第一问第一小问

clc, clear all, close all

A=xlsread('题面\附件1.xlsx', '性能数据表','C2:J115');

index=find(A(**:**,1)**==**250);

index=[index;115];

name=["A1","A2","A3","A4","A5","A6","A7","A8","A9","A10","A11","A12","A13","A14","B1","B2","B3","B4","B5","B6","B7"];

fid=fopen(['第一题1\','方程及R方.txt'],'w');*%写入文件路径*

**for** i= 1**:**21

    l=index(i);

    r=index(i**+**1)**-**1;

    B=A(l**:**r,**:**);

    T=B(**:**,1);

    yicun=B(**:**,2);

    C4=B(**:**,4);

    [f1,R1]=fit(T,yicun,'poly2');

    [f2,R2]=fit(T,C4,'poly2');

    fprintf(fid,'%s',"乙醇转化率(%): ");

    fprintf(fid,'%s',"y1 = "**+**f1.p1**+**"**\***x**^**2 **+**"**+**f1.p2**+**"**\***x **+**"**+**f1.p3**+**" rsquare**:** ");

    fprintf(fid,"%4f\n",R1.rsquare);

    fprintf(fid,'%s',"C4烯烃选择性(%): ");

    fprintf(fid,'%s',"y2 = "**+**f2.p1**+**"**\***x**^**2 **+**"**+**f2.p2**+**"**\***x **+**"**+**f2.p3**+**" rsquare**:** ");

    fprintf(fid,"%4f\n",R2.rsquare);

    figure

    subplot(2,1,1)

    hold on

    title(name(i))

    scatter(T,yicun);

    plot(f1);

    xlabel("T(温度)");

    ylabel("乙醇转化率(%)");

    subplot(2,1,2)

    hold on

    scatter(T,C4);

    plot(f2);

    xlabel("T(温度)");

    ylabel("C4烯烃选择性(%)");

    saveas(i, "第一题1\"**+**name(i)**+**".png")

**end**

fclose(fid);

第一问第二小问

clc, clear all, close all

A=xlsread('题面\附件2.xlsx', '稳定性测试','A4:H10');*% 红葡萄酒*

fid=fopen(['第一题2\','方程及R方.txt'],'w');*%写入文件路径*

name=["乙醇转化率(%)","乙烯选择性","C4烯烃选择性","乙醛选择性","碳数为4-12脂肪醇","甲基苯甲醛和甲基苯甲醇","其他"];

t=A(**:**,1);

**for** i=1**:**7

    y=A(**:**,i**+**1);

    [f,R]=fit(t,y,'poly2');

    fprintf(fid,'%s',name(i)**+**"**:** ");

    fprintf(fid,'%s',"y = "**+**f.p1**+**"**\***x**^**2 **+**"**+**f.p2**+**"**\***x **+**"**+**f.p3**+**" rsquare**:** ");

    fprintf(fid,"%4f\n",R.rsquare);

    figure

    hold on

    title(name(i))

    scatter(t,y);

    plot(f);

    xlabel("t(时间)");

    ylabel(name(i));

    saveas(i, "第一题2\"**+**name(i)**+**".png")

**end**

fclose(fid);

F=[A(**:**,4),A(**:**,3),A(**:**,5**:**8)];

F=corr(F);

name1=[name(3),name(2),name(4**:**7)];

h = heatmap(name1,name1,F, 'FontSize',10);

colormap("summer")

saveas(1, "第一题2\C4烯烃热力图.png")

第二问

clc, clear all, close all

*%% 导入B组*

x=xlsread('B组/B组.xlsx','sheet1','A2:E41');

yicun=xlsread('B组/B组.xlsx','sheet1','F2:F41');

c4=xlsread('B组/B组.xlsx','sheet1','G2:G41');

*%% 导入A组*

*%x=xlsread('整理后.xlsx', '自变量1','A2:E75');*

*%x=[x(1:54,:);x(60:end,:)];*

*%yicun=xlsread('整理后.xlsx', '因变量1','A2:A75');*

*%yicun=[yicun(1:54,:);yicun(60:end,:)];*

*%c4=xlsread('整理后.xlsx', '因变量1','B2:B75');*

*%c4=[c4(1:54,:);c4(60:end,:)];*

*%y=[yicun,c4];*

*%% 归一化*

t=**x'**;

t=mapminmax(t,0,1);

x=**t'**;

*%% 二次型*

xx=x;

**for** i=1**:**4

**for** j=i**+**1**:**5

        tem=x(**:**,i)**.\***x(**:**,j);

        xx=[xx,tem];

**end**

**end**

**for** i=1**:**5

    tem=x(**:**,i)**.\***x(**:**,i);

    xx=[xx,tem];

**end**

*%% 计算*

stepwise(xx,yicun)

stepwise(xx,c4)

第三问

构建神经网络

**import** pandas **as** pd

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**from** sklearn.neural\_network **import** MLPRegressor

**from** sko.GA **import** GA

plt.rcParams['font.sans-serif'] **=** ['SimHei']

plt.rcParams['axes.unicode\_minus'] **=** False

x**=**pd.read\_excel("A组/A组.xlsx",**skiprows=**[55,56,57,58,59])

y**=**pd.read\_excel("A组/A组.xlsx",**skiprows=**[55, 56, 57, 58, 59])

x **=** x.iloc[:, 0:5]

y **=** y.iloc[:, 7]

x **=** pd.read\_excel("B组/B组.xlsx")

y **=** pd.read\_excel("B组/B组.xlsx")

x **=** x.iloc[:, 0:5]

y **=** y.iloc[:, 7]

model **=** MLPRegressor(**hidden\_layer\_sizes=**(

    50,), **learning\_rate\_init=**0.1,**random\_state=**5,**max\_iter=**500)  *# BP神经网络回归模型*

model.fit(x,y)  *# 训练模型*

遗传算法

**def** fitness1(**p**):

    x1, x2, x3, x4, x5 **=** p

    x **=** pd.DataFrame({"co/sio2的质量百分比": [x1],

                       "co/sio2和HAP装料比": [x2],

                       "co/sio2质量": [x3],

                       "乙醇进气速率": [x4],

                       "温度": [x5]})

    y**=**model.predict(x)

    y**=**float(y)

**return** **-**1 **\*** y  *# 转化为求最小值*

ga **=** GA(**func=**fitness1, **n\_dim=**5, **size\_pop=**50, **max\_iter=**800, **lb=**[0.005, 0.5, 33, 0.3, 250],

**ub=**[0.0500, 2.0000, 200.0000, 2.1000, 450.0000], **precision=**1e-7)

best\_x, best\_y **=** ga.run()

**print**('best\_x:', best\_x, '\n', 'best\_y:', best\_y)

ga **=** GA(**func=**fitness1, **n\_dim=**5, **size\_pop=**50, **max\_iter=**800, **lb=**[0.005, 0.5, 33, 0.3, 250],

**ub=**[0.0500, 2.0000, 200.0000, 2.1000, 350.0000], **precision=**1e-7)

best\_x, best\_y **=** ga.run()

**print**('best\_x:', best\_x, '\n', 'best\_y:', best\_y)