iykra

HOME CREDIT
Kamu Bisa!

Special Crafted by

Bernardus Ari Kuncoro

Home Credit Data Science Bootcamp

Applied Machine Learning: Marketing

In this class, we will learn and code the machine learning application in marketing that cover market basket analysis using apriori algorithm, collaborative filtering, content-based filtering and hybrid.

Updated
Q4.2019

Trainer

Hi! I am **Bernardus Ari Kuncoro (Ari)**,
Head of Analytics COE at IYKRA.

My background is Electrical Engineering and Computer Science. In recent 5 years I have worked as a Data scientist in consultancy, ecommerce, and telecommunication companies. I absolutely and utterly passionate about Data Science and teaching, thus I am looking forward to sharing my knowledge with you! Please connect with me via the following digital platforms.

http://arikuncoro.xyz

@arikunc0r0

Bernardus Ari Kuncoro

Agenda

Session 1: Overview of Market Basket Analysis, Apriori Algorithm (90")

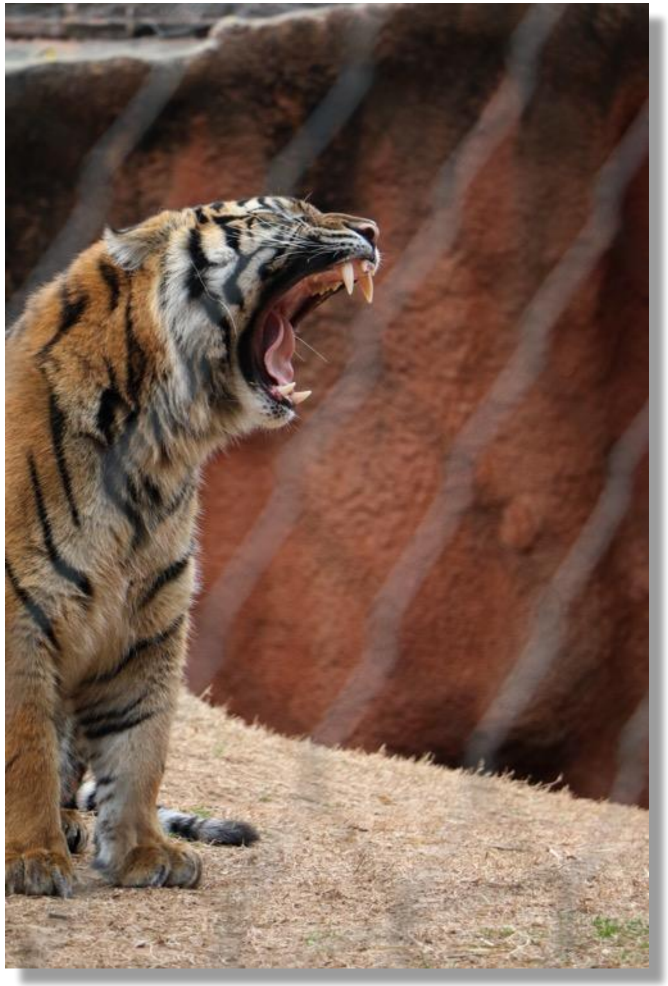
Session 2: Collaborative Filtering (60"), Practice (60")

Lunch break

Session 3: Content Based Filtering (60"), Hybrid Algorithm (60"), Practice (60")

Session 4: Exercise

Let's set the rule: ROAR



Respect Time
One focus at a time
Actively Participate
Respect Others

Objectives

To understand the concept of market basket analysis

To understand the concept and hands on the Apriori Algorithm

To understand the concept and hands on the collaborative filtering

To understand the concept and hands on the content based filtering

Session 1

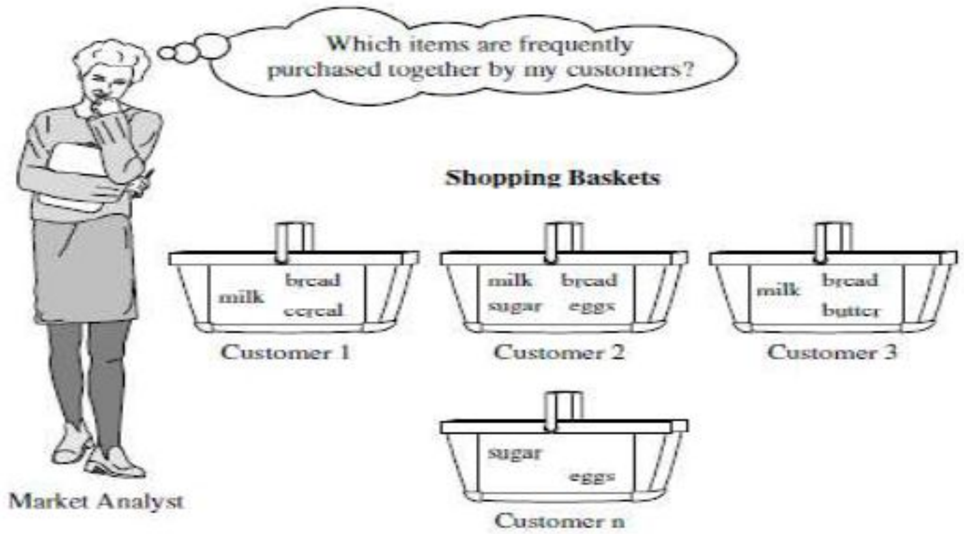
Overview of Market Basket Analysis Apriori Algorithm

Why Market Basket Analysis?

- A large number of people **buy products online and offline**.
- **Massive amounts of data** continuously being **collected and stored** as transactions
- Those **collected data** can be **very useful** from the business prospective

Objectives

- To find frequently purchased item sets from large transactional database
- To determine what products customers purchase together.



