

专业课程实验报告

课程名称： Python程序设计

开课学期： 2020 至 2021 学年 第 2 学期

专业: 软件工程 年级班级： 2019.3

学生姓名： 冯春霖 学号：222019321062074

实验教师：王超

计算机与信息科学学院 软件学院

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验项目名称 | | 使用scikit-learn构建模型 | | | |
| 实验时间 | | 2021年6月10日 | 实验类型 | | □验证性 **□**设计性 □综合性 |
| 一、实验目的  1. 掌握使用sklearn处理数据集的相关方法  2. 构建K-means聚类模型  3. 构建SVM分类模型  4. 构建回归模型  二、实验要求  通过对wine和wine\_quality数据集的分析，使用scikit-learn库构建模型 | | | | | |
| 三、实验内容与设计（主要内容，操作步骤、算法描述或程序代码）  # 实训1  from sklearn.metrics import explained\_variance\_score, mean\_absolute\_error, \      mean\_squared\_error, median\_absolute\_error, r2\_score  from sklearn.svm import SVC  from sklearn.metrics import fowlkes\_mallows\_score  from sklearn.preprocessing import StandardScaler  from sklearn.ensemble import GradientBoostingRegressor  from sklearn.linear\_model import LinearRegression  from sklearn.metrics import classification\_report  from sklearn.preprocessing import MinMaxScaler  from sklearn.metrics import calinski\_harabasz\_score  import matplotlib.pyplot as plt  from sklearn.metrics import silhouette\_score  from sklearn.cluster import KMeans  from sklearn.decomposition import PCA  from sklearn.model\_selection import train\_test\_split  import pandas as pd  wine = pd.read\_csv('./wine.csv', encoding='UTF-8')  WQ = pd.read\_csv('./winequality.csv', sep=';', encoding='UTF-8')  # 拆分数据与标签  wineData = wine.iloc[:, 1:]  WQData = WQ.iloc[:, :-1]  wineTarget = wine['Class']  WQTarget = WQ['quality']  # 划分数据集为训练集和测试集  wineDataTrain, wineDataTest, wineTargetTrain, wineTargetTest \      = train\_test\_split(wineData, wineTarget, test\_size=0.1, random\_state=6)  WQDataTrain, WQDataTest, WQTargetTrain, WQTargetTest \      = train\_test\_split(WQData, WQTarget, test\_size=0.1, random\_state=6)  # 对数据集进行标准化  stdScale = StandardScaler().fit(wineDataTrain)  wineTrainScaler = stdScale.transform(wineDataTrain)  wineTestScaler = stdScale.transform(wineDataTest)  stdScale = StandardScaler().fit(WQDataTrain)  WQTrainScaler = stdScale.transform(WQDataTrain)  WQTestScaler = stdScale.transform(WQDataTest)  # PCA降维  pca = PCA(n\_components=5).fit(wineTrainScaler)  wineTrainPca = pca.transform(wineTrainScaler)  wineTestPca = pca.transform(wineTestScaler)  pca = PCA(n\_components=5).fit(WQTrainScaler)  WQTrainPca = pca.transform(WQTrainScaler)  WQTestPca = pca.transform(WQTestScaler)  # 实训2  # 构建模型  kmeans = KMeans(n\_clusters=3, random\_state=1).fit(wineTrainScaler)  print('构建的KMeans模型为：\n', kmeans)  # 求FMI  score = fowlkes\_mallows\_score(wineTargetTrain, kmeans.labels\_)  print("wine数据集的FMI:%f" % (score))  # 在聚类数目为2-10类时，确定最优聚类数目  for i in range(2, 11):      kmeans = KMeans(n\_clusters=i, random\_state=123).fit(wineTrainScaler)      score = fowlkes\_mallows\_score(wineTargetTrain, kmeans.labels\_)      print('聚类数目为%d类时FMI评价分值为：%f' % (i, score))  # 求取模型的轮廓系数，绘制轮廓系数折线图，确定最优聚类数目  silhouettteScore = []  for i in range(2, 11):      kmeans = KMeans(n\_clusters=i, random\_state=1).fit(wine)      score = silhouette\_score(wine, kmeans.labels\_)      