|  |  |
| --- | --- |
| 参数 | 详述 |
| KNN中的K值n\_neighbors | K值的选择与样本分布有关，一般选择一个较小的K值，可以通过交叉验证来选择一个比较优的K值，默认值是5。如果数据是三维一下的，如果数据是三维或者三维以下的，可以通过可视化观察来调参。 |
| 近邻权weights | 主要用于标识每个样本的近邻样本的权重，如果是KNN，就是K个近邻样本的权重，如果是限定半径最近邻，就是在距离在半径以内的近邻样本的权重。可以选择"uniform","distance" 或者自定义权重。选择默认的"uniform"，意味着所有最近邻样本权重都一样，在做预测时一视同仁。如果是"distance"，则权重和距离成反比例，即距离预测目标更近的近邻具有更高的权重，这样在预测类别或者做回归时，更近的近邻所占的影响因子会更加大。当然，我们也可以自定义权重，即自定义一个函数，输入是距离值，输出是权重值。这样我们可以自己控制不同的距离所对应的权重。  一般来说，如果样本的分布是比较成簇的，即各类样本都在相对分开的簇中时，我们用默认的"uniform"就可以了，如果样本的分布比较乱，规律不好寻找，选择"distance"是一个比较好的选择。如果用"distance"发现预测的效果的还是不好，可以考虑自定义距离权重来调优这个参数。 |
| 停止建子树的叶子节点阈值leaf\_size | 这个值控制了使用KD树或者球树时， 停止建子树的叶子节点数量的阈值。这个值越小，则生成的KD树或者球树就越大，层数越深，建树时间越长，反之，则生成的KD树或者球树会小，层数较浅，建树时间较短。默认是30. 这个值一般依赖于样本的数量，随着样本数量的增加，这个值必须要增加，否则不光建树预测的时间长，还容易过拟合。可以通过交叉验证来选择一个适中的值。  如果使用的算法是蛮力实现，则这个参数可以忽略。 |
| 距离度量metric |  |
| 距离度量附属参数p | p是使用距离度量参数 metric 附属参数，只用于闵可夫斯基距离和带权重闵可夫斯基距离中p值的选择，p=1为曼哈顿距离， p=2为欧式距离。默认为2 |
| 并行处理任务数n\_jobs | 主要用于多核CPU时的并行处理，加快建立KNN树和预测搜索的速度。一般用默认的-1就可以了，即所有的CPU核都参与计算。 |
| KNN使用的算法algorithm | 算法一共有三种，第一种是蛮力实现，第二种是KD树实现，第三种是球树实现。这三种方法在[K近邻法(KNN)原理小结](http://www.cnblogs.com/pinard/p/6061661.html)中都有讲述，如果不熟悉可以去复习下。对于这个参数，一共有4种可选输入，‘brute’对应第一种蛮力实现，‘kd\_tree’对应第二种KD树实现，‘ball\_tree’对应第三种的球树实现， ‘auto’则会在上面三种算法中做权衡，选择一个拟合最好的最优算法。需要注意的是，如果输入样本特征是稀疏的时候，无论我们选择哪种算法，最后scikit-learn都会去用蛮力实现‘brute’。  个人的经验，如果样本少特征也少，使用默认的 ‘auto’就够了。 如果数据量很大或者特征也很多，用"auto"建树时间会很长，效率不高，建议选择KD树实现‘kd\_tree’，此时如果发现‘kd\_tree’速度比较慢或者已经知道样本分布不是很均匀时，可以尝试用‘ball\_tree’。而如果输入样本是稀疏的，无论你选择哪个算法最后实际运行的都是‘brute’。 |