Python KNN算法

数据分析与数据挖掘

3.k-近邻算法电影分类

应用：

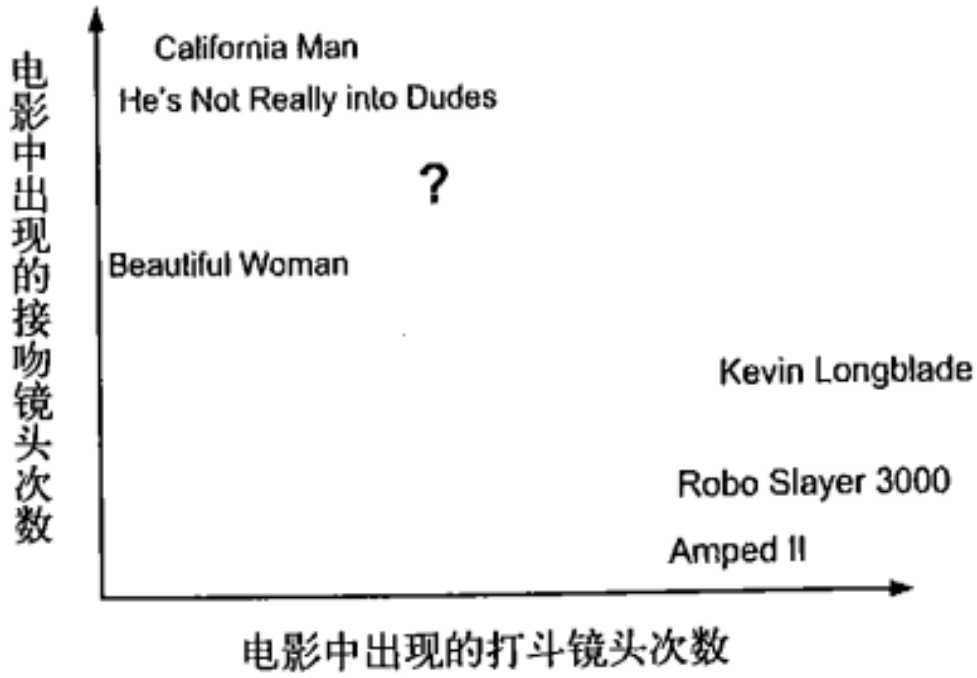


图2-1　使用打斗和接吻镜头数分类电影

3.k-近邻算法电影分类

表2-1　每部电影的打斗镜头数、接吻镜头数以及电影评估类型

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 II	98	2	动作片
?	18	90	未知

表2-2　已知电影与未知电影的距离

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 II	118.9

4.用K-均值聚类来探索顾客细分--算法流程

Kmeans 聚类原理：K-means 算法是最为经典的基于**划分**的聚类方法，是十大经典ML算法之一。

K-means基本思想：以空间中k个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。

4.用K-均值聚类来探索顾客细分--算法流程

假设要把样本集分为**k个类别**，**算法描述**如下：

(1)从N个点随机选取K个点作为质心；

(2)对剩余的每个点测量其到每个质心的距离，并把它归到最近的质心的类；

(3)重新计算已经得到的各个类的质心；

(4)迭代2～3步直至新的质心与原质心相等或小于指定阈值，算法结束。

算法的**最大优势**：简洁&快速。

算法的**关键**：初始中心的选择&距离公式。

4.用K-均值聚类来探索顾客细分--算法流程

k-means算法的优点：

（1）k-means算法是解决聚类问题的一种经典算法，算法**简单、快速**。

（2）对处理大数据集，该算法是相对可伸缩的和高效率的，因为它的复杂度大约是O(nkt)，其中n是所有对象的数目，k是簇的数目,t是迭代的次数。通常k<<n。这个算法通常**局部收敛**。

