第五章 K近邻KNN

一、距离

1. 基本理念
   所谓的K近邻就是与记录数据x有k个最近的距离的数据点
   - K近邻（KNN）算法基于相似性原理运行，**它通过考量训练数据集中与其距离最近的K个数据点的标签或值来预测新数据点的标签或值**。然后，数据点的类别或数值将由其K个邻近点的多数投票结果或平均值来确定。
   - K近邻既能用于分类也能用于回归
     对于分类问题：对新样本，根据其k个最近邻的训练样本的类别，通过多数表决等方式进行预测
     对于回归问题：对新样本，根据其k个最近邻的训练样本的标签值的均值作为预测值
   - 相似性：如何计算相似度——选择一些记录之间的“距离函数”
   - K近邻算法：如何识别K
2. k近邻法的三要素：
   - 存储的记录集合
   - 计算记录之间距离的矩阵
   - K的值，即要获取的近邻的数量
3. k近邻步骤
   分类一个未知的记录
   计算与其他训练记录的距离
   识别最近的K个近邻
   使用最近的近邻的类标号来决定未知记录的类标号
4. 距离度量：
   - 距离度量，是一种衡量数据集中元素之间关系(相似性/差异性)的方法，它通过定义距离函数来实现，这个函数为数据集中每个元素提供了一种相互关系的度量。这个距离函数帮助我们量化数据集中任意两个元素之间的差异
   - 距离度量函数应该满足的属性
     设$d(x,y)$代表点x到点y之间的距离，一个合理的距离公式应该满足以下属性：
     - 合理定义的：针对两个点x,y满足$d(x,y) >= 0$
     - 身份保持的：对于任何一个点x满足$d(x,x) = 0$
     - 对称性：对于任何两个点x,y满足$d(x,y) = d(y,x)$
     - 三角不等式：对于任何三个点x,y,z满足$d(x,z) <= d(x,y) + d(y,z)$
5. 常见的距离公式
   - **欧式距离Euclidean Distance**：

$$d(x,y) = (\sum_i |x_i - y_i|^2)^{\frac{1}{2}}$$

- 曼哈顿距离Manhattan Distance：

$$d(x,y)=\sum_i |x_i-y_i|$$

- 闵可夫斯基距离Minkowski Distance：

$$d(x,y)=(\sum_i |x_i-y_i|^p)^{\frac{1}{p}}$$

6. 标准化方法：
   不同的特征可能以不同的方式进行测量，这种测量方式可能会导致观测结果之间的差异变得毫无意义（或者意义变得不大）。这就需要进行缩放或标准化处理，以避免在分析过程中出现任何问题。
   - Min-Max：

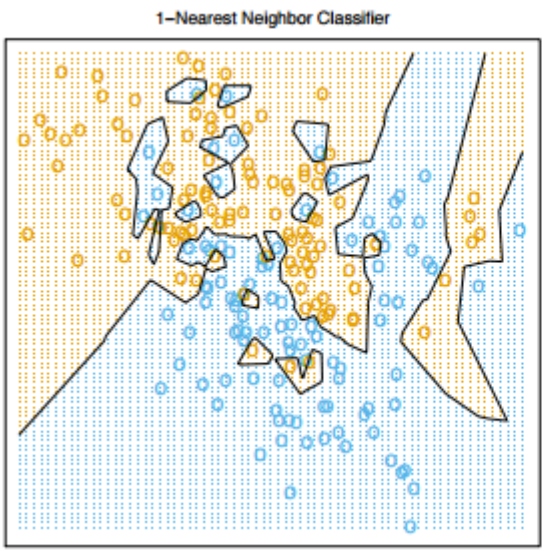
$$x'_{ij}=\frac{x_{ij}-min(x_i)}{max(x_i)-min(x_i)}$$

