K-NN算法的实现

纪元正

2019 年 7 月 21 日

- K-NN算法简介
  存在一个样本数据集合，也称为训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每一数据与所属分类对应的关系。输入没有标签的数据后，将新数据中的每个特征与样本集中数据对应的特征进行比较，提取出样本集中特征最相似数据（最近邻）的分类标签。一般来说，我们只选择样本数据集中前k个最相似的数据，这就是k近邻算法中k的出处，通常k是不大于20的整数。最后选择k个最相似数据中出现次数最多的分类作为新数据的分类。K-NN算法没有现实的训练过程，是“懒惰学习”的代表。

- 实例（sklearn实现）
  利用Day6的数据集，现想通过给定的成员的年龄（age）和估计工资（Estimated Salary）预测该成员是否会购买新款SUV （输出0/1）。
  首先计算预测点和k最近点之间的距离，这里k=5，计算出该对象对标记的对象之间的距离，确定其k近邻点，然后使用周边数量最多的最近邻点的类标签来确定该对象的类标签（是否购买SUV）。

- 步骤
  - 导入相关库(numpy matplotlib.pyplot pandas)
  - 导入数据集
  - 将数据划分成训练集和测试集

– 特征缩放

– 使用K-NN对训练集数据进行训练

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n_neighbors = 5, metric = ‘minkowski’, p = 2)

classifier.fit(X_train, y_train)

– 预测测试集

输出

[0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 1 0
0 0 0
0 0 1 0 0 0 0 1 0 0 1 0 1 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0
0 0
0 0 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 0 0 0 0 0 0 1 1]

– 生成混淆矩阵

[[65 3]

[8 24]]