**北京科技大学**

**《模式识别基础》实验报告**

学院： 计通 专业： 计科 班级： 计213班

姓名： 王佳 学号： U202142840 实验日期： 2024.04.10

**实验名称：**

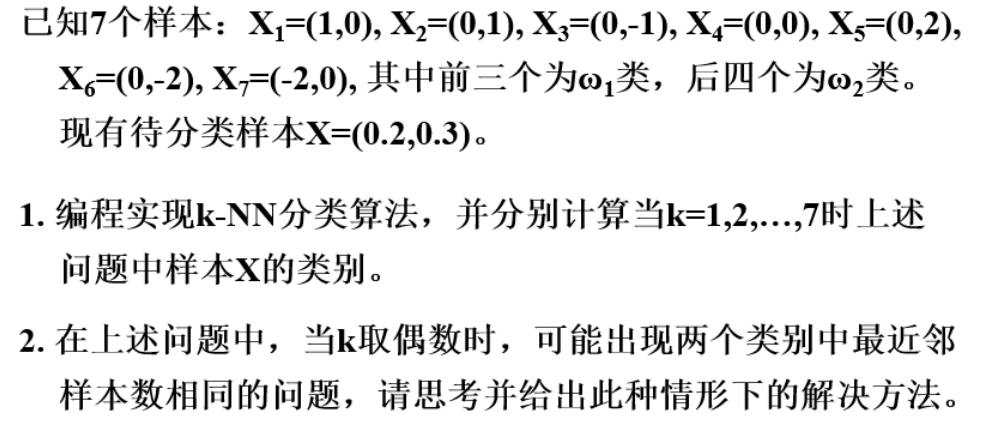
利用scikit-learn实现最近邻分类

**实验原理：**

KNN算法是有监督学习中的分类算法，其原理是当预测一个新样本的类别时，根据它距离最近的 K 个样本点是什么类别来判断该新样本属于哪个类别。

**实验内容：**

1. 实现第三章作业中的最近邻分类问题（可视化结果展示）



2）已知如下电影镜头中的镜头信息与其对应的电影类型



问题：对于电影《唐人街探案》，其包含搞笑镜头 23 个、拥抱镜头 6 个，打

斗镜头 21 个，用近邻法判断其所属的电影类型（可视化结果，并讨论 K 值

对结果的影响）

**实验过程及代码：**

（1）

1. **import** numpy as np
2. **import** matplotlib.pyplot as plt
3. **from** matplotlib.colors **import** ListedColormap
4. **from** sklearn **import** neighbors
5. n\_neighbors= 7
6. X=np.array([(1,0),(0,1),(0,-1),(0,0),(0,2),(0,-2),(-2,0)])
7. y=np.array([1,1,1,2,2,2,2])
8. h=.02
9. cmap\_light=ListedColormap(['orange','cyan'])
10. cmap\_bold=ListedColormap(['darkorange','c'])
11. **for** weights **in** ['uniform','distance']:
12. clf=neighbors.KNeighborsClassifier(n\_neighbors,weights=weights)
13. clf.fit(X,y)
14. x\_min,x\_max=X[:,0].min()-1,X[:,0].max()+1
15. y\_min,y\_max=X[:,1].min()-1,X[:,1].max()+1
16. #生成网格点坐标
17. xx,yy=np.meshgrid((np.arange(x\_min,x\_max,h)),np.arange(y\_min,y\_max,h))
18. #xx.ravel()将网格点坐标展平为一维数组，np.c\_()按列连接数组，得到一个二维数组，每一行是一个网格点坐标
19. #clf.predict()对于传入的展平后的网格点坐标，用于对每个网格点进行分类预测
20. #Z是一个一维数组，表示对每个网格点的预测结果
21. Z=clf.predict(np.c\_[xx.ravel(),yy.ravel()])
22. #将分类结果Z调整为与网格点坐标一样的形状，以便与网格点坐标对应
23. Z=Z.reshape(xx.shape)
24. plt.figure()
25. #根据网格中的数值对应到颜色映射cmap\_light中的颜色并将颜色填充到网格中
26. plt.pcolormesh(xx, yy, Z, cmap=cmap\_light, shading='auto', vmin=1, vmax=2)
27. #绘制训练数据
28. plt.scatter(X[:,0],X[:,1],c=y,cmap=cmap\_bold,edgecolor='k',s=20)
29. #测试数据
30. X\_test=np.array([(0.2,0.3)])
31. y\_pred=clf.predict(X\_test)
32. test\_color=cmap\_bold.colors[y\_pred[0]-1]
33. **print**(y\_pred)
34. plt.scatter(X\_test[:,0],X\_test[:,1],c=test\_color,edgecolor='k',s=80,marker='s',label='Test data')
35. #限定x轴显示的范围
36. plt.xlim(xx.min(),xx.max())
37. plt.ylim(yy.min(),yy.max())
38. plt.title("3-Class classification(k=%i,weights='%s')"%(n\_neighbors,weights))
39. plt.show()