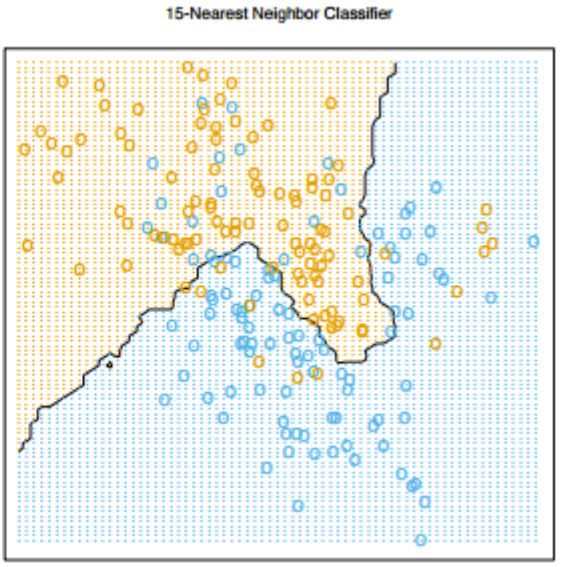
   - Z-score：

$$z=\frac{x-\mu}{\sigma}$$

二、KNN

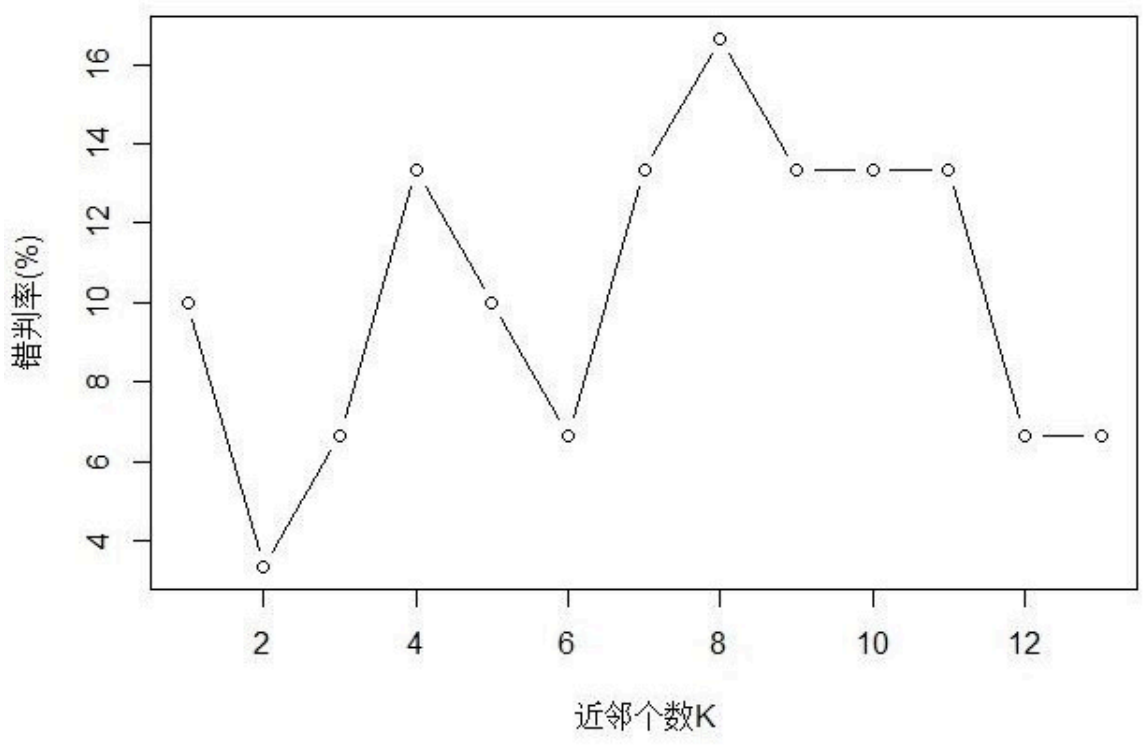
1. KNN 步骤:
   - 给K赋予一个值。
   - 计算新数据与所有现有数据之间的距离。将它们按距离从小到大进行排序。
   - 根据计算出的距离找到与新数据距离最近的K个数据点。
   - 将新数据分配给最近邻中的多数类别。
2. **高k值与低k值的区别：**