Business Application

- A store could use this information **to place products frequently sold together** into the same area
- An ecommerce/online shop merchant could use it to determine **the layout of their catalog** and order form.
- Direct marketers could use the basket analysis results to **determine what new products to offer** their prior customers.

What is Market Basket Analysis?

- The process of discovering frequent item sets in large transactional database is called market basket analysis.
- Frequent item set mining leads to the discovery of associations and correlations among items.

Apriori Algorithm

- One of the algorithm for market basket analysis
- Proposed by Agrawal and Srikant in 1994
- Designed to operate on databases containing transactions
- To understand this algorithm, terminologies you should know: support, confidence, lift and conviction. (we'll discuss later)

Main Terminologies

Terminologies: Support, Confidence, Lift and Conviction

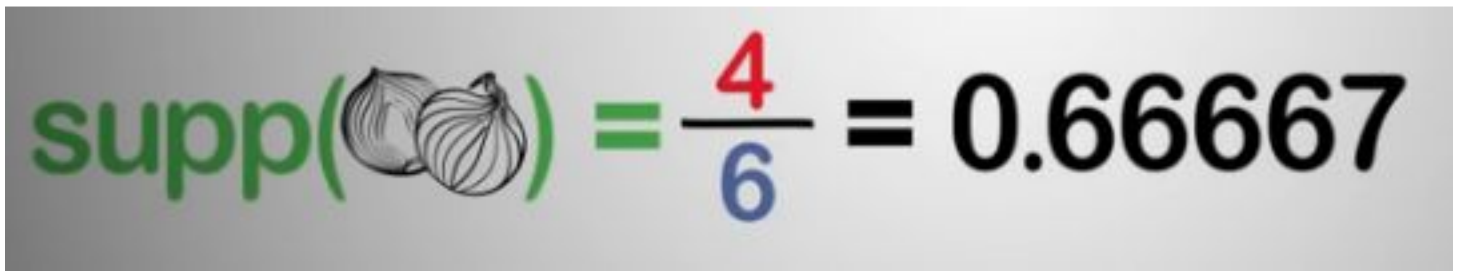
Transaction ID	Onion	Potato	Burger	Milk	Beer
t1	1	1	1	0	0
t2	0	1	1	1	0
t3	0	0	0	1	1
t4	1	1	0	1	0
t5	1	1	1	0	1
t6	1	1	1	1	1

Rule: Onion + Potato Burger

Rule: Potato + Burger Milk

Support

$$\text{supp}(X) = \frac{\text{Number of transactions in which X appears}}{\text{Total Number of transactions}}$$



Support is an indication of how frequently the itemset appears in the dataset.

Confidence

$$\mathrm{conf}(X \dashrightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X)}$$

Confidence is an indication of how often the rule has been found to be true.

Lift

$$lift(X \dashrightarrow Y) = \frac{supp(X \cup Y)}{supp(X) * supp(Y)}$$

> 1 Y is likely to be bought with X

the ratio of the observed support to that expected if X and Y were independent.

Conviction

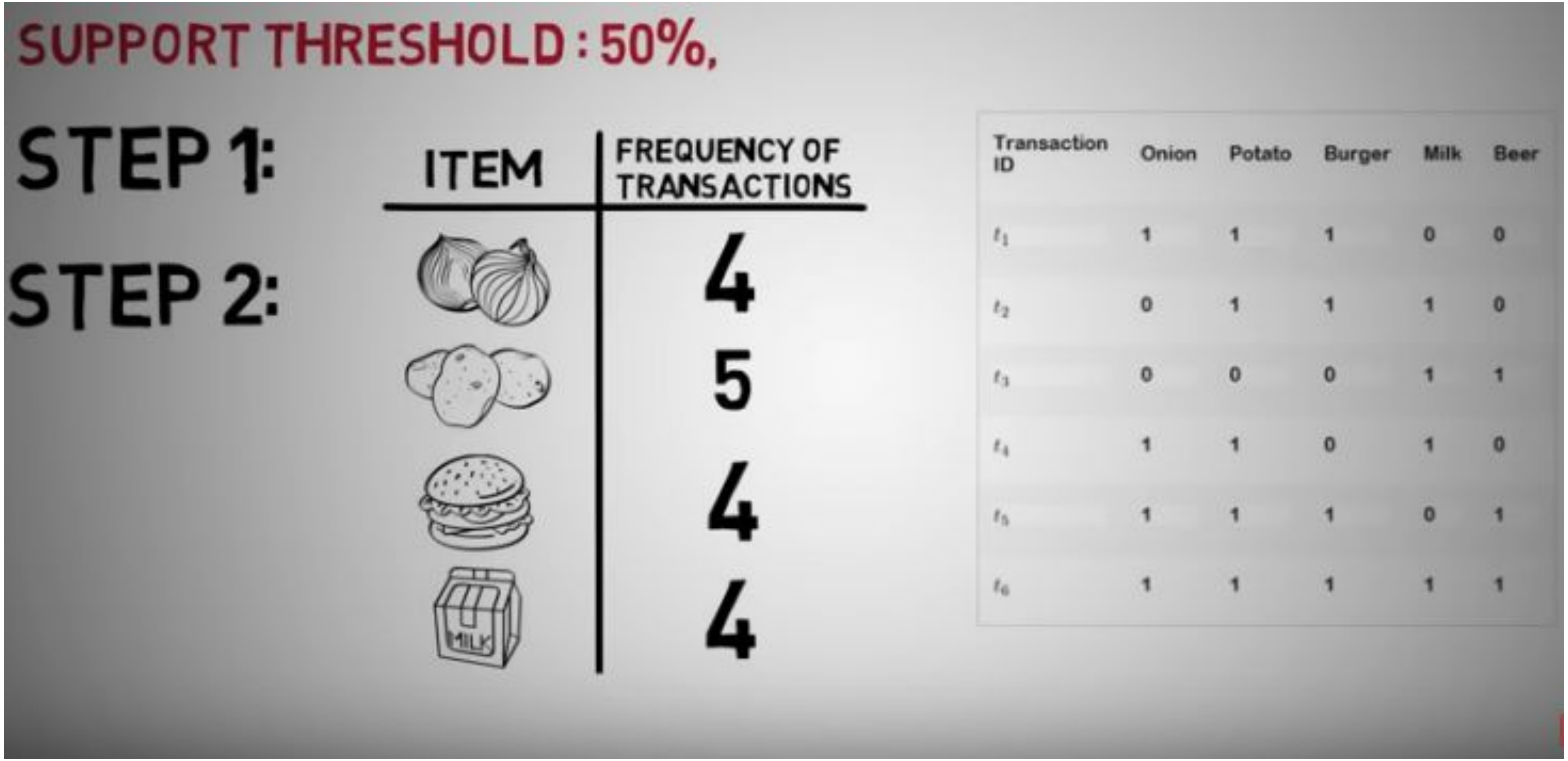
$$\mathrm{conv}(X \dashrightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \dashrightarrow Y)}$$