（2）

1. **import** numpy as np
2. **import** matplotlib.pyplot as plt
3. **from** mpl\_toolkits.mplot3d **import** Axes3D
4. **from** matplotlib.colors **import** ListedColormap
5. **from** sklearn **import** neighbors
6. n\_neighbors= 1
7. #X[0]搞笑镜头X[1]拥抱镜头X[2]打斗镜头
8. X=np.array([(45,2,9),(21,17,5),(54,9,11),(39,0,31),(5,2,57),(3,2,65),(2,3,55),(6,4,21),(7,46,4),(9,39,8),(9,38,2),(8,34,17)])
9. #喜剧1，动作2，爱情3
10. y=np.array([1,1,1,1,2,2,2,2,3,3,3,3])
11. h=3
12. cmap\_light=ListedColormap(['orange','cyan','cornflowerblue'])
13. cmap\_bold=ListedColormap(['darkorange','c','darkblue'])
14. **for** weights **in** ['uniform','distance']:
15. clf=neighbors.KNeighborsClassifier(n\_neighbors,weights=weights)
16. clf.fit(X,y)
17. fig=plt.figure()
18. ax=fig.add\_subplot(111,projection='3d')
19. #绘制训练数据
20. ax.scatter(X[:,0],X[:,1],X[:,2],c=y,cmap=cmap\_bold,edgecolor='k',s=20)
21. X\_test=np.array([(23,6,21)])
22. y\_pred=clf.predict(X\_test)
23. test\_color=cmap\_bold.colors[y\_pred[0]-1]
24. **print**(y\_pred)
25. #绘制测试数据 ax.scatter(X\_test[:,0],X\_test[:,1],X\_test[:,2],c=test\_color,cmap=cmap\_bold,edgecolor='k',s=80,marker='s')
26. #分类边界
27. x\_min,x\_max=X[:,0].min()-1,X[:,0].max()+1
28. y\_min,y\_max=X[:,1].min()-1,X[:,1].max()+1
29. z\_min,z\_max=X[:,2].min()-1,X[:,2].max()+1
30. #生成网格点坐标
31. xx,yy,zz=np.meshgrid((np.arange(x\_min,x\_max,h)),np.arange(y\_min,y\_max,h),np.arange(z\_min,z\_max,h)
32. Z=clf.predict(np.c\_[xx.ravel(),yy.ravel(),zz.ravel()])
33. #将分类结果Z调整为与网格点坐标一样的形状，以便与网格点坐标对应
34. Z=Z.reshape(xx.shape)
35. #绘制网格点
36. ax.scatter(xx, yy, zz, c=Z, cmap=cmap\_light, edgecolor='k', alpha=0.1)
37. ax.set\_title("3-Class classification(k=%i,weights='%s')"%(n\_neighbors,weights))
38. plt.show()

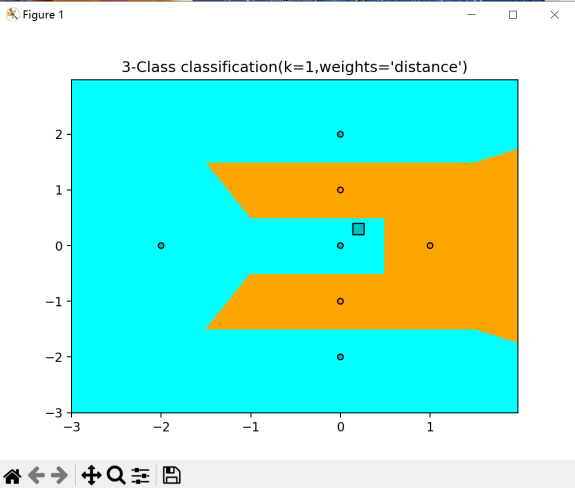
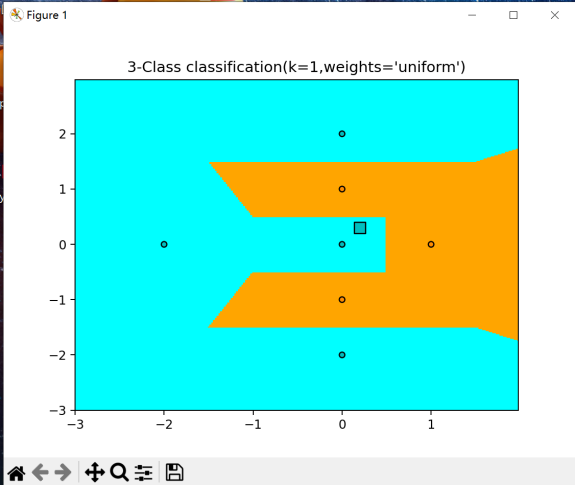
**实验结果与分析：**