silhouettteScore.append(score)  plt.figure(figsize=(10, 6))  plt.plot(range(2, 11), silhouettteScore, linewidth=1.5, linestyle="-")  plt.show()  # 求取 Calinski-Harabasz指数，确定最优聚类数  for i in range(2, 11):      kmeans = KMeans(n\_clusters=i, random\_state=1).fit(wine)      score = calinski\_harabasz\_score(wine, kmeans.labels\_)      print('聚类数目为%d时calinski\_harabaz指数为：%f' % (i, score))  # 实训3  # 使用离差标准化方法标准化wine数据集  stdScale = MinMaxScaler().fit(wineDataTrain)  wineTrainScaler = stdScale.transform(wineDataTrain)  wineTestScaler = stdScale.transform(wineDataTest)  # 构建SVM模型预测测试集结果  svm = SVC().fit(wineTrainScaler, wineTargetTrain)  wineTargetPred = svm.predict(wineTestScaler)  print('预测前10个结果为：\n', wineTargetPred[:10])  # 打印出分类报告,评价分类模型性能  print('使用SVM预测iris数据的分类报告为：\n', classification\_report        (wineTargetTest, wineTargetPred))  # 实训4  # 构建线性回归模型  clf = LinearRegression().fit(WQTrainPca, WQTargetTrain)  y\_pred = clf.predict(WQTestPca)  print('线性回归模型预测前10个结果为：', '\n', y\_pred[:10])  # 构建梯度提升回归模型  GBR\_wine = GradientBoostingRegressor().\      fit(WQTrainPca, WQTargetTrain)  wine\_target\_pred = GBR\_wine.predict(WQTestPca)  print('梯度提升回归模型预测前10个结果为：', '\n', wine\_target\_pred[:10])  print('真实标签前十个预测结果为：', '\n', list(WQTargetTest[:10]))  # 结合真实评分和预测评分，计算均方误差、中值绝对误差、可解释方差值并判定模型的性能优劣  print('线性回归模型评价结果：')  print('winequality数据线性回归模型的平均绝对误差为：',        mean\_absolute\_error(WQTargetTest, y\_pred))  print('winequality数据线性回归模型的均方误差为：',        mean\_squared\_error(WQTargetTest, y\_pred))  print('winequality数据线性回归模型的中值绝对误差为：',        median\_absolute\_error(WQTargetTest, y\_pred))  print('winequality数据线性回归模型的可解释方差值为：',        explained\_variance\_score(WQTargetTest, y\_pred))  print('winequality数据线性回归模型的R方值为：',        r2\_score(WQTargetTest, y\_pred))  print('梯度提升回归模型评价结果：')  print('winequality数据梯度提升回归树模型的平均绝对误差为：',        mean\_absolute\_error(WQTargetTest, wine\_target\_pred))  print('winequality数据梯度提升回归树模型的均方误差为：',        mean\_squared\_error(WQTargetTest, wine\_target\_pred))  print('winequality数据梯度提升回归树模型的中值绝对误差为：',        median\_absolute\_error(WQTargetTest, wine\_target\_pred))  print('winequality数据梯度提升回归树模型的可解释方差值为：',        explained\_variance\_score(WQTargetTest, wine\_target\_pred))  print('winequality数据梯度提升回归树模型的R方值为：',        r2\_score(WQTargetTest, wine\_target\_pred)) | | | | | |
| 三、测试数据和执行结果 （在给定数据下，执行操作、算法和程序的结果，可使用数据、图表、截图等给出）  **实训2：**    观察运行结果可知：聚类数目为3类时FMI评价分值最高，故聚类为3类时K-means聚类效果最好，聚类数目为10时Calinski-Harabaz指数最大，考虑到数据的实际情况，Calinski-Harabaz可能不适宜作为wine数据集的聚类模型  **实训3：**    训练集与测试集比例为9：1时，各项指标均为1.00，故可认为9：1是性能较好的划分方法  **实训4：** | | | | | |
| 四、实验结果分析及总结（对实验的结果是否达到预期进行分析，总结实验的收获和存在的问题等）  本次实验完成了题目要求的所有代码，但对实验结果的分析仍有部分问题。如在实训4中，对回归模型的结果不能很好的理解，无法判断联众回归模型的优劣，体现出我在数据分析和建模相关知识上还有所欠缺。除实训4外，其他任务均较好的完成。通过本次实验我对使用Python的scikit-learn库建立模型进行数据分析有了一定的了解，学会了使用库中一些简单的方法和模型，如K-means聚类模型、SVM分类模型和回归模型，并通过实践加深了对模型使用的理解。 | | | | | |
| 教  师  评  阅 | 实验内容和设计（A-E）： | | |  | |
| 操作过程、算法或代码（A-E）： | | |  | |
| 实验结果（A-E）： | | |  | |
| 实验分析和总结（A-E）： | | |  | |
| 实验成绩（A-E）：  反馈评语： | | | | |