



Machine Learning Applications

Recommendation Systems

What are Recommendation Systems?

- A Recommendation System is a type of information filtering system that predicts or suggests items (such as products, movies, music, or articles) to users based on their preferences, historical behavior, and similarities with other users.
- It utilizes various algorithms and data analysis techniques to generate personalized recommendations, aiming to enhance user experience, increase engagement, and facilitate decision-making processes.

Why are Recommendation Systems needed?

- **Information Overload:** Too many choices; users need help to filter out the noise and focus on items that tailored to their interests and preferences
- **Improved User Experience:** Save the users' time and effort in finding items that suit them
- **Discovery of New Items:** Suggest new and diverse items that users may not have encountered otherwise
- **Increased Engagement and Revenue:** User engagement can be significantly increased through recommendations, thus increasing the likelihood of conversions and purchases

- Editorial - New Arrivals/Releases, Featured

amazon.sg

Deliver to Tan Singapore 400009

All Search Amazon.sg

EN

Hello, Tan Account and Lists

Returns & Orders

Cart

All

Health & Personal Care

Fresh & Fast

Gift Cards

Best Sellers

Tan's Amazon.sg

Customer Service

Today's Deals

Computers

Fashion

Baby

Best Sellers

Hot New Releases

Movers and Shakers

Most Wanted For

Most Gifted

Amazon Hot New Releases

Our bestselling new and future releases. Updated frequently.

Any Category

Automotive

Baby Products

Beauty

Books

Computers

DIY & Tools

Electronics

Fashion

Garden

Health, Household and Personal Care

Home

Industrial & Scientific

Kitchen

Luxury Beauty

Movies & TV

Music

Musical Instruments

Office Products

Pet Supplies

Sporting Goods

Toys


Video Games

Hot New Releases in Musical Instruments

See More

Page 1 of 7


#1



Toyvian 1 Set Snare Drum Drum Percussion Instrument Drum Small Drum Children Drum

S\$13.79


#2



ARPWC Shure RPW204

S\$33.20

#3



Shure RPW204


S\$491.16

Hot New Releases in Toys

See More

Page 1 of 7

#1




MG Mobile Suit Gundam NT Narrative Gundam C Equipment Ver.Ka 1/100 Scale Color Coded Plastic Model

★★★★☆ 12

S\$66.49

#2




LEGO Animal Crossing Nook's Cranny & Rosie's House Creative Toy 77050 (535 Pieces)

★★★★★ 106

S\$79.00

#3



Nerf Elite Junior Racer Easy Play Dart Blaster, 4 Nerf Elite Darts, Nerf Blaster Outdoor Toys For 6 Year Old Boys & Girls & Up

S\$6.39

STEAM

STORE

COMMUNITY

ABOUT

SUPPORT

Install Steam

login

language

Your Store

New & Noteworthy

Categories

Points Shop

News


Labs

search

All Products > New Releases


NEW RELEASES

LIVE



Popular this month


S\$22.00



Popular this month


-10% S\$14.50 S\$13.05

NEW TOP SELLERS Released This Month




S\$26.00


LIVE



-10% S\$28.00 S\$26.10



-10% S\$14.50 S\$13.05



-25% S\$49.00 S\$36.75

- Simple Aggregation - Top Viewed / Liked / Comments, Trending

amazon.sg

Deliver to Tan Singapore 400009

All

Search Amazon.sg

EN

Hello, Tan

Account and Lists

Returns & Orders

0 Cart

AllHealth & Personal CareFresh & FastGift CardsBest SellersTan's Amazon.sgCustomer ServiceToday's DealsComputersFashionBaby

Best SellersHot New ReleasesMovers and ShakersMost Wishd ForMost Gifted

Amazon Best Sellers


Our most popular products based on sales. Updated frequently.

Any Category


- Automotive
- Baby Products
- Beauty
- Books
- Computers
- DIY & Tools
- Electronics
- Fashion
- Garden
- Gift Cards
- Grocery
- Health, Household and Personal Care
- Home
- Industrial & Scientific
- Kitchen
- Luxury Beauty
- Movies & TV
- Music
- Musical Instruments
- Office Products
- Pet Supplies
- Software
- Sporting Goods
- Toys
- Video Games

Best Sellers in Health, Household and Personal Care


See More

#1

RENPHO Body Fat Scale Smart BMI Scale Digital Bathroom Wireless Weight Scale, Body Composition Analyzer with Smartphone App...
★★★★☆ 282,953
S\$30.09

#2


LMNT Recharge Electrolyte Hydration Powder | Formulated by Robb Wolf and Ketogains | Keto & Paleo | No Sugar, No Artificial...
★★★★☆ 6,751
S\$97.50

