北风网
IBEIFENG.COM

人工智能之机器学习

K近邻算法（KNN）

上海育创网络科技股份有限公司

主讲人：赵翌臣

课程内容

- KNN算法介绍
- scikit-learn对KNN的实现方法
- 案例分析

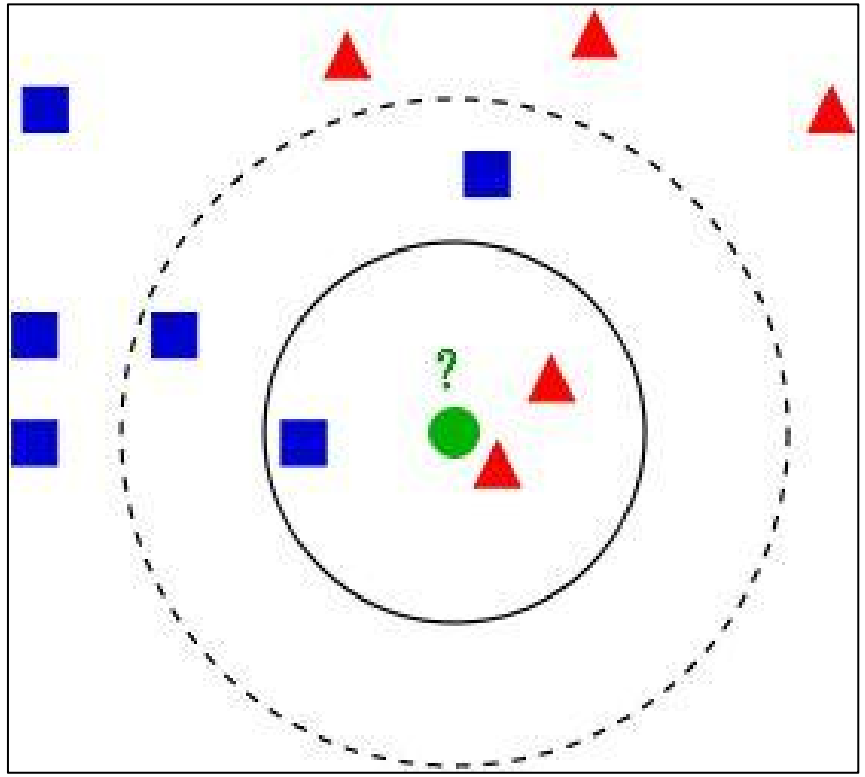
KNN直观解释

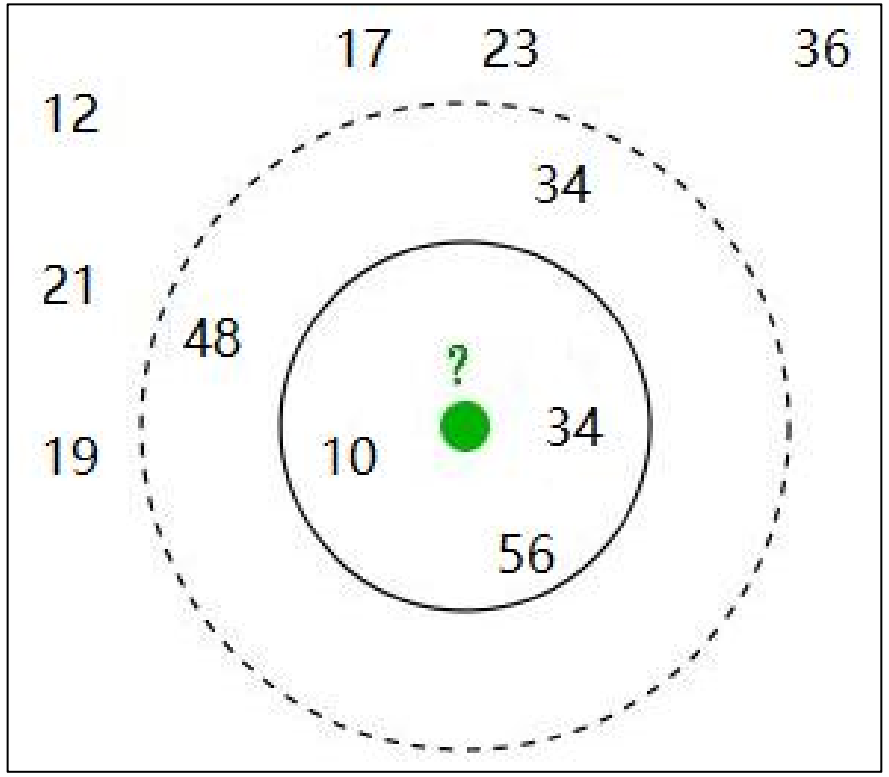

迟到党
睡神
LOL队友
深藏功与名
考神
女神
学霸
忧伤少年
妹子区
伪学霸区


讲 台
左护法
右护法
户外个人VIP休闲场所
高级阳光SPA专区
学霸区
VIP休息
VIP娱乐
高级避暑VIP专区
至尊VIP清凉楼道专区
免费苍蝇按摩区
超级VIP休闲专区

