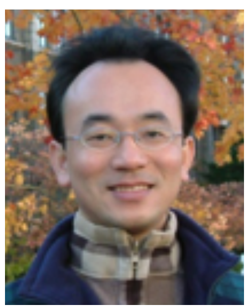
資料探勘
(Data Mining)

非監督學習：
關聯分析，購物籃分析
(Unsupervised Learning: Association Analysis, Market Basket Analysis)

1092DM05
MBA, IM, NTPU (M5026) (Spring 2021)
Tue 2, 3, 4 (9:10-12:00) (B8F40)



Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021-03-23

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

1 2021/02/23 資料探勘介紹 (Introduction to data mining)

2 2021/03/02 ABC：人工智慧，大數據，雲端運算
(ABC: AI, Big Data, Cloud Computing)

3 2021/03/09 Python資料探勘的基礎
(Foundations of Data Mining in Python)

4 2021/03/16 資料科學與資料探勘：發現，分析，可視化和呈現數據
(Data Science and Data Mining:
Discovering, Analyzing, Visualizing and Presenting Data)

5 2021/03/23 非監督學習：關聯分析，購物籃分析
(Unsupervised Learning: Association Analysis,
Market Basket Analysis)

6 2021/03/30 資料探勘個案研究 I
(Case Study on Data Mining I)

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

7 2021/04/06 非監督學習：集群分析，行銷市場區隔
(Unsupervised Learning: Cluster Analysis, Market Segmentation)

8 2021/04/13 監督學習：分類和預測
(Supervised Learning: Classification and Prediction)

9 2021/04/20 期中報告 (Midterm Project Report)

10 2021/04/27 監督學習：分類和預測
(Supervised Learning: Classification and Prediction)

11 2021/05/04 機器學習和深度學習
(Machine Learning and Deep Learning)

12 2021/05/11 卷積神經網絡
(Convolutional Neural Networks)

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

13 2021/05/18 資料探勘個案研究 II
(Case Study on Data Mining II)

14 2021/05/25 遞歸神經網絡
(Recurrent Neural Networks)

15 2021/06/01 強化學習
(Reinforcement Learning)

16 2021/06/08 社交網絡分析
(Social Network Analysis)

17 2021/06/15 期末報告 I (Final Project Report I)

18 2021/06/22 期末報告 II (Final Project Report II)

Unsupervised Learning: Association Analysis, Market Basket Analysis

Outline

- Unsupervised Learning
- Association Analysis
- Market Basket Analysis
- Recommender System
- Apriori algorithm
  - Frequent Itemsets
  - Association Rules

Data Mining Tasks & Methods

Unsupervised Learning:
Association Analysis
Market-basket

Association

Data Mining Tasks & Methods	Data Mining Algorithms	Learning Type
Prediction		
Classification	Decision Trees, Neural Networks, Support Vector Machines, kNN, Naïve Bayes, GA	Supervised
Regression	Linear/Nonlinear Regression, ANN, Regression Trees, SVM, kNN, GA	Supervised
Time series	Autoregressive Methods, Averaging Methods, Exponential Smoothing, ARIMA	Supervised
Association		
Market-basket	Apriori, OneR, ZeroR, Eclat, GA	Unsupervised
Link analysis	Expectation Maximization, Apriori Algorithm, Graph-Based Matching	Unsupervised
Sequence analysis	Apriori Algorithm, FP-Growth, Graph-Based Matching	Unsupervised
Segmentation		
Clustering	*k*-means, Expectation Maximization (EM)	Unsupervised
Outlier analysis	*k*-means, Expectation Maximization (EM)	Unsupervised

Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Transaction Database

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

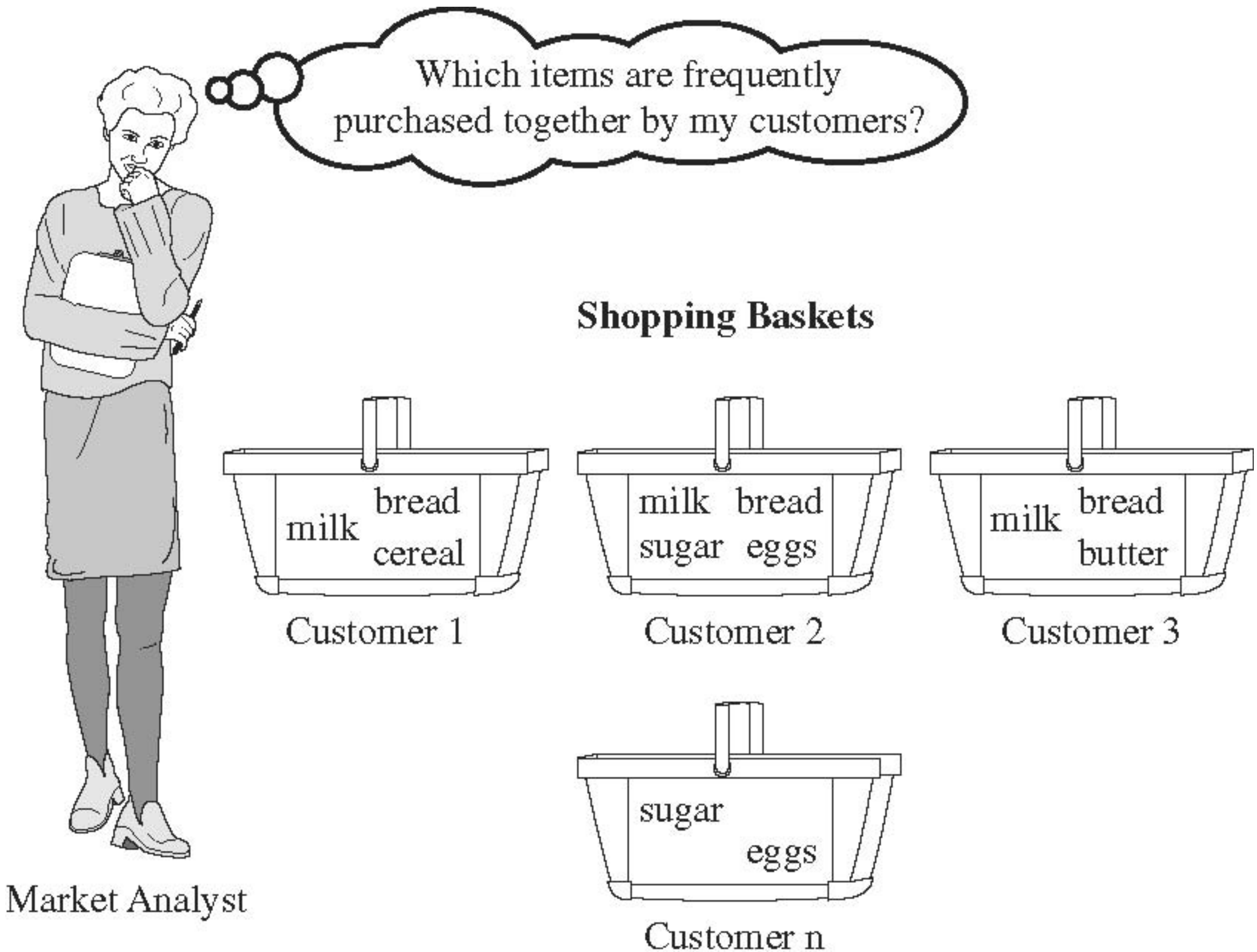
Association Analysis

Association Analysis: Mining Frequent Patterns, Association and Correlations

- Association Analysis
- Mining Frequent Patterns
- Association and Correlations
- Apriori Algorithm

Source: Han & Kamber (2006)

Market Basket Analysis



Source: Han & Kamber (2006)

Association Rule Mining

- Apriori Algorithm

Raw Transaction Data

Transaction No	SKUs (Item No)
1	1, 2, 3, 4
1	2, 3, 4
1	2, 3
1	1, 2, 4
1	1, 2, 3, 4
1	2, 4

One-item Itemsets

Itemset (SKUs)	Support
1	3
2	6
3	4
4	5

Two-item Itemsets

Itemset (SKUs)	Support
1, 2	3
1, 3	2
1, 4	3
2, 3	4
2, 4	5
3, 4	3

Three-item Itemsets

Itemset (SKUs)	Support
1, 2, 4	3
2, 3, 4	3

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- A very popular DM method in business
- Finds interesting relationships (affinities) between variables (items or events)
- Part of machine learning family
- Employs unsupervised learning
- There is no output variable
- Also known as market basket analysis
- Often used as an example to describe DM to ordinary people, such as the famous “relationship between diapers and beers!”

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- Input: the simple point-of-sale transaction data
- Output: Most frequent affinities among items
- Example: according to the transaction data…

  "Customer who bought a laptop computer and a virus protection software, also bought extended service plan 70 percent of the time."
- How do you use such a pattern/knowledge?
  - Put the items next to each other for ease of finding
  - Promote the items as a package (do not put one on sale if the other(s) are on sale)
  - Place items far apart from each other so that the customer has to walk the aisles to search for it, and by doing so potentially seeing and buying other items

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- A representative applications of association rule mining include
  - In business: cross-marketing, cross-selling, store design, catalog design, e-commerce site design, optimization of online advertising, product pricing, and sales/promotion configuration
  - In medicine: relationships between symptoms and illnesses; diagnosis and patient characteristics and treatments (to be used in medical DSS); and genes and their functions (to be used in genomics projects)…

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- Are all association rules interesting and useful?

A Generic Rule: **X ⇒ Y [S%, C%]**

X, Y: products and/or services

X: Left-hand-side (LHS)

Y: Right-hand-side (RHS)

S: Support: how often **X** and **Y** go together

C: Confidence: how often **Y** go together with the **X**

Example: {Laptop Computer, Antivirus Software} ⇒ {Extended Service Plan} [30%, 70%]

Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- Algorithms are available for generating association rules
  - Apriori
  - Eclat
  - FP-Growth
  - + Derivatives and hybrids of the three
- The algorithms help identify the frequent item sets, which are, then converted to association rules

