import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn import model\_selection

from sklearn.linear\_model import Lasso, LassoCV

diabetes = pd.read\_excel('.\\筛选后数据2.xlsx') # 读取数据集d

print(diabetes)

# 构造自变量，剔除患者性别，年龄和因变量

predictors = diabetes.columns[1: -1]

X\_train, X\_test, y\_train, y\_test = model\_selection.train\_test\_split(diabetes[predictors], diabetes['传统燃油汽车销量'], test\_size=0.2,

random\_state=1234)

Lambdas = np.logspace(-5, 2, 200)

lasso\_cofficients = []

for Lambda in Lambdas:

lasso = Lasso(alpha=Lambda, max\_iter=10000)

lasso.fit(X\_train, y\_train)

lasso\_cofficients.append(lasso.coef\_)

# 绘制Lambda与回归系数的这线图

plt.plot(Lambdas, lasso\_cofficients)

plt.xscale('log')

plt.xlabel('Lambda')

plt.ylabel('Cofficients')

plt.show()

# 交叉验证

lasso\_cv = LassoCV(alphas=Lambdas, cv = 10, max\_iter = 10000)

lasso\_cv.fit(X\_train, y\_train)

# 输出最佳的lambda数值

lasso\_best\_alpha = lasso\_cv.alpha\_

print('最佳lambda值：',lasso\_best\_alpha)

# 模型的预测

# 系数输出

lasso = Lasso(alpha=lasso\_best\_alpha, max\_iter=10000)

lasso\_model = lasso.fit(X\_train, y\_train)

res = pd.Series(index = ['Intercept'] + X\_train.columns.tolist(),

data = [lasso.intercept\_] + lasso.coef\_.tolist())

print('模型系数：')

print(res)

from sklearn.metrics import mean\_squared\_error

# 预测

lasso\_predict = lasso.predict(X\_test)

RMSE = np.sqrt(mean\_squared\_error(y\_test, lasso\_predict))

print('预测数据：',RMSE)