

星伯反 (Winnipeg) · 謝廷：江巴台人..

$$(X, Y) \Rightarrow Z:$$

设最小支持度为50%, 最小可信度为 50...

提升度 (lift)：物品集A的出现对物品集...

Leverage 与 Conviction的作用和lift类似...

概念搞定之后，来看一看怎么用Python...

设置支持度 (support) 来选择频繁项集.

计算规则

所有指标的计算公式:

再来加载一份商品数据集

电影数据集关联分析

- 支持度: 交易中包含{X、Y、Z}的可能性
- 置信度: 包含{X、Y}的交易中也包含Z的条件概率

设最小支持度为50%, 最小可信度为 50%, 则可得到:

- $A \Rightarrow C$ (50%, 66.6%)
- $C \Rightarrow A$ (50%, 100%)

若关联规则 $X \rightarrow Y$ 的支持度和置信度分别大于或等于用户指定的最小支持率 minsupport 和最小置信度 minconfidence , 则称关联规则 $X \rightarrow Y$ 为强关联规则, 否则称关联规则 $X \rightarrow Y$ 为弱关联规则。

提升度 (lift)：物品集A的出现对物品集B的出现概率发生了多大的变化

- $\text{lift}(A \Rightarrow B) = \text{confidence}(A \Rightarrow B) / \text{support}(B) = p(A|B)/p(B)$
- 现在有一“1000”个消费者，有“500”人购买了茶叶，其中有“450人同时”购买了咖啡，另“50人”没有。由于“ $\text{confidence}(\text{茶叶} \Rightarrow \text{咖啡}) = 450/500 = 90\%$ ”，由此可能会让人喜欢喝茶的人往往喜欢喝咖啡。但如果另外没有购买茶叶的“500人”，其中同样有“450人”购买了咖啡，同样是很高的“置信度90%”，由此，得到不爱喝茶的也爱喝咖啡。这样看来，其实是否购买咖啡，与有没有购买茶叶并没有关联，两者是相互独立的，其“提升度 $90\% / (450 + 450) / 1000 = 1$ ”。

由此可见, lift正是弥补了confidence的这一缺陷, if lift=1,X与Y独立, X对Y出现的可能性没有提升作用, 其值越大(lift>1), 则表明X对Y的提升程度越大, 也表明关联性越强。

X	1	1	1	1	0	0	0	0	rule	Support	Lift
Y	1	1	0	0	0	0	0	0	$X \Rightarrow Y$	25%	2.00
Z	0	1	1	1	1	1	1	1	$X \Rightarrow Z$	37.50%	0.86
									$Y \Rightarrow Z$	12.50%	0.57

Leverage 与 Conviction的作用和lift类似，都是值越大代表越关联

- Leverage : $P(A,B)-P(A)P(B)$
- Conviction: $P(A)P(!B)/P(A,!B)$

概念搞定之后，来看看怎么用Python玩玩关联规则

使用mixtend工具包得出频繁项集与规则

- `pip install mlxtend`

```
1 import pandas as pd
2 from mlxtend.frequent_patterns import apriori
3 from mlxtend.frequent_patterns import association_rules
```

自定义一份购物数据集

```
1 data = {'ID': [1,2,3,4,5,6],
2         'Onion': [1,0,0,1,1,1],
3         'Potato': [1,1,0,1,1,1],
4         'Burger': [1,1,0,0,1,1],
5         'Milk': [0,1,1,1,0,1],
6         'Beer': [0,0,1,0,1,0]}
```

```
1 df = pd.DataFrame(data)
2 df = df[['ID', 'Onion', 'Potato', 'Burger', 'Milk', 'Beer' ]]
```

	ID	Onion	Potato	Burger	Milk	Beer
0	1	1	1	1	0	0
1	2	0	1	1	1	0
2	3	0	0	0	1	1
3	4	1	1	0	1	0
4	5	1	1	1	0	1
5	6	1	1	1	1	0

设置支持度 (*support*) 来选择频繁项集。

- 选择最小支持度为50%
- `apriori(df, min_support=0.5, use_colnames=True)`

```
1 itemsets = apriori(df[['Onion', 'Potato', 'Burger', 'Milk', 'Beer' ]], min_support=0.50, use_c
```

	support	itemsets
0	0.666667	(Onion)
1	0.833333	(Potato)
2	0.666667	(Burger)
3	0.666667	(Milk)
4	0.666667	(Potato, Onion)
5	0.500000	(Burger, Onion)
6	0.666667	(Burger, Potato)
7	0.500000	(Milk, Potato)
8	0.500000	(Burger, Potato, Onion)

