什么是KNN

KNN是一种基本的监督学习方法。给定测试样本，基于某种距离度量从训练集中找出最邻近的K个训练样本，然后利用这K个样本的信息来进行预测。对于分类任务来说，通常采用多数投票法，即选择这K个样本中出现次数最多的类别作为预测结果；对于回归任务来说，通常采用平均法，即将这K个样本的实值输出的平均值作为预测结果。还可基于距离远近进行加权投票或者加权平均，距离越近的样本权重越大。

三个要素

距离的度量、K值的选取和决策规则的制定

距离的度量

一般采用L_p距离。当$p=2$时，为欧式距离；当$p=1$时，为曼哈顿距离；当$p=\infty$时，为各个坐标距离的最大值。

K值的选取

K值的选取需要权衡模型的偏差和方差。K值越小，模型越复杂，容易过拟合；K值越大，模型越简单，但容易欠拟合。通常采用交叉验证法来选取最优的K值。

决策规则的制定

对于分类任务来说，通常采用多数投票或者根据距离远近来加权投票；

对于回归任务来说，通常采用平均法或者根据距离远近来加权平均。

KNN的实现

当特征空间的维数很大和训练数据量很大时，用线性扫描来搜索K近邻会很耗时，可以采用kd树的树形存储结构，便于快速检索。

kd树

kd树是一种对k维空间中的实例点进行存储以便对其进行快速检索的树形数据结构。kd是二叉树，kd树的每个结点对应于一个k维超矩形区域。

kd树的构造： 不断地用垂直于坐标轴的超平面将k维空间切分，构成一系列的k维超矩形区域。

搜索**kd**树： 以最近邻搜索为例。首先找到包含目标点的叶结点，以此叶结点为当前最近点；然后从该叶结点出发，依次回退到父结点，在父结点的另一子结点区域寻找，如果存在比当前最近点更近的点，则更新为当前最近点，否则转到更上一级的父结点，这样递归地向上回退；当回退到根结点时结束。

复杂度分析： 当实例点是随机分布的，kd树搜索的平均计算复杂度是O(logN)，N是训练集实例点个数。kd树适用于训练实例数远大于空间维数时的k近邻搜索。

KNN的说明

特点

关于KNN算法，有以下几点说明：

- KNN算法简单，易于理解，易于实现；
- KNN没有显式的学习过程，无需训练，无需估计参数；
- KNN是一种惰性学习算法，测试时计算开销很大；
- 当特征空间的维度很高时，可能会对距离之间的差异性产生影响，从而影响预测效果；
- 近邻间的距离会被大量不相关的特征属性所支配

改进

具体实现中，可以做的有以下几个方面：

- 距离度量方面
    - 针对不同的问题选择合适的距离度量
    - 计算距离之前先对特征进行标准化
    - 对于特征维度很高的情况，可以先进行降维
    - 计算实例间距离时对每个特征属性进行加权，减小不相关属性的影响
- 在最后预测时，根据距离远近加权输出。