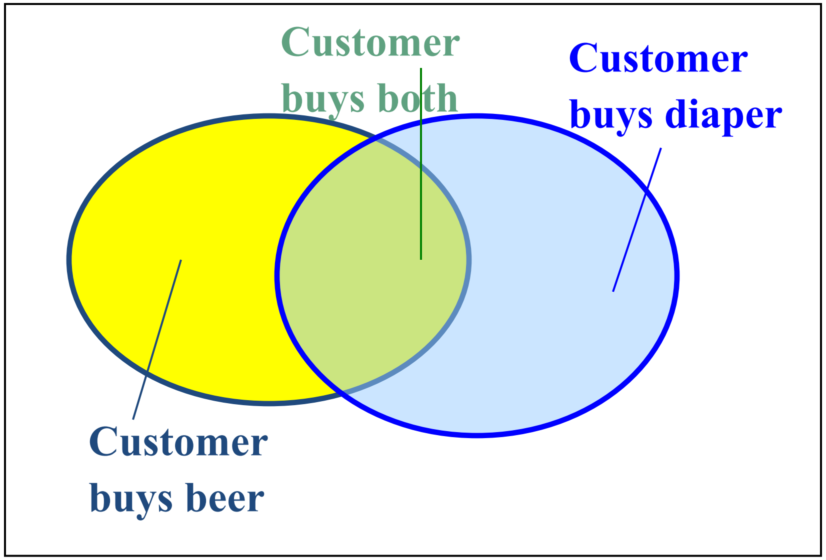
Source: Turban et al. (2011), Decision Support and Business Intelligence Systems

Association Rule Mining

- Apriori Algorithm
  - Finds subsets that are common to at least a minimum number of the itemsets
  - uses a bottom-up approach
    - frequent subsets are extended one item at a time (the size of frequent subsets increases from one-item subsets to two-item subsets, then three-item subsets, and so on), and
    - groups of candidates at each level are tested against the data for minimum

Basic Concepts: Frequent Patterns and Association Rules

Transaction-id	Items bought
10	A, B, D
20	A, C, D
30	A, D, E
40	B, E, F
50	B, C, D, E, F



- Itemset X = $\{x_1, \ldots, x_k\}$
- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, *s*, probability that a transaction contains $X \cup Y$
  - confidence, *c*, conditional probability that a transaction having X also contains *Y*

Let $sup_{min} = 50\%$, $conf_{min} = 50\%$

Freq. Pat.: {A:3, B:3, D:4, E:3, AD:3}

Association rules:

$A \rightarrow D$ (60%, 100%)

$D \rightarrow A$ (60%, 75%)

$A \rightarrow D$ (support = 3/5 = 60%, confidence = 3/3 =100%)

$D \rightarrow A$ (support = 3/5 = 60%, confidence = 3/4 = 75%)

Source: Han & Kamber (2006)

Market basket analysis

- Example
  - Which groups or sets of items are customers likely to purchase on a given trip to the store?
- Association Rule
  - *Computer → antivirus_software [support = 2%; confidence = 60%]*
    - A support of 2% means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.
    - A confidence of 60% means that 60% of the customers who purchased a computer also bought the software.

Source: Han & Kamber (2006)

Association rules

- Association rules are considered interesting if they satisfy both
  - a minimum support threshold and
  - a minimum confidence threshold.

Source: Han & Kamber (2006)

Frequent Itemsets, Closed Itemsets, and Association Rules

Let $I = \{I_1, I_2, \ldots, I_m\}$ be a set of items. Let D, the task-relevant data, be a set of database transactions where each transaction T is a set of items such that $T \subseteq I$. Each transaction is associated with an identifier, called TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an implication of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \phi$. The rule $A \Rightarrow B$ holds in the transaction set D with **support** s, where s is the percentage of transactions in D that contain $A \cup B$ (i.e., the *union* of sets A and B, or say, both A and B). This is taken to be the probability, $P(A \cup B)$.[1] The rule $A \Rightarrow B$ has **confidence** c in the transaction set D, where c is the percentage of transactions in D containing A that also contain B. This is taken to be the conditional probability, $P(B|A)$. That is,

$$\text{Support}\ (A \rightarrow B) = P(A \cup B)$$

$$\text{Confidence}\ (A \rightarrow B) = P(B|A)$$

Source: Han & Kamber (2006)

Support (A → B) = P(A ∪ B)
Confidence (A → B) = P(B|A)

- The notation *P(A ∪ B) indicates the probability that a transaction contains the union of set A and set B*
  - *(i.e., it contains every item in A and in B).*
- *This should not be confused with P(A or B),* which indicates the probability that a transaction contains either *A or B.*

Source: Han & Kamber (2006)

Does diaper purchase predict beer purchase?