返回的3种项集均是支持度 $\geq 50\%$

计算规则

- `association_rules(df, metric='lift', min_threshold=1)`
- 可以指定不同的衡量标准与最小阈值

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Potato)	(Onion)	0.833333	0.666667	0.666667	0.80	1.200	0.111111	1.666667
1	(Onion)	(Potato)	0.666667	0.833333	0.666667	1.00	1.200	0.111111	inf
2	(Burger)	(Onion)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333
3	(Onion)	(Burger)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333
4	(Burger)	(Potato)	0.666667	0.833333	0.666667	1.00	1.200	0.111111	inf
5	(Potato)	(Burger)	0.833333	0.666667	0.666667	0.80	1.200	0.111111	1.666667
6	(Burger, Potato)	(Onion)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333
7	(Burger, Onion)	(Potato)	0.500000	0.833333	0.500000	1.00	1.200	0.083333	inf
8	(Potato, Onion)	(Burger)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333
9	(Burger)	(Potato, Onion)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333
10	(Potato)	(Burger, Onion)	0.833333	0.500000	0.500000	0.60	1.200	0.083333	1.250000
11	(Onion)	(Burger, Potato)	0.666667	0.666667	0.500000	0.75	1.125	0.055556	1.333333

返回的是各个的指标的数值，可以按照感兴趣的指标排序观察,但具体解释还得参考实际数据的含义。

```
1 | rules [ (rules['lift'] >1.125) & (rules['confidence']> 0.8) ]
```


	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
1	(Onion)	(Potato)	0.666667	0.833333	0.666667	1.0	1.2	0.111111	inf
4	(Burger)	(Potato)	0.666667	0.833333	0.666667	1.0	1.2	0.111111	inf
7	(Burger, Onion)	(Potato)	0.500000	0.833333	0.500000	1.0	1.2	0.083333	inf

这几条结果就比较有价值了：

- （洋葱和马铃薯）（汉堡和马铃薯）可以搭配着来卖
- 如果洋葱和汉堡都在购物篮中，顾客买马铃薯的可能性也比较高，如果他篮子里面没有，可以推荐一下。

所有指标的计算公式：

measure	definition	interpretation	
support	$\text{supp}_T(A \Rightarrow B)$	$P(A \cap B)$	
confidence	$\frac{\text{supp}_T(A \Rightarrow B)}{\text{supp}_T(A)}$	$P(B / A)$	
lift	$\frac{\text{conf}_T(A \Rightarrow B)}{\text{supp}_T(B)}$	$\frac{P(B / A)}{P(B)}$	
leverage	$\text{supp}_T(A \Rightarrow B) - \text{supp}_T(A) \text{supp}_T(B)$	$P(A \cap B) - P(A) P(B)$	
conviction	$\frac{1 - \text{supp}_T(B)}{1 - \text{conf}_T(A \Rightarrow B)}$	$\frac{1 - P(B)}{1 - P(B / A)}$	
measure	min value, incompatibility	value at independence	max value, logical rule
support	0	$\text{supp}_T(A) \text{supp}_T(B)$	$\text{supp}_T(A)$
confidence	0	$\text{supp}_T(B)$	1
lift	0	1	$\frac{1}{\text{supp}_T(B)}$
leverage	$-\text{supp}_T(A) \text{supp}_T(B)$	0	$\text{supp}_T(A) (1 - \text{supp}_T(B))$
conviction	$1 - \text{supp}_T(B)$	1	∞

<https://blog.csdn.net/hangyu>

<https://blog.csdn.net/tangyudi>

再来加载一份商品数据集

此处需要大家注意如何进行数据预处理，使用工具包一定得按照人家要求来才可以！

```
1 | retail_shopping_basket = {'ID':[1,2,3,4,5,6],
2 |                           'Basket':['Beer', 'Diaper', 'Pretzels', 'Chips', 'Aspirin'],
3 |                           ['Diaper', 'Beer', 'Chips', 'Lotion', 'Juice', 'BabyFood',
4 |                           ['Soda', 'Chips', 'Milk'],
5 |                           ['Soup', 'Beer', 'Diaper', 'Milk', 'IceCream'],
6 |                           ['Soda', 'Coffee', 'Milk', 'Bread'],
7 |                           ['Beer', 'Chips']
8 |                           ]
9 | }
```

```
1 | retail = pd.DataFrame(retail_shopping_basket)
2 | retail = retail[['ID', 'Basket']]
3 | pd.options.display.max_colwidth=100
```

