实验或实习报告

|  |
| --- |
| **实验一**   1. 实验题目及内容   课后作业（一）1：安装python机器学习库到自己的机器，并输入自己的第一行python代码，输出：Hello python！  二、实验过程步骤  1、以前安装过python（此处使用Pycharm编译）    2.成功安装anaconda（之前用的pip）    3.设定run/debug configurations     1. 实验结果及分析   编译“hello，python”    **实验二**  一、实验题目及内容  课后作业（一）2：安装常用机器学习算法Xgboost和LightGbm，并在python中输入导入算法命令验证安装成功，同时在网上查阅相关博客，学会简单的导入与调用。   1. 实验过程步骤   *因为是用的macOS系统，实验指导书提供的方法不能使用，自己在网上找的攻略*   1. 安装Xgboost   第一步：在终端中输入  pip3 install xgboost  显示：    成功安装  但是在pycharm中import xgboost显示没有相关模块，查询资料后显示应该是interpreter选择不对。  第三步：进入pycharm-> preference-> project：projeceName-> project interpreter    找到正确的interpreter路径（选择anaconda下的路径，之前默认的系统自带python2的路径），apply之后编译成功   1. 安装LightGbm   想继续用pip来安装lightgbm，结果    百度了一下之后按着新的方法安装：  第一步：安装cmake和gcc（需要首先安装homebrew）  brew install cmake  brew install gcc –without-multilib  第二步：  git clone –recursive http://github.com/Microsoft/LightGBM;  cd LightGBM  export CXX=g++-7 CC=gcc-7  mkdir build;  cd build  cmake ..  make -j4  第三步：回到根目录  pip3 install lightgbm     1. 实验结果及分析     会产生warning但不影响编译  **实验三**  一、实验题目及内容  KNN算法的代码实现（提交源代码）、数据（至少测试5个UCI数据集）测试结果展示  二、实验过程步骤   1. 理解knn算法： 2. 分类算法   2）如何分类（测试集每个例子的label是什么）？计算离例子最近的k个训练集里的点的距离（欧式距离），并记录他们的label。在这k个点中选取出现次数最多的label，作为此例子的label。   1. 关键代码：     要点：   1. 训练集和测试集数据作为参数导入eludSim（）里时，是str类型，需要转换为float才能继续计算。     要点：   1. 新建二维list需要写成   list = [[] for i in range(n)]  不然会报错：list index out of range   1. np.argsort(data)可以将data里的数据从小到大排序，并按从小到大的顺序返回data里的index 2. max（data，key=data.count）可以返回data里出现次数最多的值   三、实验结果及分析（截取一部分数据，红色标记为不正确的分类）   1. 第一组数据：     正确率：100%   1. 第二组数据：     正确率：90%   1. 第三组数据     此组数据随k值变化，其结果也有相对的不同，当k=3时，正确率达到100%。   1. 第四组数据     正确率：99.2%   1. 第五组数据     正确率：80%   1. 结果分析： 2. 训练样本分类不平衡时会对结果产生影响（如第四组数据） 3. K值的选取与正确率相关，在百度上看到的关于k值选取的建议： k值通常选取奇数；现在可用的判定k值取值的方法有Cross Validation，贝叶斯或bootstrap（待研究）。   **实验四**  一、实验题目及内容  课后作业（二）1：用Python实现F1-score，并自己构建一个简易数据集进行测试  二、实验过程步骤及实验结果及分析  代码如下：    x簇代表真实情况，y簇代表预测情况  结果如下：    **实验五**  一、实验题目及内容  课后作业（二）2：用python中的for循环和列表推导式，分别实现计算：1+2+3+…+1000  二、实验过程步骤（注意是主要关键步骤，适当文字+截图说明）、实验结果及分析  代码如下：    结果如下：    **实验六**  一、实验题目及内容  课后作业（二）3：随机生成一个五列十行的dataframe的数据类型，行列索引自定义，绘制出对应的柱状图、散点图，以及在查询官网学习绘制一个课程未讲解（即除柱状图、饼图、散点图、箱线图以外的图形）的数据分析的图形  二、实验过程步骤（注意是主要关键步骤，适当文字+截图说明）、实验结果及分析  1.代码如下：    2.新学习的折线图：  plot(x,y,style,color,linewidth)       title(“图的标题”)  参数说明：style,画线的样式，color,画线的颜色，linewidth，线的宽度   1. 结果如下：     **实验七**   1. 实验题目及内容   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](http://archive.ics.uci.edu/ml/index.php)）测试结果   1. 实验过程步骤 2. 算法理解： 3. 聚类算法 4. 如何聚类？定下k值（数据集被分为k个clusters），创建k个初始中心点；计算每个样本分别到k个中心点的距离，并将此样本分配给距离最近的cluster；对于每个cluster计算坐标均值作为新的中心点；重复以上步骤知道新的中心点没有变化。 5. 伪代码设计：      1. 关键代码实现： 2. 关于初始中心点的选择：     *初始设计：随机分配*  因为发现每次结果差异较大，还会出现一个cluster的点非常少的情况，遂在网上找到选择中心点的方法改进如下：    *最终设计：选取数据集中相对较远的k个点作为初始中心点*  两种方式产生结果对比：    其实感觉上差距不太大，但是后一种方式产生的分类更稳定，多次运行的结果不易变化。   1. KMeans算法：     要点：设计一个boolean变量change来控制循环的开始与结束   1. 可视化     遇到的错误：忘记dataset里面存储的值是str格式，画出来的图坐标抽重叠；需要float（）一下   1. 降维   最初在实现整个算法时，为了方便直接用的iris.data里的前两位数据，画出来的图就很奇怪，且没有理由只选择前两位进行运算；在网上搜索后发现可以用降维的方法，将4维的原数据集降到2维；现在常用的两种降维方法有：PCA、SVD；鉴于sklearn库中可直接使用pca（），便采取了PCA进行降维（时间紧急，对降维了解不多）。  代码实现：    要点：   1. From sklearn import decomposition 2. Fit\_transform（）中的参数只能是matrix或者array；我之前用的list存放数据集，所以需要前面的步骤来转换。（直接用np.array()/ np.mat()数据会报错，显示list没有对齐）   降维后数据：    降维前后结果对比：     1. 实验结果及分析 2. 第一组数据      1. 第二组数据      1. 第三组数据      1. 第四组数据      1. 第五组数据      1. 结果分析： 2. 多次运行之后分类还是会有很大变化 3. 我是在已知k的值（在UCI上下载的数据集自带label）的情况下进行的聚类，那如果不知道k的值的情况下如何确定k的值（待解决）。 |