北京邮电大学实践课程实验报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **课程名称** | **创新创业实践** | **学院** | **计算机学院** | **指导老师：吴起凡** |
| **知识模块** | **机器学习** | **完成时间** | | **2023年11月10日** |
| **班级** | **序号** | **学号** | **学生姓名** | **成绩** |
| **2023211307** | **31** | **2023211088** | **罗浩** | **100** |

**实验一 Python机器学习入门：有监督学习**

实验目的：

1.了解机器学习的基本概念，了解机器学习的应用方法。

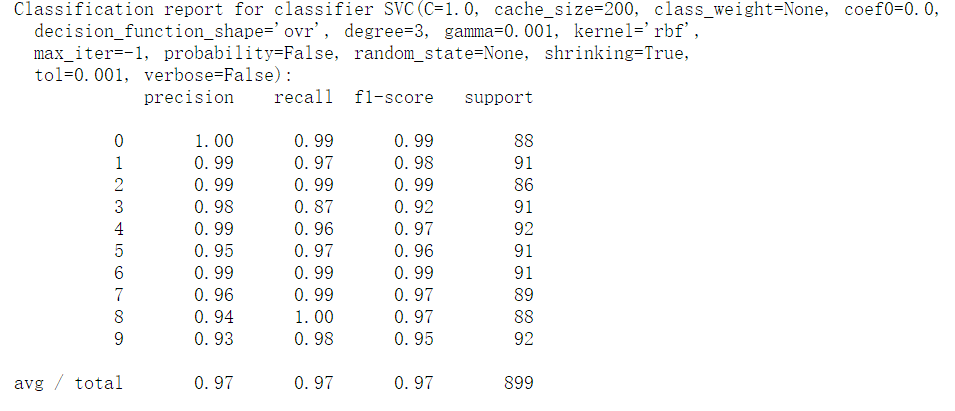
2.通过实验掌握机器学习预测任务的基本流程。

实验内容与实验步骤：

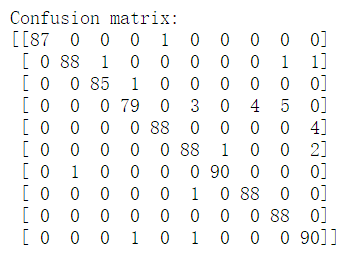
1. 进入Ubuntu系统后，同时按下Ctrl+Alt+T打开终端Terminal窗口。
2. 在终端窗口中输入jupyter notebook命令打开Jupyter Notebook编译环境。
3. 键入Enter回车键后等待，浏览器会自动打开编译环境。
4. 点击页面右上方区域按钮New->Python3。
5. 将exper1.txt文件中代码复制入In[]:后光标中，键入Shift+Enter运行。
6. 代码文件exper1.txt中实现了以下步骤：
7. 导入实验依赖模块.
8. 载入示例数据集。载入Scikit-learn自带数据集手写数字识别集（Handwritten Digits Data Set）。
9. 查看数据集。使用matplotlib显示数据集图片。
10. 数据预处理。使用numpy将图片展开成向量。
11. 构建分类器模型。使用Scikit-learn中的分类器SVM。
12. 训练分类器模型。使用一半数据集进行模型的训练。
13. 使用训练好的分类器模型预测另一半数据集。
14. 检查分类器的预测效果。使用Scikit-learn自带metrics检查预测准确率、召回率及混淆矩阵（Confusion Matrix）等。

实验结果与实验总结：

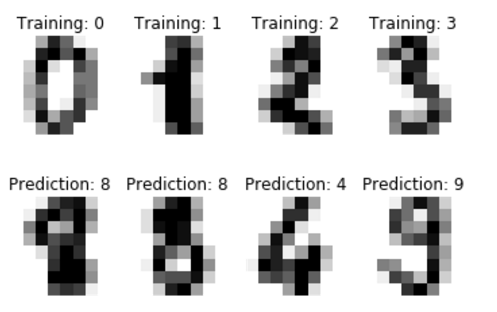
1. 分类器说明和每个分类的准确率precision，召回率recall，F1分数f1-score和各类别参与训练的样本数。