KNN直观解释

- K近邻法(k-NearestNeighbor)是一种很基本的机器学习方法，能做分类和回归任务





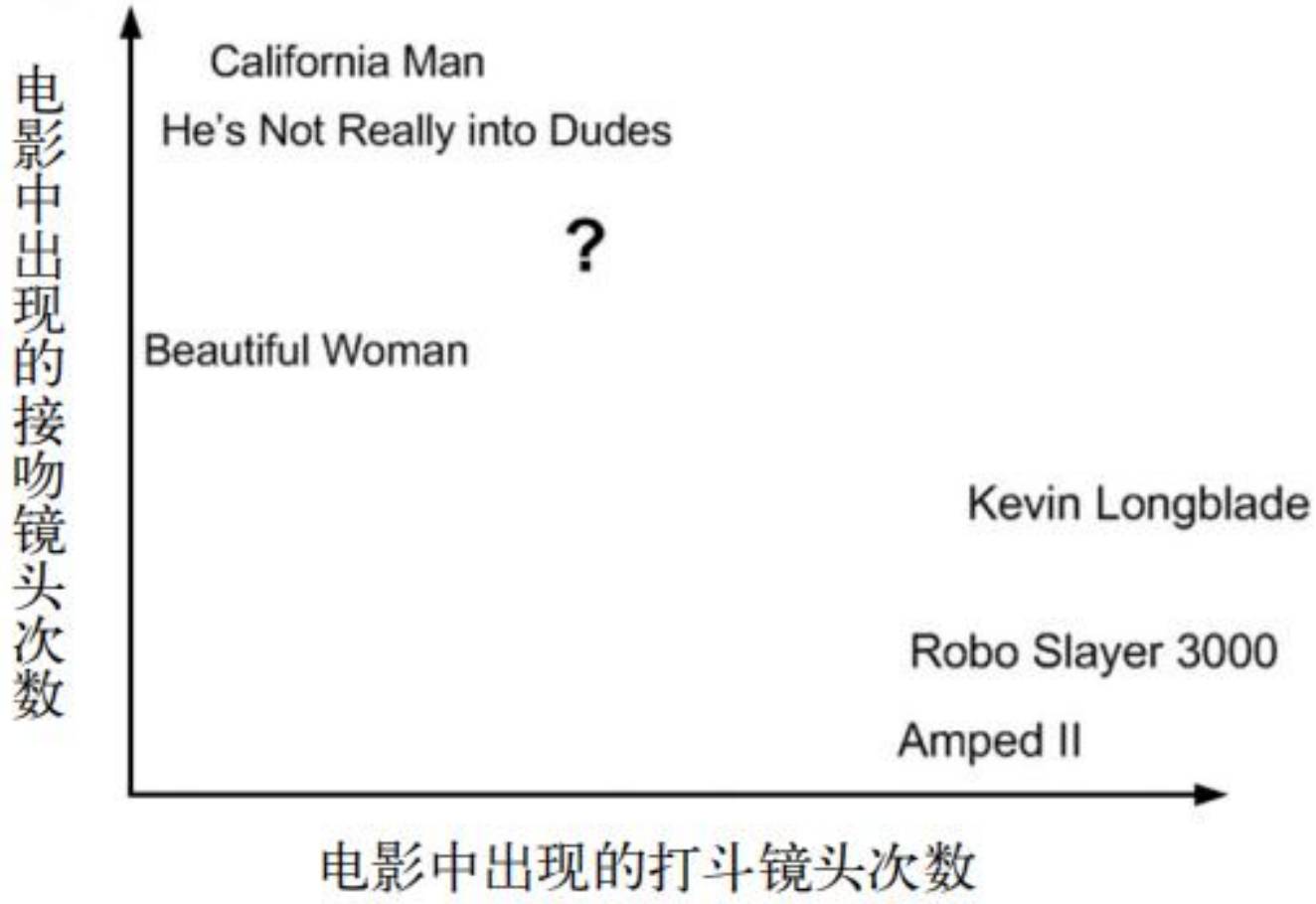
KNN直观解释

- 假设有训练集，请根据训练集训练一个KNN模型，预测最后一部影片的电影类型。

电影名称	打斗镜头	接吻镜头	电影类型
California Man	3	104	爱情片
He's Not Really into Dudes	2	100	爱情片
Beautiful Woman	1	81	爱情片
Kevin Longblade	101	10	动作片
Robo Slayer 3000	99	5	动作片
Amped ll	98	2	动作片
?	18	90	未知

KNN直观解释

- 第一步：将训练集中的所有样例画入坐标系，也将待测样例画入



KNN直观解释

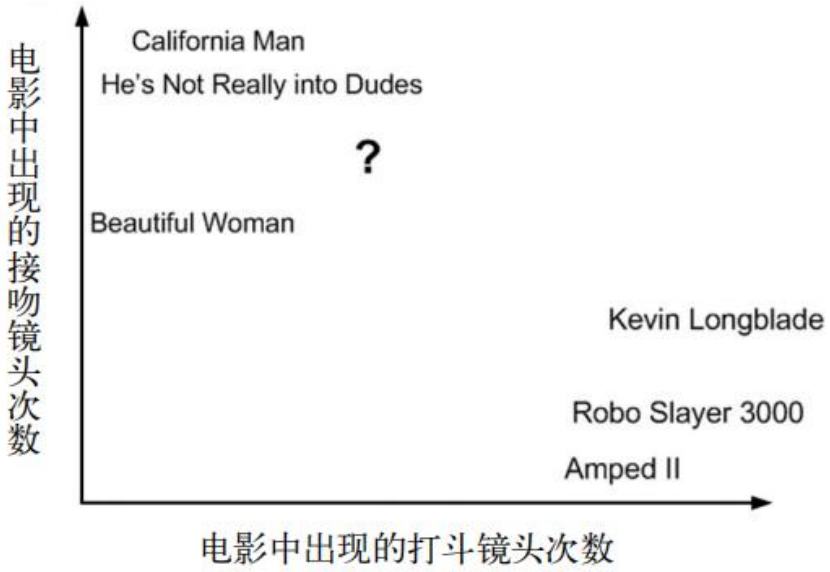
- 第二步：计算待测分类的电影与所有已知分类的电影的欧式距离

电影名称	与未知电影的距离
California Man	20.5
He's Not Really into Dudes	18.7
Beautiful Woman	19.2
Kevin Longblade	115.3
Robo Slayer 3000	117.4
Amped ll	118.9

- 第三步：将这些电影按照距离升序排序，取前k个电影，假设k=3，那么我们得到的电影依次是《He's Not Really Into Dudes》、《Beautiful Woman》和《California Man》。而这三部电影全是爱情片，因此我们判定未知电影是爱情片。