	ID	Basket
0	1	[Beer, Diaper, Pretzels, Chips, Aspirin]
1	2	[Diaper, Beer, Chips, Lotion, Juice, BabyFood, Milk]
2	3	[Soda, Chips, Milk]
3	4	[Soup, Beer, Diaper, Milk, IceCream]
4	5	[Soda, Coffee, Milk, Bread]
5	6	[Beer, Chips]

<https://blog.csdn.net/tangyudi>

数据集中都是字符串组成的，需要转换成数值编码

```
1 | retail_id = retail.drop('Basket', 1)
```

	ID
0	1
1	2
2	3
3	4
4	5
5	6

```
1 | retail_Basket = retail.Basket.str.join(',')
```



举报



```
0      Beer,Diaper,Pretzels,Chips,Aspirin
1  Diaper,Beer,Chips,Lotion,Juice,BabyFood,Milk
2                        Soda,Chips,Milk
3      Soup,Beer,Diaper,Milk,IceCream
4                        Soda,Coffee,Milk,Bread
5                        Beer,Chips
Name: Basket, dtype: object
```

```
1 | retail_Basket = retail_Basket.str.get_dummies(',')
```

	Aspirin	BabyFood	Beer	Bread	Chips	Coffee	Diaper	IceCream	Juice	Lotion	Milk	Pretzels	Soda	Soup
0	1	0	1	0	1	0	1	0	0	0	0	1	0	0
1	0	1	1	0	1	0	1	0	1	1	1	0	0	0
2	0	0	0	0	1	0	0	0	0	0	0	1	0	0
3	0	0	1	0	0	0	1	1	0	0	1	0	0	1
4	0	0	0	1	0	1	0	0	0	0	1	0	1	0
5	0	0	1	0	1	0	0	0	0	0	0	0	0	0

```
1 | retail = retail_id.join(retail_Basket)
```

	ID	Aspirin	BabyFood	Beer	Bread	Chips	Coffee	Diaper	IceCream	Juice	Lotion	Milk	Pretzels	Soda	Soup
0	1	1	0	1	0	1	0	1	0	0	0	0	1	0	0
1	2	0	1	1	0	1	0	1	0	1	1	1	0	0	0
2	3	0	0	0	0	1	0	0	0	0	0	1	0	1	0
3	4	0	0	1	0	0	0	1	1	0	0	1	0	0	1
4	5	0	0	0	1	0	1	0	0	0	0	1	0	1	0
5	6	0	0	1	0	1	0	0	0	0	0	0	0	0	0

```
1 | frequent_itemsets_2 = apriori(retail.drop('ID',1), use_colnames=True)
```

	support	itemsets
0	0.666667	(Beer)
1	0.666667	(Chips)
2	0.500000	(Diaper)
3	0.666667	(Milk)
4	0.500000	(Chips, Beer)
5	0.500000	(Diaper, Beer)

如果光考虑支持度support($X > Y$), [Beer, Chips] 和 [Beer, Diaper] 都是很频繁的, 哪一种组合更相关呢?

```
1 | association_rules(frequent_itemsets_2, metric='lift')
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Chips)	(Beer)	0.666667	0.666667	0.5	0.75	1.125	0.055556	1.333333
1	(Beer)	(Chips)	0.666667	0.666667	0.5	0.75	1.125	0.055556	1.333333
2	(Diaper)	(Beer)	0.500000	0.666667	0.5	1.00	1.500	0.166667	inf
3	(Beer)	(Diaper)	0.666667	0.500000	0.5	0.75	1.500	0.166667	2.000000

显然(Diaper, Beer)更相关一些

电影数据集关联分析

```
1 | movies = pd.read_csv('ml-latest-small/movies.csv')
2 | movies.head(10)
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
1	2	Jumanji (1995)	Adventure Children Fantasy
2	3	Grumpier Old Men (1995)	Comedy Romance
3	4	Waiting to Exhale (1995)	Comedy Drama Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action Crime Thriller
6	7	Sabrina (1995)	Comedy Romance
7	8	Tom and Huck (1995)	Adventure Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action Adventure Thriller

数据中包括电影名字与电影类型的标签, 第一步还是先转换成one-hot格式

```
1 | movies_ohc = movies.drop('genres',1).join(movies.genres.str.get_dummies())
2 | pd.options.display.max_columns=100
3 | movies_ohc.head()
```



举报



专栏目录

movieId	title	(no genres listed)	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	IMAX	Musical	Mystery	Romance
0	1	Toy Story (1995)	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0
1	2	Jumanji (1995)	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0
2	3	Grumpier Old Men (1995)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
3	4	Waiting to Exhale (1995)	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
4	5	Father of the Bride Part II (1995)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

```
1 | movies_ohc.shape
```