[](https://camo.githubusercontent.com/57f1bd035d51b0484768d1e19b2b92832d3108bc/68747470733a2f2f692e696d6775722e636f6d2f5064556859374d2e706e67)

1. 混淆矩阵：可看到测试数据集被分类的情况。

[](https://camo.githubusercontent.com/a85af5583639e1441721fccc884ebebd3a80b9ba/68747470733a2f2f692e696d6775722e636f6d2f6d435675626c362e706e67)

1. 训练和测试情况：

[](https://camo.githubusercontent.com/47053c09a6e5d1b79dc0131477e09bbe608550bf/68747470733a2f2f692e696d6775722e636f6d2f324735376c30502e706e67)

**实验二 Python机器学习入门：无监督学习**

实验目的：

1.了解机器学习有监督和无监督的区别。

2.通过实验掌握简单无监督算法使用方式。

实验内容与实验步骤：

1.打开编译环境。如实验一打开Jupyter Notebook，新建New->Python3交互窗口。

2.代码文件exper2.txt中实现了以下步骤：

i.导入实验依赖模块。

ii.载入示例数据集。载入Scikit-learn自带数据集手写数字识别集（Handwritten Digits Data Set）。

iii.数据预处理。标准化数据，并获取数据集相关信息。

iv.定义bench\_k\_means函数用以完成模型的训练及打印模型的聚类评估指标。

v.分别构建三种不同的K-means聚类器。

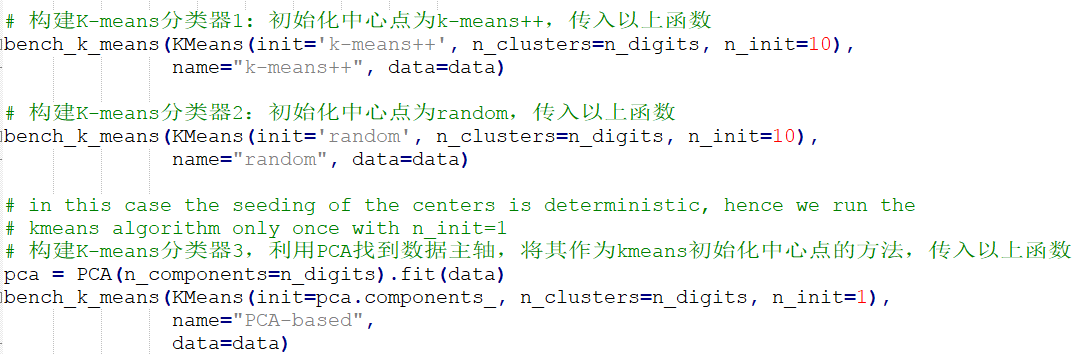
vi.聚类可视化。使用matplotlib可视化聚类结果（PCA降维到2维以便平面显示）。

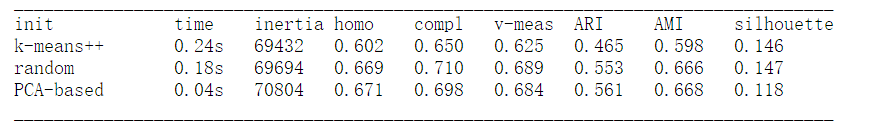
实验结果与实验总结：

1.数据集包含10个分类（手写数字1-10），1797个样本，特征维度为64维。

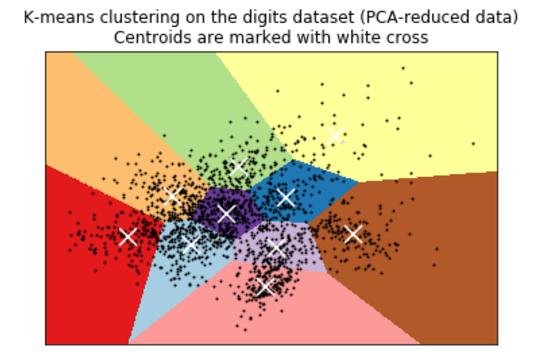
[https://camo.githubusercontent.com/3e7a174eb17aac7ae68bf5bf9c0616e9b85a9c16/68747470733a2f2f692e696d6775722e636f6d2f707249514a59532e706e67](https://camo.githubusercontent.com/3e7a174eb17aac7ae68bf5bf9c0616e9b85a9c16/68747470733a2f2f692e696d6775722e636f6d2f707249514a59532e706e67)

