Lab 6 实验题目

学 号	17363011	学 院	专 业
姓 名	陈政培	智能工程学院	智能科学与技术

1、实验目的： 利用 python 实现 KNN 分类器

2、实验环境：vs code、python3.6.8

3、实验步骤：

① Knn1 就是最简单的根据距离排序得到的聚类结果

- 我们在 knn.classify 测试分类器时，调整了用于分类的输入向量，当输入为【0，1】时输出结果为 B，输入为【0，1.1】时输出结果为 A。证明分类器正常工作

② Knn2 改进约会网站的配对效果

- 根据 knn2 教程编写了对 datingTestSet2.txt 的分类，分为'not at all' ,'in small doses' ,'in large doses'三个种类并通过 knn2.classifyPerson 测试结果
- 为了方便我们调试不同 k 值下对错误率的影响我们对代码进行了一点更改，直接在 knn2.py 文件中为 datingClassTest 和 classifyPerson 函数加入了一个参数 k，并且给定了原文件中的预设 k 值
- 只需要在主函数文件调用对应函数时传递不同的 k 值即可观看效果
- 让 knn 分类器使用曼哈顿距离进行计算，需要在 knn2.py 中的 classify0 函数中修改计算距离的代码

```
#计算距离
dataSetSize = dataSet.shape[0]
diffMat = tile(inX,(dataSetSize,1)) - dataSet
sqDiffMat = diffMat ** 1
sqDistances = sqDiffMat.sum(axis=1)
distances = sqDistances ** 1
sortedDistIndicies = distances.argsort()
```

将平方项和开根项改为 1 即可

- 为了探索随机选取训练样本，对错误率的影响需要在 knn2.py 中的 datingClassTest 函数中修改测试样本的代码

```
    #测试样本的数量
    numTestVecs = int(m*hoRatio)
    errorCount = 0.0
    for count in range(numTestVecs):
        i = int(np.random.uniform(0,1000))
        classifierResult = classify0(normMat[i,:],normMat[numTestVecs:m,:],
                                     datingLabels[numTestVecs:m],k)
        #print("the classifier came back with: %d, the real answer is :%d"
        if (classifierResult != datingLabels[i]): errorCount +=1.0
```

把原来按顺序循环的变量 i 变成有 numpy 提供的 random 函数从 1000 个样本中获取到的随机数

- 测试不同样本数目对错误率的影响就比较简单只需要修改 knn2.py 中的 datingClassTest 函数中测试样本比例参数 hoRatio 的大小

③ 使用 knn 识别手写体

- 实例中只提供了 knn3.py 函数，并没有给出主函数操作，所以需要在主函数中自主调用
- 因为 knn3.py 中 createDataSet 函数和 trainingDataSet 函数并没有区别，且 handwritingTest 函数中包含了调用 createDataSet 训练分类器的语句，所以主函数仅仅需要调用

```
knn3.handwritingTest()
```

④ sklearn 实现 knn2 和 knn3

- knn2 直接重写了一套程序，主函数和函数都在 sklearnknn2.py 文件中
- knn3 则是仿照 knn3.py，把在 classify 过程中直接调用 KNN 分类器

⑤

4、实验结果与分析：

① Knn1 实验结果

```
[[1.  1.1]
 [1.  1. ]
 [0.  0. ]
 [0.  0.1]]
['A', 'A', 'B', 'B']
A
PS C:\Users\93744\Deskt
/data_and_code/knnapp.p
[[1.  1.1]
 [1.  1. ]
 [0.  0. ]
 [0.  0.1]]
['A', 'A', 'B', 'B']
B
```

② Knn2 实验结果

- 分类测试

```
percentage of time spent playing video games?0.5
frequent flier miles earned per year?40000
liters of ice cream consumed per year?7
You will probably like this person: in large doses
percentage of time spent playing video games?0.9
frequent flier miles earned per year?20000
liters of ice cream consumed per year?9
You will probably like this person: in small doses
```

结果符合理论，游戏时间少里程多的人更加有吸引力

③ Knn3 手写体测试结果

```
the total number of tests is: 946
the total number of errors is: 17
the total error rate is: 0.017970
Cost time: 0.00min, 27.8331s.
```

错误率仅有 1.797%

5、作业：

（1） k 值对错误率的影响

```
k=5
the total error rate is: 0.050000
k=4
the total error rate is: 0.040000
k=3
the total error rate is: 0.050000
```

K=4 时，分类器错误率达到最低

（2） 使用曼哈顿距离对错误率的影响

```
k=6
the total error rate is: 0.610000
k=5
the total error rate is: 0.610000
k=4
the total error rate is: 0.610000
k=3
the total error rate is: 0.610000
k=2
the total error rate is: 0.610000
```

K 值将不会再产生影响，使用曼哈顿距离错误率比欧氏距离要高

（3） 随机选取训练样本，测试不同样本数目对错误率的影响

```
k=4
the total error rate is: 0.020000
PS C:\Users\93744\Desktop\data_and_
de/knnapp.py
k=4
the total error rate is: 0.060000
```

同样的代码随机选取样本，两次运行结果错误率不同

```
k=4
hoRatio= 0.1
the total error rate is: 0.040000
PS C:\Users\93744\Desktop\data_and_c
de/knnapp.py
k=4
hoRatio= 0.2
the total error rate is: 0.080000
```

取消随机样本后，调整测试样本比例参数，测试样本比例越大，错误率越高

（4） 将 knn1，knn2，knn3 中的语句

```
from numpy import *
```

修改成

```
import numpy as np
```

并让代码正常运行

只需要在所有 numpy 提供的函数前加上 np. 引用即可

主要是 zeros，tile，array 等函数

（5） 使用 sklearn 实现 knn2 和 knn3 效果上由于机制原理相同没有什么差异，但是使用 sklearn 不再需要自己编写算法，可以更加简单方便的解决问题

（6） 程序代码完成了，但是 tensorflow 出现了问题无法正常运行

6、实验总结：

- 本次实验初期使用的是最新版本的 python3.8.0，在安装 matplotlib、sklearn 等库时出现安装失败的现象。回退老版本 python3.6 或 3.7 系列即可解决问题
- 上几次实验用到 matplotlib 绘图的部分无法正常显示汉字问题得到解决，只需要在程序开头加上

```
plt.rcParams['font.sans-serif']=['SimHei']
```

- 在用 tensorflow 实现 knn 的时候遇到了问题

```
Traceback (most recent call last):
  File "c:/Users/93744/Desktop/data_and_code/tf.py", line 45, in <module>
    traindata_tensor=tf.placeholder('float',[None,3])
AttributeError: module 'tensorflow' has no attribute 'placeholder'
```

无法在 tensorflow 中找到 placeholder，这个问题没有得到解决

参考文献:

1. https://blog.csdn.net/asialee_bird/article/details/81051281
2. https://www.cnblogs.com/CXZzero/p/10747636.html
3. https://www.jianshu.com/p/ab296440b0de