#3


ASAKUKI 500ml Essential Oil Diffuser, Premium 5 In 1 Ultrasonic Aromatherapy Scented Oil Diffuser Vaporizer Humidifier, Timer and...
★★★★☆ 8,168
S\$26.64

Best Sellers in Office Products


See More

#1

Ohuhu Markers for Adult Coloring Books: 60 Colors Dual Brush Fine

#2

Moleskine Classic Notebook, Hard Cover, Large (5" x 8.25")

#3

Faber-Castell GW589150 90-Pieces Tack-it Adhesive, White, 50 g

STEAM

STORECOMMUNITYABOUTSUPPORT

Install Steamloginlanguage

Your StoreNew & NoteworthyCategoriesPoints ShopNewsLabs

search

Charts Overview

TOP SELLERS

Top 100 selling games right now, by revenue

Singapore

LIVE CHARTS

Top Sellers

Most Played

WEEKLY CHARTS

Apr 30, 2024

Apr 23, 2024

Apr 16, 2024




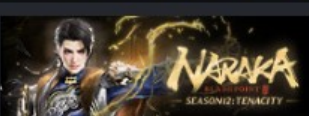




TOP NEW RELEASES

February 2024

January 2024

December 2023

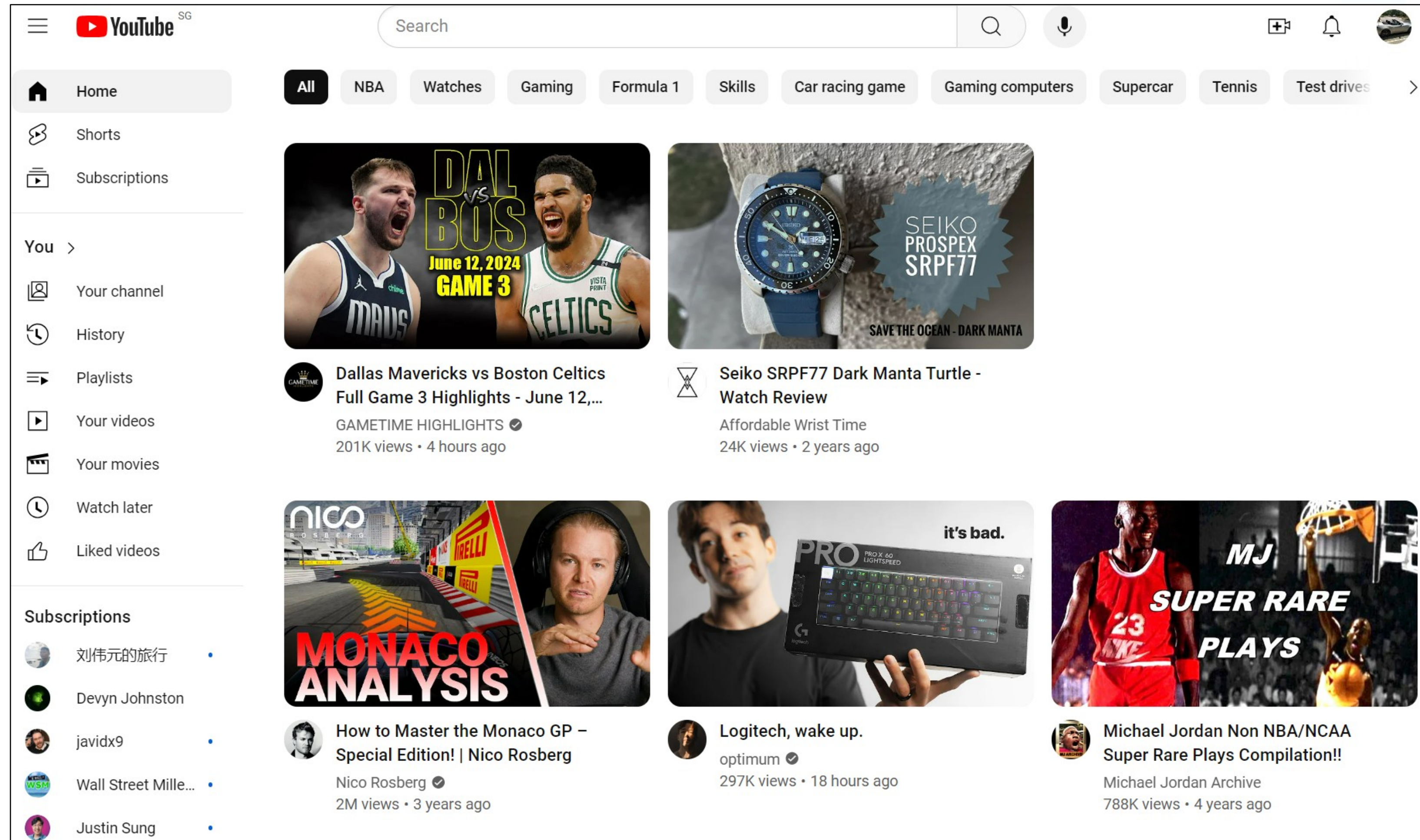
BEST OF YEAR

RANK		PRICE	CHANGE	WEEKS