The ratio of the expected frequency that X occurs without Y

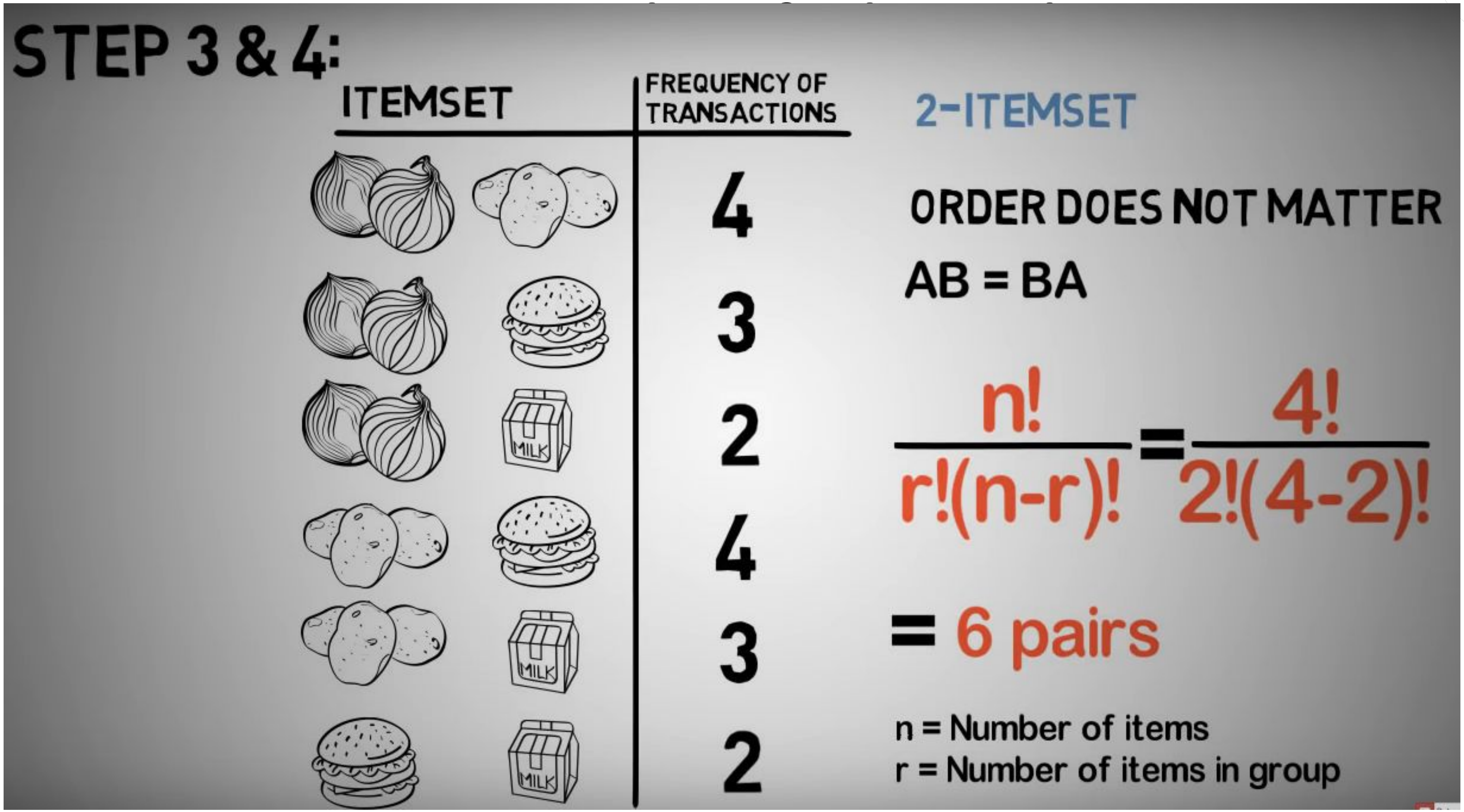
Example of Algorithm

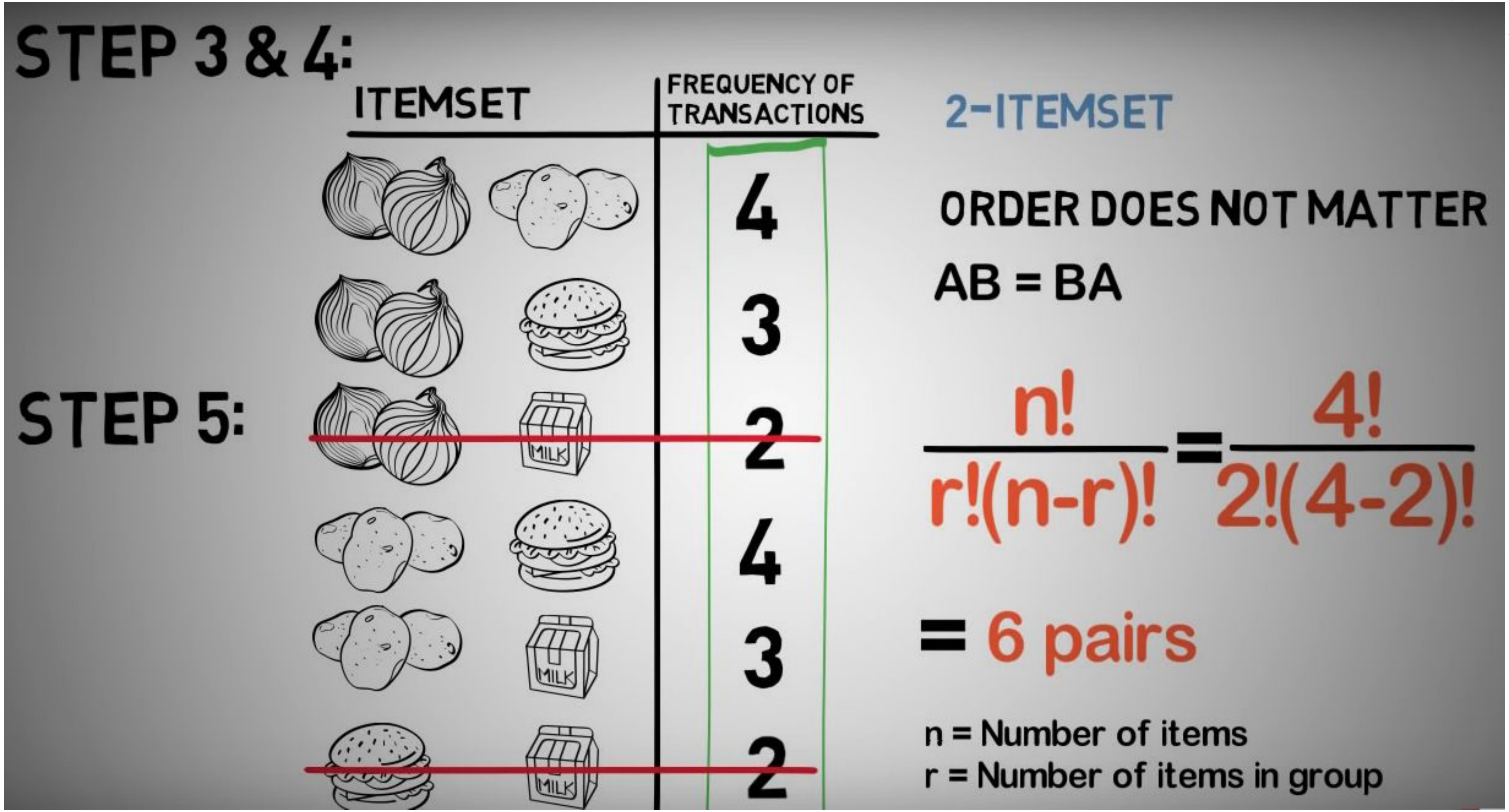
Transaction ID	Onion	Potato	Burger	Milk	Beer
t_1	1	1	1	0	0
t_2	0	1	1	1	0
t_3	0	0	0	1	1
t_4	1	1	0	1	0
t_5	1	1	1	0	1
t_6	1	1	1	1	1

Example of Algorithm



Transaction ID	Onion	Potato	Burger	Milk	Beer
t_1	1	1	1	0	0
t_2	0	1	1	1	0
t_3	0	0	0	1	1
t_4	1	1	0	1	0
t_5	1	1	1	0	1
t_6	1	1	1	1	1