（1）

K=1

Uniform:2

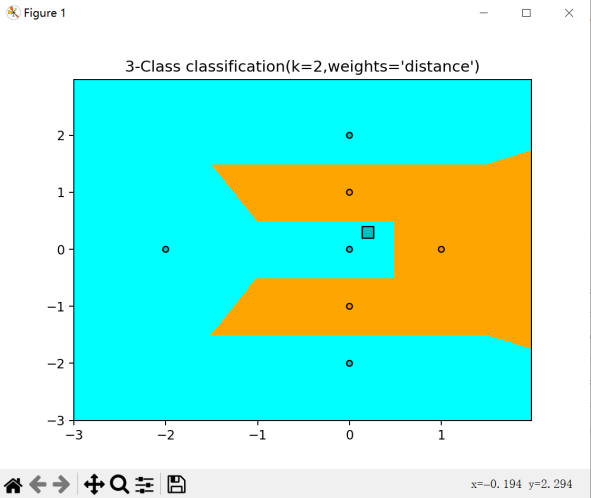
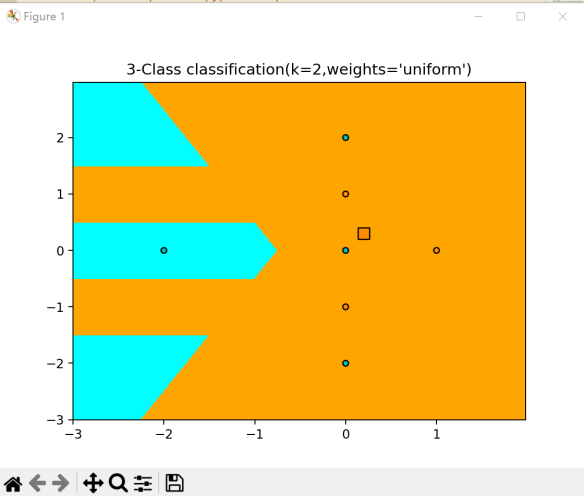
Distance:2



K=2

Uniform:1

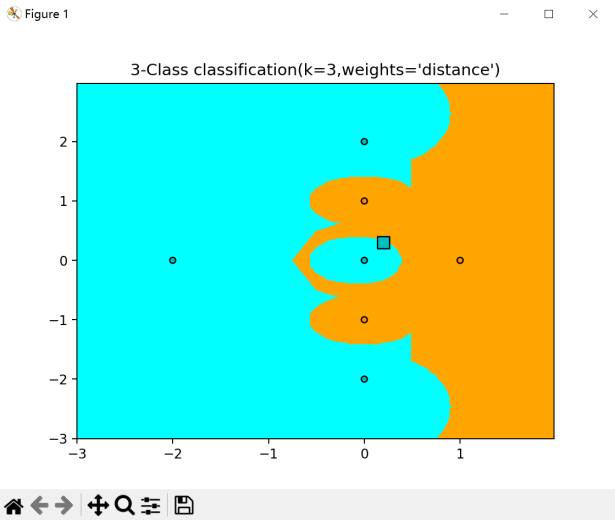
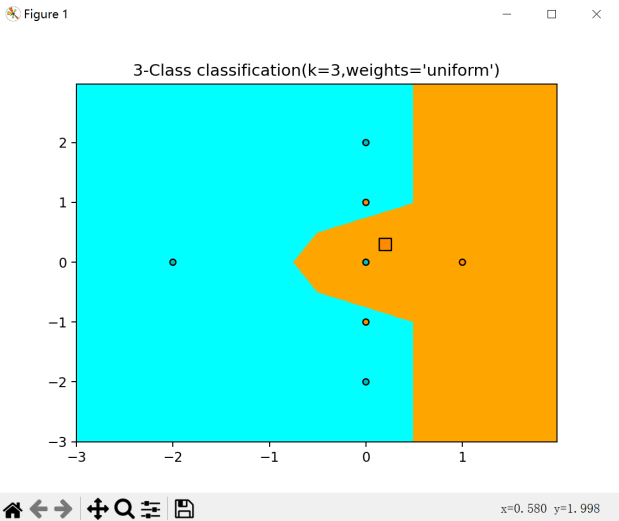
Distance:2



