#Method 1

import pandas as pd

import numpy as np

from scipy.optimize import least\_squares as lsq

from scipy.optimize import minimize

xls = pd.ExcelFile('midterm.xlsx')

AGEbiggerthan60 = pd.read\_excel(xls,'Sheet2')

AGESAMLLERthan21 = pd.read\_excel(xls,'Sheet3')

AGEbiggerthan60 = AGEbiggerthan60.drop([0])

AGEbiggerthan60.columns = AGEbiggerthan60.loc[1]

AGEbiggerthan60 = AGEbiggerthan60.drop([1])

x = AGEbiggerthan60.loc[:,['Age','job1','job2','job3','rent',\

'free','little','moderate','rich','Credit amount','Duration']]

y = AGEbiggerthan60['risk encode'].astype("int")

AGESAMLLERthan21.columns = AGESAMLLERthan21.loc[0]

AGESAMLLERthan21 = AGESAMLLERthan21.iloc[1:,:]

x\_test = AGESAMLLERthan21.iloc[:,1:12]

y\_test = AGESAMLLERthan21['risk encode'].astype('int')

X0 = pd.DataFrame(np.ones(47),columns=['intercept'])

X0 = X0.drop(0)

X0 = X0.drop(1)

x = pd.concat((X0,x.iloc[:,0:11]),axis = 1).astype("int")

def log\_likelihood (Beta):

exp\_x\_beta = np.exp(np.dot(x,Beta))

pi = exp\_x\_beta/(1+exp\_x\_beta)

each\_row = y\*np.log(pi)+(1-y)\*np.log(1-pi)

return np.sum(each\_row)\*(-1)

lr\_us\_1 = minimize(log\_likelihood , [0,0,0,0,0,0,0,0,0,0,0,0], method='BFGS')

#lr\_us\_2 = lsq(log\_likelihood , [0,0,0,0,0,0,0,0,0,0,0,0], method='dogbox')

print(lr\_us\_1.x)

#print(lr\_us\_2.x)

#Method 2

import pandas as pd

import numpy as np

xls = pd.ExcelFile('midterm.xlsx')

AGEbiggerthan60 = pd.read\_excel(xls,'Sheet2')

AGESAMLLERthan21 = pd.read\_excel(xls,'Sheet3')

AGEbiggerthan60 = AGEbiggerthan60.drop([0])

AGEbiggerthan60.columns = AGEbiggerthan60.loc[1]

AGEbiggerthan60 = AGEbiggerthan60.drop([1])

AGESAMLLERthan21.columns = AGESAMLLERthan21.loc[0]

AGESAMLLERthan21 = AGESAMLLERthan21.iloc[1:,:]

x\_test = AGESAMLLERthan21.iloc[:,1:12]

y\_test = AGESAMLLERthan21['risk encode'].astype('int')

x = AGEbiggerthan60.loc[:,['Age','job1','job2','job3','rent',\

'free','little','moderate','rich','Credit amount','Duration']]

y = AGEbiggerthan60['risk encode'].astype("int")

#practice logistic

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import cross\_val\_score

#

logreg = LogisticRegression(solver='newton-cg')

logit = logreg.fit(x,y)

logit.coef\_

#cross\_val\_score(logreg,x,y.values.ravel(),cv=5,scoring = 'accuracy').mean()

#AGEbiggerthan60.loc[[]

#test

logit.predict(x\_test)

logit.score(x\_test,y\_test)