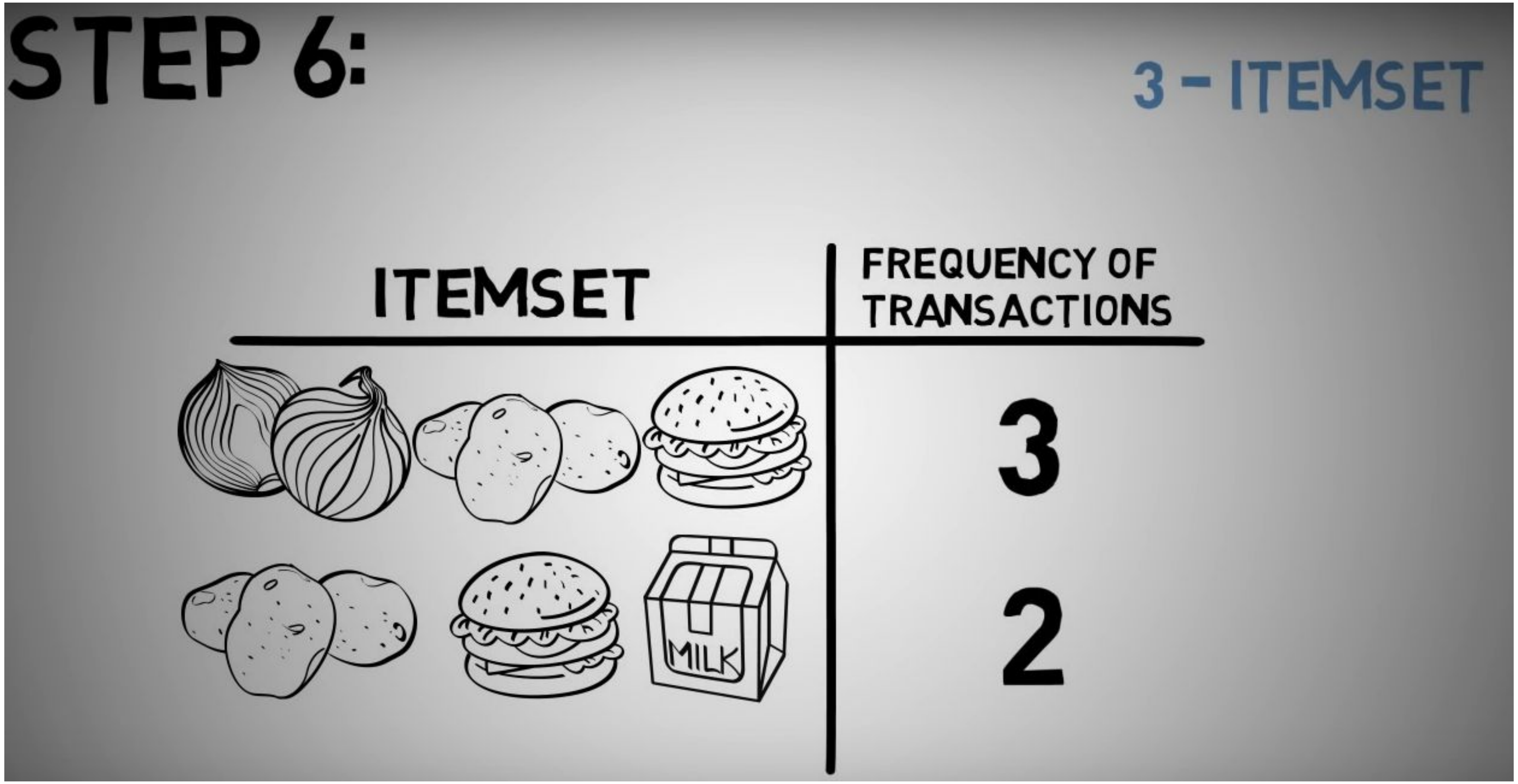

STEP 3 & 4:
ITEMSET
FREQUENCY OF TRANSACTIONS
4
3
2
4
3
2
2-ITEMSET
ORDER DOES NOT MATTER
AB = BA
$\frac{n!}{r!(n-r)!} = \frac{4!}{2!(4-2)!}$
= 6 pairs
n = Number of items
r = Number of items in group



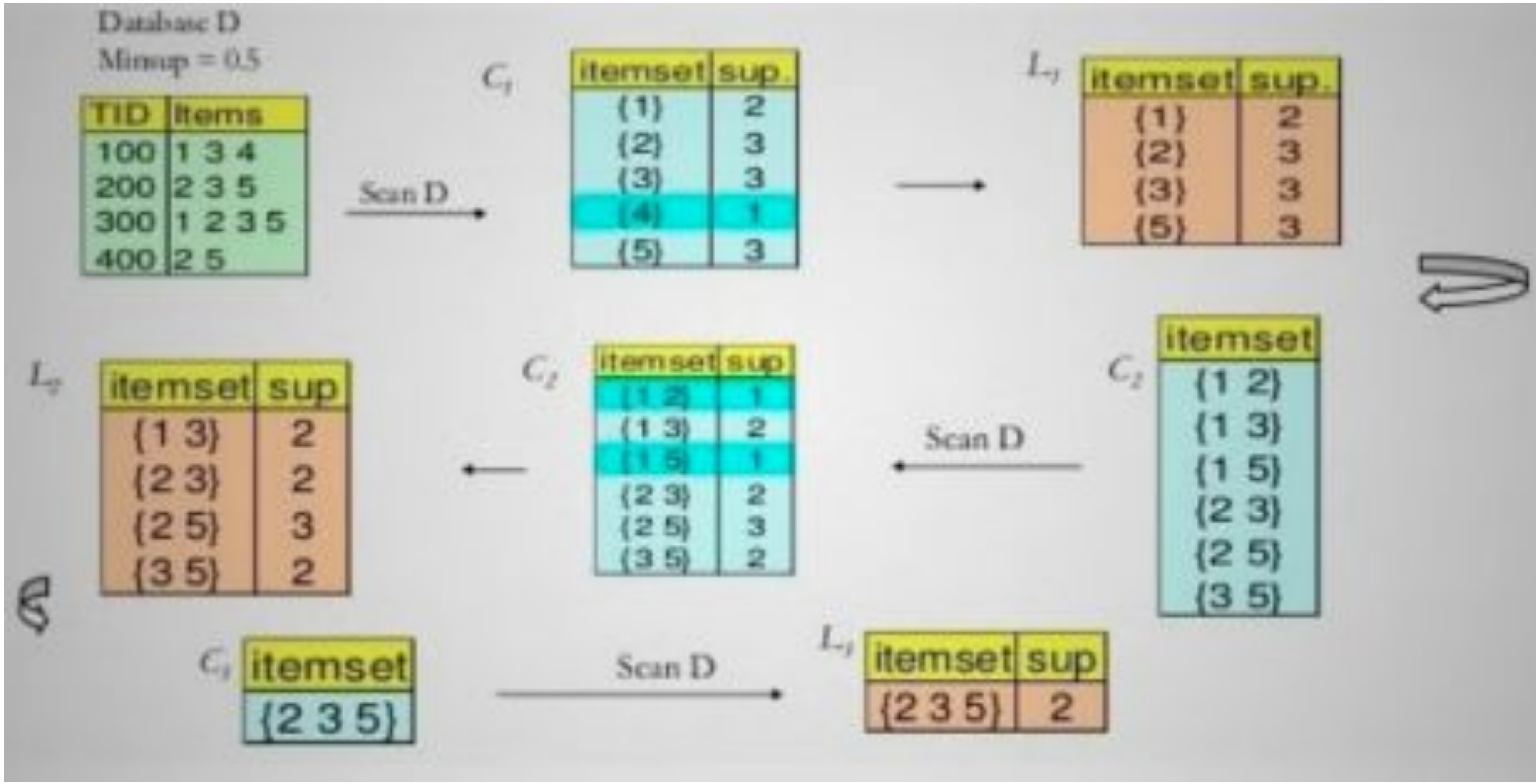
STEP 3 & 4:
ITEMSET
FREQUENCY OF TRANSACTIONS
4
3
MILK
2
STEP 5:
4
MILK
3
MILK
2
2-ITEMSET
ORDER DOES NOT MATTER
AB = BA
$\frac{n!}{r!(n-r)!} = \frac{4!}{2!(4-2)!}$
= 6 pairs
n = Number of items
r = Number of items in group

iykra

HOME CREDIT
Kamu Bisa!