（3）算法尝试找出使平方误差函数值最小的k个划分。当簇是密集的、球状或团状的，且簇与簇之间区别明显时，聚类效果较好。

k-means算法的缺点：

（1）k-平均方法只有在**簇的平均值**被定义的情况下才能使用，且对有些分类属性的数据不适合。

（2）要求用户必须事先给出要生成的**簇的数目k**。

（3）对**初值敏感**，对于不同的初始值，可能会导致不同的聚类结果。

（4）不适合于发现非凸面形状的簇，或者大小差别很大的簇。

（5）对于"噪声"和孤立点**数据敏感**，少量的该类数据能够对平均值产生极大影响。

4.用K-均值聚类来探索顾客细分--算法流程

4.用K-均值聚类来探索顾客细分--应用分析

核心：能够识别不同类型的客户，然后知道如何找到更多这样的人，获得更多的客户！

```
1 import pandas as pd
2 df_offers = pd.read_excel("./WineKMC.xlsx", sheetname=0)
3 df_offers.columns = ["offer_id", "campaign", "varietal", "min_qty", "discount", "origin", "past_peak"]
4 df_offers.head()
```

	offer_id	campaign	varietal	min_qty	discount	origin	past_peak
0	1	January	Malbec	72	56	France	False
1	2	January	Pinot Noir	72	17	France	False
2	3	February	Espumante	144	32	Oregon	True
3	4	February	Champagne	72	48	France	True
4	5	February	Cabernet Sauvignon	144	44	New Zealand	True

表：数据集

4.用K-均值聚类来探索顾客细分--应用分析

```
df_transactions = pd.read_excel("./WineKMC.xlsx", sheetname=1)
df_transactions.columns = ["customer_name", "offer_id"]
df_transactions['n'] = 1
df_transactions.head()
```

	customer_name	offer_id	n
0	Smith	2	1
1	Smith	24	1
2	Johnson	17	1
3	Johnson	24	1
4	Johnson	26	1

表：交易层面的数据

offer_id	customer_name	cluster	x	y
0	Adams	2	-1.007580	0.108215
1	Allen	4	0.287539	0.044715
2	Anderson	1	0.392032	1.038391
3	Bailey	2	-0.699477	-0.022542
4	Baker	3	-0.088183	-0.471695