- Contingency tables

	Beer: Yes	Beer: No	
No diapers	6	94	100
diapers	40	60	100

DEPENDENT (yes)

	Beer: Yes	Beer: No
No diapers	23	77
diapers	23	77

INDEPENDENT (no predictability)

Source: Dickey (2012) http://www4.stat.ncsu.edu/~dickey/SAScode/Encore_2012.ppt

$$\text{Support}(A \rightarrow B) = P(A \cup B)$$

$$\text{Confidence}(A \rightarrow B) = P(B|A)$$

$$\text{Conf}(A \rightarrow B) = \text{Supp}(A \cup B) / \text{Supp}(A)$$

$$\text{Lift}(A \rightarrow B) = \text{Supp}(A \cup B) / (\text{Supp}(A) \times \text{Supp}(B))$$

Lift (Correlation)

$$\text{Lift}(A \rightarrow B) = \text{Confidence}(A \rightarrow B) / \text{Support}(B)$$

Source: Dickey (2012) http://www4.stat.ncsu.edu/~dickey/SAScode/Encore_2012.ppt

Lift

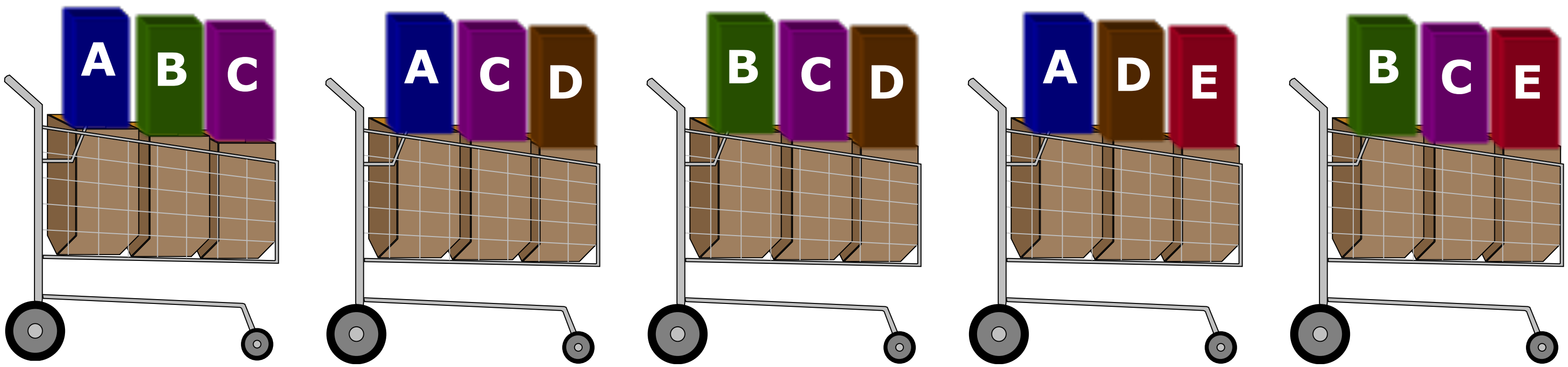
Lift = Confidence / Expected Confidence if Independent

Checking → Saving ↓	No (1500)	Yes (8500)	(10000)
No	500	3500	4000
Yes	1000	5000	6000

SVG=>CHKG Expect 8500/10000 = 85% if independent
Observed Confidence is 5000/6000 = 83%
Lift = 83/85 < 1.
Savings account holders actually LESS likely than others to have checking account !!!

Source: Dickey (2012) http://www4.stat.ncsu.edu/~dickey/SAScode/Encore_2012.ppt

Support & Confidence



Rule	Support	Confidence
$A \Rightarrow D$	2/5	2/3
$C \Rightarrow A$	2/5	2/4
$A \Rightarrow C$	2/5	2/3
$B \,\&\, C \Rightarrow D$	1/5	1/3

Source: SAS Enterprise Miner Course Notes, 2014, SAS

Support & Confidence & Lift

		Checking Account: No	Checking Account: Yes	
Saving Account	No	500	3500	4,000
	Yes	1000	5000	6,000
				10,000

Support(SVG $\Rightarrow$ CK) = 50%=5,000/10,000

Confidence(SVG $\Rightarrow$ CK) = 83%=5,000/6,000

Expected Confidence(SVG $\Rightarrow$ CK) = 85%=8,500/10,000

Lift (SVG $\rightarrow$ CK) = Confidence/Expected Confidence = 0.83/0.85 < 1

Source: SAS Enterprise Miner Course Notes, 2014, SAS

Support (A→B)

Confidence (A→B)

Expected Confidence (A→B)

Lift (A→B)

Support (A → B) = P(A ∪ B)

Count(A&B)/Count(Total)

Confidence (A → B) = P(B|A)

Conf (A → B) = Supp (A ∪ B)/ Supp (A)

Count(A&B)/Count(A)

Expected Confidence (A →B) = Support(B)

Count(B)

Lift (A → B) = Confidence (A →B) / Expected Confidence (A →B)

Lift (A → B) = Supp (A ∪ B) / (Supp (A) x Supp (B))

Lift (Correlation)

Lift (A →B) = Confidence (A →B) / Support(B)

Lift (A→B)

- Lift (A→B)
  = Confidence (A→B) / Expected Confidence (A→B)
  = Confidence (A→B) / Support(B)
  = (Supp (A&B) / Supp (A)) / Supp(B)
  = Supp (A&B) / Supp (A) x Supp (B)

Minimum Support and Minimum Confidence

- Rules that satisfy both a minimum support threshold (*min_sup*) and a minimum confidence threshold (*min_conf*) are called **strong**.
- By convention, we write support and confidence values so as to occur between 0% and 100%, rather than 0 to 1.0.

Source: Han & Kamber (2006)

K-itemset

- itemset
  - A set of items is referred to as an itemset.
- K-itemset
  - An itemset that contains *k items is a k*-itemset*.*
- Example:
  - The set {*computer, antivirus software*} is a 2-itemset.

