 2.94, 4.53, 5.96, 7.88, 9.02, 10.94, 12.14, 13.96, 14.74, 16.68, 17.79, 19.67, 21.20, 22.07,
23.75, 25.22, 27.17, 28.84, 29.84, 31.78];
plot(x,y,'.','MarkerSize',15); %以散点图的形式画出该20组数据点
hold on %不清除图像，使散点图和直线图在同一图中绘制出来'
parameter=polyfit(x,y,1); %调用polyfit函数对数据拟合，由于是线性，因此多项式的阶设为1
k=parameter(1); %斜率k
b=parameter(2); %截距b
line=k*x+b;
plot(x,line); %绘制拟合直线
str=strcat('y=',num2str(k),'x+',num2str(b));
text(4,25,str,'interpreter','latex','fontsize',30) %使用text函数在图上标注出直线方程
```

运行结果：

名称	值
b	1.6168
k	1.4942
line	*1x20 double*
parameter	[1.4942,1.6168]
str	'y=1.4942x+1.6168'
x	*1x20 double*
y	*1x20 double*

运行以上代码，可以求出直线斜率 k 为 1.4942，截距 b 为 1.6168，得到拟合的直线方程为：y=1.4942x+1.6168



将 20 组数据的散点图和拟合的直线绘制在同一张图上，如图所示，可以看出数据点基本上都均匀分布在拟合直线的两侧，拟合结果与数据接近。

2.请使用线性回归模型来拟合 bodyfat 数据。数据集介绍可阅读：https://www.mathworks.com/help/nnet/examples/body-fat-estimation.html

在 matlab 中，在命令行中输入[X,Y] = bodyfat_dataset; 即可获得一个拥有 13 个属性，252 个样本的数据集。使用前 200 个样本来获得模型，并写出你所获得的模型。使用后 52 个样本做测试，汇报你所获得的泛化误差。

解答：这是一个多元线性回归问题，共有 13 个属性。可以设拟合模型的表达式为 y=b+w1x1+w2x2+w3x3+w4x4+w5x5+w6x6+w7x7+w8x8+w9x9+w10x10+w11x11+w12x12+w13x13

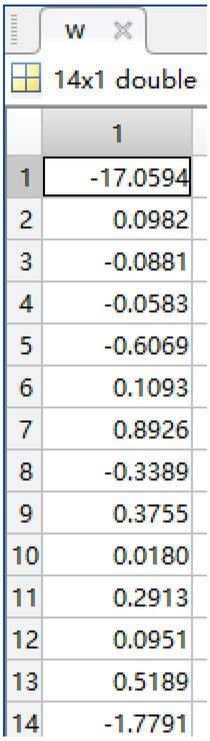
其中 b 为常数项，wi（i=1,2,…13）分别是 xi 对应的系数，待求系数共 14 个。

将前 200 个样本作为训练集（注意要将行向量转置变为列向量，首列全为 1 对应常数项 b），调用 matlab 中的 regress 函数可求得拟合直线方程的系数列向量 w。剩余的 52 个测试数据矩阵与系数列向量 w 即可求得拟合结果。采用均方误差作为泛化误差，对测试结果与真实结果的误差进行评估。

Matlab 代码：

