

lab 1

1.1

题目

编写程序，利用classify0函数（groups, labels, k=3）；测试[0,0]、[0.8,0.7]等点的类别

代码

MY_KNN.py文件代码

```
from cProfile import label
from tokenize import group
from numpy import *
import operator

def creatDataSet():
    group = array([[1.0, 1.1], [1.0, 1.0], [0, 0], [0, 0.1]])
    labels = ['A', 'A', 'B', 'B']
    return group, labels

def classify0(inX, dataSet, labels, k):
    dataSetSize = dataSet.shape[0]
    diffMat = tile(inX, (dataSetSize, 1)) - dataSet
    sqDiffMat = diffMat ** 2
    sqDistances = sqDiffMat.sum(axis=1)
    distances = sqDistances ** 0.5
    sortedDistIndicies = distances.argsort()
    classCount = {}
    for i in range(k):
        voteLabel = labels[sortedDistIndicies[i]]
        classCount[voteLabel] = classCount.get(voteLabel, 0) + 1
    sortedClassCount = sorted(classCount.items(),
                              key=operator.itemgetter(1), reverse=True)
    return sortedClassCount[0][0]
```

test.py文件代码

```
import MY_KNN
import numpy as np

group, labels = MY_KNN.creatDataSet()
print(MY_KNN.classify0([0, 0], group, labels, 3))
print(MY_KNN.classify0([0.8, 0.7], group, labels, 3))
```

运行结果截图

```
1.1 > test.py > ...
1 import MY_KNN
2 import numpy as np
3
4 group, labels = MY_KNN.creatDataSet()
5 print(MY_KNN.classify0([0,0], group, labels, 3))
6 print(MY_KNN.classify0([0.8, 0.7], group, labels, 3))
7

File "G:\mine\may_useful_softwares\anaconda3\lib\site-packages\numpy\lib\_datasource.py", line 100, in _open(path, mode, encoding encoding, newline newline)
    raise IOError("%s not found." % path)
OSError: train.txt not found.

[Done] exited with code=1 in 0.382 seconds

[Running] python -u "f:\桌面\一些文件\主修课程\大二下\人工智能实验\作业\lab1\1.1\test.py"
B
A

[Done] exited with code=0 in 0.952 seconds
```

1.2

题目

写程序，利用classify0函数（k=3）；系统性的实现datingTestSet2.txt中10% 数据的测试，并打印出结果；

代码

KNN.py文件代码

```
from cProfile import label
from tokenize import group
from numpy import *
import operator
import numpy as np

def creatDataSet():
    group = np.loadtxt('train.txt', usecols=(0,1,2))
    labels = np.loadtxt('train.txt', usecols=(3))
    #group = array([[1.0, 1.1], [1.0, 1.0], [0, 0], [0, 0.1]])
    #labels = ['A', 'A', 'B', 'B']
```

```

        return group, labels

def classify0(inX, dataSet, labels, k):
    dataSetSize = dataSet.shape[0]
    diffMat = tile(inX, (dataSetSize, 1)) - dataSet
    sqDiffMat = diffMat ** 2
    sqDistances = sqDiffMat.sum(axis=1)
    distances = sqDistances ** 0.5
    sortedDistIndicies = distances.argsort()
    classCount = {}
    for i in range(k):
        voteLabel = labels[sortedDistIndicies[i]]
        classCount[voteLabel] = classCount.get(voteLabel, 0) + 1
    sortedClassCount = sorted(classCount.items(),
                              key=operator.itemgetter(1), reverse=True)
    return sortedClassCount[0][0]

```

test.py文件代码

```

import KNN
import numpy as np

group, labels = KNN.creatDataSet()
print("group:", group)
print("labels:", labels)
print("testData:", np.loadtxt("test.txt", usecols=(0, 1, 2)))
testData = np.loadtxt("test.txt", usecols=(0, 1, 2)).reshape(100, -1)
trueData = np.loadtxt("test.txt", usecols=(3))
print("testData:", testData[0])
trueNum = 0
for i in range(100):
    print("the [" + str(i) + "] data, " + "predict result:", KNN.classify0(testData[i], group, labels, 3), "true result:", trueData[i])
    if (KNN.classify0(testData[i], group, labels, 3) == trueData[i]):
        trueNum += 1
print("trueNum:", trueNum)
print("trueRate:", trueNum/100)

```

运行结果

```
the [ 67 ] data, predict result: 2.0 true result: 2.0
the [ 68 ] data, predict result: 3.0 true result: 3.0
the [ 69 ] data, predict result: 1.0 true result: 1.0
the [ 70 ] data, predict result: 2.0 true result: 2.0
the [ 71 ] data, predict result: 3.0 true result: 3.0
the [ 72 ] data, predict result: 2.0 true result: 2.0
the [ 73 ] data, predict result: 2.0 true result: 2.0
the [ 74 ] data, predict result: 1.0 true result: 1.0
the [ 75 ] data, predict result: 3.0 true result: 3.0
the [ 76 ] data, predict result: 1.0 true result: 1.0
the [ 77 ] data, predict result: 1.0 true result: 1.0
the [ 78 ] data, predict result: 3.0 true result: 3.0
the [ 79 ] data, predict result: 3.0 true result: 3.0
the [ 80 ] data, predict result: 1.0 true result: 1.0
the [ 81 ] data, predict result: 2.0 true result: 2.0
the [ 82 ] data, predict result: 3.0 true result: 3.0
the [ 83 ] data, predict result: 1.0 true result: 1.0
the [ 84 ] data, predict result: 3.0 true result: 3.0
the [ 85 ] data, predict result: 1.0 true result: 1.0
the [ 86 ] data, predict result: 2.0 true result: 2.0
the [ 87 ] data, predict result: 2.0 true result: 2.0
the [ 88 ] data, predict result: 1.0 true result: 1.0
the [ 89 ] data, predict result: 1.0 true result: 1.0
the [ 90 ] data, predict result: 3.0 true result: 3.0
the [ 91 ] data, predict result: 2.0 true result: 3.0
the [ 92 ] data, predict result: 1.0 true result: 1.0
the [ 93 ] data, predict result: 2.0 true result: 2.0
the [ 94 ] data, predict result: 1.0 true result: 1.0
the [ 95 ] data, predict result: 1.0 true result: 3.0
the [ 96 ] data, predict result: 3.0 true result: 3.0
the [ 97 ] data, predict result: 2.0 true result: 2.0
the [ 98 ] data, predict result: 1.0 true result: 1.0
the [ 99 ] data, predict result: 1.0 true result: 1.0
trueNum: 86
trueRate: 0.86
```

注

如果要运行1.2文件的话最好确保目录下没有中文，如果有中文的话可能导致 `np.loadtxt` 报错

并且请手动把原来的data划分成900个数据的train和100个数据的test文件