2.可以看到3个不同kmeans初始化中心点方法的聚类器的效果，注意使用PCA-based方法初始化中心点速度极快，因为中心点更新次数少。

[](https://camo.githubusercontent.com/7d034448022471d2f5a18cbac68e3c19aa9c5728/68747470733a2f2f692e696d6775722e636f6d2f665957384636682e706e67)

[](https://camo.githubusercontent.com/1ce73a58439757e1b8537736670ab5c7b60dd428/68747470733a2f2f692e696d6775722e636f6d2f59706e3543754b2e706e67)

3.最后在图中可以看到PCA降维到2维的数据聚类情况。

[](https://camo.githubusercontent.com/86aa939f9d4ffa56e91f49d293571b0994ded1d1/68747470733a2f2f692e696d6775722e636f6d2f3668504970704a2e706e67)

**实验三 Python深度学习入门：单层神经网络**

实验目的：

1.了解深度学习的基本概念。

2.通过实验学会使用框架实现简单神经网络。

实验内容与实验步骤：

1.打开编译环境。如实验一打开Jupyter Notebook，新建New->Python3交互窗口。

2.代码文件exper3.txt中实现了以下步骤：

i.导入实验所需模块。

ii.载入示例数据集。载入Tensorflow自带数据集手写数字识别集（MNIST Data）。

iii.构建神经网络，利用Tensorflow构建简单神经网络。

iv.构建损失函数和优化器。

v.构建TensorFlow会话并初始化变量。

vi.进行模型的训练，在给定的训练样本上运行以上神经网络，每隔100轮打印一次训练情况，观察交叉熵（Cross Entropy）误差error值的变化。

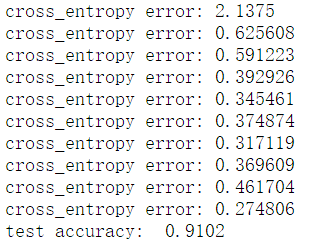
vii.检查模型预测效果。定义准确率accuracy计算方式，检查模型预测准确率。

viii.选择图片进行测试，选择一张图片，用训练好的模型对新的图片进行预测。

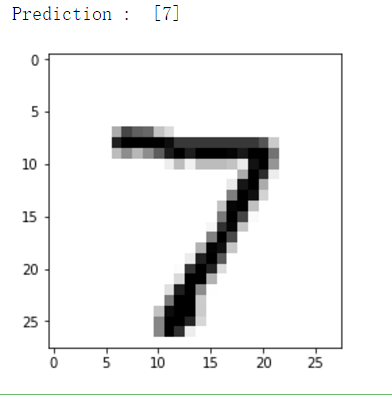
ix.打印模型预测结果，并可视化测试图片。

实验结果与实验总结：

1.随着迭代的进行，神经网络在数据集上的交叉熵（Cross Entropy）误差error值越来越小，代表正在慢慢拟合训练数据，最后在测试集上的测试准确率accuracy为91.02%。

[](https://camo.githubusercontent.com/e4d348c8a8d3156be6b28e928429c311bf2b5bde/68747470733a2f2f692e696d6775722e636f6d2f496776377354342e706e67)

2.选取测试集图片，进行预测：

[](https://camo.githubusercontent.com/0d986a673a8c73008c0e074b30d7be1e766da2ac/68747470733a2f2f692e696d6775722e636f6d2f626e686f3347662e706e67)