```
[X,Y]=bodyfat_dataset; %获得一个拥有13个属性，252个样本的数据集。X：13*252，Y1*252
trainingX=[ones(200,1),X(:,1:200)’]; %取前200个样本进行训练。注意第一列的1对应拟合直线的常数项，并将行向量转置变为列向量
trainingY=Y(:,1:200)’; %取前200个样本进行训练，并将行向量转置变为列向量
%X训练集：200*14，Y训练集：200*1
testX=[ones(52,1),X(:,201:end)’]; %取剩余52个样本进行测试，并将行向量转置变为列向量
testY=Y(:,201:end)’; %取剩余52个样本进行测试，并将行向量转置变为列向量
%X测试集：52*14，Y测试集：52*1
w=regress(trainingY,trainingX); %调用regress函数进行多元线性回归拟合，计算出系数列向量w
test=testX*w; %X测试集与系数列向量w相乘，得出模型拟合结果集
%用剩余的52个样本计算均方误差
E=0;
for i=1:52
```

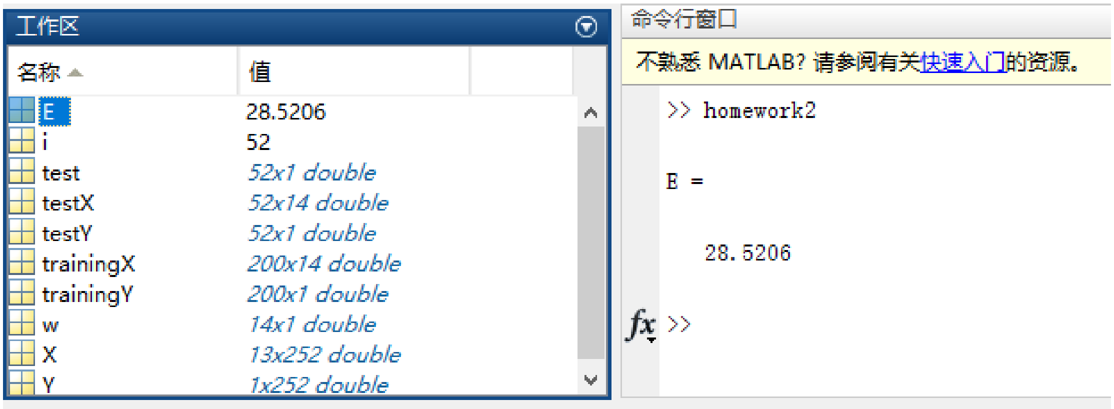
```
    E=E+(test(i)-testY(i))^2;
end
E=E/52
```



w 14x1 double	1
1	-17.0594
2	0.0982
3	-0.0881
4	-0.0583
5	-0.6069
6	0.1093
7	0.8926
8	-0.3389
9	0.3755
10	0.0180
11	0.2913
12	0.0951
13	0.5189
14	-1.7791

运行以上代码，可以求得系数列向量 w，各系数的具体值如图所示。因此，可以得到拟合直线方程为 y=-17.0594+0.0982x1-0.0881x2-0.0583x3-0.6069x4+0.1093x5+0.8926x6-0.3389x7+0.3755x8+0.0180x9+0.2913x10+0.0951x11+0.5189x12-1.7791x13

其中，x1-年龄（年），x2-重量（磅），x3-高度（英寸），x4 颈部周长（厘米），x5-胸围（厘米），x6-腹部 2 周长（厘米），x7-臀部周长（厘米），x8-大腿周长（厘米），x9-膝关节周长（厘米），x10-踝周长（厘米），x11-二头肌（延长）周长（厘米），x12-前臂周长（厘米），x13-腕周长（厘米）



使用拟合的直线方程对后 52 个样本进行测试，计算得均方误差 E=28.5206

3. 编程实现对数回归，并给出教材 89 页上的西瓜数据集 3.0 上的结果。要求采用 4 折交叉验证法来评估结果。因为此处一共 17 个样本，你可以去掉最后一个样本，也可以用所有数据，然后测试用 5 个样本。在汇报结果时，请说明你的选择。请在二维图上画出你的结果（用两种不同颜色或者形状来标注类别），同时打印出完整的代码。

解答：这是一个二分类问题，属性有 2 种（密度、含糖率），结果只有 2 种（好瓜与非好瓜，分别用 1 和 0 表示）。采用逻辑回归模型（Logistic Regression），

取 $g(z)=\frac{1}{1+e^{-z}}$ （Sigmoid Function），代价函数（Cost Function）为

$$J(\theta)=-\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right)+\left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right]$$

，是一个连续可导凸函数，可以使用牛顿法或者梯度下降法等数值方法求解使上式最小的θ值。Fminunc 函数是 matlab 中带的一个最小值优化函数，使用时我们需要提供代价函数和每个参数的求导。若采用梯度下降法，求导可得梯度下降的θ更新的表达式为：

$$\theta_j:=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)x_j^{(i)}$$

。借助 fminunc 函数可以求得使代价函数 J 最小的θ的取值。

选择去掉最后一个样本，数据集共 16 个样本，正、负例各占 1/2.因此，在 4 折交叉验证过程中，采用分层抽样，每个子集含正例负例各 2 个，保证数据分布的一致性。每次选 3 个子集作为训练集，余下的子集作为测试集求正确率，4 次结果取均值。采取 10 次 4 折交叉验证，10 次结果的平均值作为最终评估结果。

Matlab 代码：

```
clear all
clc
dataset=[1  0.697   0.460   1;
    2   0.774   0.376   1;
    3   0.634   0.264   1;
    4   0.608   0.318   1;
    5   0.556   0.215   1;
    6   0.403   0.237   1;
    7   0.481   0.149   1;
    8   0.437   0.211   1;
    9   0.666   0.091   0;
    10  0.243   0.267   0;
    11  0.245   0.057   0;
    12  0.343   0.099   0;
    13  0.639   0.161   0;
    14  0.657   0.198   0;
    15  0.360   0.370   0;
    16  0.593   0.042   0;
    17  0.719   0.103   0];
X=dataset(:,(2:3)); %X取密度与含糖率两个属性
Y=dataset(:,4); %1表示是好瓜，0表示不是好瓜