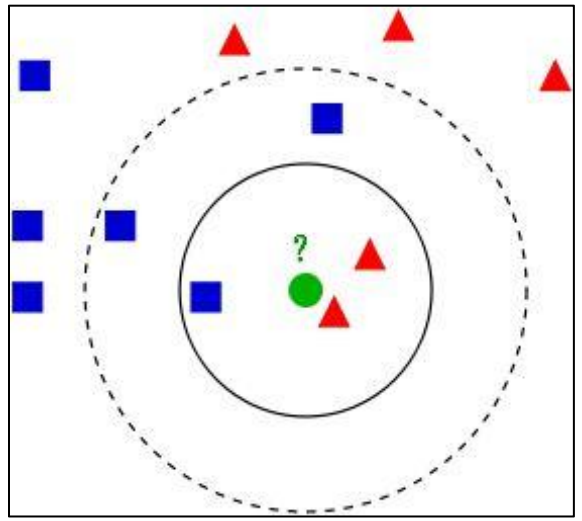
KNN的三个基本要素

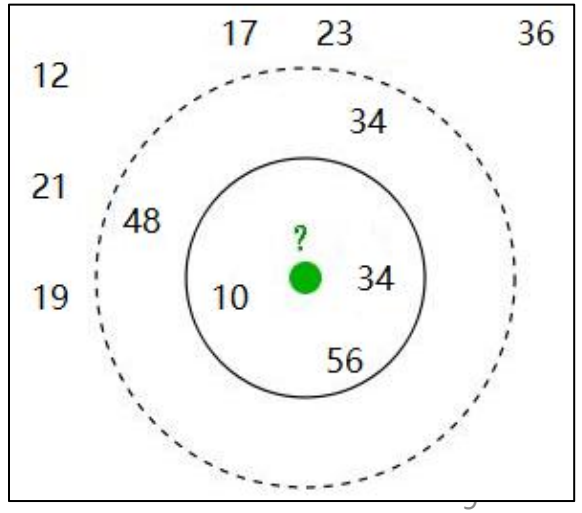
- kNN的三个基本要素：距离度量、k值的选择和决策规则
- （1）距离度量
  - 在引例中所画的坐标系，可以叫做特征空间。特征空间中两个实例点的距离是两个实例点相似程度的反应（距离越近，相似度越高）。kNN模型使用的距离一般是欧氏距离，但也可以是其他距离如：曼哈顿距离



KNN的三个基本要素

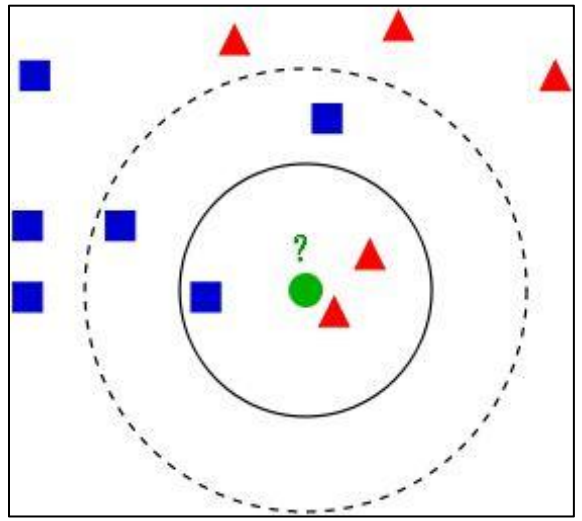
- （2）k值的选择
  - k值的选择会对kNN模型的结果产生重大影响。选择较大的k值，相当于用较大邻域中的训练实例进行预测，模型会考虑过多的邻近点实例点，甚至会考虑到大量已经对预测结果没有影响的实例点，会让预测出错；选择较小的k值，相当于用较小邻域中的训练实例进行预测，会使模型变得敏感（如果邻近的实例点恰巧是噪声，预测就会出错）。
  - 在应用中，k值一般取一个比较小的数值。通常采用一些验证方法来选取最优的k值。

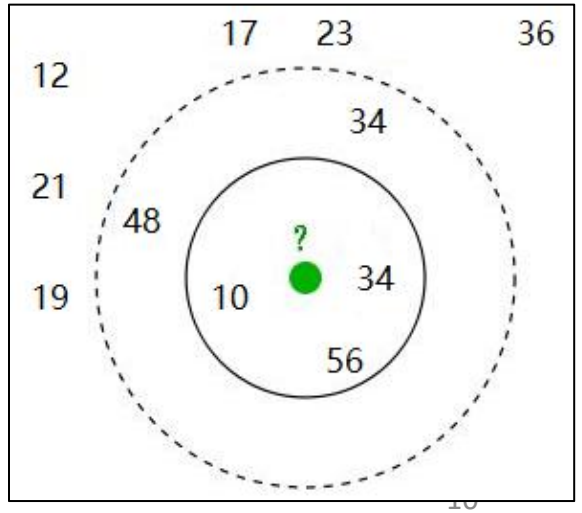




KNN的三个基本要素

- （3）决策规则
  - 分类：往往是多数表决，即由输入实例的k个邻近的训练实例中的多数类决定待测实例的类。或带权投票
  - 回归：取平均值。或带权取平均值





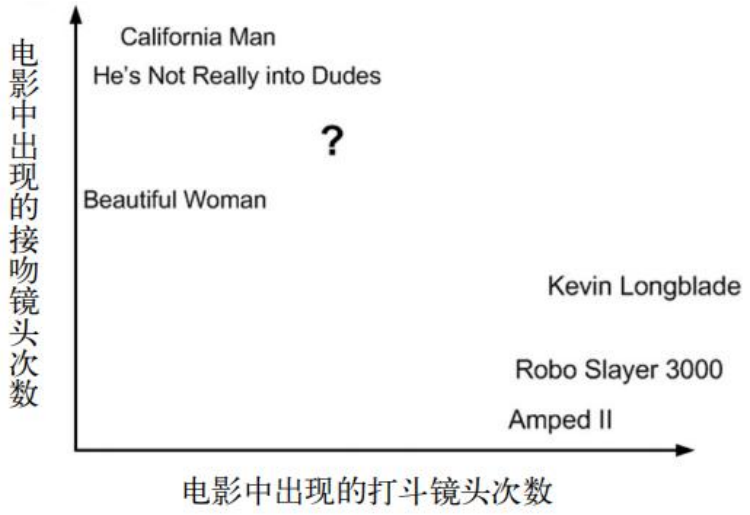
思考

- 训练KNN模型需要进行特征标准化吗?

