

**实验报告册**

|  |  |
| --- | --- |
| 学年学期： | 2023 -2024学年 第1学期 |
| 课程名称： | 数据工程综合实践 |
| 学生学院： | 计算机科学与技术学院/人工智能学院 |
| 专业班级： | 04012104 |
| 学生学号： | 2021212151 |
| 学生姓名： | 周昭光 |
| 联系电话： | 15826301649 |

**重庆邮电大学教务处制**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 周昭光 | 学生学号 | 2021212151 | 专业班级 | 04012104 |
| 校内指导教师 | 王进 | | | | |
| 实验日期 | 2023.12 | 实验地点 | 综合实验楼B501/B502 | | |
| 实验名称 | 数据工程综合实践-任务一 | | | | |
| 评阅人签字 | 王进 | | | 成绩 |  |
| **一、实验目的和要求** | | | | | |
| 1. 完成 Python 机器学习库与未集成算法的安装   2. 了解机器学习基本概念  3. 学习基础、中级 Python 编程  4. 学习使用第三方 Python 库  5. 了解O2O 优惠券使用预测算法基本框架  6. 完成O2O 优惠券使用预测项目解读  7. 学习数据分析与预处理方法  8. 学习数据划分与打标方法  9. 完成O2O 优惠券使用预测 Baseline | | | | | |
| **二、实验设备** | | | | | |
| 1、操作系统：Windows  2、基本硬件配置要求：处理器Intel i3、内存8G；  3、Python及其组件版本：  Python 3.6.5  NumPy 1.14.3  Pandas 0.23.0  Scikit-learn 0.19.1  XGBoost 0.6  LightGBM 2.1.0 | | | | | |
| **三、实验内容** | | | | | |
| **实验一 课后作业（二.3）**   1. 实验要求   随机生成一个五列十行的Dataframe的数据类型，行列索引自定义，绘制出相应的柱状图、散点图，以及自学绘制一个课程未讲解的数据分析图形。   1. 实验方法和步骤 2. 代码   import pandas as pd  import numpy as np  #随机生成十行五列数据  arr=np.random.rand(10,5)  #调整数据类型和范围，便于操作  arr=arr\*10  arr=arr.astype(int)  #设置行列标签  row\_tag=["day{}".format(i) for i in range(1,11)]  col\_tag=["00:00","07:00","12:00","14:00","20:00"]  #生成Dataframe  df=pd.DataFrame(data=arr,index=row\_tag,columns=col\_tag)  #输出df数据  print(df)  #绘制柱状图  ax=df.plot(kind="bar")  ax.set\_xlabel("date")  ax.set\_ylabel("temperature")  #绘制散点图  ax=df.plot(x="00:00",y="12:00",kind="scatter")  ax.set\_xlabel("00:00")  ax.set\_ylabel("12:00")  #绘制折线图  ax=df.plot(title="10-day temperature change chart")  ax.set\_xlabel("date")  ax.set\_ylabel("temperature")  #绘制箱线图  ax=df.plot.box(title="Temperature box")  ax.set\_xlabel("time")  ax.set\_ylabel("temperature")  #绘制区域图（面积图）  ax=df.plot(kind="area")  ax.set\_xlabel("date")  ax.set\_ylabel("temperature area")   1. 关键步骤 2. 随机生成Dataframe   用np.random.rand( )随机生成所需形状的数组，再用该数组生成目标数据结构，并添加标签。    我模拟的是一个10天不同时刻的气温表。   1. 绘图   Pandas封装了matplotlib的部分绘图功能，调用即可。  