Summary of Apriori Algorithm



Pros and Cons of Apriori Algorithm

+ Easy to understand
+ Easy to implement
+ It can be used for large dataset and easy to be parallelized

- Computationally expensive in choosing the sets and support.

iykra

HOME CREDIT
Kamu Bisa!

Pick at least 3 items on your basket!

1

LOGITECH MX Master 2S Wireless Mouse [910-...

~~Rp 1.461.000~~ -27%

Rp 1.069.000

2

XIAOMI Redmi 5 (32GB/3GB) Black

~~Rp 2.199.000~~ -9%

Rp 1.999.000

3

TP-LINK 300Mbps Wireless N Speed Router TL-...

Rp 166.500

4

GOPRO HERO7 Black

~~Rp 6.999.000~~ -14%

Rp 5.999.000

5

FITBIT Versa Special Edition Ruby Rose Gold

Rp 4.299.000

5

YONEX Nanoray Z-Speed Lime Yellow

Rp 2.110.000

6

Alesis Sample Pad Pro

~~Rp 5.850.000~~ -29%

Rp 4.150.000

7

DIADORA Backpack 81101 [DIABPU81101N] - Navy

~~Rp 259.000~~ -25%

Rp 195.000

8

NIVEA Nivea Men White Oil Clear Anti-Shine Facial...

Rp 29.000

9

GRANDE Acoustic Electric Guitar GCE-8NA Natural

Rp 1.350.000

Python code of apriori

https://github.com/coorty/apriori-agorithm-python

Session 2

Collaborative Filtering

Concept of Collaborative Filtering

Collaborative Filtering
The process of information filtering by collecting human judgments (ratings)
“word of mouth”

User
Any individual who provides ratings to a system

Items
Anything for which a human can provide a rating

Let's fill up this survey

http://bit.ly/HCI-Movie

Concept of Collaborative Filtering

	Star Wars	Hoop Dreams	Contact	Titanic
Joe	5	2	5	4
John	2	5		3
Al	2	2	4	2
Nathan	5	1	5	?

The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given a set of historical preference judgments for a community of users.

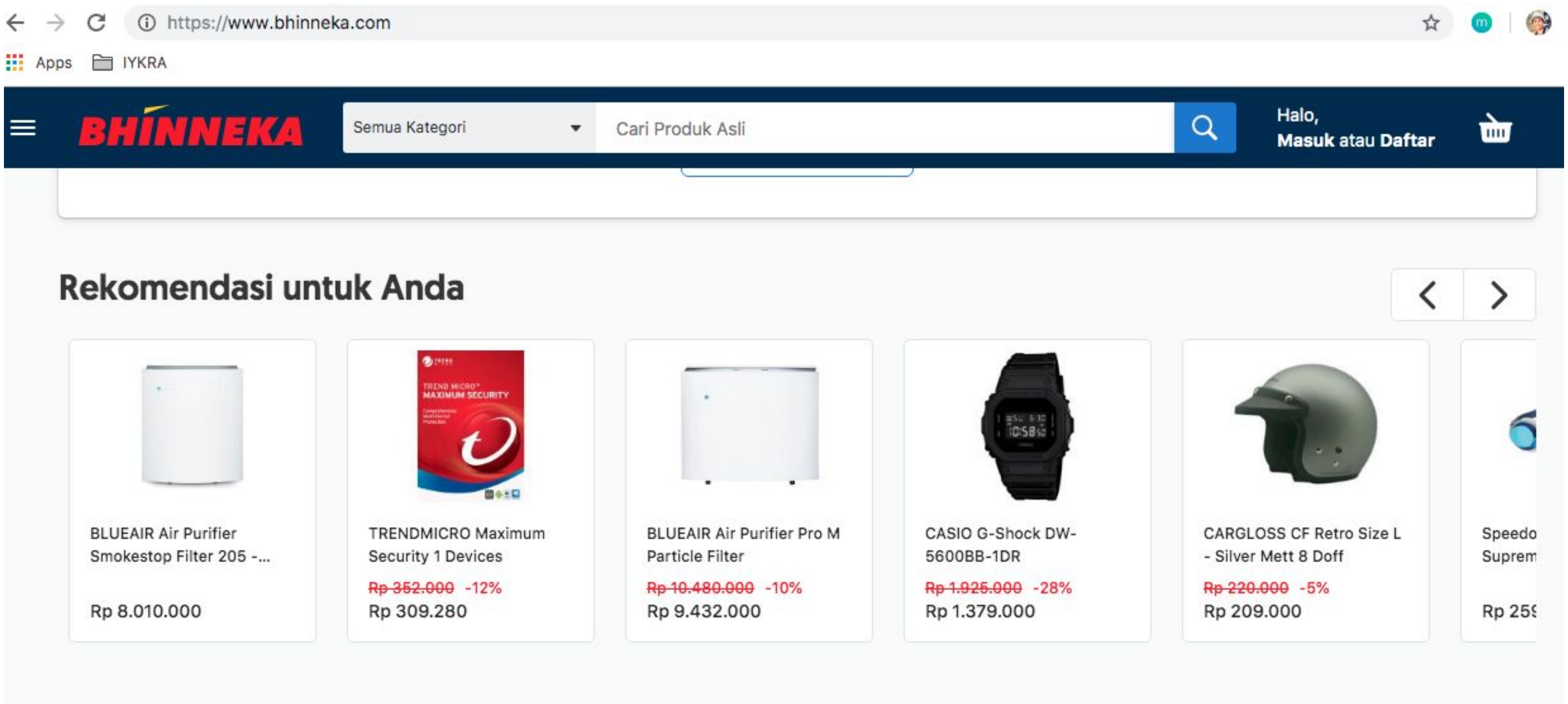
Uses for CF : User Tasks

- What tasks users may wish to accomplish
  - Help me find new items I might like
  - Advise me on a particular item
  - Help me find a user (or some users) I might like
  - Help our group find something new that we might like
  - Domain-specific tasks
  - Help me find an item, new or not