编程——经典面试题KNN的实现

- 使用Python的二维List对KNN进行实现（使用等权投票），然后对未知影片的类型进行预测

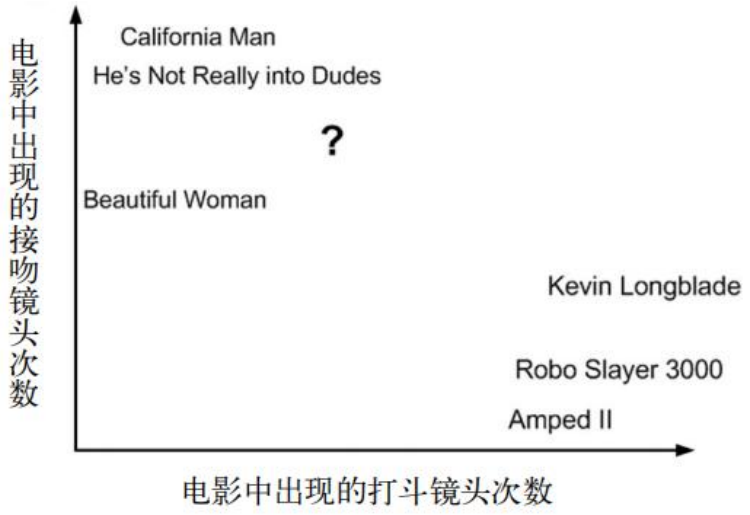
电影名称	打斗镜头	接吻镜头	电影类型
California Man	3	104	爱情片
He's Not Really into Dudes	2	100	爱情片
Beautiful Woman	1	81	爱情片
Kevin Longblade	101	10	动作片
Robo Slayer 3000	99	5	动作片
Amped ll	98	2	动作片
?	18	90	未知



编程——经典面试题KNN的实现

- 使用Python的二维List对KNN进行实现（使用**带权投票：所有邻居节点的投票权重与距离成反比**），然后对未知影片的类型进行预测

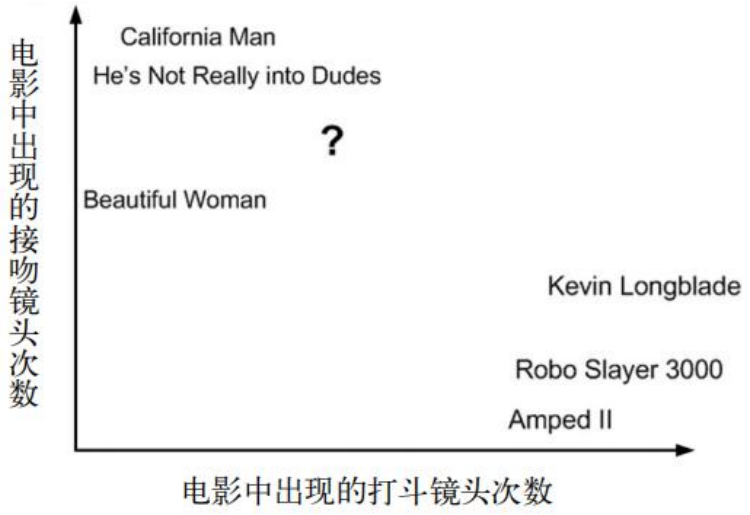
电影名称	打斗镜头	接吻镜头	电影类型
California Man	3	104	爱情片
He's Not Really into Dudes	2	100	爱情片
Beautiful Woman	1	81	爱情片
Kevin Longblade	101	10	动作片
Robo Slayer 3000	99	5	动作片
Amped ll	98	2	动作片
?	18	90	未知



编程——经典面试题KNN的实现

- 使用Python的二维List对KNN进行实现（使用等权均值），然后对未知影片的甜蜜指数进行预测

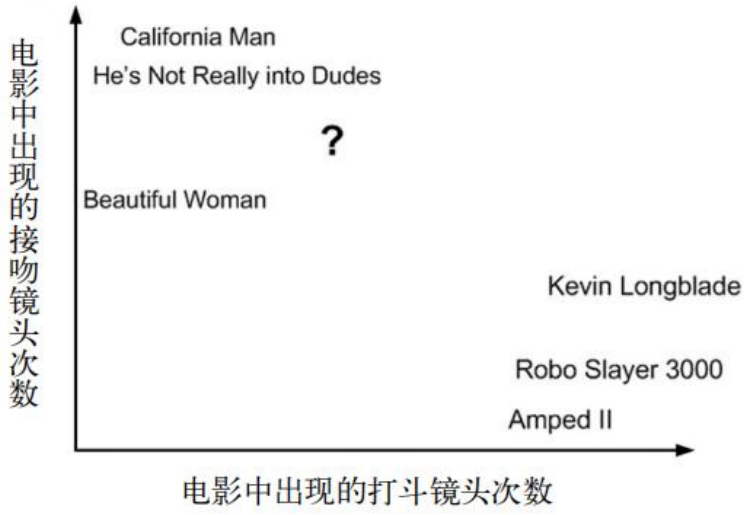
电影名称	打斗镜头	接吻镜头	甜蜜指数
California Man	3	104	98
He's Not Really into Dudes	2	100	93
Beautiful Woman	1	81	95
Kevin Longblade	101	10	16
Robo Slayer 3000	99	5	8
Amped II	98	2	7
?	18	90	未知