DataFrame.plot( )函数原型：  DataFrame.plot(x=None, y=None, kind='line', ax=None, subplots=False,  sharex=None, sharey=False, layout=None, figsize=None,  use\_index=True, title=None, grid=None, legend=True,  style=None, logx=False, logy=False, loglog=False,  xticks=None, yticks=None, xlim=None, ylim=None, rot=None,  fontsize=None, colormap=None, position=0.5, table=False, yerr=None,  xerr=None, stacked=True/False, sort\_columns=False,  secondary\_y=False, mark\_right=True, \*\*kwds)  参数如下：   * x和y：表示标签或者位置，用来指定显示的索引，默认为None * kind：表示绘图的类型，默认为line，折线图   line：折线图  bar/barh：柱状图（条形图），纵向/横向  pie：饼状图  hist：直方图（数值频率分布）  box：箱型图  kde：密度图，主要对柱状图添加Kernel 概率密度线  area：区域图（面积图）  scatter：散点图  hexbin：蜂巢图   * ax：子图，可以理解成第二坐标轴，默认None * subplots：是否对列分别作子图，默认False * sharex：共享x轴刻度标签。若ax为None，则默认为True，若传入ax，则默认为False * sharey：共享y轴刻度、标签 * layout：子图的行列布局，(rows, columns) * figsize：图形尺寸大小，(width, height) * use\_index：用索引做x轴，默认True * title：图形的标题 * grid：图形是否有网格，默认None * legend：子图的图例 * style：对每列折线图设置线的类型，list or dict * logx：设置x轴刻度是否取对数，默认False * loglog：同时设置x，y轴刻度是否取对数，默认False * xticks：设置x轴刻度值，序列形式（比如列表） * xlim：设置坐标轴的范围。数值，列表或元组（区间范围） * rot：轴标签（轴刻度）的显示旋转度数，默认None * fontsize : int, default None#设置轴刻度的字体大小 * colormap：设置图的区域颜色 * colorbar：柱子颜色 * position：柱形图的对齐方式，取值范围[0,1]，默认0.5（中间对齐） * table：图下添加表，默认False。若为True，则使用DataFrame中的数据绘制表格 * yerr：误差线 * stacked：是否堆积，在折线图和柱状图中默认为False，在区域图中默认为True * sort\_columns：对列名称进行排序，默认为False * secondary\_y：设置第二个y轴（右辅助y轴），默认为False * mark\_right : 使用secondary\_y轴时，图例中自动用“(right)”标记列标签 ，默认True * x\_compat：适配x轴刻度显示，默认为False。设置True可优化时间刻度的显示  1. 实验结果及分析 2. 生成的数据      1. 绘制的图形   柱状图    散点图，表现的是10天的00：00和12：00的气温关系    折线图，表现的是10天不同时刻的气温变化    箱型图    区域图（自学绘制的图形）    **实验二 课后作业（五.2）**  一、实验要求  对测试集ccf\_offline\_stagel\_test\_revised做分析与数据观察。  二、实验方法和步骤   1. 代码   import pandas as pd  import matplotlib.pyplot as plt  from datetime import datetime  #测试数据分析  def test\_count(df):  test=df.copy()    sample\_size=test.shape[0] #样本数  user\_size=test["User\_id"].value\_counts().size #用户数  merchant\_size=test["Merchant\_id"].value\_counts().size #商户数    coupon\_size=test["Coupon\_id"].count() #优惠券发放数  coupon\_type\_size=test["Coupon\_id"].value\_counts().size#优惠券种类数  distance\_max=test["Distance"].max() #用户与商户最远距离  