

KNN算法实现

实验目标

通过本案例的学习和课后作业的练习：

1. 了解KNN算法的基本思想；
2. 能够使用SKlearn实现KNN算法。

你也可以将本案例相关的 ipynb 学习笔记分享到 [AI Gallery Notebook \(https://marketplace.huaweicloud.com/markets/aihub/notebook/list/\)](https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看[此文档 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

案例内容介绍

KNN是通过测量不同特征值之间的距离进行分类。它的思路是：如果一个样本在特征空间中的k个最相似(即特征空间中最邻近)的样本中的大多数属于某一个类别，则该样本也属于这个类别，其中K通常是不大于20的整数。KNN算法中，所选择的邻居都是已经正确分类的对象。该方法在定类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

本案例推荐的理论学习视频：

- [《AI技术领域课程--机器学习》 KNN \(https://education.huaweicloud.com/courses/course-v1:HuaweiX+CBUCNXE086+Self-paced/courseware/b2d3d567f5e64e32a2e3a186af64e11a/a8d199817f254372842f19f0e71b001a/\)](https://education.huaweicloud.com/courses/course-v1:HuaweiX+CBUCNXE086+Self-paced/courseware/b2d3d567f5e64e32a2e3a186af64e11a/a8d199817f254372842f19f0e71b001a/)

注意事项

1. 如果您是第一次使用 JupyterLab，请查看 [《ModelArts JupyterLab使用指导》 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=03676d0a-0630-4a3f-b62c-07fba43d2857\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=03676d0a-0630-4a3f-b62c-07fba43d2857) 了解使用方法；
2. 如果您在使用 JupyterLab 过程中碰到报错，请参考 [《ModelArts JupyterLab常见问题解决办法》 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9ad8ce7d-06f7-4394-80ef-4dbf6cfb4be1\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9ad8ce7d-06f7-4394-80ef-4dbf6cfb4be1) 尝试解决问题。

实验步骤

1、导入实验所需的sklearn包，导入numpy，进行矩阵计算

```
In [1]: # 从sklearn 导入KNeighborsClassifier方法
        from sklearn.neighbors import KNeighborsClassifier
        import numpy as np
```

2、提供简单的数据结构进行后续的KNN算法验证

```
In [2]: X = np.array([[1, 1], [1, 1.5], [2, 2.5], [2.5, 3], [1.5, 1], [3, 2.5]])
        y = ['A', 'A', 'B', 'B', 'A', 'B']
```

3、KNN算法的内容

根据sklearn提供的包 创建knn算法对象

n_neighbors=5

int 型参数 knn算法中指定以最近的几个最近邻样本具有投票权，默认参数为5

algorithm='auto'

str参数 即内部采用什么算法实现。有以下几种选择参数：

'ball_tree':球树、'kd_tree':kd树、'brute':暴力搜索、'auto':自动根据数据的类型和结构选择合适的算法。

默认情况下是'auto'。暴力搜索就不用说了大家都知道。具体前两种树型数据结构哪种好视情况而定。KD树是对依次对K维坐标轴，以中值切分构造的树,每一个节点是一个超矩形，在维数小于20时效率最高

ball tree 是为了克服KD树高维失效而发明的，其构造过程是以质心 C和半径r分割样本空间，每一个节点是一个超球体。一般低维数据用 kd_tree速度快，用ball_tree相对较慢。超过20维之后的高维数据用kd_tree效果反而不佳，而ball_tree效果要好，具体构造过程及优劣势的理论大家有兴趣可以去具体学习。

4、调用KNN算法

```
In [3]: model = KNeighborsClassifier(n_neighbors=4, algorithm='ball_tree')
```

5、训练数据

```
In [4]: """
fit 就是训练模型
fit()
训练函数，它是最主要的函数。接收参数只有1个，就是训练数据集，
每一行是一个样本，每一列是一个属性。它返回对象本身，即只是修
改对象内部属性，因此直接调用就可以了，后面用该对象的预测函数
取预测自然及用到了这个训练的结果。其实该函数并不是
KNeighborsClassifier这个类的方法，
而是它的父类SupervisedIntegerMixin继承下来的方法。
"""
model.fit(X, y)

Out[4]: KNeighborsClassifier(algorithm='ball_tree', leaf_size=30, metric='min
kowski',
metric_params=None, n_jobs=None, n_neighbors=4, p=2,
weights='uniform')
```

6、预测结果，可以查看分类结果

```
In [5]: # 预测
print(model.predict([[1.75, 1.75]])) # 输出分类结果
print(model.predict_proba([[1.75, 1.75]])) # 返回预测属于某标签的概率
print(model.score(X, y)) # 输出模型训练结果

['A']

[[0.75 0.25]]

1.0
```

以上是 KNN 的实现方法，受限于篇幅原因，本案例未完全覆盖 KNN 的全部操作，欢迎你将更全面的 KNN 学习笔记分享到 [AI Gallery Notebook \(https://marketplace.huaweicloud.com/markets/aihub/notebook/list/\)](https://marketplace.huaweicloud.com/markets/aihub/notebook/list/) 版块获得成长值 (https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=9b8d7e7a-a150-449e-ac17-2dcf76d8b492)，分享方法请查看[此文档 \(https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb\)](https://marketplace.huaweicloud.com/markets/aihub/article/detail/?content_id=8afec58a-b797-4bf9-acca-76ed512a3acb)。

作业

请你利用本实验中学到的知识点，完成以下编程题：

1. 请你尝试修改 KNeighborsClassifier() 函数的 n_neighbors (选取几个邻居) 参数的不同取值, 看看该参数的修改对模型会有怎样的影响。 (<https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=412a2f98-6e5d-4c54-8415-74c2bb8a9d60>)
2. 请你尝试修改 KNeighborsClassifier() 函数的 weights (权重) 参数的不同取值, 看看该参数的修改对模型会有怎样的影响。 (<https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=591528b4-ab5b-4e07-be6c-34523fc24930>)
3. 使用鸢尾花数据集, 基于Sklearn框架实现一个KNN鸢尾花分类模型。 (<https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=a8b0f4f3-8c98-45d2-98f0-8823a06bb0f3>)