编程——经典面试题KNN的实现

- 使用Python的二维List对KNN进行实现（使用**带权均值**），然后对未知影片的甜蜜指数进行预测

电影名称	打斗镜头	接吻镜头	甜蜜指数
California Man	3	104	98
He's Not Really into Dudes	2	100	93
Beautiful Woman	1	81	95
Kevin Longblade	101	10	16
Robo Slayer 3000	99	5	8
Amped II	98	2	7
?	18	90	未知



思考

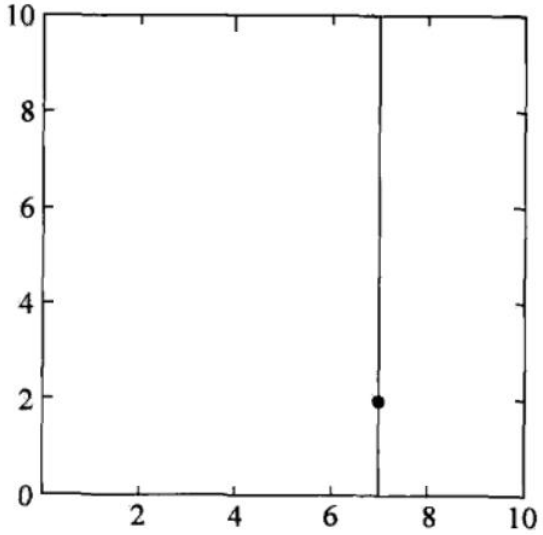
- 你认为这种实现方式的缺点是什么?

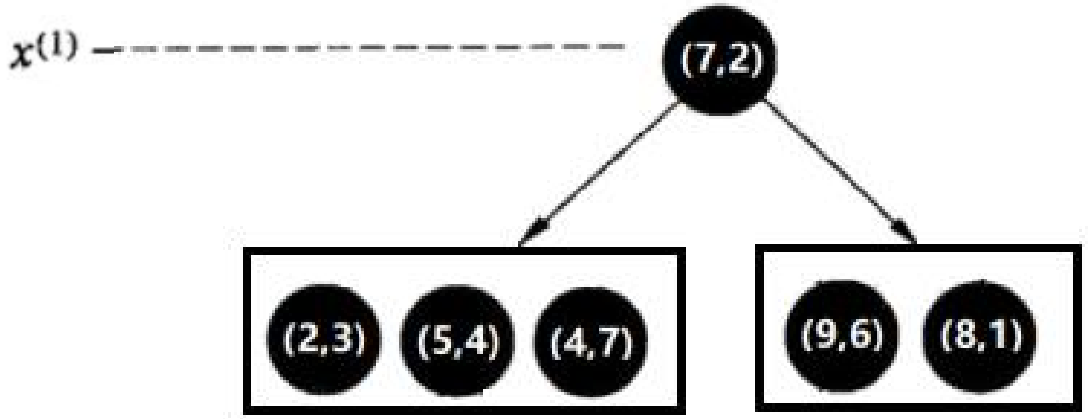
scikit-learn对KNN的实现方法

- 蛮力法(brute-force)
  - k近邻法最简单的实现方式是线性扫描，需要计算待测实例与每个实例的距离，在大数据上不可行。
- KD树(KDTree)
  - 为了提高k近邻搜索效率，考虑使用特殊的结构存储训练数据，以减少计算距离的次数，可以使用kd树（kd tree）方法。kd树分为两个过程——构造kd树（使用特殊结构存储训练集）、搜索kd树（减少搜索计算量）
- 球树(BallTree)

构造kd树

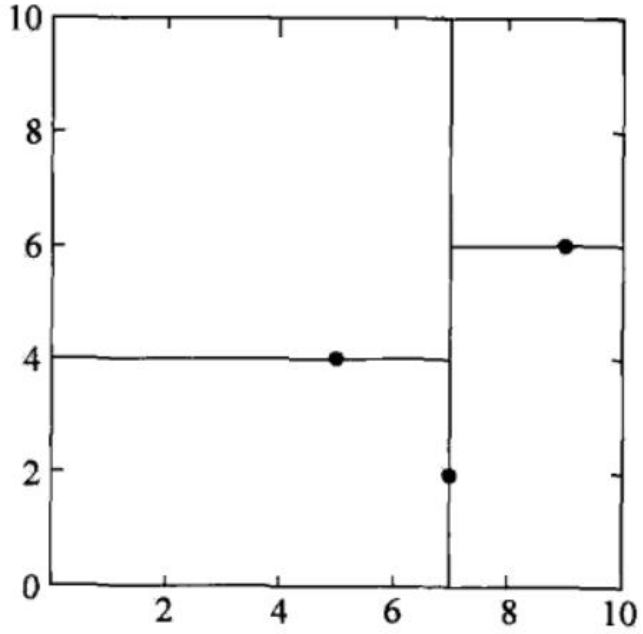
- 给定一个二维空间的数据集（含有标记的一般叫训练集，不含标记的一般叫数据集）：$T=\{(2,3)^T,(5,4)^T,(9,6)^T,(4,7)^T,(8,1)^T,(7,2)^T\}$ ，请画出：特征空间的划分过程、kd树的构造过程。
- 第一步：选择$x^{(1)}$轴，6个数据点的$x^{(1)}$坐标上的数字分别是2,5,9,4,8,7。取中位数7（不是严格意义的中位数，取较大的数），以$x^{(1)}=7$将特征空间分为两个矩形：