(9125, 22)
数据集包括9125部电影，一共有22种不同类型

```
1 | movies_ohc.set_index(['movieId','title'],inplace=True)
2 | movies_ohc.head()
```

movieId	title	(no genres listed)	Action	Adventure	Animation	Children	Comedy	Crime	Documentary	Drama	Fantasy	Film-Noir	Horror	IMAX	Musical	Mystery	Romance
1	Toy Story (1995)	0	0	1	1	1	1	0	0	0	1	0	0	0	0	0	0
2	Jumanji (1995)	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0
3	Grumpier Old Men (1995)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
4	Waiting to Exhale (1995)	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
5	Father of the Bride Part II (1995)	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

```
1 | frequent_itemsets_movies = apriori(movies_ohc,use_colnames=True, min_support=0.025)
```

support	itemsets
0.169315	(Action)
0.122411	(Adventure)
0.048986	(Animation)
0.063890	(Children)
0.363288	(Comedy)
0.120548	(Crime)
0.054247	(Documentary)
0.478356	(Drama)
0.071671	(Fantasy)
0.096110	(Horror)
0.043178	(Musical)

```
1 | rules_movies = association_rules(frequent_itemsets_movies, metric='lift', min_threshold=1.2)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Action)	(Adventure)	0.169315	0.122411	0.058301	0.344337	2.812955	0.037575	1.338475
1	(Adventure)	(Action)	0.122411	0.169315	0.058301	0.476276	2.812955	0.037575	1.586111
2	(Action)	(Crime)	0.169315	0.120548	0.038247	0.225890	1.873860	0.017836	1.136081
3	(Crime)	(Action)	0.120548	0.169315	0.038247	0.317273	1.873860	0.017836	1.216716
4	(Sci-Fi)	(Action)	0.086795	0.169315	0.040986	0.472222	2.789015	0.026291	1.573929
5	(Action)	(Sci-Fi)	0.169315	0.086795	0.040986	0.242071	2.789015	0.026291	1.204870
6	(Action)	(Thriller)	0.169315	0.189479	0.062904	0.371521	1.960746	0.030822	1.289654
7	(Thriller)	(Action)	0.189479	0.169315	0.062904	0.331984	1.960746	0.030822	1.243510
8	(Adventure)	(Children)	0.122411	0.063890	0.029260	0.239033	3.741299	0.021439	1.230158
9	(Children)	(Adventure)	0.063890	0.122411	0.029260	0.457876	3.741299	0.021439	1.619096
10	(Adventure)	(Fantasy)	0.122411	0.071671	0.030685	0.250671	3.497518	0.021912	1.238881

```
1 | rules_movies[(rules_movies.lift>4)].sort_values(by=['lift'], ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
14	(Children)	(Animation)	0.063890	0.048986	0.027068	0.423671	8.648758	0.023939	1.650122
15	(Animation)	(Children)	0.048986	0.063890	0.027068	0.552573	8.648758	0.023939	2.092205

Children和Animation 这两题材是最相关的了，常识也可以分辨出来。

python 爬虫从入门到实战

12-29

1. 什么是爬虫 2. 为什么要爬取网络数据 3. 网页基础简介 4. python入门简介 5. python爬虫工作流程 6. 网络元素解析 7. python爬虫实例

使用Python进行数据关联分析_热门推荐

冬之晓 4万+

关联分析 选择函数包 关联分析属于数据挖掘的一大类。我发现的python语言实现的包有两个： pymining：根据Apriori算法进行关联规则...



请发表有价值的评论。 博客评论欢迎灌水，良好的社区氛围需大家一起维护。



评论



Amamiya yuuko: 博主能分享下电影的数据集吗 3月前 回复 **



点赞



Greatest Chili: 楼主，置信度的定义那里，“sigma(X和Y的并集)”这里是不是写错了？应该是“sigma(X和Y的交集)”，面的条件概率公式保持一致 5月前 回复 **



点赞

使用Python进行数据关联分析_冬之晓_python关联分析

11-24

使用Python进行数据关联分析 关联分析 选择函数包 经过分析,我决定使用Oranges进行关联规则的实现,原因如下: FP-growth,Apriori...

python机器学习案例系列教程——关联分析(Apriori、FP...

11-27

python数据挖掘系列教程 关联分析的基本概念 关联分析(Association Analysis):在大规模数据集寻找有趣的关系。频繁项集(Frequent I...