distance\_min=test["Distance"].min() #用户与商户最近距离    received\_size=test["Date\_received"].count() #优惠券领取数  date\_received\_max=test["Date\_received"].max() #领取优惠卷最晚日期  date\_received\_min=test["Date\_received"].min() #领取优惠卷最早日期      print("样本数: ",sample\_size,"\n")  print("用户数: ",user\_size,"\n")  print("商户数: ",merchant\_size,"\n")    print("优惠券发放数: ",coupon\_size)  print("优惠券种类数: ",coupon\_type\_size,"\n")    print("最远距离: ",distance\_max)  print("最近距离: ",distance\_min,"\n")    print("优惠券领取数: ",received\_size)  print("领取优惠卷最晚日期: ",date\_received\_max)  print("领取优惠卷最早日期: ",date\_received\_min)  #数据观察与绘图  def data\_plot(df):  ds=df.copy()      #用户领卷次数饼状图    #按User\_id分组统计领卷次数  user\_coupon=ds.groupby(by="User\_id")["Coupon\_id"].count()  #按领卷次数分组统计用户数  user\_coupon=user\_coupon.groupby(user\_coupon).count()  #处理数据  coupon1=user\_coupon.values  coupon1\_indexs=["once","twice","Three or more times"]  coupon1\_values=[coupon1[0],coupon1[1],sum(coupon1[2:])]  user\_coupon=pd.Series(data=coupon1\_values,index=coupon1\_indexs,name="user\_number")  #作饼状图  user\_coupon.plot(kind="pie",autopct="%.2f")      #商户发卷能力饼状图    #按Merchant\_id分组统计发卷次数  merchant\_coupon=ds.groupby(by="Merchant\_id")["Coupon\_id"].count()  #处理数据  merchant\_coupon=merchant\_coupon.sort\_values()  coupon2=merchant\_coupon.values  coupon2\_indexs=["Other merchants","Top 20 merchants"]  coupon2\_values=[sum(coupon2[0:len(coupon2)-20]),sum(coupon2[len(coupon2)-20:])]  #作饼状图  plt.figure()  plt.pie(coupon2\_values,labels=coupon2\_indexs,autopct="%.2f")  plt.title("coupon\_number")  plt.show()      #不同折扣消费卷数量直方图  #将Discount\_rate转换为统一的优惠率  rate\_coupon=ds["Discount\_rate"].apply(  lambda r:float(r) if ':' not in r else  (float(str(r).split(':')[0])-float(str(r).split(':')[1]))/float(str(r).split(':')[0])  )  #作直方图  plt.figure()  plt.hist(rate\_coupon, bins=10)  plt.xlabel("discount\_rate")  plt.ylabel("coupon\_number")  plt.show()      #满减型与非满减型优惠卷占比饼状图    #判断优惠卷是否为满减类型  off\_or\_discount=ds["Discount\_rate"].str.contains(':')  #处理数据  off\_or\_discount=off\_or\_discount.groupby(off\_or\_discount).count()  #作饼状图  plt.figure()  plt.pie(off\_or\_discount.values,labels=["discount","off"],autopct="%.2f")  