# 1.学习目标

|  |
| --- |
|  |

## 注意:这一节可以学习到解决KNN算法k值的取值问题,可以选择正确是k值

# 2.什么是交叉验证

|  |
| --- |
|  |
|  |

# 3.为什么需要交叉验证

|  |
| --- |
|  |

# 4.超参数网格搜索Grid Search

|  |
| --- |
|  |

# 5.算法API模块

|  |
| --- |
|  |

## 案例,给上一节的鸢尾花案例添加应该k值调优功能,代码如下

|  |
| --- |
|  |
| *""" KNN案例1,鸢尾花种类预测,添加网格搜索和交叉验证. """* from sklearn.datasets import load\_iris from sklearn.model\_selection import train\_test\_split, GridSearchCV from sklearn.preprocessing import StandardScaler from sklearn.neighbors import KNeighborsClassifier  # 1.获取数据,这里使用sklearn里面自带的数据集 iris\_data = load\_iris() # 2.划分数据集,random\_state参数值会影响KNN算法的效果 x\_train, x\_test, y\_train, y\_test = train\_test\_split(iris\_data.data, iris\_data.target, random\_state=6) # 3.特征工程:标准化处理,需要给训练集和测试集都做标准化处理 transformer = StandardScaler() x\_train = transformer.fit\_transform(x\_train) # 给训练集做标准化处理 x\_test = transformer.transform(x\_test) # 这里使用transform是因为我们需要使用训练集的平均值和标准差来对测试集做处理 # 4.KNN算法预估器 estimator = KNeighborsClassifier() # 这里不需要k值因为网格搜索就是为了确定最好的k值 # 模型调优把预估器作为其中应该参数传递给网格搜索函数,返回调优后的类似预估器的对象 estimator = GridSearchCV(estimator, {"n\_neighbors": [1, 3, 5, 7, 9, 11, 13]}, cv=10) estimator.fit(x\_train, y\_train) # 5.模型评估 # 方法1,直接比对真实值与预测值 y\_predict = estimator.predict(x\_test) print("比对真实值与预测值的结果", y\_test == y\_predict) # 方法2.计算准确率 accuracy = estimator.score(x\_test, y\_test) print("准确率为:", accuracy) print("最佳参数:",estimator.best\_params\_) print("最佳结果:",estimator.best\_score\_) print("最佳预估器:",estimator.best\_estimator\_) print("交叉验证结果:",estimator.cv\_results\_) |

### 结果

|  |
| --- |
| D:\programs\Python310\python.exe F:\Projects\_f\Scikit-learn-proj\heima-ml-3days-basic-course-code\day02\_codes\ex1-iris-knn-demo-cross-validation.py  比对真实值与预测值的结果 [ True True True True True True True True True True True True  True True True False True True True True True True True True  True True True True True True True True True True False True  True True]  准确率为: 0.9473684210526315  最佳参数: {'n\_neighbors': 11}  最佳结果: 0.9734848484848484  最佳预估器: KNeighborsClassifier(n\_neighbors=11)  交叉验证结果: {'mean\_fit\_time': array([0.00130074, 0.00140035, 0.0009002 , 0.00120006, 0.00109982,  0.00100036, 0.0009001 ]), 'std\_fit\_time': array([0.00045908, 0.00066394, 0.00030007, 0.00040032, 0.00030041,  0.00044755, 0.00030003]), 'mean\_score\_time': array([0.00419962, 0.00420048, 0.00410006, 0.00360012, 0.00370057,  0.00399995, 0.00370014]), 'std\_score\_time': array([0.00107701, 0.00074893, 0.00083102, 0.00066383, 0.00045841,  0.00126542, 0.00064034]), 'param\_n\_neighbors': masked\_array(data=[1, 3, 5, 7, 9, 11, 13],  mask=[False, False, False, False, False, False, False],  fill\_value=999999), 'params': [{'n\_neighbors': 1}, {'n\_neighbors': 3}, {'n\_neighbors': 5}, {'n\_neighbors': 7}, {'n\_neighbors': 9}, {'n\_neighbors': 11}, {'n\_neighbors': 13}], 'split0\_test\_score': array([1., 1., 1., 1., 1., 1., 1.]), 'split1\_test\_score': array([0.91666667, 0.91666667, 1. , 0.91666667, 0.91666667,  0.91666667, 0.91666667]), 'split2\_test\_score': array([1., 1., 1., 1., 1., 1., 1.]), 'split3\_test\_score': array([1. , 1. , 1. , 1. , 0.90909091,  1. , 1. ]), 'split4\_test\_score': array([1., 1., 1., 1., 1., 1., 1.]), 'split5\_test\_score': array([0.90909091, 0.90909091, 1. , 1. , 1. ,  1. , 1. ]), 'split6\_test\_score': array([1., 1., 1., 1., 1., 1., 1.]), 'split7\_test\_score': array([0.90909091, 0.90909091, 0.90909091, 0.90909091, 1. ,  1. , 1. ]), 'split8\_test\_score': array([1., 1., 1., 1., 1., 1., 1.]), 'split9\_test\_score': array([0.90909091, 0.81818182, 0.81818182, 0.81818182, 0.81818182,  0.81818182, 0.81818182]), 'mean\_test\_score': array([0.96439394, 0.95530303, 0.97272727, 0.96439394, 0.96439394,  0.97348485, 0.97348485]), 'std\_test\_score': array([0.04365767, 0.0604591 , 0.05821022, 0.05965639, 0.05965639,  0.05742104, 0.05742104]), 'rank\_test\_score': array([6, 7, 3, 4, 4, 1, 1])}  Process finished with exit code 0 |

## 扩展:上面我们提到了3总距离,怎么查看当前使用哪一个?我们只需要按住Ctrl点击KNeighborsClassifier(),就可以打开应该选项卡,在里面看到下面的内容

|  |
| --- |
|  |

### 可见,默认是使用欧式距离,也就是明可夫斯基距离当p=2的情况,当明可夫斯基距离的p=1,就是指曼哈顿距离

# 6案例:Facebook签到位置预测(p05)

|  |
| --- |
|  |
|  |
|  |

## 由于我的电脑性能不好,我这里只能使用缩小的数据集,我打算使用10%左右的数据复制到xx2.csv中

## 流程分析

|  |
| --- |
|  |

## 实际操作

|  |
| --- |
|  |