

# Ассоциативные правила

# Ассоциативные правила

**Ассоциативные правила** - метод поиска взаимосвязей или ассоциаций в датасетах (точнее в itemsets).

**Идея:** «Кто купил х, также купил у».

В основе – анализ транзакций.

Ищем правила совместных покупок и оцениваем силу правил.

Хороший пост на Хабр: <https://habr.com/ru/company/ods/blog/353502/>

# Описание Association rule

$D$  – датасет или коллекция транзакций

$d$  — уникальная транзакция-*itemset* (например, кассовый чек)

$d1 = [\{Пиво: 1\}, \{Вода: 0\}, \{Кола: 1\}, \{\dots\}], d2 = [\{Пиво: 0\}, \{Вода: 1\}, \{Кола: 1\}, \{\dots\}]$

Таким образом, датасет представляет собой разреженную матрицу со значениями {1,0} – **бинарный датасет**.

# Support (поддержка)

**Support** - это показатель «частотности» данного itemset в транзакциях.

$$supp(X) = \frac{|\{t \in T; X \in t\}|}{|T|}$$

$X$  — itemset

$|T|$  — количество транзакций

# Пример: пиво с подгузниками

Transaction	Кола	Пиво	Подгузники
0	1	1	1
1	2	0	0
2	3	1	1
3	4	1	1
4	5	0	1

$$supp = \frac{\text{Транзакции с пивом и подгузниками}}{\text{Все транзакции}} = P(\text{Пиво} \cap \text{Подгузники})$$

$$supp = \frac{3}{5} = 60\%$$

# Confidence (достоверность)

**Confidence** - это показатель того, как часто наше правило («*кто покупает x, тот покупает y*») срабатывает для датасета.

$$conf(x_1 \cup x_2) = \frac{supp(x_1 \cup x_2)}{supp(x_1)}$$

$$conf(\text{Пиво} \cup \text{Подгузники}) = \frac{supp(\text{Пиво} \cup \text{Подгузники})}{supp(\text{Пиво})} = P(\text{Подгузники} \mid \text{Пиво})$$

$$conf = \frac{3}{4} = 75\%$$

Transaction	Кола	Пиво	Подгузники	
0	1	1	1	1
1	2	0	0	0
2	3	1	1	0
3	4	1	1	1
4	5	0	1	1

# Lift (поддержка)

**Lift** показывает насколько items зависят друг от друга.

$$lift(x_1 \cup x_2) = \frac{supp(x_1 \cup x_2)}{supp(x_1) \times supp(x_2)}$$

$lift = 1$  – items независимы

$lift > 1$  – items покупают вместе

$lift < 1$  – с одним другое обычно не покупают

$$lift = \frac{\text{Confidence}}{\text{Expected confidence}} = \frac{P(\text{Подгузники} \mid \text{Пиво})}{P(\text{Подгузники})}$$

$$lift = \frac{\frac{3}{4}}{\frac{3}{5}} = 1,25$$

Transaction	Кола	Пиво	Подгузники	
0	1	1	1	1
1	2	0	0	0
2	3	1	1	0
3	4	1	1	1
4	5	0	1	1

# Conviction (убедительность)

**Conviction** — это «частотность ошибок» нашего правила.

$$conv(x_1 \cup x_2) = \frac{1 - supp(x_2)}{1 - conf(x_1 \cup x_2)}$$

$$conv(\text{Пиво} \cup \text{Подгузники}) = \frac{1 - supp(\text{Подгузники})}{1 - conf(\text{Пиво} \cup \text{Подгузники})} = \frac{1 - 0.6}{1 - 0.75} = 1,6$$