E=0;
for times=1:10
    posrand=randperm(8);
    negrand=randperm(8,8)+8;
    %去掉最后一个样本采用4折交叉验证，一共需要划分4个子集
    %去掉最后一个样本后，正负例均为8个，因此每个子集应包含正负例各2个
    subset1=[dataset(posrand(1,1:2),2:4);dataset(negrand(1,1:2),2:4)];% 子集1
    subset2=[dataset(posrand(1,3:4),2:4);dataset(negrand(1,3:4),2:4)];% 子集2
    subset3=[dataset(posrand(1,5:6),2:4);dataset(negrand(1,5:6),2:4)];% 子集3
```

```
subset4=[dataset(posrand(1,7:8),2:4);dataset(negrand(1,7:8),2:4)];% 子集4

%绘制正负例分布图
figure;
hold on;
pos=find(Y==1); %正例
neg=find(Y==0); %负例
plot(X(pos,1),X(pos,2),'k+','LineWidth',2,'MarkerSize',7); %画正例点，用“+”表示
plot(X(neg,1),X(neg,2),'o','MarkerFaceColor','r','MarkerSize',7); %画负例点，用“。”表示
xlabel('密度')
ylabel('含糖率')

training1=[subset1;subset2;subset3]; %第一次，取前3个子集作训练集，第4个子集作测试集
X1=training1(:,1:2); %X取密度与含糖率两个属性
Y1=training1(:,3); %1表示是好瓜，0表示不是好瓜
[m,n]=size(X1);
X1=[X1,ones(m,1)]; %在最右端添加1列1以拟合常数项b
initial_theta=zeros(n+1,1); %初始化系数theta
options=optimset('GradObj','on','MaxIter',400);
%调用fminunc函数求解逻辑回归的最佳参数，也就是使costFunction达到最小值的对应参数theta
[theta1,cost1]=fminunc(@(t)(costFunction(t, X1, Y1)),initial_theta,options);
%第4个子集用于测试
X4=subset4(:,1:2);
X4=[X4,ones(4,1)];
%计算正确率
count=0;
for i=1:4
    if((X4(i,:)*theta1>0&&subset4(i,3)==1)||(X4(i,:)*theta1<0&&subset4(i,3)==0))
        count=count+1;
    end
end
e1=count/4;

%绘制判定边界直线
x=0:0.1:0.8;
line=(-theta1(3,1)-theta1(1,1)*x)/theta1(2,1); %判定边界直线方程为:
theta1*x1+theta2*x2+theta3=0，可以反解出theta2的斜截式方程
plot(x,line);
str=strcat('y=',num2str(-theta1(1,1)/theta1(2,1)),'x+',num2str(-theta1(3,1)/theta1(2,1)));
text(0.1,0.45,str,'interpreter','latex','fontsize',20) %使用text函数在图上标注出直线方程


%绘制正负例分布图
figure;
hold on;
pos=find(Y==1); %正例
neg=find(Y==0); %负例
plot(X(pos,1),X(pos,2),'k+','LineWidth',2,'MarkerSize',7); %画正例点，用“+”表示
plot(X(neg,1),X(neg,2),'o','MarkerFaceColor','r','MarkerSize',7); %画负例点，用“。”表示
```

```
xlabel('密度')
ylabel('含糖率')

training2=[subset1;subset2;subset4];%第二次，取1 2 4子集作训练集，第3个子集作测试集
X2=training2(:,1:2);%X取密度与含糖率两个属性
Y2=training2(:,3);%1表示是好瓜，0表示不是好瓜
[m,n]=size(X2);
X2=[X2,ones(m,1)];%在最右端添加1列1以拟合常数项b
initial_theta=zeros(n+1,1);%初始化系数theta
options=optimset('GradObj','on','MaxIter',400);
%调用fminunc函数求解逻辑回归的最佳参数，也就是使costFunction达到最小值的对应参数theta
[theta2,cost2]=fminunc(@(t)(costFunction(t, X2, Y2)),initial_theta,options);
%第3个子集用于测试
X3=subset3(:,1:2);
X3=[X3,ones(4,1)];
%计算正确率
count=0;
for i=1:4
    if((X3(i,:)*theta2>0&&subset3(i,3)==1)||(X3(i,:)*theta2<0&&subset3(i,3)==0))
        count=count+1;
    end
end
e2=count/4;