1	 Hades II	NEW S\$26.00	NEW	1
2	 Counter-Strike 2	Free To Play	▲ 1	613
3	 HELLDIVERS™ 2	S\$54.90	▲ 1	17
4	 NARAKA: BLADEPOINT	Free To Play	▲ 6	147
5	 Dota 2	Free To Play	▼ 3	623
6	 V Rising	NEW -10% S\$29.00 S\$26.10	NEW	1
7	 Apex Legends™	Free To Play	▼ 2	183
8	 Rabbit and Steel	NEW -10% S\$14.50 S\$13.05	NEW	1

5

Types of Recommendations

- **Personalized** – Content-Based Filtering, Collaborative Filtering



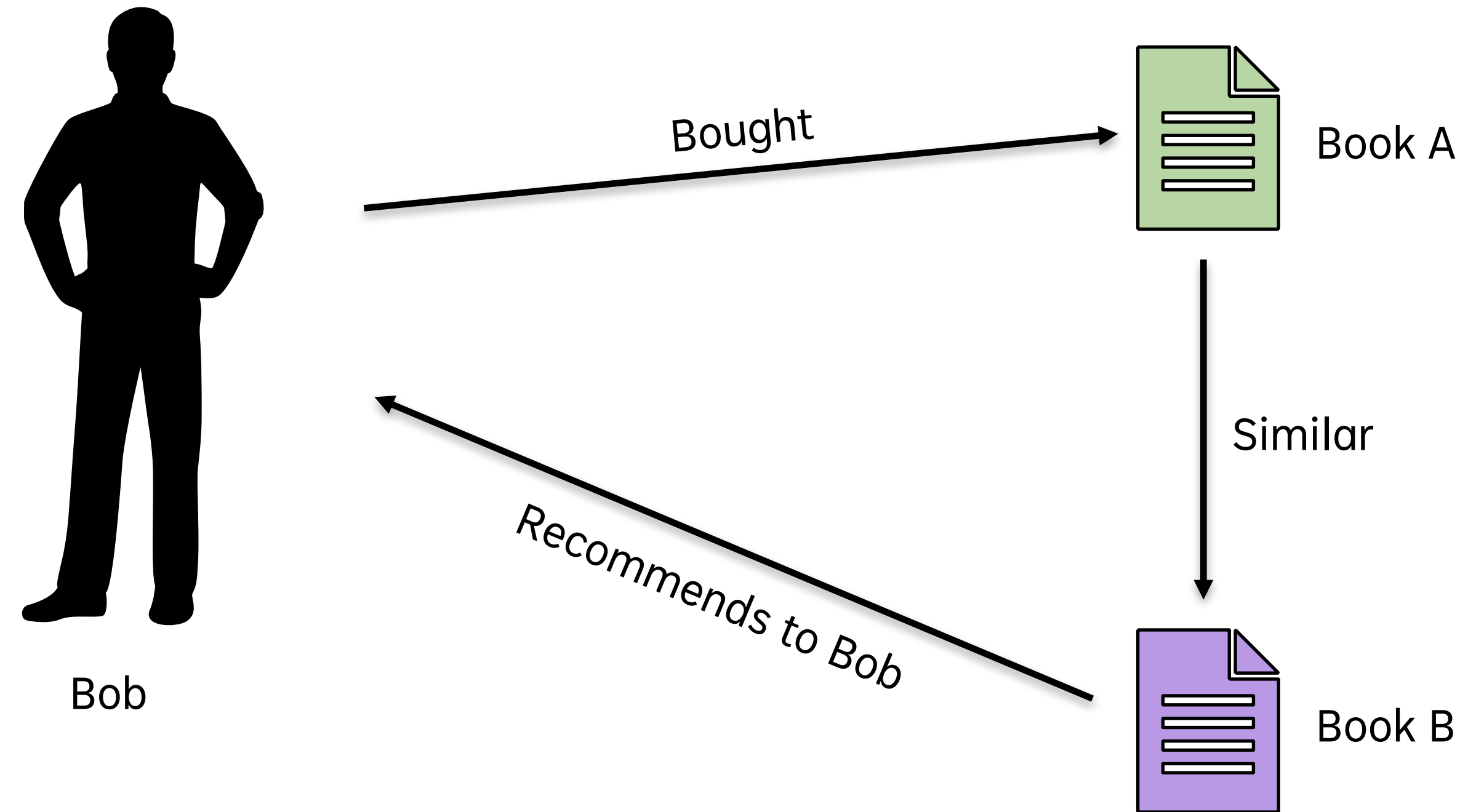
CONTENT-BASED FILTERING

Content-Based Filtering

- Content-based filtering recommends items to a user based on the **attributes (or features) of items** they have previously interacted with
- It focuses on **individual user-item interactions** and does not consider other users' behavior when making recommendations
- It is a more traditional, knowledge-based approach that leverages explicit data about the items being recommended