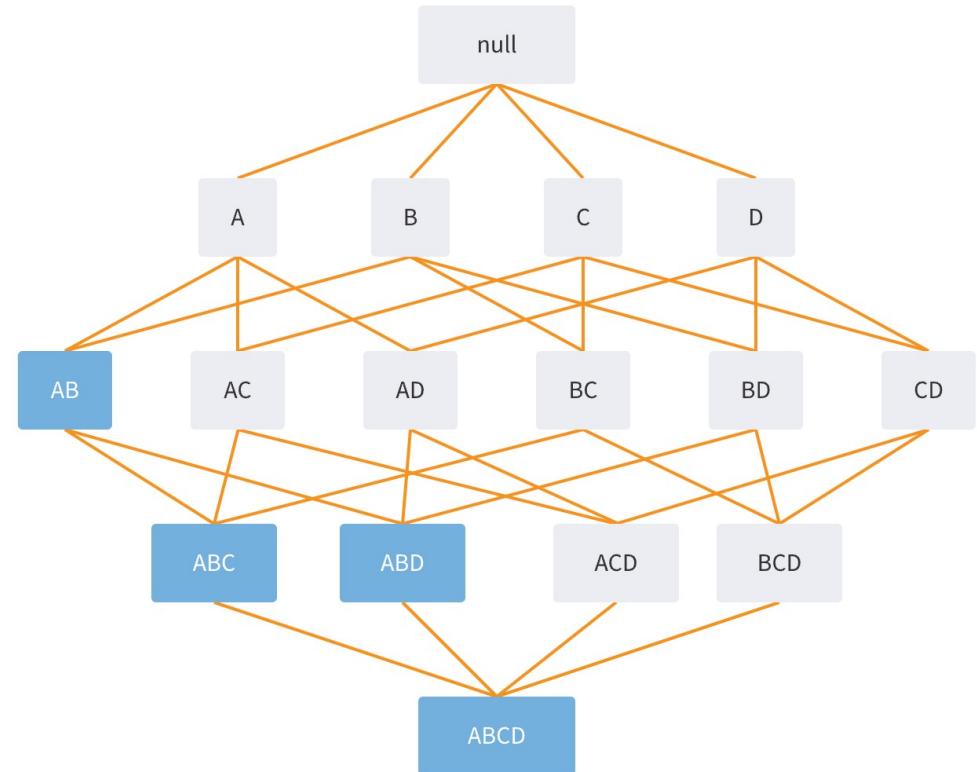
Transaction	Кола	Пиво	Подгузники	
0	1	1	1	1
1	2	0	0	0
2	3	1	1	0
3	4	1	1	1
4	5	0	1	1

# Apriori algorithm

## Свойство анти-монотонности:

**Формулировка 1:** поддержка любого набора элементов не может превышать минимальной поддержки любого из его подмножеств.

**Формулировка 2:** с ростом размера набора элементов поддержка уменьшается, либо остается такой же.

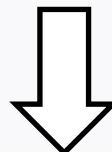


# Apriori algorithm

## Свойство анти-монотонности:

**Формулировка 1:** поддержка любого набора элементов не может превышать минимальной поддержки любого из его подмножеств.

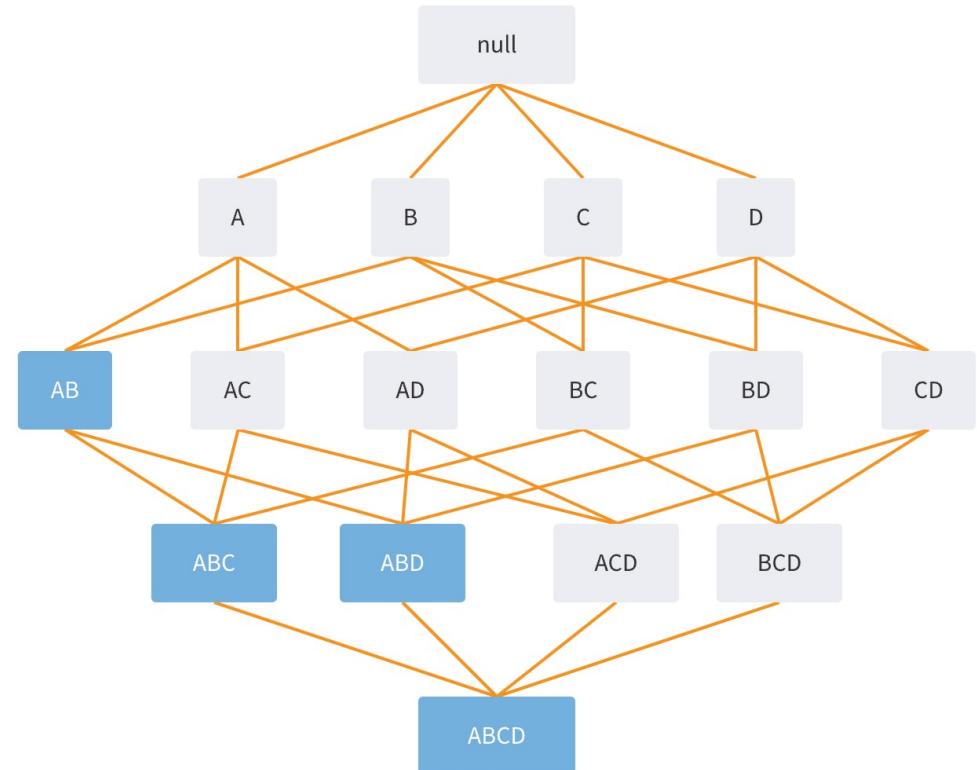
**Формулировка 2:** с ростом размера набора элементов поддержка уменьшается, либо остается такой же.