K=3

Uniform:1

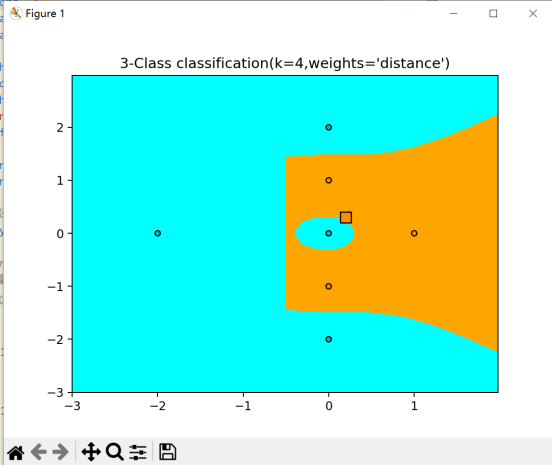
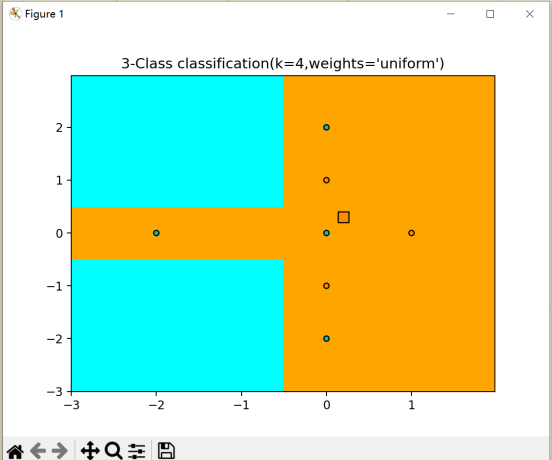
Distance:2



K=4

Uniform:1

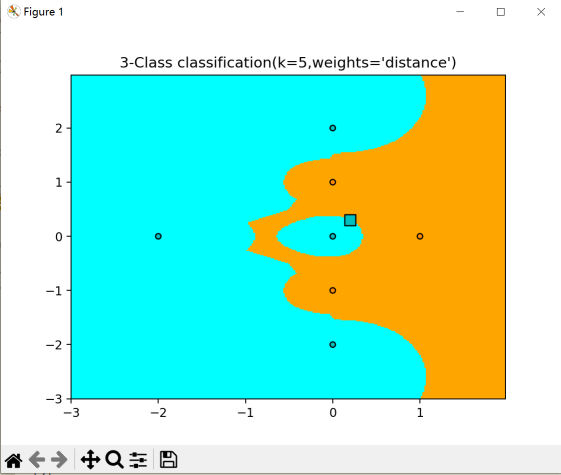
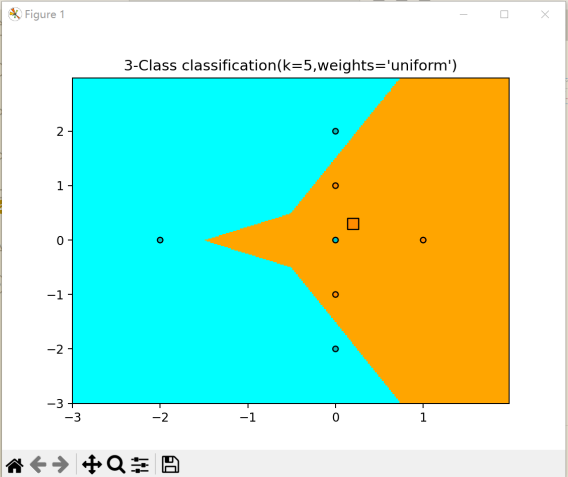
Distance:1



K=5

Uniform:1

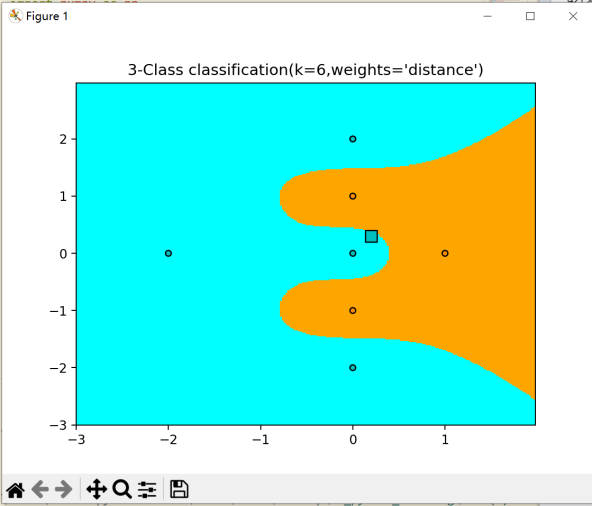
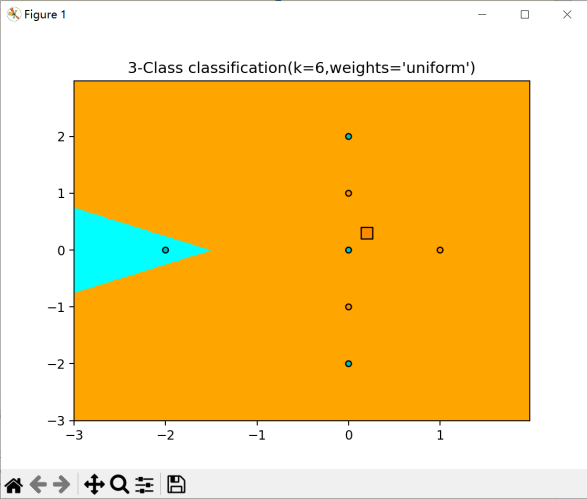
Distance:2