Source: Han & Kamber (2006)

Absolute Support and Relative Support

- Absolute Support
  - The occurrence frequency of an itemset is the number of transactions that contain the itemset
    - frequency, support count, or count of the itemset
  - Ex: 3
- Relative support
  - Ex: 60%

Source: Han & Kamber (2006)

Frequent Itemset

- If the relative support of an itemset *I satisfies a prespecified minimum support threshold, then I is a* frequent itemset.
  - *i.e., the absolute* support of *I satisfies the corresponding minimum support count threshold*
- The set of frequent *k-itemsets is commonly denoted by* L_K

Source: Han & Kamber (2006)

Confidence

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support(A \cup B)}{support(A)} = \frac{support\_count(A \cup B)}{support\_count(A)}$$

- the confidence of rule *A → B can be easily derived from the* support counts of *A and A ∪ B.*
- once the support counts of *A, B, and A ∪ B are* found, it is straightforward to derive the corresponding association rules *A →B and B →A* and check whether they are strong.
- Thus the problem of mining association rules can be reduced to that of mining frequent itemsets.

Source: Han & Kamber (2006)

Association rule mining: Two-step process

1. Find all frequent itemsets
   - By definition, each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min_sup.*
2. Generate strong association rules from the frequent itemsets
   - By definition, these rules must satisfy minimum support and minimum confidence.

Source: Han & Kamber (2006)

Efficient and Scalable Frequent Itemset Mining Methods

- The Apriori Algorithm
  - Finding Frequent Itemsets Using Candidate Generation

Source: Han & Kamber (2006)

Apriori Algorithm

- **Apriori** is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.
- The name of the algorithm is based on the fact that the algorithm uses *prior knowledge of frequent itemset properties, as we shall* see following.

Source: Han & Kamber (2006)

Apriori Algorithm

- Apriori employs an iterative approach known as a *level-wise search, where k-itemsets are used to explore (k+1)-itemsets.*
- *First, the set of frequent 1-itemsets is found* by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted L_1.
- *Next,* L_1 *is used to find* L_2, the set of frequent 2-itemsets, which is used to find L_3, *and so on, until no more frequent k-itemsets can be found.*
- *The finding of each* L_k *requires one full scan of the database.*

Source: Han & Kamber (2006)

Apriori Algorithm

- To improve the efficiency of the level-wise generation of frequent itemsets, an important property called the Apriori property.
- Apriori property
  - *All nonempty subsets of a frequent itemset must also be frequent.*

Source: Han & Kamber (2006)

Apriori algorithm
(1) Frequent Itemsets
(2) Association Rules

Transaction Database

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

Table 1 shows a database with 10 transactions.
Let *minimum support = 20%* and *minimum confidence = 80%*.
Please use **Apriori algorithm** for generating **association rules** from frequent itemsets.

Table 1: Transaction Database

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

Apriori Algorithm
$C_1 \rightarrow L_1$

Step 1-1

C_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

minimum support = 20% = 2 / 10
Min. Support Count = 2

L_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

Apriori Algorithm

$C_2 \rightarrow L_2$

Step 1-2

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

L_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

C_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3
D, E	1

minimum support = 20% = 2 / 10
Min. Support Count = 2

L_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3

Apriori Algorithm
$C_3 \rightarrow L_3$

Step **1-3**

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

L_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3

C_3

Itemset	Support Count
A, B, C	1
A, B, D	2
A, B, E	1
A, C, D	1
A, C, E	2
B, C, D	2
B, C, E	2

minimum support = 20% = 2 / 10
Min. Support Count = 2

L_3

Itemset	Support Count
A, B, D	2
A, C, E	2
B, C, D	2
B, C, E	2

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

Generating Association Rules

Step 2-1

minimum confidence = 80%

L_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

L_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3

Association Rules Generated from L_2

A→B: 3/6	B→A: 3/7
A→C: 4/6	C→A: 4/6
A→D: 3/6	D→A: 3/7
A→E: 2/6	E→A: 2/3
B→C: 3/7	C→B: 3/6
B→D: 6/7=85.7% *	D→B: 6/7=85.7% *
B→E: 2/7	E→B: 2/3
C→D: 3/6	D→C: 2/7
C→E: 3/6	E→C: 3/3=100% *

Generating Association Rules

Step 2-2

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

minimum confidence = 80%

Association Rules Generated from L_3

A→BD: 2/6	B→CD: 2/7
B→AD: 2/7	C→BD: 2/6
D→AB: 2/7	D→BC: 2/7
AB→D: 2/3	BC→D: 2/3
AD→B: 2/3	BD→C: 2/6
BD→A: 2/6	CD→B: 2/3
A→CE: 2/6	B→CE: 2/7
C→AE: 2/6	C→BE: 2/6
E→AC: 2/3	E→BC: 2/3
AC→E: 2/4	BC→E: 2/3
AE→C: 2/2=100%*	BE→C: 2/2=100%*
CE→A: 2/3	CE→B: 2/3

