**K-Nearest Neighbor 最邻近分类算法：**

简称KNN，最简单的机器学习算法之一，核心思想俗称“随大流”。是一种分类算法，基于实例的学习（instance-based learning）和懒惰学习（lazy learning）。懒惰学习：指的是在训练是仅仅是保存样本集的信息，直到测试样本到达是才进行分类决策。**核心想法：**

在距离空间里，如果一个样本的最接近的k个邻居里，绝大多数属于某个类别，则该样本也属于这个类别。

**范例：**假设，我们有这样一组电影数据：



**Python实现方式：**

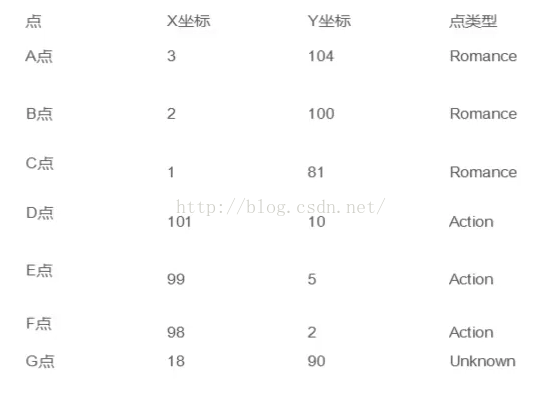
**[python]** [view plain](https://blog.csdn.net/jerry81333/article/details/53137919) [copy](https://blog.csdn.net/jerry81333/article/details/53137919)

1. **import** numpy as np
2. **from** sklearn **import** neighbors
4. knn = neighbors.KNeighborsClassifier() #取得knn分类器
5. data = np.array([[3,104],[2,100],[1,81],[101,10],[99,5],[98,2]]) #data对应着打斗次数和接吻次数
6. labels = np.array([1,1,1,2,2,2]) #labels则是对应Romance和Action
7. knn.fit(data,labels) #导入数据进行训练
8. **print**(knn.predict([18,90]))

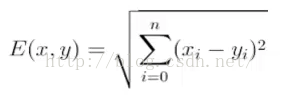
需要加载numpy，sklearn包，这两个都是机器学习或数据挖掘常用的包。

原理分析：

由数据可以看出，我们有上述6部电影的数据及分类，最后一部“未知”的是需要预测处于哪个分类中。然后，我们将数据中的“打斗次数”属性标记为X，“接吻次数”标记为Y，这样上述数据都能化为坐标轴中的一点：



之后便是将所有点与“未知”的点G进行距离计算，因为这个例子是二维的，因此这里我们使用E(x,y)=sqr((x2-x1)^2+(y2-y1)^2)，如果是多维的话，可以使用：



最后可得到结果，这里我省略到int：

a：20

b：18

c：19

d：115

e：117

f：118

因此可以看出，最近的三个点是ABC三点，而ABC三点都是Romance类型。

**选择方式：**

根据上述例子，如果ABC中三个电影分类有一个不是Romance怎么办。这里我们遵循少数服从多数的投票法则（majority-voting），让未知实例归类为最邻近样本中最多数的类别。

**其他距离衡量方式：**

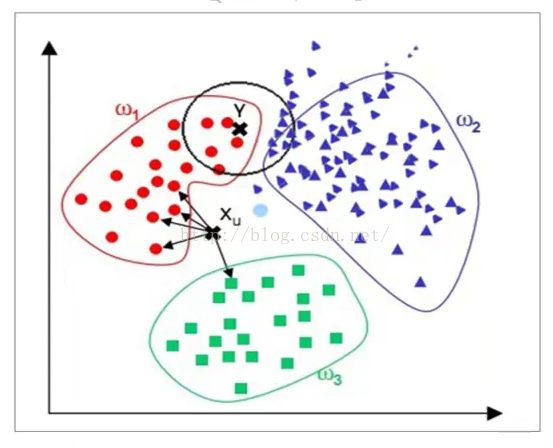
亦可使用余弦值（cos），相关度（correlation），曼哈顿距离等。

**优点：**

简单，易于实现，易于理解，通过对K的选择能一定程度上的具备丢噪音数据的健壮性（增大K值）

**缺点：**

需要大量的空间存储已知实例，算法复杂度高（需要比较所有已知实例）。当样本分布不平均时，比如其中一个样本实例过多，容易被归纳为实例多的样本，如下图Y点：



**解决方法：**

给距离增加权重，越近的距离权重越高，能一定程度的避免上述样本分布不平均的问题。