K=6

Uniform:1

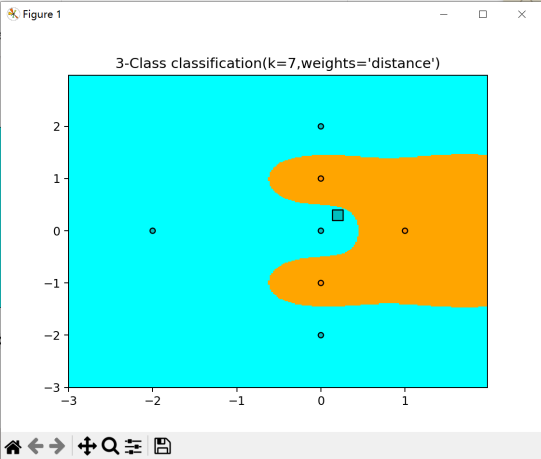
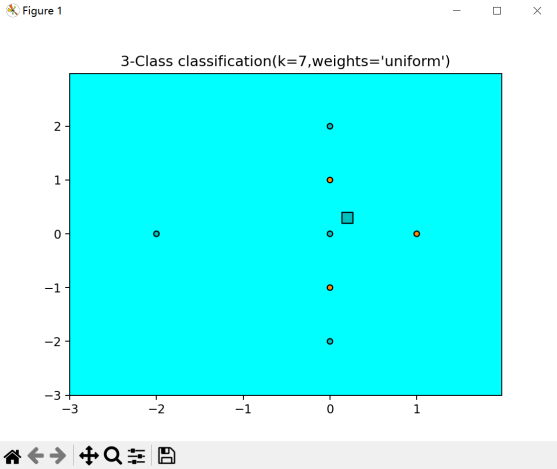
Distance:2



K=7

Uniform:2

Distance:2



1）可见随着K取值的不同，待测样本所属类别不同。当K取值为1时，新增未标注样本点会被归到距离他最近的样本所在的类中，这可能存在噪声点问题。当K值过大，这是可能会把较远的邻居考虑进去，这也是我们所不希望的情况。

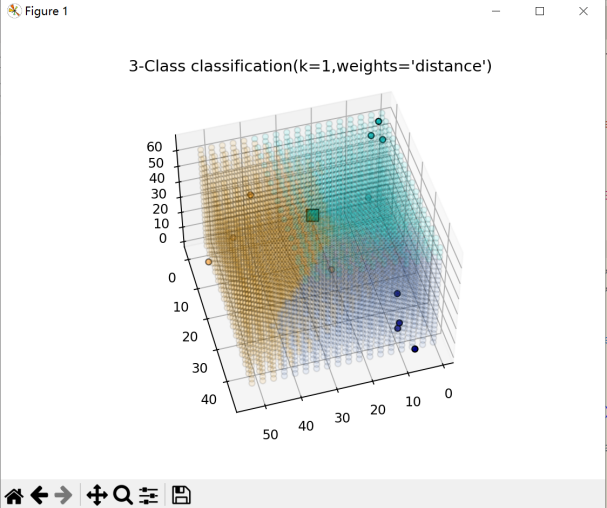
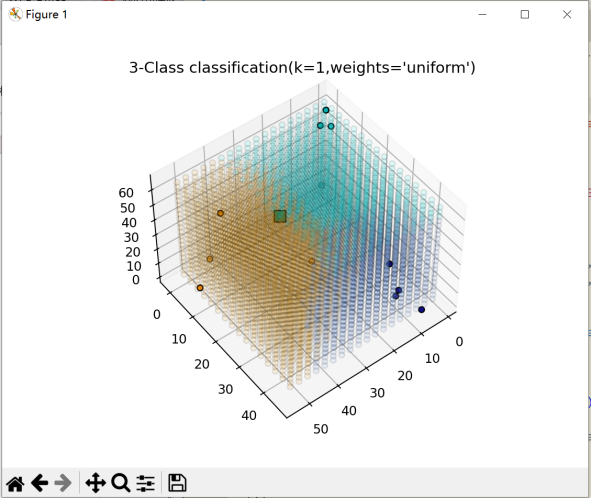
2）当K值为偶数时，就可能出现两个类出现次数一样的情况，即上方使用uniform参数，代表均匀的权重。当k取值为6时，待测样本最近样本3个属于第一类，3个属于第二类，无法判断该样本属于哪一类，这种情况下函数默认按训练样本出现次序来给待测样本分类。这时需要给每个近邻增加权重，即上面所用的distance参数，重量点距离的倒数，这种情况下，查询点的较近邻居将比距离远的邻居具有更大影响力。故把待测样本按照距离倒数的权重正确分类。