迪哥有点稳了

关注



12



2



51



专栏目录

09-28

商品 关联性分析 （python算法） 电商，物流，存储，仓储，商品 关联性分析 ，python，Apriori	06-27
python数据 关联分析_关联分析 (Apriori)详解和python实... 关联分析 关联关系是一种非常有用的数据挖掘算法,它可以 分析 出数据内在的 关联 关系。其中比较著名的是啤酒和尿不湿的案例 交易号 ...	11-15
python机器学习案例系列—— 关联分析 (Apriori、FP-grow... return list(map(frozenset, C1)) #map(frozenset, C1)的语义是将C1由Python列表转换为不变集合(frozenset,Python中的数据结构) #找...	10-16
无监督学习- 关联分析 apriori 原理 与python代码 关联分析 是一种无监督学习，它的目标就是从大数据中找出那些经常一起出现的东西，不管是商品还是其他什么 item，然后靠这些结果...	weixin_37825814的博客 206
python机器学习之 关联分析 （Apriori） 在机器学习中，除了聚类算法外，Apriori算法也是在数据集中寻找数据之间的某种 关联 关系，通过该算法，我们可以在大规模的数据中发...	最新发布 杨小葱的博客 1188
python 关联分析 算法的包_Python 极简 关联分析 (购物篮 分析) 关联分析 ,也称购物篮 分析 ,本文目的: 基于订单表,用最少的python代码完成数据整合及 关联分析 文中所用数据下载地址: 使用Python Anac...	11-23
Python——机器学习 实战 ——Apriori算法进行 关联分析 本代码主要利用Python工具实现Apriori算法进行 关联分析 ，简单明了，易于理解	08-27
Apriori 关联性分析 python实现(含数据集) Apriori 关联性分析 python实现(含数据集)，结构清晰易懂	12-15
Python数据 分析 基础之 关联分析 FP_growth 作者：蛰虫始航 来源：蛰虫始航上篇文章我们了解了 关联分析 的基本概念和应用场景，以及挖掘数据集中 关联 规则的Apriori算法，通...	数据森麟 438
python机器学习案例系列教程—— 关联分析 （Apriori、FP-growth） 全栈工程师开发手册（原创）（腾讯内推）	1万+
python 关联分析 案例_基于Python实现相关 分析 案例 基于Python实现相关 分析 案例mp.weixin.qq.com节选自《Python预测之美：数据 分析 与算法 实战 》相关关系是一种非确定的关系，就好...	weixin_39862669的博客 417
关联分析-从算法到实战 apriori - 频繁项集的产生 http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/ association_rules - 关联 规则的生成 associa...	weixin_43962871的博客 4951
张正友标定法-- 从原理到实战 源.cpp 张正友标定处于什么水平，为啥提到相机标定，就不得不提他张博士的方法？ 简单介绍一下张博士 他的方法优缺点，有没有替代方案？ ...	11-29
Partial Dependence Plots 从原理到实战 Partial Dependence：用来解释某个特征和目标值y的关系，一般是通过画出Partial Dependence Plots(PDP)来体现。PDP是依赖于模型...	weixin_43962871的博客 2026
十分钟搞定PCA主成分 分析 在数据建模当中我们会经常听到一个词叫做降维，首先咱们先来唠一唠数据为啥要降维呢？最主要的原因还是在于一方面使得我们需要...	迪哥有点愁 7854
新手如何快速入门深度学习 如何快速入门深度学习本篇学习笔记对应深度学习入门视频教程博客地址：http://blog.csdn.net/tangyudi 欢迎转载 深度学习入门必备基...	迪哥有点愁 6779
AI时代-人工智能入学指南 【导读】：本篇文章旨在帮助大家建立一份人工智能的学习计划以及我的一些个人建议，希望大家在AI之路都能早日成为大神！人工智...	迪哥有点愁 7810
机器学习故事汇-线性回归算法 机器学习故事汇-线性回归算法 【咱们的目标】 系列算法讲解旨在用最简单易懂的故事情节帮助大家掌握晦涩无趣的机器学习，适合对数...	迪哥有点愁 6474
clementine中Apriori参数解读Maximum number of antecedents 数据挖掘课后作业需要用到这个软件，老师将基本参数设置好了，但是具体各项是什么意思呢？ 下面是老师预设的参数， Run Apriori o...	K's Blog 455

©2021 CSDN 皮肤主题: 大白 设计师: CSDN官方博客 返回首页

关于我们 招贤纳士 广告服务 开发助手 400-660-0108 kefu@csdn.net 在线客服 工作时间 8:30-22:00
公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 网络110报警服务 中国互联网举报中心 家长监护 Chrome商店下载 ©1999-2021北京创新乐知网络技术有限公司 版权与免责声明 版权申诉 出版物许可证 营业执照