%绘制判定边界直线
x=0:0.1:0.8;
line=(-theta2(3,1)-theta2(1,1)*x)/theta2(2,1);%判定边界直线方程为：
theta1*x1+theta2*x2+theta3=0，可以反解出theta2的斜截式方程
plot(x,line);
str=strcat('y=',num2str(-theta2(1,1)/theta2(2,1)),'x+',num2str(-theta2(3,1)/theta2(2,1)));
text(0.1,0.45,str,'interpreter','latex','fontsize',20) %使用text函数在图上标注出直线方程

%绘制正负例分布图
figure;
hold on;
pos=find(Y==1);%正例
neg=find(Y==0);%负例
plot(X(pos,1),X(pos,2),'k+','LineWidth',2,'MarkerSize',7);%画正例点，用“+”表示
plot(X(neg,1),X(neg,2),'o','MarkerFaceColor','r','MarkerSize',7);%画负例点，用“。”表示
xlabel('密度')
ylabel('含糖率')

training3=[subset1;subset3;subset4];%第三次，取1 3 4作训练集，第2个子集作测试集
X3=training3(:,1:2);%X取密度与含糖率两个属性
Y3=training3(:,3);%1表示是好瓜，0表示不是好瓜
[m,n]=size(X3);
X3=[X3,ones(m,1)];%在最右端添加1列1以拟合常数项b
initial_theta=zeros(n+1,1);%初始化系数theta
options=optimset('GradObj','on','MaxIter',400);
```

```
%调用fminunc函数求解逻辑回归的最佳参数，也就是使costFunction达到最小值的对应参数theta
[theta3,cost3]=fminunc(@(t)(costFunction(t, X3, Y3)),initial_theta,options);
%第2个子集用于测试
X2=subset2(:,1:2);
X2=[X2,ones(4,1)];
%计算正确率
count=0;
for i=1:4
    if((X2(i,:)*theta3>0&&subset2(i,3)==1)||(X2(i,:)*theta3<0&&subset2(i,3)==0))
        count=count+1;
    end
end
e3=count/4;

%绘制判定边界直线
x=0:0.1:0.8;
line=(-theta3(3,1)-theta3(1,1)*x)/theta3(2,1);%判定边界直线方程为:
theta1*x1+theta2*x2+theta3=0，可以反解出theta2的斜截式方程
plot(x,line);
str=strcat('y=',num2str(-theta3(1,1)/theta3(2,1)),'x+',num2str(-theta3(3,1)/theta3(2,1)));
text(0.1,0.45,str,'interpreter','latex','fontsize',20) %使用text函数在图上标注出直线方程

%绘制正负例分布图
figure;
hold on;
pos=find(Y==1);%正例
neg=find(Y==0);%负例
plot(X(pos,1),X(pos,2),'k+','LineWidth',2,'MarkerSize',7);%画正例点，用“+”表示
plot(X(neg,1),X(neg,2),'o','MarkerFaceColor','r','MarkerSize',7);%画负例点，用“。”表示
xlabel('密度')
ylabel('含糖率')