$k$ -элементный набор будет часто встречающимся



все его  $(k-1)$ -элементные подмножества будут часто встречающимися

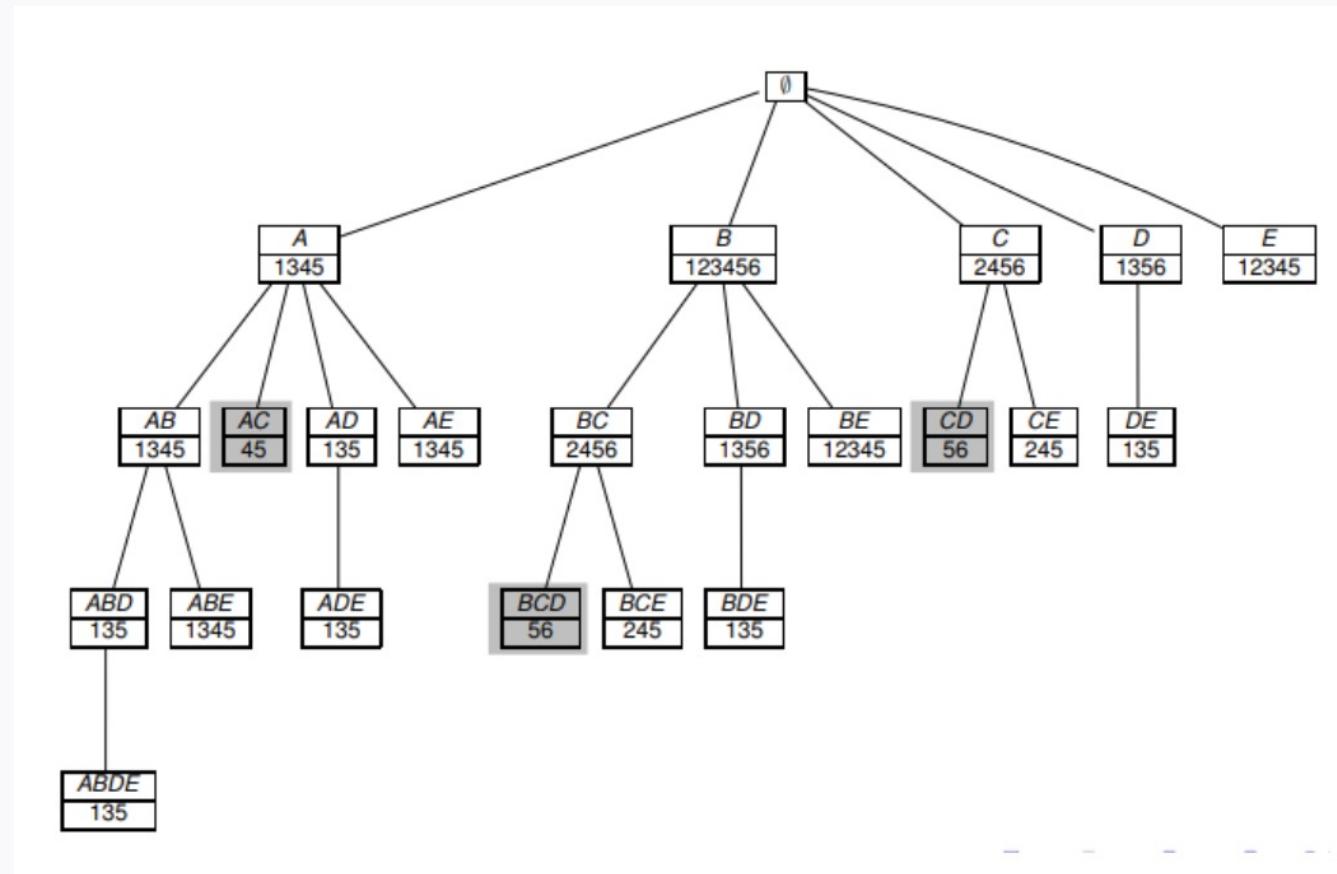


# Apriori algorithm

## Apriori

Apriori алгоритм по уровням проходит по префиксному дереву и рассчитывает частоту встречаемости подмножеств X в D.

- исключаются редкие подмножества и все их супермножества
- рассчитывается  $supp(X)$  для каждого подходящего кандидата X размера  $k$  на уровне  $k$



# Apriori algorithm

**1.Объединение.** Просмотр базы данных и определение частоты вхождения отдельных товаров.

**2.Отсечение.** Те наборы, которые удовлетворяют поддержке и достоверности, переходят на следующую итерацию.

**3.Повторение.** Предыдущие два шага повторяются для каждой величины набора, пока не будет повторно получен ранее определенный размер.

*Apriori*( $T, \varepsilon$ )

```
 $L_1 \leftarrow \{ \text{large 1-itemsets that appear in more than } \varepsilon \text{ transactions} \}$ 
 $k \leftarrow 2$ 
    while  $L_{k-1} \neq \emptyset$ 
         $C_k \leftarrow \text{Generate}(L_{k-1})$ 
        for transactions  $t \in T$ 
             $C_t \leftarrow \text{Subset}(C_k, t)$ 
            for candidates  $c \in C_t$ 
                 $\text{count}[c] \leftarrow \text{count}[c] + 1$ 
         $L_k \leftarrow \{ c \in C_k \mid \text{count}[c] \geq \varepsilon \}$ 
         $k \leftarrow k + 1$ 
    return  $\bigcup_k L_k$ 
```

# Практика Python

# Гибридные рекомендательные системы

# Гибридные системы

- Разные подходы имеют свои преимущества и недостатки, т. е. свою область применения
- Гибридные системы позволяют одновременно использовать преимущества разных подходов

# Виды гибридных систем

- Weighted
- Switching
- Mixed
- Feature combination
- Cascade
- Feature augmentation

# Weighted

Разные рекомендательные системы независимо генерируют оценки для каждого объекта.