L_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

L_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3

L_3

Itemset	Support Count
A, B, D	2
A, C, E	2
B, C, D	2
B, C, E	2

Frequent Itemsets and Association Rules

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

L_1

Itemset	Support Count
A	6
B	7
C	6
D	7
E	3

L_2

Itemset	Support Count
A, B	3
A, C	4
A, D	3
A, E	2
B, C	3
B, D	6
B, E	2
C, D	3
C, E	3

L_3

Itemset	Support Count
A, B, D	2
A, C, E	2
B, C, D	2
B, C, E	2

minimum support = 20%
minimum confidence = 80%

Association Rules:

B→D (60%, 85.7%) (Sup.: 6/10, Conf.: 6/7)
D→B (60%, 85.7%) (Sup.: 6/10, Conf.: 6/7)
E→C (30%, 100%) (Sup.: 3/10, Conf.: 3/3)
AE→C (20%, 100%) (Sup.: 2/10, Conf.: 2/2)
BE→C (20%, 100%) (Sup.: 2/10, Conf.: 2/2)

Table 1 shows a database with 10 transactions.
Let *minimum support = 20%* and *minimum confidence = 80%*.
Please use **Apriori algorithm** for generating **association rules** from frequent itemsets.

Transaction ID	Items bought
T01	A, B, D
T02	A, C, D
T03	B, C, D, E
T04	A, B, D
T05	A, B, C, E
T06	A, C
T07	B, C, D
T08	B, D
T09	A, C, E
T10	B, D

Association Rules:

B→D (60%, 85.7%) (Sup.: 6/10, Conf.: 6/7)
D→B (60%, 85.7%) (Sup.: 6/10, Conf.: 6/7)
E→C (30%, 100%) (Sup.: 3/10, Conf.: 3/3)
AE→C (20%, 100%) (Sup.: 2/10, Conf.: 2/2)
BE→C (20%, 100%) (Sup.: 2/10, Conf.: 2/2)

Python mlxtend Association Rules

```
# !pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

dataset = [['A', 'B', 'D'],
           ['A', 'C', 'D'],
           ['B', 'C', 'D', 'E'],
           ['A', 'B', 'D'],
           ['A', 'B', 'C', 'E'],
           ['A', 'C'],
           ['B', 'C', 'D'],
           ['B', 'D'],
           ['A', 'C', 'E'],
           ['B', 'D']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.2, use_colnames=True)
association_rules(frequent_itemsets, metric="confidence", min_threshold=0.8)
```

https://tinyurl.com/aintpupython101

Python mlxtend Association Rules

```python
# !pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
dataset = [['A', 'B', 'D'],
           ['A', 'C', 'D'],
           ['B', 'C', 'D', 'E'],
           ['A', 'B', 'D'],
           ['A', 'B', 'C', 'E'],
           ['A', 'C'],
           ['B', 'C', 'D'],
           ['B', 'D'],
           ['A', 'C', 'E'],
           ['B', 'D']]
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.2, use_colnames=True)
rules = association_rules(frequent_itemsets, metric="confidence", min_threshold=0.8)
rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(B)	(D)	0.7	0.7	0.6	0.857143	1.224490	0.11	2.1
1	(D)	(B)	0.7	0.7	0.6	0.857143	1.224490	0.11	2.1
2	(E)	(C)	0.3	0.6	0.3	1.000000	1.666667	0.12	inf
3	(E, A)	(C)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf
4	(E, B)	(C)	0.2	0.6	0.2	1.000000	1.666667	0.08	inf

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

Table of contents

+ Code + Text RAM Disk Editing

Association Rules Generation from Frequent Itemsets

- mlxtend: https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

```python
# !pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)
frequent_itemsets
```

	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)

https://tinyurl.com/aintpupython101

```
# ! pip install mlxtend
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

frequent_itemsets = apriori(df, min_support=0.6,
use_colnames=True)
```

```
# ! pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)

frequent_itemsets
```

https://tinyurl.com/aintpupython101

```
# ! pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules

dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]

te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6,
use_colnames=True)

frequent_itemsets
```

```python
frequent_itemsets = apriori(df,
min_support=0.6,
use_colnames=True)
```

	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)
5	0.8	(Eggs, Kidney Beans)
6	0.6	(Onion, Eggs)
7	0.6	(Milk, Kidney Beans)
8	0.6	(Onion, Kidney Beans)
9	0.6	(Yogurt, Kidney Beans)
10	0.6	(Onion, Eggs, Kidney Beans)

https://tinyurl.com/aintpupython101

association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)

```
1 association_rules(frequent_itemsets, metric="confidence", min_threshold=0.7)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Eggs)	(Kidney Beans)	0.8	1.0	0.8	1.00	1.00	0.00	inf
1	(Kidney Beans)	(Eggs)	1.0	0.8	0.8	0.80	1.00	0.00	1.000000
2	(Onion)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
3	(Eggs)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
4	(Milk)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
5	(Onion)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
6	(Yogurt)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
7	(Onion, Eggs)	(Kidney Beans)	0.6	1.0	0.6	1.00	1.00	0.00	inf
8	(Onion, Kidney Beans)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
9	(Eggs, Kidney Beans)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
10	(Onion)	(Eggs, Kidney Beans)	0.6	0.8	0.6	1.00	1.25	0.12	inf
11	(Eggs)	(Onion, Kidney Beans)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000

```
rules =
association_rules(frequent_itemsets,
metric="lift", min_threshold=1.2)
rules
```

```
rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1.2)
rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(Onion)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
1	(Eggs)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
2	(Onion, Kidney Beans)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf
3	(Eggs, Kidney Beans)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000
4	(Onion)	(Eggs, Kidney Beans)	0.6	0.8	0.6	1.00	1.25	0.12	inf
5	(Eggs)	(Onion, Kidney Beans)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000