training4=[subset2;subset3;subset4];%第四次，取2 3 4作训练集，第1个子集作测试集
X4=training4(:,1:2);%X取密度与含糖率两个属性
Y4=training4(:,3);%1表示是好瓜，0表示不是好瓜
[m,n]=size(X4);
X4=[X4,ones(m,1)];%在最右端添加1列1以拟合常数项b
initial_theta=zeros(n+1,1);%初始化系数theta
options=optimset('GradObj','on','MaxIter',400);
%调用fminunc函数求解逻辑回归的最佳参数，也就是使costFunction达到最小值的对应参数theta
[theta4,cost4]=fminunc(@(t)(costFunction(t, X4, Y4)),initial_theta,options);
%第1个子集用于测试
X1=subset1(:,1:2);
X1=[X1,ones(4,1)];
%计算正确率
count=0;
for i=1:4
    if((X1(i,:)*theta4>0&&subset1(i,3)==1)||(X1(i,:)*theta4<0&&subset1(i,3)==0))
        count=count+1;
```

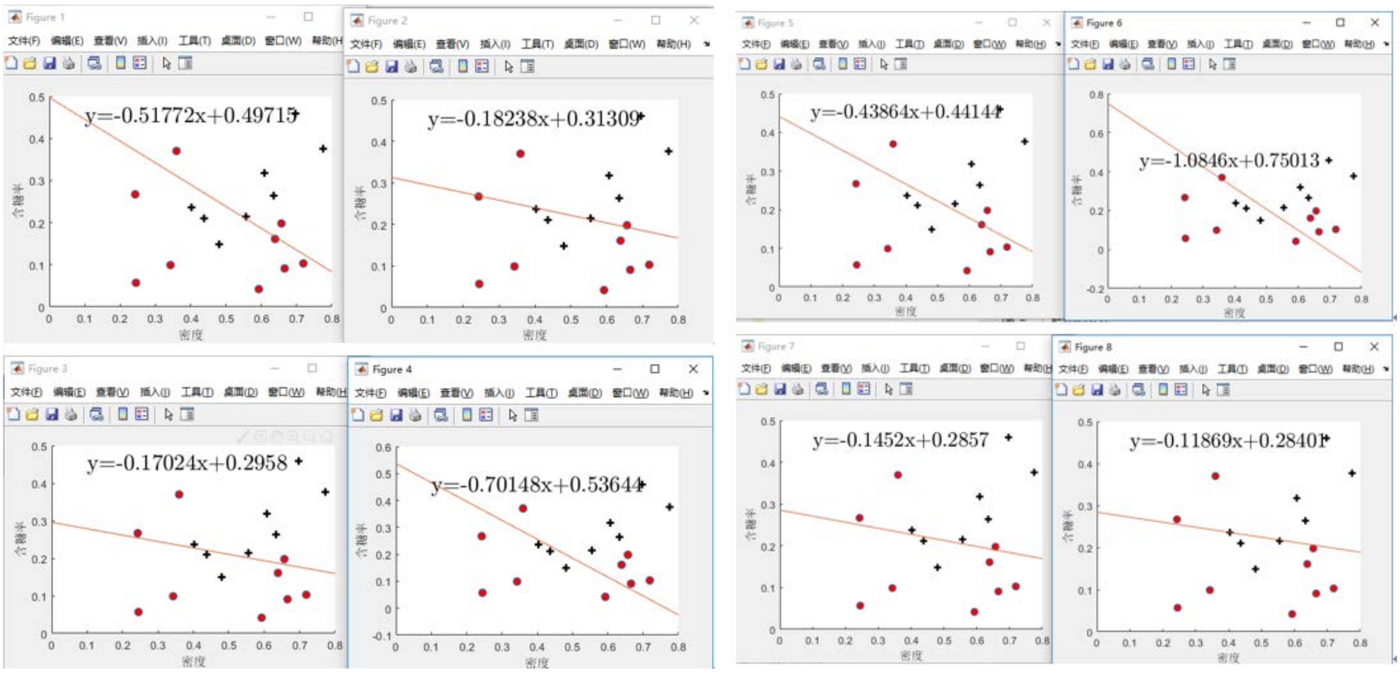
```
        end
    end
    e4=count/4;

    %绘制判定边界直线
    x=0:0.1:0.8;
    line=(-theta4(3,1)-theta4(1,1)*x)/theta4(2,1); %判定边界直线方程为:
theta1*x1+theta2*x2+theta3=0, 可以反解出theta2的斜截式方程
    plot(x,line);
    str=strcat('y=',num2str(-theta4(1,1)/theta4(2,1)),'x+',num2str(-theta4(3,1)/theta4(2,1)));
    text(0.1,0.45,str,'interpreter','latex','fontsize',20) %使用text函数在图上标注出直线方程