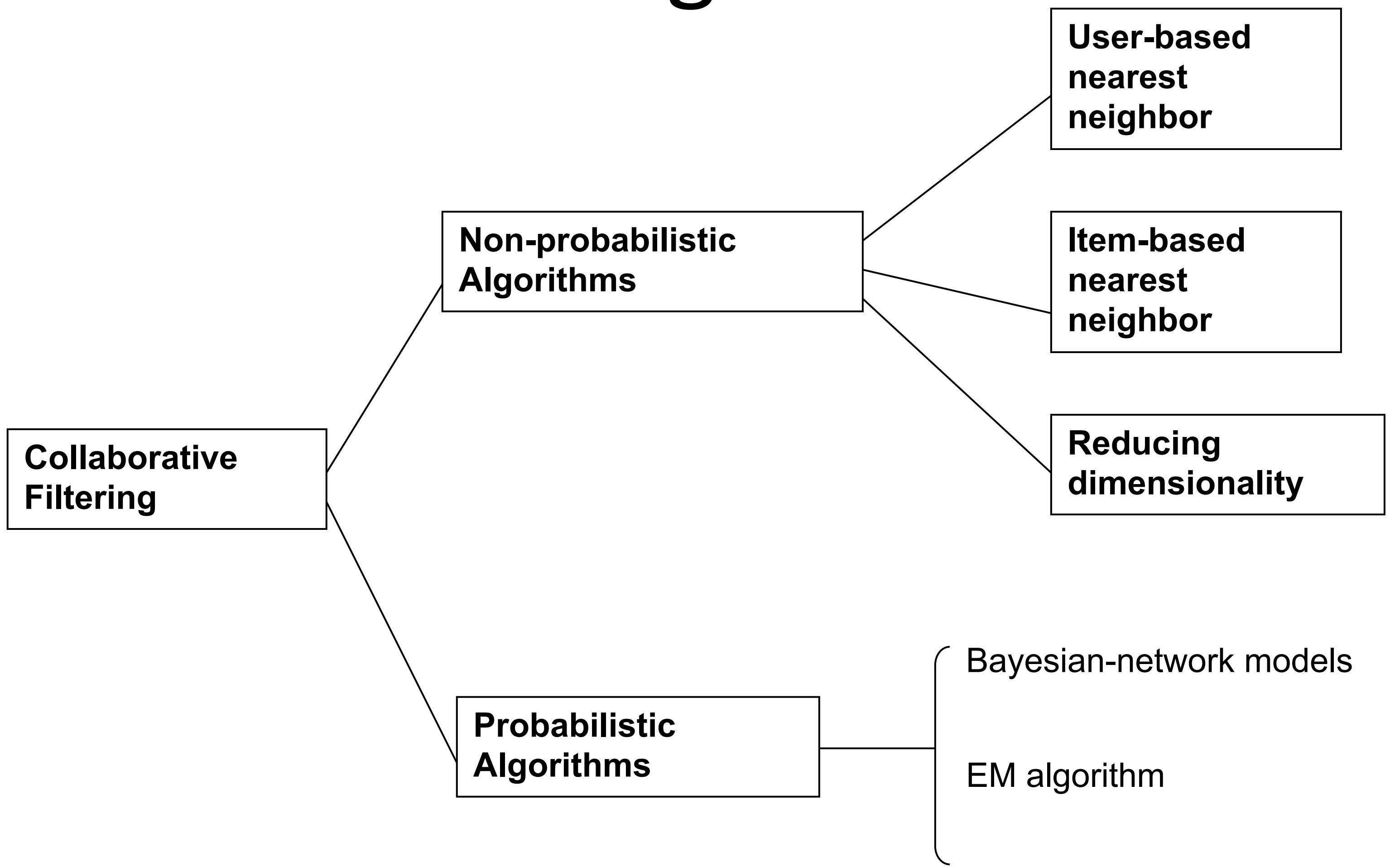
Uses for CF : System Tasks

- What CF systems support
  - Recommend items
    - Eg. Amazon.com, Bhinneka.com
  - Predict for a given item
  - Constrained recommendations
    - Recommend from a set of items

Bhinneka.Com Recommendation



Algorithms


Collaborative Filtering
Non-probabilistic Algorithms
User-based nearest neighbor
Item-based nearest neighbor
Reducing dimensionality
Probabilistic Algorithms
Bayesian-network models
EM algorithm

Algorithms : Non-probabilistic

- **User-Based Nearest Neighbor**
  - Neighbor = similar users
  - Generate a prediction for an item *i* by analyzing ratings for *i* from users in *u*'s neighborhood

$$pred(u,i) = \bar{r}_u + \frac{\sum_{n \subset neighbors(u)} sim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \subset neighbors(u)} sim(u,n)}$$

Algorithms : Non-probabilistic

- **Item-Based Nearest Neighbor**
  - Generate predictions based on similarities between items.
  - Prediction for a user *u* and item *i* is composed of a weighted sum of the user *u*'s ratings for items most similar to *i*.

$$pred(u,i) = \frac{\sum_{j \in ratedItems(u)} sim(i,j) \cdot r_{ui}}{\sum_{j \in ratedItems(u)} sim(i,j)}$$