```
rules["antecedent_len"] =
rules["antecedents"].apply(lambda x: len(x))
rules
```

```
rules["antecedent_len"] = rules["antecedents"].apply(lambda x: len(x))
rules
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len
0	(Onion)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf	1
1	(Eggs)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000	1
2	(Onion, Kidney Beans)	(Eggs)	0.6	0.8	0.6	1.00	1.25	0.12	inf	2
3	(Eggs, Kidney Beans)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000	2
4	(Onion)	(Eggs, Kidney Beans)	0.6	0.8	0.6	1.00	1.25	0.12	inf	1
5	(Eggs)	(Onion, Kidney Beans)	0.8	0.6	0.6	0.75	1.25	0.12	1.600000	1

```
rules[ (rules['antecedent_len'] >= 2) &
       (rules['confidence'] > 0.75) &
       (rules['lift'] > 1.2) ]
```

```
rules[ (rules['antecedent_len'] >= 2) &
       (rules['confidence'] > 0.75) &
       (rules['lift'] > 1.2) ]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len
2	(Onion, Kidney Beans)	(Eggs)	0.6	0.8	0.6	1.0	1.25	0.12	inf	2

```
rules[rules['antecedents'] ==
{'Eggs', 'Kidney Beans'}]
```

```
1 rules[rules['antecedents'] == {'Eggs', 'Kidney Beans'}]
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	antecedent_len
3	(Eggs, Kidney Beans)	(Onion)	0.8	0.6	0.6	0.75	1.25	0.12	1.6	2

Wes McKinney (2017), "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", 2nd Edition, O'Reilly Media.

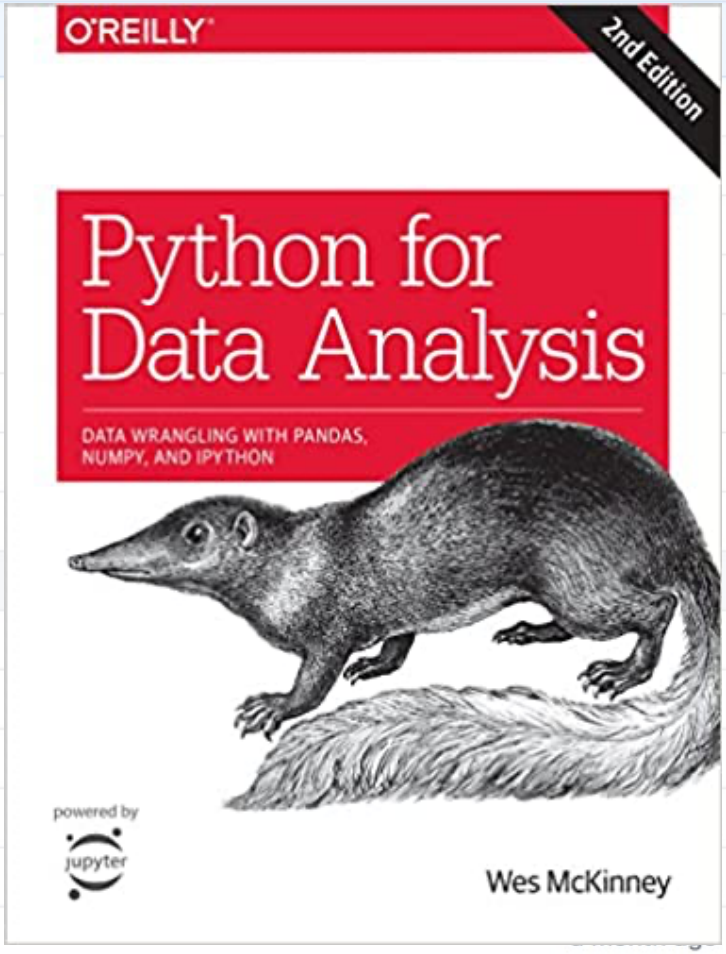
Materials and IPython notebooks for "Python for Data Analysis" by Wes McKinney, published by O'Reilly Media

52 commits | 2 branches | 0 releases | 6 contributors

Branch: 2nd-edition | New pull request | Find file | Clone or download

betatim committed with wesm Add requirements (#71)

datasets	Add Kaggle titanic dataset
examples	Remove sex column from tips dataset
.gitignore	Add gitignore
COPYING	Use MIT license for code examples
README.md	Add launch in Azure Notebooks button (#70)
appa.ipynb	Make more cells markdown instead of raw
ch02.ipynb	Make more cells markdown instead of raw
ch03.ipynb	Make more cells markdown instead of raw
ch04.ipynb	Convert all notebooks to v4 format
ch05.ipynb	Make more cells markdown instead of raw
ch06.ipynb	Make more cells markdown instead of raw
ch07.ipynb	Convert all notebooks to v4 format
ch08.ipynb	Make more cells markdown instead of raw
ch09.ipynb	Make more cells markdown instead of raw
ch10.ipynb	Make more cells markdown instead of raw