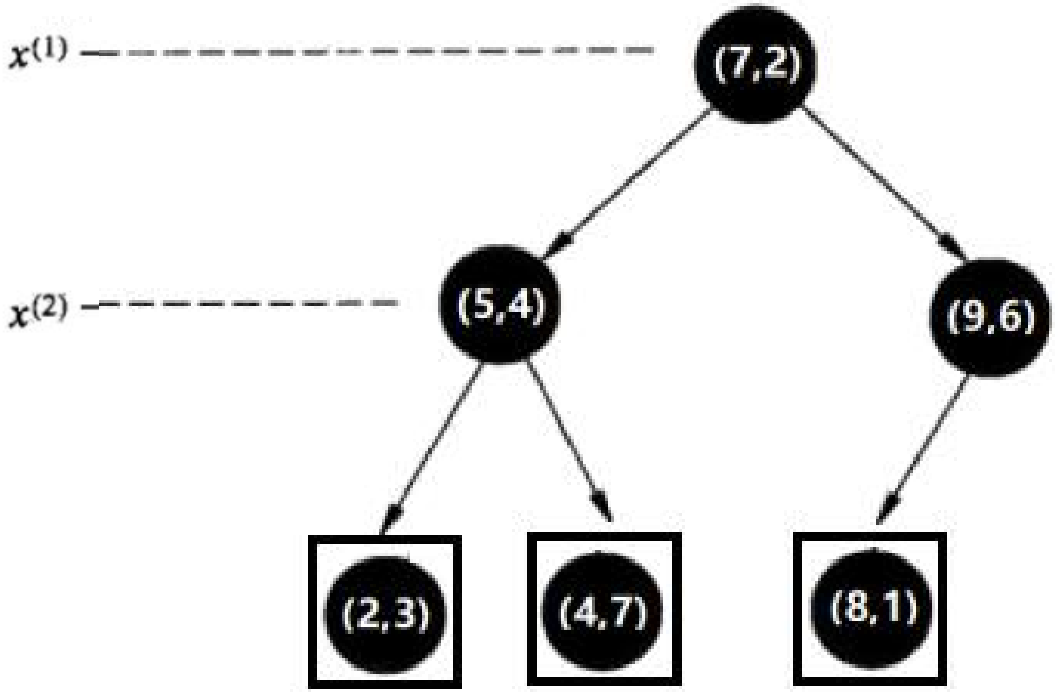




构造kd树

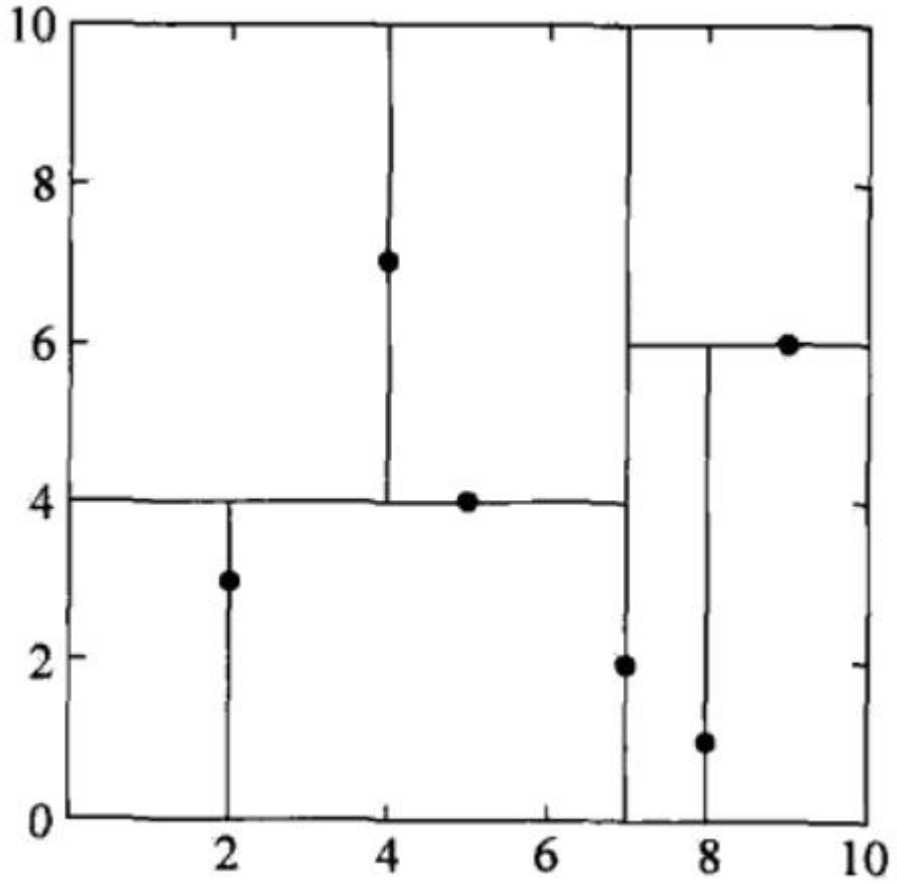
- 第二步：选择$x^{(2)}$轴，处理左子树，3个数据点的$x^{(2)}$坐标上的数字分别是3,4,7。取中位数4，以$x^{(2)}=4$将左子树对应的特征空间分为两个矩形；处理右子树，2个数据点的$x^{(2)}$坐标上的数字分别是6,1。取6，以$x^{(2)}=6$将右子树对应的特征空间分为两个矩形：