plt.title("coupon\_number")  plt.show()      #用户与商户距离直方图  distance=ds["Distance"].values  plt.figure()  plt.hist(distance, bins=10)  plt.xlabel("distance")  plt.ylabel("user\_number")  plt.show()      #每日消费卷领取数量折线图  received\_daily=ds["Date\_received"]  received\_daily=received\_daily.groupby(received\_daily).count()  plt.figure()  plt.plot(received\_daily.values)  plt.xlabel("date")  plt.ylabel("received\_number")  plt.show()      #工作日与周末领卷均值柱状图  received\_mean=ds["Date\_received"]  #统计每日的领卷数  received\_mean=received\_mean.groupby(received\_mean).count()  #数据处理  date=received\_mean.index #日期列表  weekday\_size=0 #工作日天数  weekend\_size=0 #周末天数  received\_weekday=0 #工作日领卷数  received\_weekend=0 #周末领卷数  for i in date:  #日期转换为星期  week=datetime.strptime(str(i), '%Y%m%d').weekday()  if 0<=week<=4:  weekday\_size+=1  received\_weekday+=received\_mean[i]  else:  weekend\_size+=1  received\_weekend+=received\_mean[i]  weekday\_mean=received\_weekday/weekday\_size #工作日领卷均值  weekend\_mean=received\_weekend/weekend\_size #周末领卷均值  #作柱状图  plt.figure()  plt.bar(x=[0,1],height=[weekday\_mean,weekend\_mean],width=0.2,tick\_label=["weekday","weekend"])  plt.show()    if \_\_name\_\_ == '\_\_main\_\_':  off\_test\_path="data\\ccf\_offline\_stage1\_test\_revised.csv"  off\_test=pd.read\_csv(off\_test\_path)  test\_count(off\_test)  data\_plot(off\_test)   1. 关键步骤   a. 测试集信息  预测用户在2016年7月领取优惠券后15天以内的使用情况。    b. 测试数据分析  通过pandas提取以下10个数据信息：  样本数、用户数、商户数、优惠券发放数、优惠券种类数、用户与商户最远距离、  用户与商户最近距离、优惠券领取数、领取优惠卷最晚日期、领取优惠卷最早日期。   1. 数据观察与绘图   通过pandas和matplotlib.pyplot进行绘图。   1. 实验结果及分析 2. 测试集信息     该图说明测试数据中每个样本都领取了优惠卷，且领卷时间跨度为整个7月。   1. 用户领卷次数饼状图     该图说明90%的用户都只领取了1~2张优惠卷，10%的用户领取了较多的优惠卷，说明商户想用最少的优惠卷尽可能得到更多的用户，以获取最大的利润。   1. 商户发卷能力饼状图     该图说明不到2%的商户（商户数：1559家）发放了四分之三的优惠卷，而发放优惠卷是有一定的成本的，表明只有大型商户才有足够资本进行大量的优惠促销活动。   1. 不同折扣消费卷数量直方图     该图表明绝大多数优惠卷优惠率在八折附近，可能该优惠率能在满足用户需求的同时尽可能节约优惠促销成本，以获取最大利润。   1. 满减型与非满减型优惠卷占比饼状图     该图表明商户发放的优惠卷几乎为满减卷，有意诱导用户尽可能多消费。   1. 用户与商户距离直方图     由图可知，领取优惠卷的用户想尽可能的就近消费，所以领卷数随距离增大而减小，但当距离大于9时，领卷数又有增加，考虑优惠卷大多为大型商户所发，这些商户的服务和产品质量好，使部分用户愿意远道而来，进行消费。   1. 