(2)

K=1

Uniform:2动作片

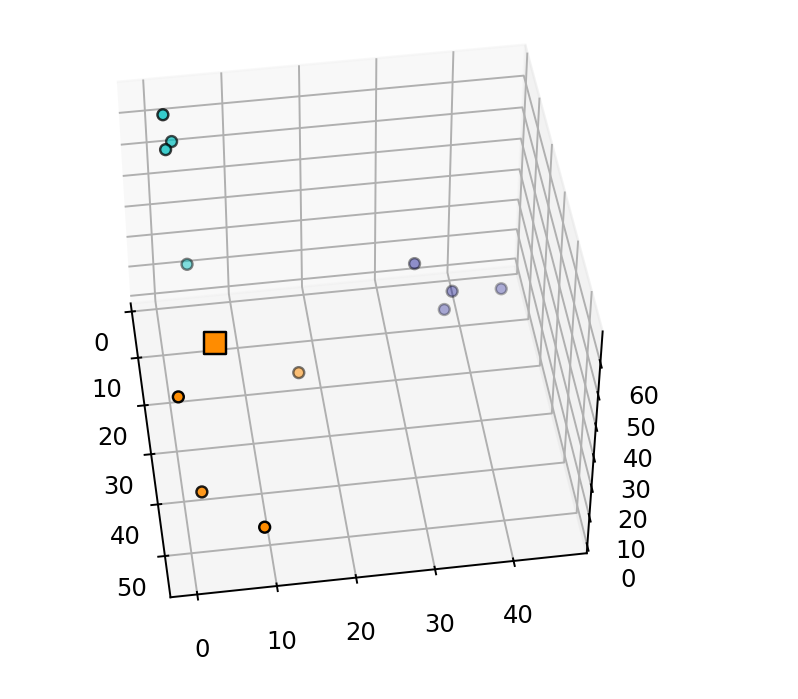
Distance:2动作片



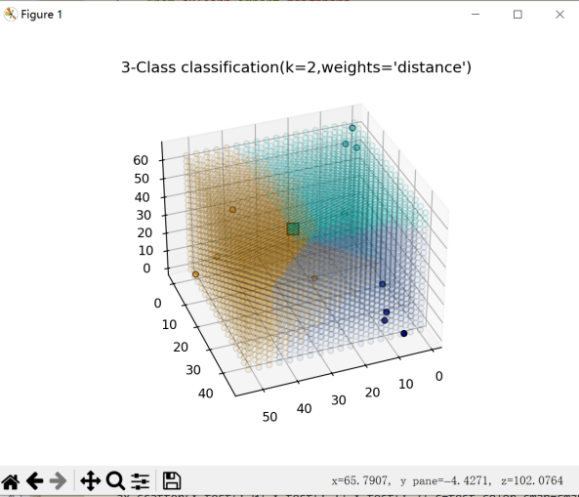
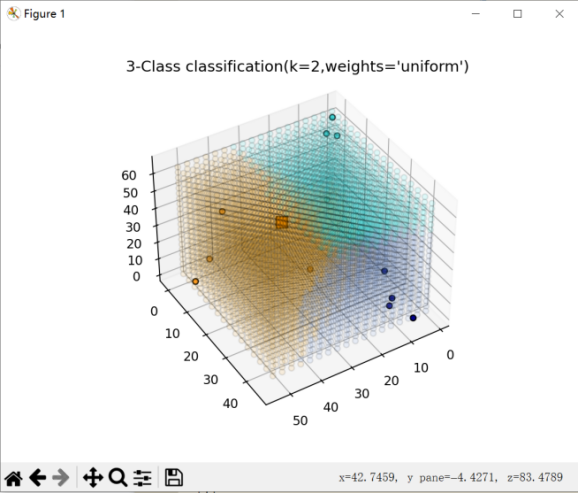
K=2