- 较低的k值能够捕捉数据中的局部结构（但也会包含噪声），使模型对噪声更加敏感，但可能会出现过拟合现象。高方差，但低偏差。
- k值越大，效果越好，能更平滑地处理数据，减少噪声，但可能会遗漏局部结构。存在较大的偏差，且方差较小。

3. K值的选择
   - 建议将k设为**奇数**，以避免分类过程中的平局情况，而交叉验证策略能够帮助您为特定数据集选择最佳的k值。



   为了确定适合数据的K值，需**多次运行KNN算法，并针对不同的K值进行测试**。
   “K”这个奇数值比偶数值更受青睐，因为这样可以避免投票时出现平局的情况。交叉验证有助于确定数据集的最优“K”值。

4. 评估

优点	**缺点**
易于实施：易于理解和执行	计算成本：尤其是在处理大型数据集时，因为需要为每个数据点计算距离。这会占用更多的内存和时间。
无需训练阶段（惰性算法）：无需单独的训练阶段。	仅限欧几里得距离：在处理非欧几里得数据（如分类数据或二进制数据）时，这可能会成为一种不利因素
参数较少：与其他机器学习算法相比，仅需要一个K值和一个距离度量标准即可。	需要合理选择K值：如果K值过小，算法可能会对数据中的噪声过于敏感；而如果K值过大，算法则可能遗漏数据中的重要模式。

三、基于变量重要性的KNN算法

1. 主要步骤

- 确定K
- 逐个排除输入变量，计算误差e_i
- 计算第i个变量的重要性：$FI_i = e_i + \frac{1}{p}$，其中p代表变量的数量
- 计算距离的权重：重要的变量有更高的权重

$$w_i = \frac{FI_i}{\sum_{j=1}^{p} FI_j}$$

2. KNN的改进
   - 普通 KNN 的假设问题
     KNN 默认 **K 个近邻对预测结果有同等的影响**。
     当输入变量是 **分类型或顺序型** 时，直接使用欧几里得距离不再恰当，因为：
     - 欧几里得距离只能衡量数值型变量的大小差异
     - 无法正确表达类别之间的相似性
   - 改进思路：加权 KNN
     核心思想：邻居的权重取决于与目标点的相似度
     相似性定义：每个观测值与 (X_0) 之间距离的非线性函数
     特点：
     - 距离越近，相似性越强 → 权重越高
     - 权重高的邻居对预测结果的影响更大
   - 常见核函数及公式
     其中(Z_i, Z_0) 为第 i 个观测点与目标点的距离，h 为带宽参数。

核函数	公式	作用 / 特点
三角核（Triangle Kernel）	$w_i = \max\left(0,\ 1 - \frac{d(Z_i, Z_0)}{h}\right)$	权重随距离线性递减
高斯核（Gaussian Kerne…	$w_i = \exp\left(-\frac{d(Z_i, Z_0)^2}{2h^2}\right)$	权重随距离指数递减，更平滑

3. 计算相似度的流程
   - **数值数据标准化**

$$z_{ij} = \frac{x_{ij}}{\sigma_{ij}}$$

     - σ_{ij} 为第 j 个变量的标准差
     - 作用：消除不同量纲对距离的影响
     - **计算欧几里得距离**

$$d(Z_i, Z_0) = \sqrt{\sum_{j=1}^{p} |Z_{ij} - Z_{0j}|^2}$$

- **根据距离计算权重**

$$w_i = K(d(Z_i, Z_0))$$

  - K 为核函数
  - 距离越近 → 权重越大
- **预测**
  - **分类问题**：加权投票
  - **回归问题**：加权平均