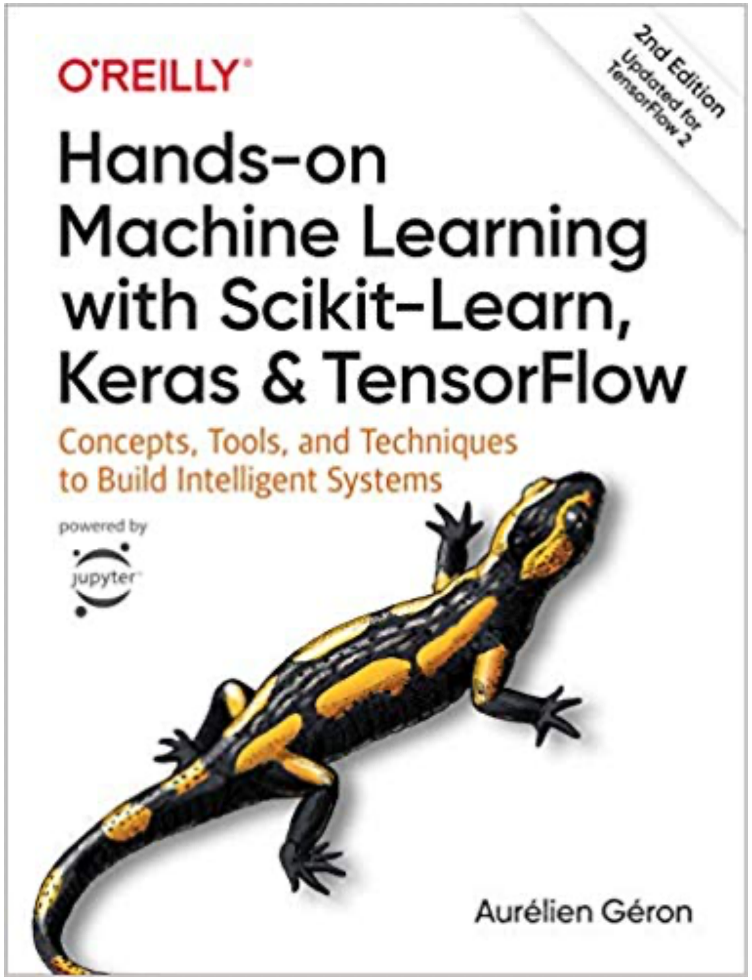


https://github.com/wesm/pydata-book

Aurélien Géron (2019),

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition

O'Reilly Media, 2019



https://github.com/ageron/handson-ml2

Source: https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1492032646/

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

Notebooks

1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. Deep Computer Vision Using Convolutional Neural Networks
15. Processing Sequences Using RNNs and CNNs
16. Natural Language Processing with RNNs and Attention
17. Representation Learning Using Autoencoders
18. Reinforcement Learning
19. Training and Deploying TensorFlow Models at Scale

https://github.com/ageron/handson-ml2

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

Table of contents

+ Code + Text RAM Disk Editing

Association Rules Generation from Frequent Itemsets

- mlxtend: https://rasbt.github.io/mlxtend/user_guide/frequent_patterns/association_rules/

```
# !pip install mlxtend
import pandas as pd
from mlxtend.preprocessing import TransactionEncoder
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
dataset = [['Milk', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Dill', 'Onion', 'Nutmeg', 'Kidney Beans', 'Eggs', 'Yogurt'],
           ['Milk', 'Apple', 'Kidney Beans', 'Eggs'],
           ['Milk', 'Unicorn', 'Corn', 'Kidney Beans', 'Yogurt'],
           ['Corn', 'Onion', 'Onion', 'Kidney Beans', 'Ice cream', 'Eggs']]
te = TransactionEncoder()
te_ary = te.fit(dataset).transform(dataset)
df = pd.DataFrame(te_ary, columns=te.columns_)
frequent_itemsets = apriori(df, min_support=0.6, use_colnames=True)
frequent_itemsets
```

	support	itemsets
0	0.8	(Eggs)
1	1.0	(Kidney Beans)
2	0.6	(Milk)
3	0.6	(Onion)
4	0.6	(Yogurt)

https://tinyurl.com/aintpupython101

Summary

- Unsupervised Learning
- Association Analysis
- Market Basket Analysis
- Recommender System
- Apriori algorithm
  - Frequent Itemsets
  - Association Rules

References

- Jiawei Han and Micheline Kamber (2006), Data Mining: Concepts and Techniques, Second Edition, Elsevier, 2006.
- Jiawei Han, Micheline Kamber and Jian Pei (2011), Data Mining: Concepts and Techniques, Third Edition, Morgan Kaufmann 2011.
- Efraim Turban, Ramesh Sharda, Dursun Delen (2011), Decision Support and Business Intelligence Systems, Ninth Edition, Pearson.
- Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Robert Layton (2017), Learning Data Mining with Python - Second Edition, Packt Publishing.
- Wes McKinney (2017), "Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython", 2nd Edition, O'Reilly Media.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O’Reilly Media.
  https://github.com/wesm/pydata-book
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101