Uniform:1喜剧片

Distance:2动作片



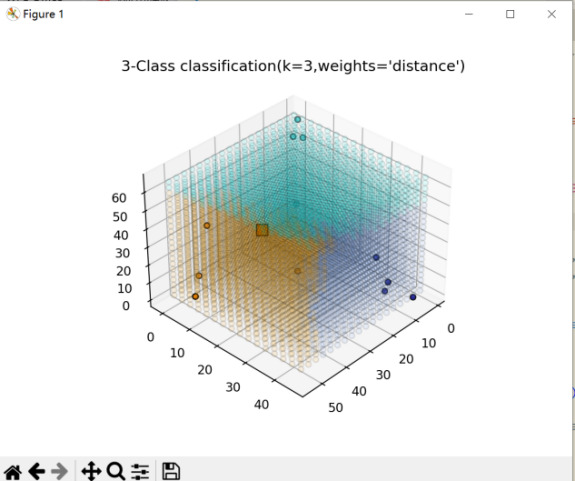
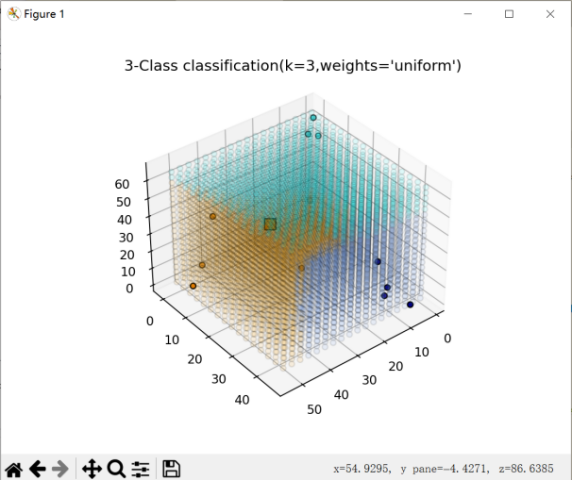
此时距离待测样本最近的一个是喜剧片，一个是动作片，但喜剧片在训练样本中的顺序在动作片前，所以当参数为uniform时，把待测样本归于喜剧片，但是当给近邻权重设置为距离的倒数时，便是归于距离最近的动作片，与k=1时相符



K=3

Uniform:1喜剧片

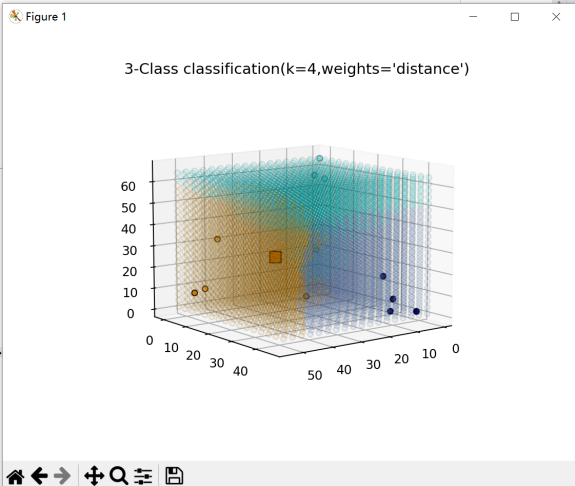
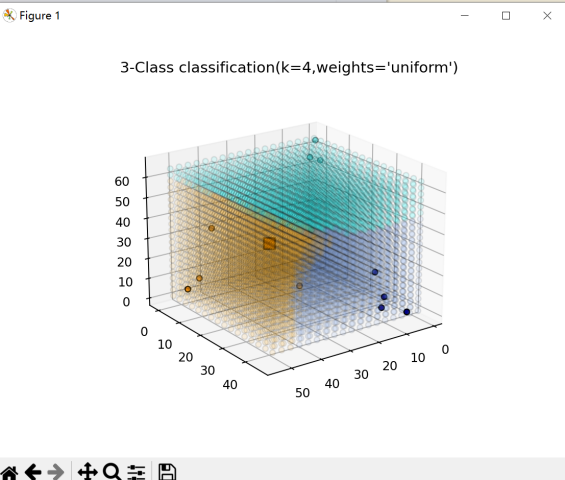
Distance:1喜剧片



K=4

Uniform:1喜剧片

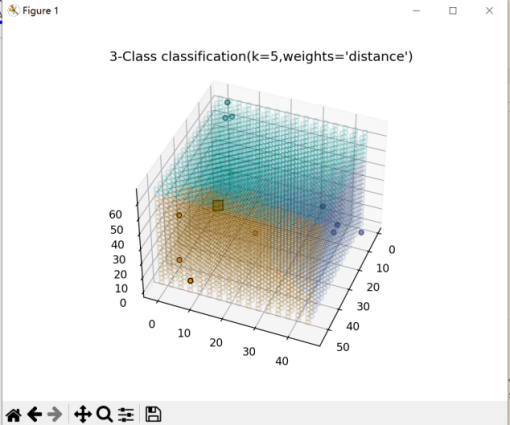
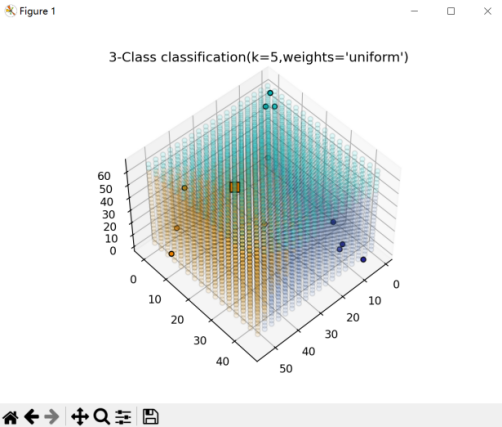
Distance:1喜剧片