每日消费卷领取数量折线图     由图可知，每日消费卷的领取数量变化起伏很大，暂无规律可循。   1. 工作日与周末领卷均值柱状图     由图可知，工作日领取优惠卷的日平均值要略高于周末领取优惠卷的日平均值，说明优惠卷的领取可能与是否为工作日的关系不大。  **实验三 课后作业（五.3）**   1. 实验要求   对测试集ccf\_offline\_stagel\_test\_revised做数据预处理。   1. 实验方法和步骤 2. 代码   import pandas as pd  def preprocessing(df):  ds=df.copy()    print("数据缺失率： ",ds.isnull().sum()/ds.shape[0])  #处理User\_id、Merchant\_id和Coupon\_id,转换为int类型  ds["User\_id"]=ds["User\_id"].apply(int)  ds["Merchant\_id"]=ds["Merchant\_id"].apply(int)  ds["Coupon\_id"]=ds["Coupon\_id"].apply(int)  #处理Discount\_rate,消费卷类型判断(off满减)  ds["is\_off"]=ds["Discount\_rate"].apply(  lambda x:1 if ':' in str(x) else 0  )  #满减类型卷，给出消费最低价格，折扣卷赋值-1  ds["min\_spend"]=ds["Discount\_rate"].apply(  lambda x: -1 if ':' not in str(x) else str(x).split(':')[0]  )  #满减型消费卷，给出优惠额度  ds["cut\_money"]=ds["Discount\_rate"].apply(  lambda x: 0 if ':' not in str(x) else str(x).split(':')[1]  )  #将消费卷优惠率统一转换为折扣率  ds["Discount\_rate"]=ds["Discount\_rate"].apply(  lambda r:float(r) if ':' not in r else  (float(str(r).split(':')[0])-float(str(r).split(':')[1]))/float(str(r).split(':')[0])  )  #处理Distance,将空值替换为-1  ds["Distance"].fillna(value=-1,inplace=True)  #处理Date\_received,转换为标准时间戳  ds["Date\_received"]=pd.to\_datetime(ds["Date\_received"],format='%Y%m%d')    return ds;  if \_\_name\_\_ == '\_\_main\_\_':  off\_test\_path="data\\ccf\_offline\_stage1\_test\_revised.csv"  off\_test=pd.read\_csv(off\_test\_path)  print("处理之后的数据:\n",preprocessing(off\_test));     1. 关键步骤   本次实验主要进行以下数据预处理：  a. 处理User\_id、Merchant\_id和Coupon\_id,转换为int类型  b. 处理Discount\_rate,消费卷类型判断(off满减)  c. 增加min\_spend列,对满减类型卷给出消费最低价格，折扣卷赋值-1  d. 增加cut\_money列,满减型消费卷，给出优惠额度  e. 处理Discount\_rate,将消费卷优惠率统一转换为折扣率  f. 处理Distance,将空值替换为-1   1. 实验结果及分析 2. 数据缺失率     该结果说明平台无法保证用户个人信息的完全收集，用户与商户距离（Distance）这一无法直接用平台收集的信息出现缺失，需对该数据进行缺失值处理。   1. 处理之后的数据     **实验四 课后作业（八）**   1. 实验要求   提取下列用户特征，完成模型训练，并在阿里云天池平台提交结果：   1. 领卷数 2. 领卷并消费数 3. 领卷未消费数 4. 领卷并消费数/消费数 5. 领取并消费优惠卷的平均折扣率 6. 领取并消费优惠卷的平均距离 7. 在多少不同商家领取并消费优惠卷 8. 在多少不同商家领取优惠卷 9. 在多少不同商家领取并消费优惠卷/在多少不同商家领取优惠卷 10. 实验方法和步骤 11. 用户领卷数      1. 用户领卷并消费数      1. 用户领卷未消费数      1. 用户领卷并消费数/消费数      1. 用户领取并消费优惠卷的平均折扣率      1. 