**一、基本原理：**

KNN是机器学习中一个比较简单的模型，它主要用于分类，也可以用于回归。KNN算法的基本思路是：认为距离越近的样本，输出越类似，因此一个样本的输出可以由与它较近的几个样本的输出估计出来。

KNN算法的流程为：给定训练样本集和待预测的样本，首先计算训练集中与待预测样本距离最小的k个样本；然后将待预测样本分类到这k个样本中样本数最多的类别。（对于回归问题，则是将这k个样本的平均值作为输出）

KNN算法没有显式的训练过程，它在给定待预测样本后，就直接进行预测的操作。

**二、参数：**

KNN算法的主要参数是k值和距离度量方式。

对于二类分类问题，为了避免出现“平局”的情况，k值一般取奇数。k值既不能取太大，也不能取太小，若k取太小，预测结果容易受噪声影响，误差较大（“估计误差”较大）；k越大，模型受噪声影响越小，模型决策边界越平滑，模型越稳定，但若k取太大，就减弱了距离因素的影响，违背了KNN的初衷，导致模型过于简单而使误差也较大（“近似误差”较大）。一般来说，k值应低于样本数的平方根。

常见的距离度量方式有：欧氏距离、曼哈顿距离、闵可夫斯基距离等，不同的距离度量方式下，样本的最近邻可能是不同的。

**三、算法的复杂度：**

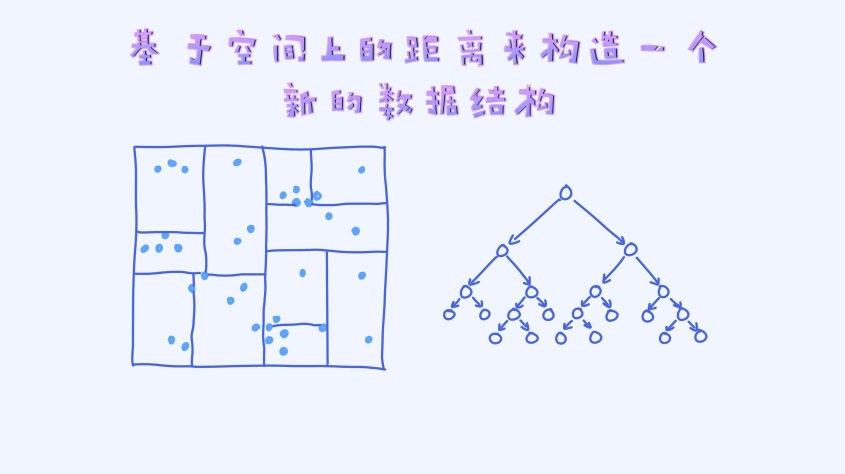
预测一个样本时，需要计算它与每个训练样本的距离。

设N是样本个数，D是样本特征的维数，则KNN的时间复杂度为：O(N\*D)。

**四、kd树：**

kd树用于KNN算法在样本集中搜索k近邻的过程，提高搜索速度。kd树的核心思想是，按照样本各维特征，将样本空间划分为多个小空间。这个划分的过程对应于kd树的分裂过程。划分空间完成后，就形成一棵kd树。搜索样本的k近邻时，只需在样本所在的小空间和相邻空间中进行搜索即可（注意，与一个样本距离最近的k个样本不一定在同一小空间中，因此不能只搜索待预测样本所在的小空间，而是还要搜索相邻空间），搜索的过程相当于在kd树上“tracking”。

kd树的缺点是当样本特征维数增加时，kd树搜索的时间复杂度会呈指数级增加，因此kd树只能用于维数较少的数据。



**五、KNN的改进：**

最基本的KNN将待预测样本分类为距离最近的k个样本中样本数最多的类别，这样做有一个缺陷，就是决策的规则太简单。虽然在确定k个决策样本时考虑了距离的因素，但是在利用这k个样本做最终决策时，没有考虑距离的影响。举个例子，有一个二类分类问题，k=3，k近邻的3个样本中，0类有两个，1类有1个，这时按照基本KNN算法，应将待预测样本分为0类，但是1类的1个样本与待预测样本之间的距离比那两个0类样本要小的多，这时按照常理应该将待预测样本分为1类。

针对此缺陷的改进方法为：“带权重的KNN”，其思路是，对于k近邻样本，计算每一类样本与待预测样本的距离的平均值，作为该类别的权重，然后比较每个类别的权重大小，将待预测样本分到权重大的那一类里。