主成分分析

# 数据处理

import pandas as pd

import numpy as np

# 绘图

import seaborn as sns

import matplotlib.pyplot as plt

df = pd.read\_excel(r"C:\Users\阿韩想养二哈\Desktop\数据集aa.xls", index\_col=0).reset\_index(drop=True)

print(df)

# Bartlett's球状检验

from factor\_analyzer.factor\_analyzer import calculate\_bartlett\_sphericity

chi\_square\_value, p\_value = calculate\_bartlett\_sphericity(df)

print(chi\_square\_value, p\_value)

# KMO检验

# 检查变量间的相关性和偏相关性，取值在0-1之间；KOM统计量越接近1，变量间的相关性越强，偏相关性越弱，因子分析的效果越好。

# 通常取值从0.6开始进行因子分析

from factor\_analyzer.factor\_analyzer import calculate\_kmo

kmo\_all, kmo\_model = calculate\_kmo(df)

print(kmo\_all)

# #标准化

# #所需库

# from sklearn import preprocessing

# #进行标准化

# df = preprocessing.scale(df)

# print(df)

# #求解系数相关矩阵

# covX = np.around(np.corrcoef(df.T),decimals=3)

# print(covX)

# #求解特征值和特征向量

# featValue, featVec= np.linalg.eig(covX.T) #求解系数相关矩阵的特征值和特征向量

# print(featValue, featVec)

#不标准化

#均值

def meanX(dataX):

return np.mean(dataX,axis=0)#axis=0表示依照列来求均值。假设输入list,则axis=1

average = meanX(df)

print(average)

#查看列数和行数

m, n = np.shape(df)

print(m,n)

#均值矩阵

data\_adjust = []

avgs = np.tile(average, (m, 1))

print(avgs)

#去中心化

data\_adjust = df - avgs

print(data\_adjust)

#协方差阵

covX = np.cov(data\_adjust.T) #计算协方差矩阵

print(covX)

#计算协方差阵的特征值和特征向量

featValue, featVec= np.linalg.eig(covX) #求解协方差矩阵的特征值和特征向量

print(featValue, featVec)

####下面没有区分#######

#对特征值进行排序并输出 降序

featValue = sorted(featValue)[::-1]

print(featValue)

#绘制散点图和折线图

# 同样的数据绘制散点图和折线图

plt.scatter(range(1, df.shape[1] + 1), featValue)

plt.plot(range(1, df.shape[1] + 1), featValue)

# 显示图的标题和xy轴的名字

# 最好使用英文，中文可能乱码

plt.title("Scree Plot")

plt.xlabel("Factors")

plt.ylabel("Value")

plt.grid() # 显示网格

plt.show() # 显示图形

#求特征值的贡献度

gx = featValue/np.sum(featValue)

print(gx)

#求特征值的累计贡献度

lg = np.cumsum(gx)

print(lg)

#选出主成分

k=[i for i in range(len(lg)) if lg[i]<0.90]

k = list(k)

print(k)

#选出主成分对应的特征向量矩阵

selectVec = np.matrix(featVec.T[k]).T

selectVe=selectVec\*(-1)

print(selectVec)

#主成分得分

#标准化第一个参数是df  
#非标准化第一个参数是data\_adjust

finalData = np.dot(data\_adjust,selectVec)

print(finalData)

#绘制热力图

plt.figure(figsize = (14,14))

ax = sns.heatmap(selectVec, annot=True, cmap="BuPu")

# 设置y轴字体大小

ax.yaxis.set\_tick\_params(labelsize=15)

plt.title("Factor Analysis", fontsize="xx-large")

# 设置y轴标签

plt.ylabel("Sepal Width", fontsize="xx-large")

# 显示图片

plt.show()