用户领取并消费优惠卷的平均距离      1. 用户在多少不同商家领取并消费优惠卷      1. 用户在多少不同商家领取优惠卷      1. 用户在多少不同商家领取并消费优惠卷/用户在多少不同商家领取优惠卷      1. 实验结果及分析   在阿里云天池平台提交初步建模生成的csv文件，由于建模很粗糙，拟合效果很差，得分仅为0.5405，后续将会不断优化模型，以取得更好的效果。  结果如图所示：    **实验五 K-means 算法的代码实现**   1. 实验要求   完成 K-means 算法的代码实现（同时提交源代码）及数据（至少测试 5  个数据集，数据集来源建议采用 UCI 数据集）测试。   1. 实验方法和步骤 2. K-means聚类算法   K-means是一种常用的聚类算法，其目标是最小化所有数据点到其所属簇中心的距离的平方和，通常称为簇内平方和（Inertia）。这意味着算法试图将数据点划分为紧密相邻的簇，使得每个簇的数据点与其簇中心的距离最小。其基本原理描述如下：  a. 选择聚类数目 k：首先，确定要将数据集分成多少个簇。这是K-means的一个超参数，通常需要通过领域知识、经验或其他方法来确定。  b. 初始化聚类中心：随机选择 k 个数据点作为初始的聚类中心。这些聚类中心将成为每个簇的代表。  c. 分配数据点到最近的聚类中心：对于每个数据点，计算其与所有聚类中心的距离，并将其分配到距离最近的聚类中心所属的簇。  d. 更新聚类中心：对每个簇，计算其所有数据点的均值，将该均值作为新的聚类中心。这一步旨在更新聚类中心，以更好地代表每个簇的数据点。  e. 重复步骤 c 和 d：重复执行步骤 c 和 d，直到聚类中心不再明显变化或达到预定的迭代次数。这样就完成了聚类的过程。  K-means算法具有以下优点：  a. 简单且高效：K-means是一种简单而高效的聚类算法，易于实现和理解。由于其线性时间复杂度，对于大规模数据集也能够有效地工作。  b. 可扩展性：K-means适用于大规模数据集，其计算复杂性随着数据规模的增加而线性增长。  c. 对均衡型簇效果好：在簇的形状相对简单、密度相对均匀的情况下，K-means表现较好。  d. 对数值型数据表现好：K-means对数值型数据的适应性较好，特别是在各个维度的尺度相差不大时。  K-means算法具有以下缺点：  a. 对初始聚类中心敏感：K-means对初始聚类中心的选择非常敏感。不同的初始中心可能导致不同的最终聚类结果，因此可能陷入局部最优解。  b. 需要事先确定聚类数目 k：算法的性能依赖于预先指定的簇数目 k，而实际中通常难以事先确定。选择不合适的 k 值可能导致不准确的聚类结果。  c. 对非球形状簇效果差：由于K-means使用欧氏距离来度量簇的紧密度，因此对非球形状簇效果较差，往往会产生偏向于球形簇的结果。  d. 对噪声和离群值敏感：K-means对噪声和离群值较为敏感，可能会影响聚类结果的稳定性。   1. 数据集介绍   本次实验采用的是sklearn数据集，sklearn提供了一些经典的示例数据集，方便用户在机器学习的实验和教学中使用。   1. iris数据集   Iris数据集是一个经典的用于机器学习和数据挖掘的数据集。这个数据集包含了三种不同种类的鸢尾花（iris）的样本，每个类别有50个样本，共计150个样本。每个样本包含四个特征，分别是花萼（sepal）的长度和宽度，以及花瓣（petal）的长度和宽度，这些特征都以厘米（cm）为单位。  Iris数据集的三个类别分别是：Setosa（山鸢尾）、Versicolor（杂色鸢尾）和Virginica（维吉尼亚鸢尾）  这个数据集是由统计学家和生物学家Ronald A. Fisher在1936年收集的，成为了模式识别中最广泛使用的数据集之一。Iris数据集的目的是通过测量不同鸢尾花的特征，将它们分为不同的品种。   1. boston数据集   Boston Housing数据集是一个用于回归问题的经典数据集，用于预测波士顿地区房屋价格的中位数。该数据集包含506个样本，每个样本有13个特征，包括犯罪率、房屋所在位置的各项指标、房间数量等。目标变量是房屋价格的中位数。  这个数据集的特征如下：   |  |  | | --- | --- | | CRIM | 该城镇的犯罪率 | | ZN | 住宅用地所占比例 | | INDUS | 非零售业务用地所占比例 | | CHAS | 查尔斯河（Charles River）的虚拟变量（如果某段房产靠近河岸，则为1；否则为0） | | NOX | 一氧化氮浓度 | | RM | 每个住宅的平均房间数 | | AGE | 1940年以前建成的自住房屋的比例 | | DIS | 距离波士顿五个就业中心的加权距离 | | RAD | 距离高速公路的便捷度指数 | | TAX | 房产税率 | | PTRATIO | 学生与教师的比例 | | B | 1000(Bk - 0.63)^2，其中Bk是城镇中黑人的比例 | | LSTAT | 低收入人群所占比例 |   目标变量：MEDV（房屋价格的中位数）。   