    averageE=(e1+e2+e3+e4)/4;
    E=E+averageE;
end
E/10

function [J, grad] = costFunction(theta, X, Y)
m = length(Y); %训练集数据数
grad = zeros(size(theta)); %初始化梯度为0
h=1.0./(1.0+exp(-1*X*theta)); %对数几率函数, 即Sigmoid函数
m=size(Y,1);
J=((-1*Y)'*log(h)-(1-Y)'*log(1-h))/m; %简化后统一形式的costFunction J
for i=1:size(theta,1),
    grad(i)=((h-Y)'*X(:,i))/m; %对costFunction J求导可得梯度的表达式
end
end
```

每组训练结果如图所示:



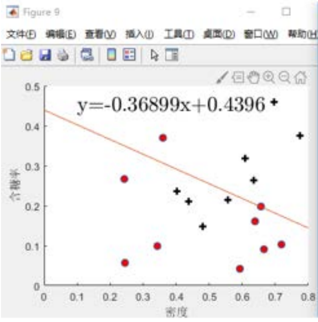


Figure 9
y=-0.36899x+0.4396
含糖率
密度

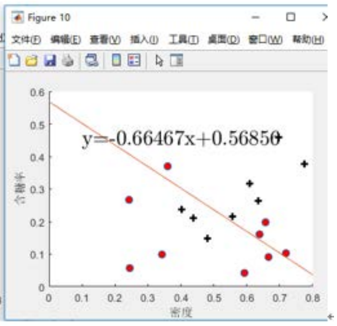



Figure 10
y=-0.66467x+0.56850
含糖率
密度

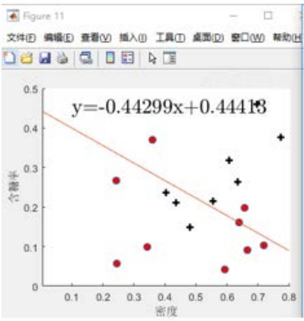



Figure 11
y=-0.44299x+0.44413
含糖率
密度

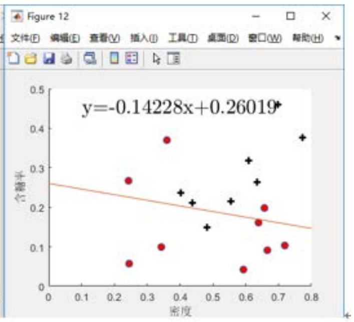



Figure 12
y=-0.14228x+0.26019
含糖率
密度

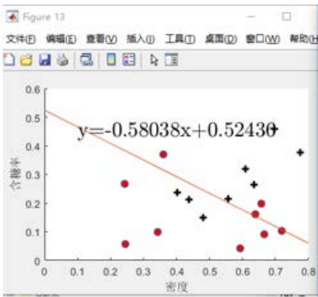



Figure 13
y=-0.58038x+0.52430
含糖率
密度

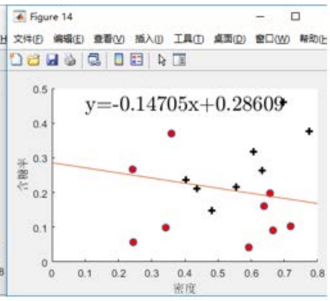



Figure 14
y=-0.14705x+0.28609
含糖率
密度

