

## KNN:

所谓K近邻算法，即是给定一个训练数据集，对新的输入实例，在训练数据集中找到与该实例最邻近的K个实例（也就是上面所说的K个邻居），这K个实例的多数属于某个类，就把该输入实例分类到这个类中。

当无法判定当前待分类点是从属于已知分类中的哪一类时，我们可以依据统计学的理论看它所处的位置特征，衡量它周围邻居的权重，而把它归为(或分配)到权重更大的那一类。这就是K近邻算法的核心思想。

### 欧氏距离:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

### 曼哈顿距离:

$$d_{12} = \sum_{k=1}^n |x_{1k} - x_{2k}|$$

### 切比雪夫距离:

$$D_{Chess} = \max(|x_2 - x_1|, |y_2 - y_1|)$$

### 闵可夫斯基距离:

$$d_{12} = \sqrt[p]{\sum_{k=1}^n |x_{1k} - x_{2k}|^p}$$

### 标准化欧氏距离:

$$d_{12} = \sqrt{\sum_{k=1}^n \left(\frac{x_{1k} - x_{2k}}{s_k}\right)^2}$$

### 马氏距离:

$$D(X) = \sqrt{(X - u)^T S^{-1} (X_i - X_j)}$$

### 巴氏距离:

$$D_B(p, q) = -\ln(BC(p, q))$$

其中：

$$BC(p, q) = \sum_{x \in X} \sqrt{p(x)q(x)}$$

**K值选择:**

1. 如果选择较小的K值，就相当于用较小的领域中的训练实例进行预测，“学习”近似误差会减小，只有与输入实例较近或相似的训练实例才会对预测结果起作用，与此同时带来的问题是“学习”的估计误差会增大，换句话说，K值的减小就意味着整体模型变得复杂，容易发生过拟合；
2. 如果选择较大的K值，就相当于用较大领域中的训练实例进行预测，其优点是可以减少学习的估计误差，但缺点是学习的近似误差会增大。这时候，与输入实例较远（不相似的）训练实例也会对预测器作用，使预测发生错误，且K值的增大就意味着整体的模型变得简单。
3.  $K=N$ ，则完全不足取，因为此时无论输入实例是什么，都只是简单的预测它属于在训练实例中最多的类，模型过于简单，忽略了训练实例中大量有用信息。

在实际应用中，K值一般取一个比较小的数值，例如采用交叉验证法（简单来说，就是一部分样本做训练集，一部分做测试集）来选择最优的K值。