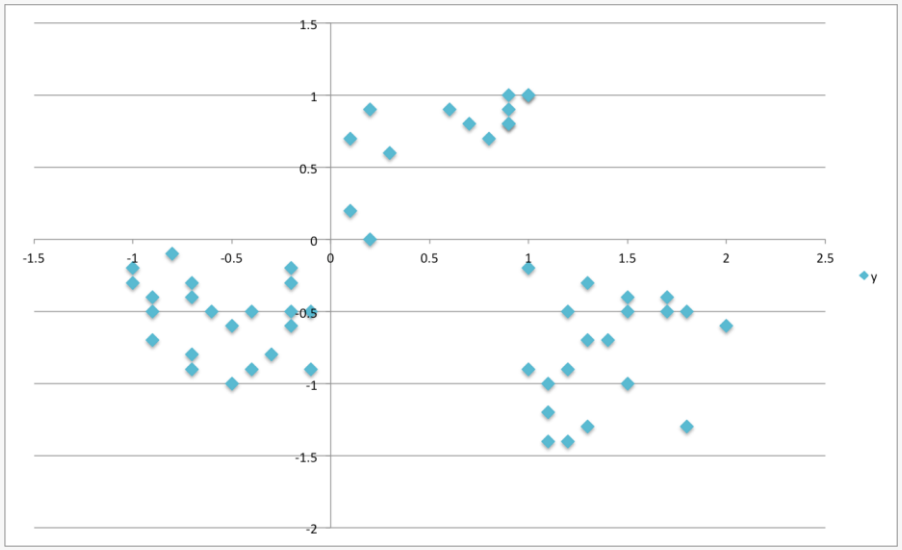
表：聚类结果

4.用K-均值聚类来探索顾客细分--应用分析

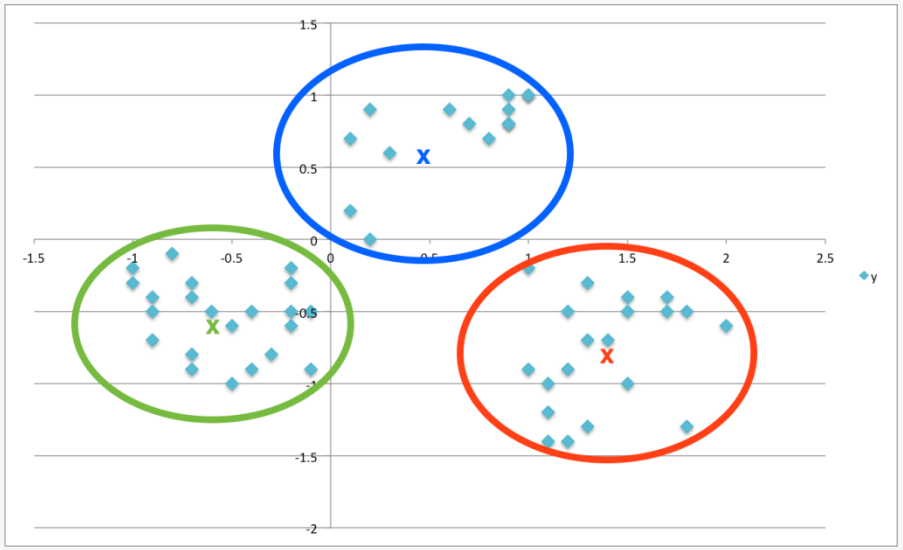
is_4	varietal	count
False	Champagne	45
	Espumante	40
	Prosecco	37
	Pinot Noir	37
	Malbec	17
	Pinot Grigio	16
	Merlot	8
	Cabernet Sauvignon	6
	Chardonnay	4
True	Champagne	36
	Cabernet Sauvignon	26
	Malbec	15
	Merlot	12
	Chardonnay	11
	Pinot Noir	7
	Prosecco	6
	Pinot Grigio	1

表：第四类数据分析

4.用K-均值聚类来探索顾客细分--应用分析



图：数据集



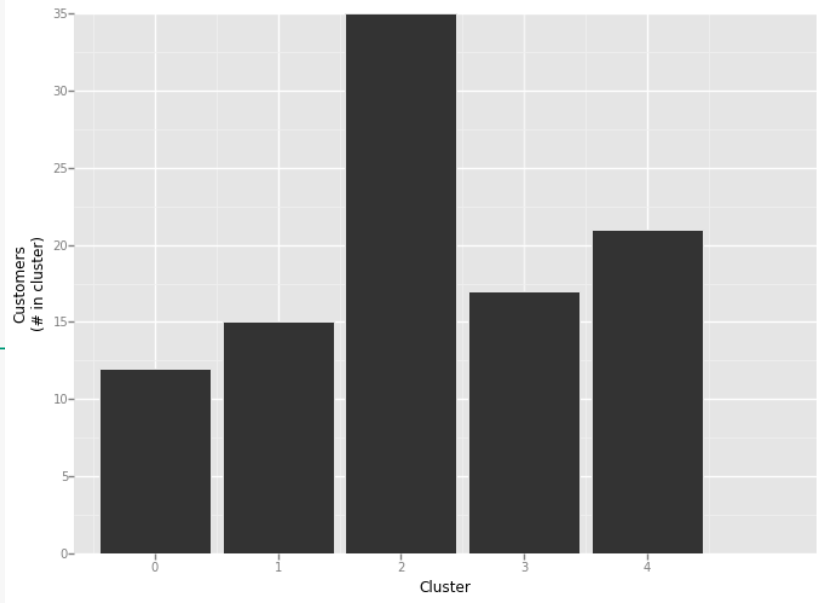
图：数据集的预期分类

4.用K-均值聚类来探索顾客细分--应用分析

```
from sklearn.cluster import KMeans

cluster = KMeans(n_clusters=5)
# slice matrix so we only include the 0/1 indicator columns in the clustering
matrix['cluster'] = cluster.fit_predict(matrix[matrix.columns[2:]])
matrix.cluster.value_counts()
2    32
1    22
4    20
0    15
3    11
dtype: int64
```