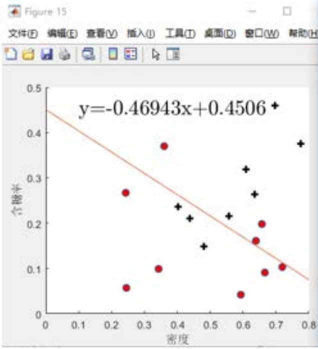



Figure 15
y=-0.46943x+0.4506
含糖率
密度

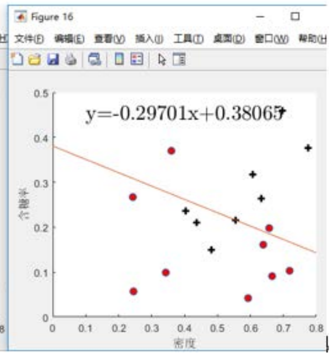



Figure 16
y=-0.29701x+0.38065
含糖率
密度

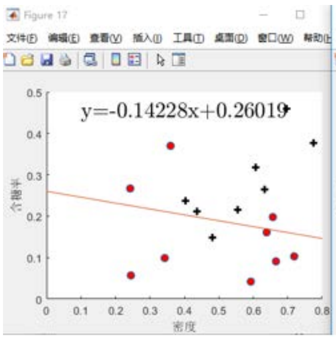



Figure 17
y=-0.14228x+0.26019
含糖率
密度

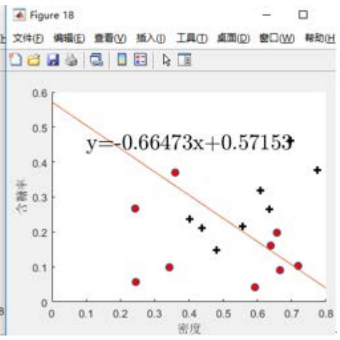



Figure 18
y=-0.66473x+0.57153
含糖率
密度

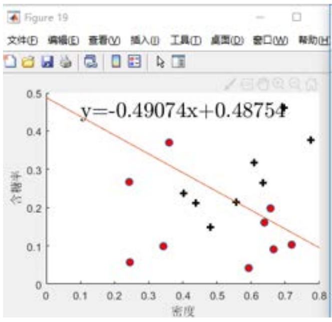



Figure 19
y=-0.49074x+0.48754
含糖率
密度

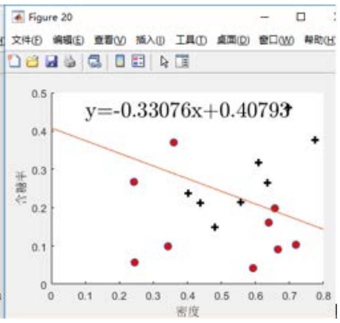



Figure 20
y=-0.33076x+0.40793
含糖率
密度

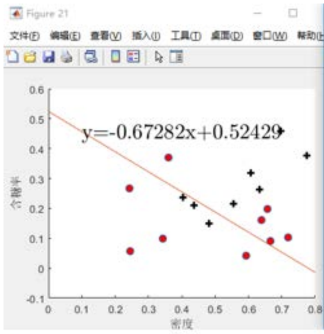



Figure 21
y=-0.67282x+0.52429
含糖率
密度

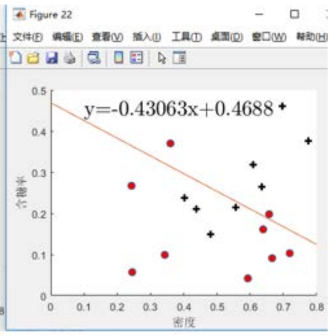



Figure 22
y=-0.43063x+0.4688
含糖率
密度

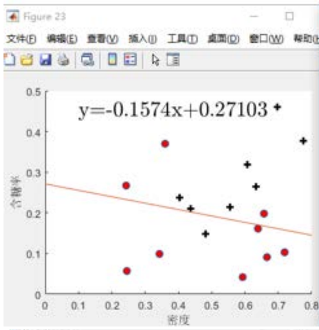



Figure 23
y=-0.1574x+0.27103
含糖率
密度

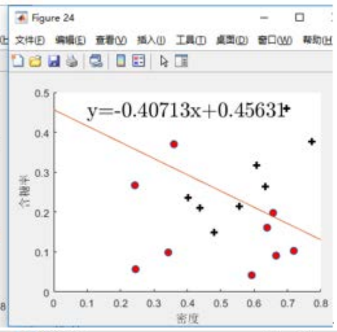



Figure 24
y=-0.40713x+0.45631
含糖率
密度

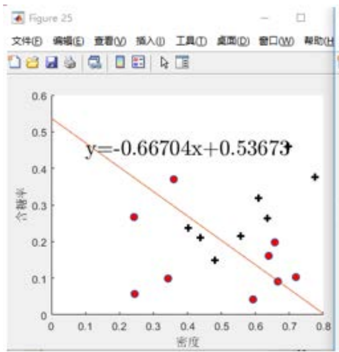



Figure 25
y=-0.66704x+0.53673