1. diabetes数据集   Diabetes（糖尿病）数据集是一个用于回归问题的经典数据集，用于预测糖尿病患者一年后的疾病进展指标。该数据集包含442个患者的10个生理特征，以及对应的一年后的疾病进展指标。Diabetes数据集通常用于研究和测试回归模型，旨在预测患者一年后的疾病进展情况。  这个数据集的特征如下：   |  |  | | --- | --- | | Age | 年龄 | | Sex | 性别 | | BMI | 体质指数 | | BP | 平均血压 | | S1 | 总胆固醇 | | S2 | 低密度脂蛋白胆固醇 | | S3 | 高密度脂蛋白胆固醇 | | S4 | 总胆固醇/高密度脂蛋白胆固醇比例 | | S5 | 对数尿素 | | S6 | 一年后的疾病进展指标 |   目标变量：Target（一年后的疾病进展指标（定量测量））。   1. wine数据集   Wine（葡萄酒）数据集是一个经典的用于分类问题的数据集，用于对葡萄酒进行分类。该数据集包含了三个不同类别的葡萄酒，每个类别有178个样本。每个样本有13个特征，包括酸度、酚含量、颜色强度等。Wine数据集通常用于演示和测试分类算法，旨在根据葡萄酒的化学特征将其分为不同的类别。  这个数据集的三个类别分别是：  Class 0：59个样本  Class 1：71个样本  Class 2： 48个样本  每个样本的特征如下：  Alcohol（酒精含量）、Malic Acid（苹果酸含量）、Ash（灰分含量）、Alcalinity of ash（灰碱度）、Magnesium（镁含量）、Total phenols（总酚含量）、Flavanoids（类黄酮含量）、Nonflavanoid phenols（非黄酮类酚含量）、Proanthocyanins（原花青素含量）、Color intensity（颜色强度）、Hue（色调）、OD280/OD315 of diluted wines（稀释葡萄酒的吸光度比值）、Proline（脯氨酸含量）   1. breast\_cancer数据集   Breast Cancer（乳腺癌）数据集是一个用于二分类问题的数据集，用于对乳腺肿瘤是良性还是恶性进行分类。该数据集包含了569个乳腺肿瘤样本，每个样本有30个特征，包括肿瘤的半径、纹理、对称性等。  这个数据集的两个类别分别是：  Malignant（恶性）：212个样本  Benign（良性）：357个样本  每个样本的30个特征如下：  mean radius（平均半径）、mean texture（平均纹理）、mean perimeter（平均周长）、mean area（平均面积）、mean smoothness（平均光滑度）、mean compactness（平均紧凑度）、mean concavity（平均凹度）、mean concave points（平均凹点）、mean symmetry（平均对称性）、mean fractal dimension（平均分形维度）、radius error（半径误差）、texture error（纹理误差）、perimeter error（周长误差）、area error（面积误差）、smoothness error（光滑度误差）、compactness error（紧凑度误差）、concavity error（凹度误差）、concave points error（凹点误差）、symmetry error（对称性误差）、fractal dimension error（分形维度误差）、worst radius（最差半径）、worst texture（最差纹理）、worst perimeter（最差周长）、worst area（最差面积）、worst smoothness（最差光滑度）、worst compactness（最差紧凑度）、worst concavity（最差凹度）、worst concave points（最差凹点）、worst symmetry（最差对称性）、worst fractal dimension（最差分形维度）   1. 评估指标   常用的测定聚类质量的方法按是否有基准可用分为两类不同的方法，分别是外在方法（有可用的基准）和内在方法（无可用基准），两种方法的具体介绍如下：  外在方法用于测定聚类算法的质量，通常通过将聚类结果与已知的真实标签（ground truth）进行比较。要求在实验开始时，数据集中包含有真实标签（ground truth）。这样的方法适用于评估聚类结果在已知类别的情况下的性能，但在实际应用中，真实标签并不总是可用。  需要注意的是，外在方法对于不同标签分配的问题可能不敏感，即使标签名称不同，只要对应的样本属于相同的真实类别，这些指标仍然可能给出较高的评分。因此，在使用外在方法时，需要综合考虑其他内在方法以及任务的具体特点。  