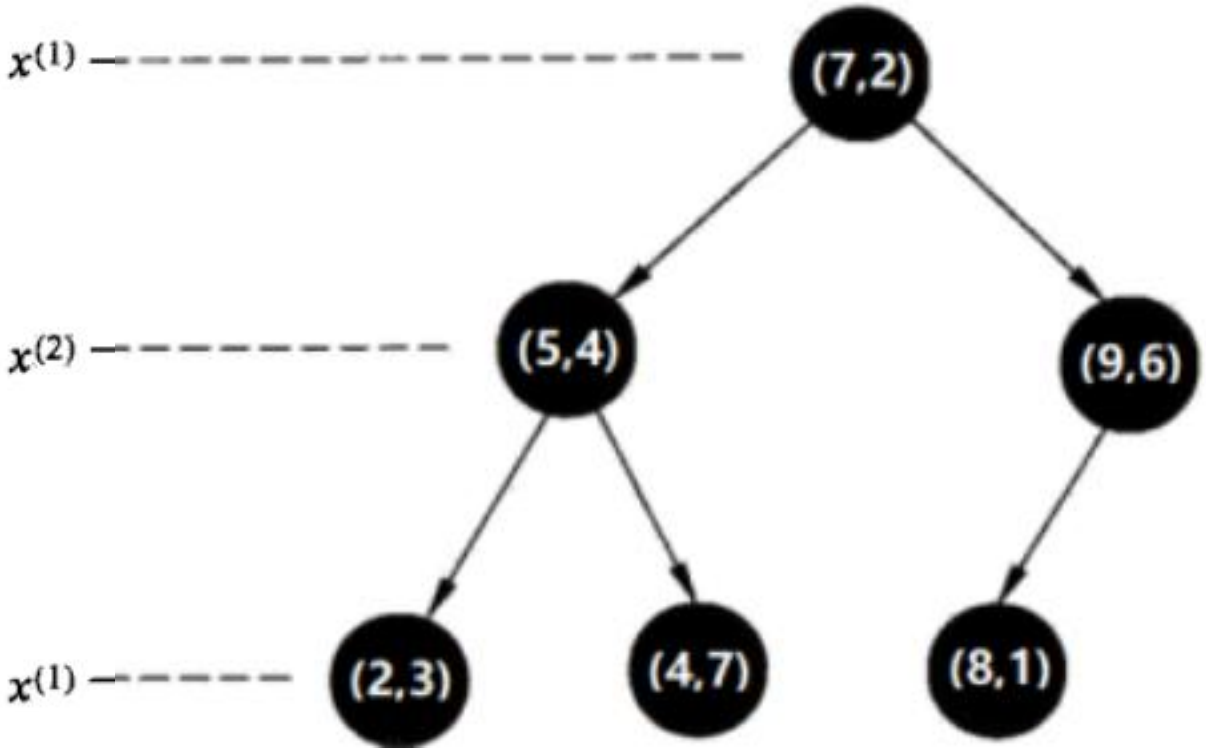




构造kd树

- 第三步：$x^{(1)}$轴，分别处理所有待处理的节点：

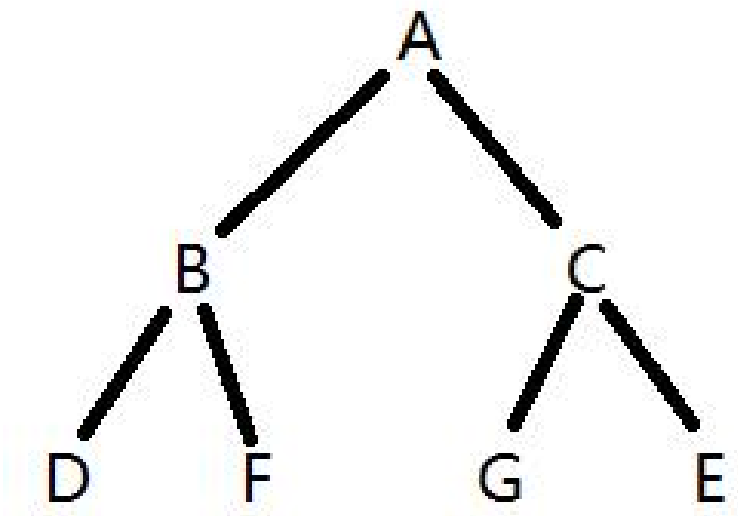


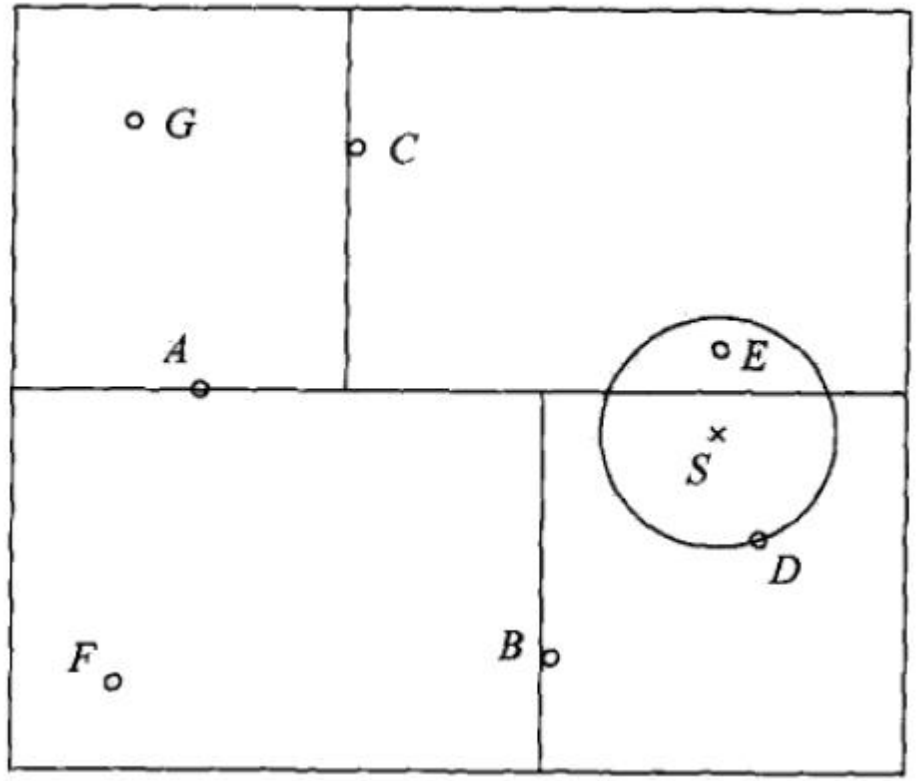


搜索kd树

例 3.3 给定一个如图 3.5 所示的 *kd* 树，根结点为 A，其子结点为 B，C 等. 树上共存储 7 个实例点；另有一个输入目标实例点 S，求 S 的最近邻.