Algorithms : Non-probabilistic

- Dimensionality Reduction
  - Reduce domain complexity by mapping the item space to a smaller number of underlying dimensions.
  - Dimension may be latent topics or tastes.
  - Vector-based techniques
    - Vector decomposition
    - Principal component analysis
    - Factor analysis

Algorithms : Probabilistic

- Represent probability distributions
- Given a user *u* and a rated item *i*, the user assigned the item a rating of *r* : *p(r|u, i)*.

$$E(r \mid u,i) = \sum_r r \cdot p(r \mid u,i)$$

- Bayesian-network models, Expectation maximization (EM) algorithm

Practical Issues : Ratings

- Rating Scales
  - Scalar ratings
    - Numerical scales
    - 1-5, 1-7, etc.
  - Binary ratings
    - Agree/Disagree, Good/Bad, etc.
  - Unary ratings
    - Good, Purchase, etc.
    - Absence of rating indicates no information

Practical Issues : Cold Start

- New user
  - Rate some initial items
  - Non-personalized recommendations
  - Describe tastes
  - Demographic info.
- New Item
  - Non-CF : content analysis, metadata
  - Randomly selecting items
- New Community
  - Provide rating incentives to subset of community
  - Initially generate non-CF recommendation
  - Start with other set of ratings from another source outside community

Evaluation Metrics

- Accuracy
  - Predict accuracy
    - The ability of a CF system to predict a user's rating for an item
    - Mean absolute error (MAE)
  - Rank accuracy
    - Precision – percentage of items in a recommendation list that the user would rate as useful
    - Half-life utility – percentage of the maximum utility achieved by the ranked list in question

Evaluation Metrics

- Novelty
  - The ability of a CF system to recommend items that the user was not already aware of.
- Serendipity
  - Users are given recommendations for items that they would not have seen given their existing channels of discovery.
- Coverage
  - The percentage of the items known to the CF system for which the CF system can generate predictions.

Evaluation Metrics

- Learning Rate
  - How quickly the CF system becomes an effective predictor of taste as data begins to arrive.
- Confidence
  - Ability to evaluate the likely quality of its predictions.
- User Satisfaction
  - By surveying the users or measuring retention and use statistics

Additional Issues : Privacy & Trust

- User profiles
  - Personalized information
- Distributed architecture

- Recommender system may break trust when malicious users give ratings that are not representative of their true preferences.

Additional Issues : Interfaces

- Explanation
  - Where, how, from whom the recommendations are generated.
  - Do not make it too much!
    - Not showing reasoning process
    - Graphs, key items
    - Reviews

Additional Issues : Interfaces

- Social Navigation
  - Make the behavior of community visible
  - Leaving “footprints” : read-wear / edit-wear
  - Attempt to mimic more accurately the social process of word-of-mouth recommendations
  - Epinions.com

Collaborative Filtering Code

https://github.com/aryankashyap0/collaborative-filtering-python

Session 3

Content-Based Filtering Hybrid

Content-based recommendation systems

- Use exclusively the history of the target user

- Items are described by features

e.g.: actors, director, category, words in the description

- Train a regression model for each of the user based on the content features

Content-based recommendation systems

ID	Name	Cuisine	Service	Cost
10001	Mike's Pizza	Italian	Counter	Low
10002	Chris's Cafe	French	Table	Medium
10003	Jacques Bistro	French	Table	High

iykra

HOME CREDIT
Kamu Bisa!

Content-based recommendation systems

- Independent from other users (no need for critical mass)
- Recommendation can be given for a single user
- The cold start problem is smaller
- No need for storing/handling a huge matrix
- It recommends from the long tail
- It can give you a „user model”

Content-based recommendation systems

- Feature engineering is domain-specific and requires external data collection
- The filter bubble problem:
  - The greatest predicted rating might be a wrong recommendation as it „overfits” to the user’s preferences
  - E.g. if the user rated only Hungarian and Chinese restaurants the system won’t recommend a Greek restaurant (even it’s the best in the town)
- A new user has to be modeled, i.e. a sufficient personal training data is needed

Additional Issues : Hybrid Approach

- CF + CB
- Content based system
  - Maintain user profile based on content analysis
- Collaborative system
  - Directly compare profiles to determine similar users for recommendation
- Fab system

Hybrid recommender systems

- Content-based → collaborative
  - We can use content-based prediction at users with many training examples and collaboration at others
  - The prediction of content-based models can be used in recursive collaborative filtering
- Collaborative → content-based
  - Features can be extracted from other users' ratings

Hybridisation (general schema)

A hybrid model of several individual models usually performs better than the best individual model (even weaker models can contribute)

- voting
  - Weights of votes can be calibrated on a validation set

- stacking
  - Predictions of the individual models can form features in a second-phase classifier

Python Code for Collaborative Filtering, Content Based + Hybrid

https://github.com/revantkumar/Collaborative-Filtering

Dataset that you can play with

https://www.kaggle.com/rounakbanik/the-movies-dataset

Summary

- Market Basket Analysis
- Apriori
- Collaborative Filtering
- Content Based Filtering
- CF + CB (Hybrid)

Session 4

Exercise

Exercise

Organize nicely your code that you've done today into one Github repository that contain 4 subtopics.

1. Market Basket Analysis with Apriori
2. Collaborative Filtering
3. Content Based
4. Hybrid

You may name your repository with: **Marketing Analytics**
Submit your answer (github link URL) on e-learning platform.

Further watching

Apriori Algorithm	https://www.youtube.com/watch?v=WGlMlS_Yydk
Recommendation System	https://www.youtube.com/watch?v=h9gpufJFF-0
Content Based Recommendation	https://www.youtube.com/watch?v=2uxXPzm-7FY
Collaborative Filtering	https://www.youtube.com/watch?v=1JRrCEgiyHM