以下是一些常用的外在方法：  a. 互信息（Mutual Information）：互信息衡量了两个分布之间的相似性，用于比较聚类结果与真实标签之间的相似性。数值范围为 [0, 1]，越接近1表示相似性越高。  b. 调整兰德指数（Adjusted Rand Index，ARI）：ARI通过比较聚类结果中的相对顺序和真实标签中的相对顺序来测量它们之间的一致性。取值范围为 [-1, 1]，越接近1表示聚类结果越好。  c. Fowlkes-Mallows指数：该指数是精确度和召回率的几何平均值，用于评估聚类结果与真实标签的相似性。  d. Jaccard相似系数：Jaccard相似系数用于比较两个集合的相似性，可以用于比较聚类结果的簇与真实标签之间的相似性。  e. V-Measure：V-Measure是均一性和完整性的调和平均值，同时考虑了簇内的均一性和簇间的完整性。数值范围为 [0, 1]，越接近1表示聚类效果越好。  内在方法用于测定聚类算法的质量，主要基于数据本身的性质，而无需外部信息（例如真实标签）。这些方法旨在度量聚类结果的紧密度和分离度等特性，从而评估聚类的质量。但需要注意的是，不同的内在方法可能对不同类型的数据和聚类结构更具有敏感性，因此在选择时需要综合考虑实际任务的特点。  以下是一些常用的内在方法：  1. 轮廓系数（Silhouette Coefficient）：轮廓系数综合考虑了簇内的紧密度和簇间的分离度。对于每个样本，计算其与同簇其他样本的平均距离（a）和与最近相邻簇的所有样本的平均距离（b），然后计算轮廓系数为 (b - a) / max(a, b)。系数取值范围为 [-1, 1]，越接近1表示聚类效果越好。  2. Calinski-Harabasz指数：通过计算簇内的离散度和簇间的相似度的比率，来评估聚类的紧密度。指数值越高，表示簇内紧密度高、簇间距离大，聚类效果越好。  3. Davies-Bouldin指数：通过计算簇内平均距离与簇间距离的比率，来评估簇的紧密度和分离度。指数值越低，表示聚类效果越好。  4. Gap统计量：通过比较原始数据集与随机生成的数据集的聚类结果，来评估聚类效果。Gap统计量计算原始数据集的轮廓系数与随机数据集的轮廓系数之差。  5. Hopkins统计量：用于衡量数据集的簇趋势性，可以用于判断数据集的聚类性质。Hopkins统计量的取值范围为 [0, 1]，越接近1表示数据集更具有聚类性。  6. Gap统计量：通过比较原始数据集的聚类效果与随机生成数据集的聚类效果，来评估聚类算法的性能。  本次实验采用的是轮廓系数作为测定聚类质量的指标。   1. 实验结果及分析 2. iris数据集聚类结果     轮廓系数：     1. boston数据集聚类结果     轮廓系数：     1. boston数据集聚类结果     轮廓系数：     1. boston数据集聚类结果     轮廓系数：     1. boston数据集聚类结果     轮廓系数：    整体上看轮廓系数都比较低，可见K-means算法对异常值和噪声数据都比较敏感，同时对分类的图形观察可以发现，k值的选取对算法结果有较大影响。 | | | | | |
| **四、学习心得** | | | | | |
| 在学习数据工程综合实践的过程中，我深刻认识到这是一个涵盖广泛技能的全面项目。首先，我成功完成了Python机器学习库和未集成算法的安装，这为后续的实践提供了强大的基础。对机器学习基本概念的了解让我能够更好地理解项目的背景和目标。  通过学习基础和中级Python编程，我提高了自己的编程技能，包括对Python语法、面向对象编程和高级编程概念的掌握。这些技能对于处理大规模数据、实现复杂算法和进行高效编程非常关键。  学习使用第三方Python库，如NumPy、Pandas、Matplotlib等，为我提供了强大的工具，使我能够更有效地进行数据处理和可视化。这对于实际项目中的数据分析、特征工程和模型评估非常有帮助。  了解O2O优惠券使用预测算法的基本框架为我提供了项目实施的整体架构。通过完成O2O优惠券使用预测项目解读，我不仅理论知识得到了锻炼，而且在实际问题解决方面也积累了丰富经验。这包括了数据清理、特征工程的实践，以及模型选择和评估的过程。  学习数据分析与预处理方法，我掌握了处理数据质量的关键技能，包括处理缺失值、异常值和选择适当的数据预处理方法。此外，学习数据划分与打标方法为我提供了在监督学习问题中正确划分数据集和打标签的指导原则。  完成O2O优惠券使用预测Baseline是项目的一个重要阶段，通过实现一个简单的预测模型，我建立了一个基准，为未来的模型改进和优化提供了起点。  总体而言，这个数据工程综合实践项目不仅深化了我的理论知识，还提高了我的实际问题解决能力。这是一个全方位的学习过程，将对我未来在数据科学和机器学习领域的工作产生深远的影响。 | | | | | |
| **五、教师评阅意见** | | | | | |
|  | | | | | |