解 首先在 *kd* 树中找到包含点 S 的叶结点 D（图中的右下区域），以点 D 作为近似最近邻. 真正最近邻一定在以点 S 为中心通过点 D 的圆的内部. 然后返回结点 D 的父结点 B，在结点 B 的另一子结点 F 的区域内搜索最近邻. 结点 F 的区域与圆不相交，不可能有最近邻点. 继续返回上一级父结点 A，在结点 A 的另一子结点 C 的区域内搜索最近邻. 结点 C 的区域与圆相交；该区域在圆内的实例点有点 E，点 E 比点 D 更近，成为新的最近邻近似. 最后得到点 E 是点 S 的最近邻. ■





思考

- 啰嗦了半天，kd树的搜索，究竟是如何减少搜索计算量的呢?

kd树的k近邻搜索

- 最近邻
  - 在最近邻搜索（k=1）中，首先存储一个自认为是“当前最近点”的节点，然后在搜索过程中，找到更近的就替换掉。
- K近邻
  - 在KD树搜索最近邻的基础上，我们选择到了第一个最近邻样本，就把它置为已选。在第二轮中，我们忽略置为已选的样本，重新选择最近邻，这样跑k次，就得到了目标的K个最近邻，然后根据多数表决法，如果是KNN分类，预测为K个最近邻里面有最多类别数的类别。如果是KNN回归，用K个最近邻样本输出的平均值作为回归预测值

KNN模型的优缺点

- 优点
  - 思想简单，能做分类和回归
  - 惰性学习，无需训练（蛮力法），KD树的话，则需要建树
  - 对异常点不敏感
- 缺点
  - 计算量大、速度慢
  - 样本不平衡的时候，对稀有类别的预测准确率低
  - KD树，球树之类的模型建立需要大量的内存
  - 相比决策树模型，KNN模型可解释性不强

编程——使用KNN模型对鸢尾花进行分类

- 基于鸢尾花数据进行分类模型构建，使用KNN算法进行构建
- 数据来源：http://archive.ics.uci.edu/ml/datasets/Iris

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1319181

Attribute Information:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm
5. class:
-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

```
5.1,3.8,1.9,0.4,Iris-setosa
4.8,3.0,1.4,0.3,Iris-setosa
5.1,3.8,1.6,0.2,Iris-setosa
4.6,3.2,1.4,0.2,Iris-setosa
5.3,3.7,1.5,0.2,Iris-setosa
5.0,3.3,1.4,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
```

附：闵可夫斯基距离

- 假设有两个样本点x_1，x_2，它们两者间的闵可夫斯基距离L_p定义为

$$L_p(x_1,x_2)=\left(\sum_{l=1}^{n}|x_i^{(l)}-x_j^{(l)}|^{p}\right)^{\frac{1}{p}}$$

- 当p=1时，称为曼哈顿距离（Manhattan distance），即

$$L_p(x_1,x_2)=\sum_{l=1}^{n}|x_i^{(l)}-x_j^{(l)}|$$

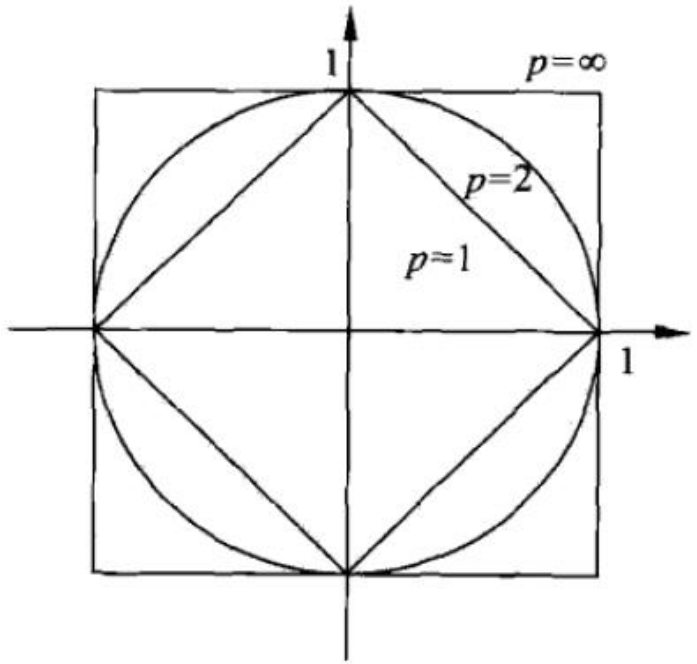
- 当p=2时，称为欧氏距离（Euclidean distance），即

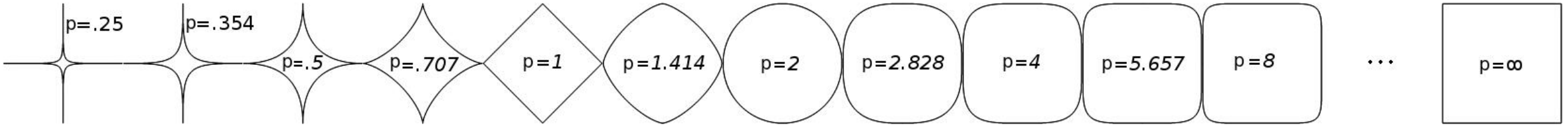
$$L_p(x_1,x_2)=\sqrt{\sum_{l=1}^{n}(x_i^{(l)}-x_j^{(l)})^2}$$

附：闵可夫斯基距离

- 当p=∞时，称为切比雪夫距离

$$L_p(x_1, x_2) = \max_l | x_i^{(l)} - x_j^{(l)} |$$





PS：带权平均数（广义线性平均值）

假设有a_1, a_2, a_3三个数，对应的权重分别是p_1, p_2, p_3

则$\overline{a} = \dfrac{p_1 a_1 + p_2 a_2 + p_3 a_3}{p_1 + p_2 + p_3}$


北风网
IBEIFENG.COM