图：创建簇（程序分类）

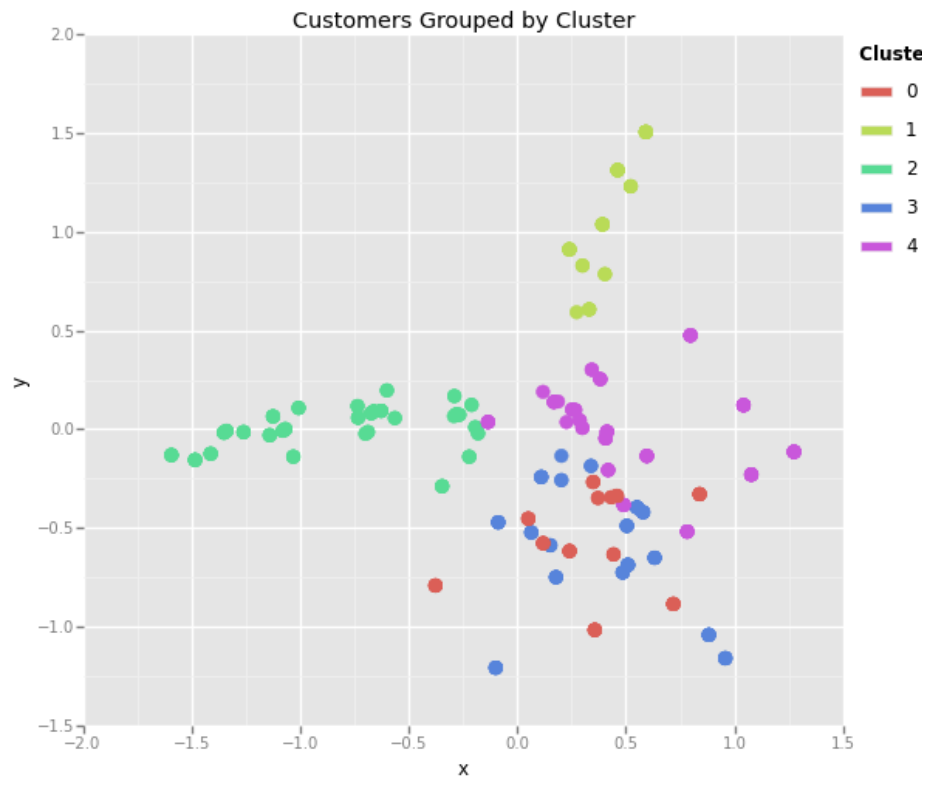


图：创建簇

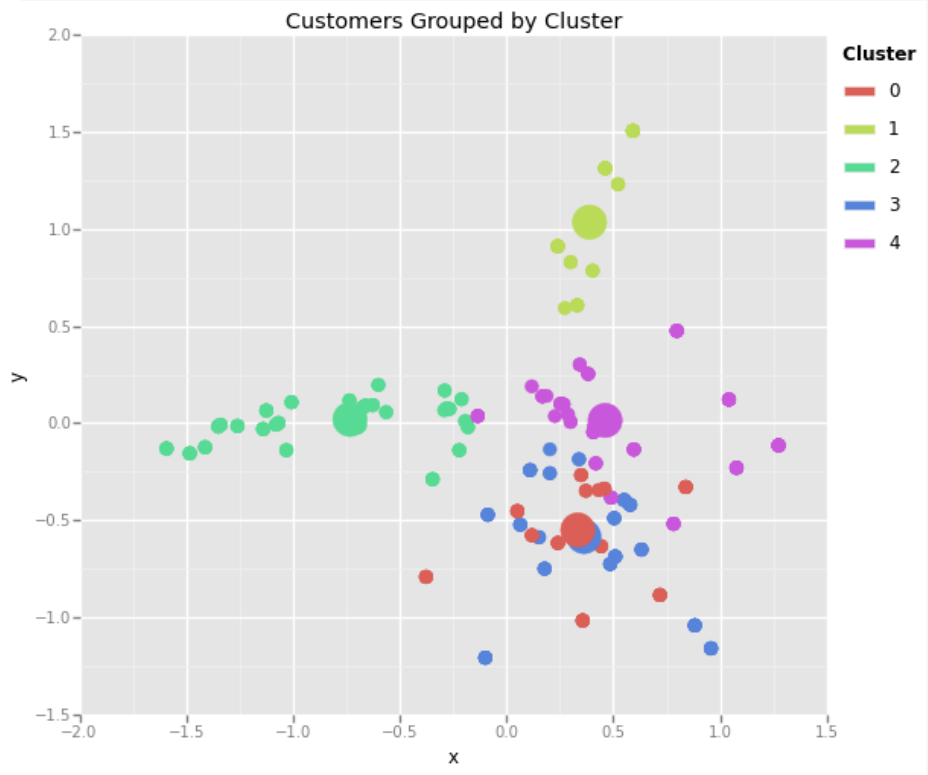
论文中，进行分类的**记录数**至少是**类别数**的 7 倍。

原则：记录数要足够多。

4.用K-均值聚类来探索顾客细分--应用分析



图：数据集的实际分类



图：数据集的实际分类
（标明簇心）