K=5

Uniform:1喜剧片

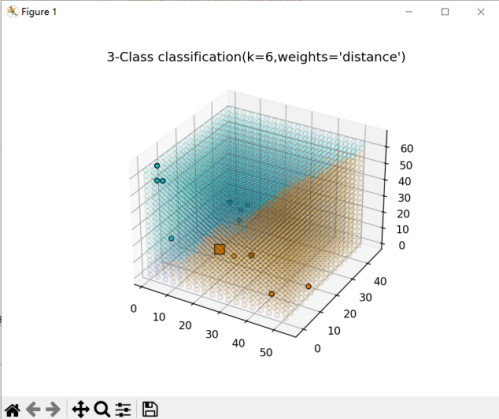
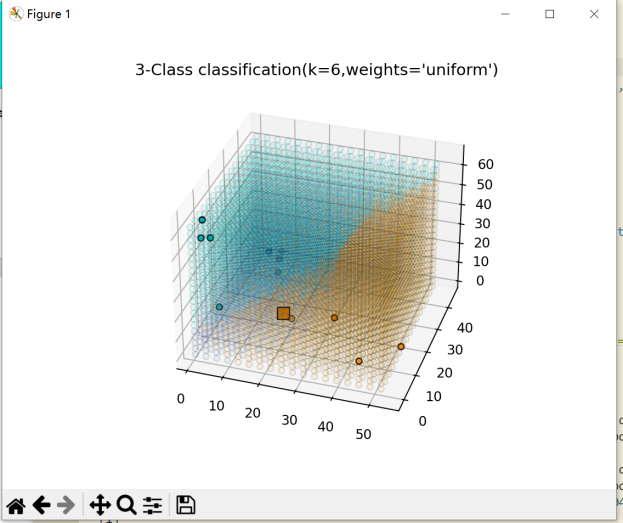
Distance:1喜剧片



K=6

Uniform:1喜剧片

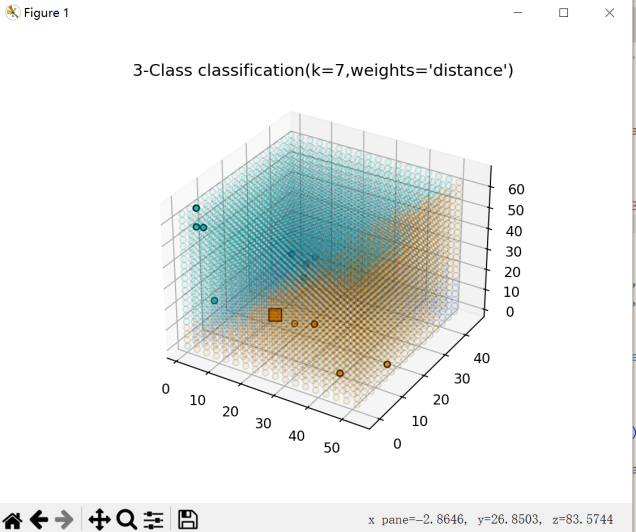
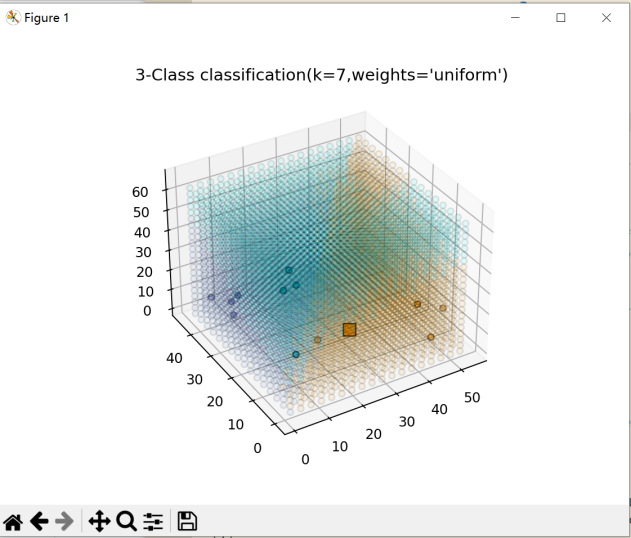
Distance:1喜剧片



K=7

Uniform:1喜剧片

Distance:1喜剧片



**实验体会与收获：**

1. 通过本次实验学会了将二维和三维数据通过KNN算法进行分类，学会了使用sklearn.neighbors.KNeighborsClassifier()来训练分类器，并通过plt来进行数据可视化
2. 更深的体会到K值的不同给分类结果造成的影响，以及不同权重给分类结果带来的改善