Финальная оценка строится на основе комбинирования доступных оценок с некоторыми весами.

**Пример:**

- *Линейная комбинация коллаборативной фильтрации и content-based*
- *Голосование для бинарных оценок*

# Switching

Разные рекомендательные системы независимо генерируют оценки для каждого объекта.

Для финальной оценки выбираем ответ одной из рекомендательных систем по некоторому критерию.

**Пример:**

- Коллаборативная фильтрация с *content-based* подходом для новых пользователей (решаем проблему холодного старта)

# Mixed

Каждая система строит свой список рекомендаций.

Итоговый список рекомендаций строится на основе смеси рекомендаций от разных систем.

Позволяет получить длинный, разнообразный список рекомендаций (feed новостей, рекомендации медиаконтента).

# Feature combination

В некоторой степени content-based подход.

**Идея:** объединить в одной обучающей выборке признаки от разных подходов.

Используем информацию о предпочтениях похожих пользователей, полученную в рамках коллаборативного подхода, в качестве признаков для content-based подхода.

# Cascade

Последовательно применяем несколько рекомендательных систем для уточнения рекомендаций

**Пример:**

1. сначала грубо отсеаем точно нерелевантные объекты
2. затем уточняем рекомендации

# Feature augmentation

Применяем к объектам и пользователям несколько рекомендательных систем, затем их выход используем как входные признаки для системы на следующем уровне

**Пример:**

*Можем оценить объекты с помощью рекомендательной системы на основе коллаборативной фильтрации, а дальше использовать эти оценки для системы на основе content-based.*

# Гибридизация

1. Часто улучшает качество
2. Иногда положительно сказывается на разнообразии
3. Не гарантирует решения всех проблем, связанных с тем или иным подходом