含糖率
密度

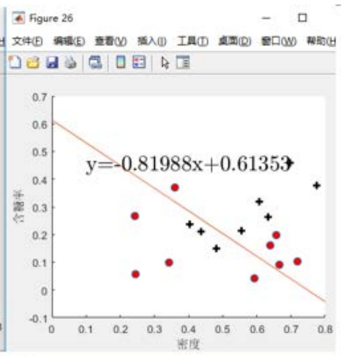



Figure 26
y=-0.81988x+0.61353
含糖率
密度

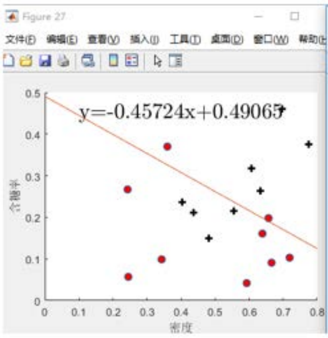



Figure 27
y=-0.45724x+0.49065
含糖率
密度

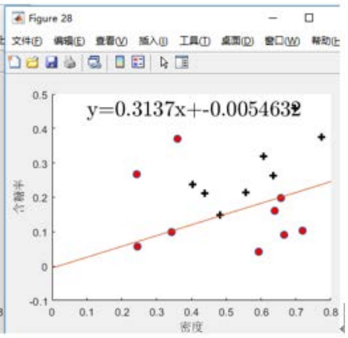



Figure 28
y=0.3137x+-0.0054632
含糖率
密度

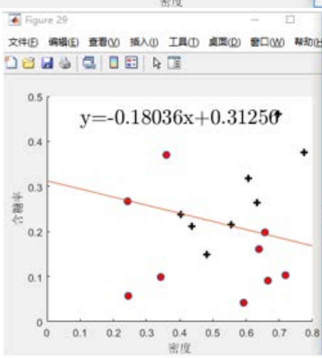



Figure 29
y=-0.18036x+0.31250
含糖率
密度

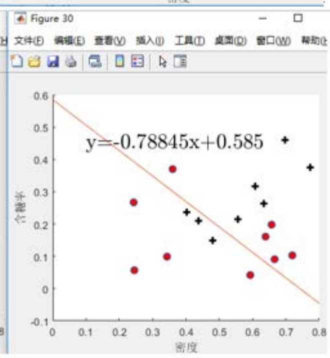



Figure 30
y=-0.78845x+0.585
含糖率
密度

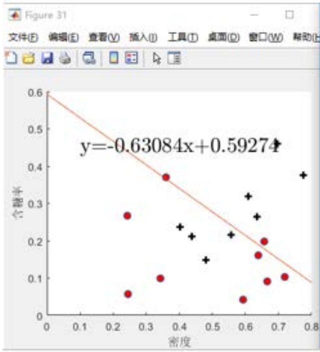



Figure 31
y=-0.63084x+0.59274
含糖率
密度

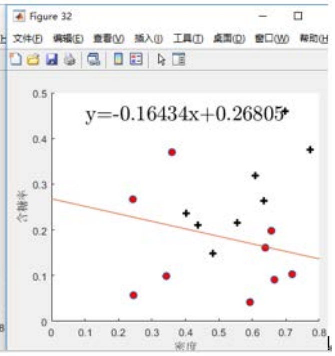
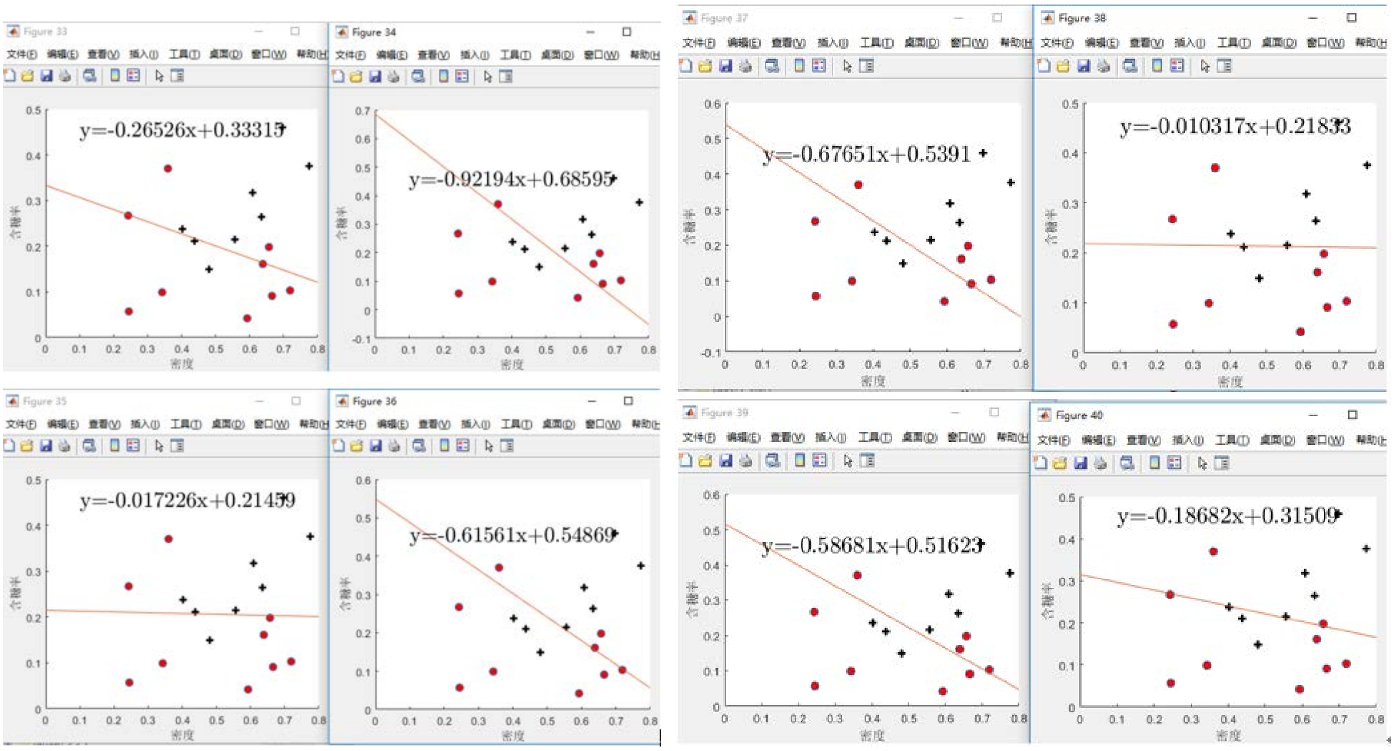



Figure 32
y=-0.16434x+0.26805
含糖率
密度




```
ans =

    0.5875
```

最终计算得平均正确率为 58.75%。正确率不高，这可能与数据集太小，样本量太少有关。

需要注意的是，由于代码采用的是随机分配子集的方式，因此，每次运行的训练、测试集数据都可能不同，得出的计算结果也不一样，本次截图仅仅代表一次运行结果（经过大量运行测试，正确率一般在 60%左右）

参考资料：

[1].《机器学习》.周志华.清华大学出版社
[2].《Machine Learning》.Andrew Ng. Coursera, Standford University