Content-Based Filtering

- For example, if Book A and Book B are similar, recommends Book B to Bob if he has bought Book A but not Book B

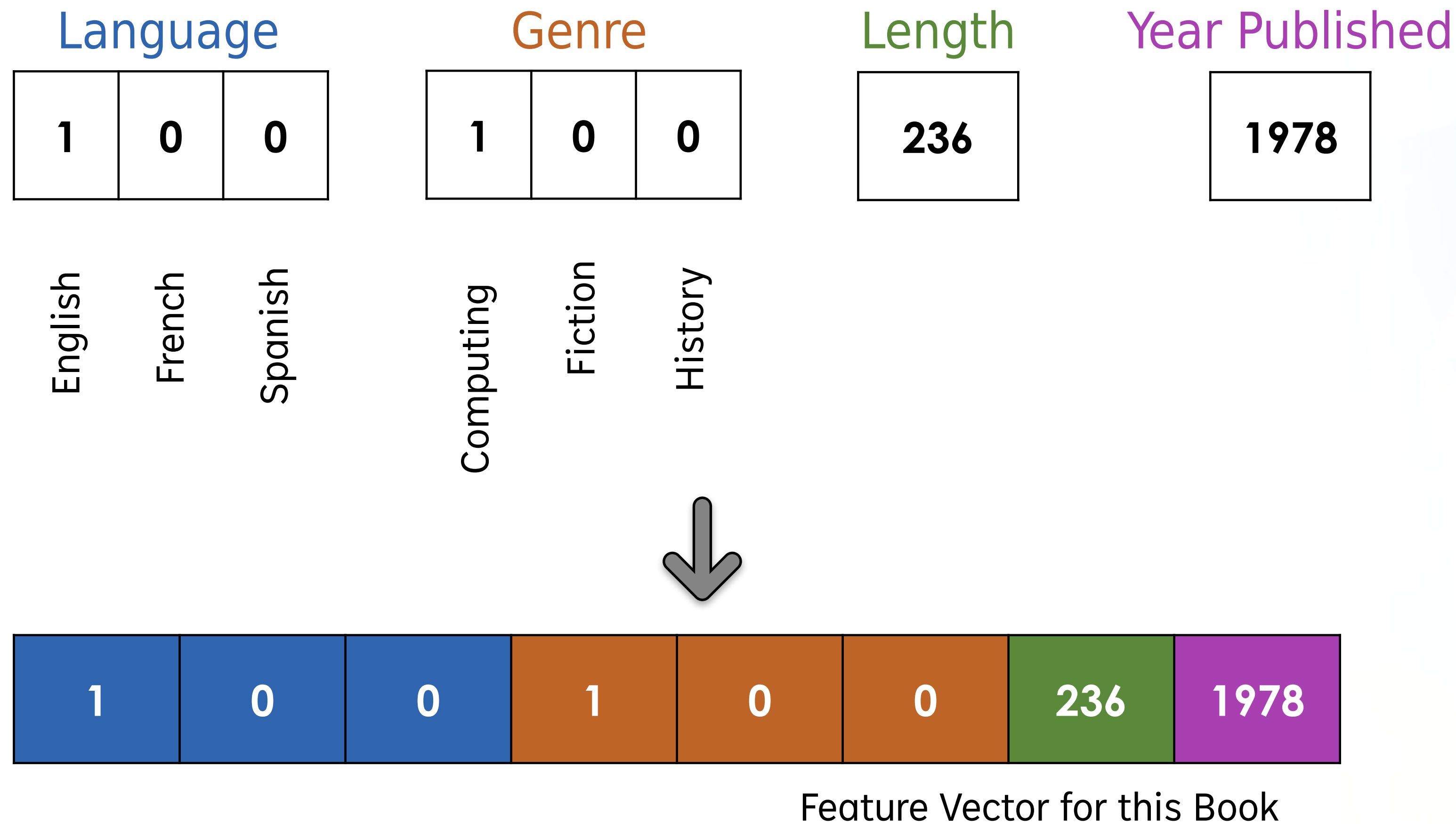


Possible Features for Books

- Language (e.g. English, French, Spanish)
- Genre (e.g. Computing, Fiction)
- Author (e.g. Dennis Ritchie)
- Length (e.g. 236 pages)
- Year Published (e.g. 1978)
- Publisher (e.g. Prentice Hall, Pearson, MIT Press)

Encoding to Feature Vectors

- A simple illustration on **encoding** of **features** for our books

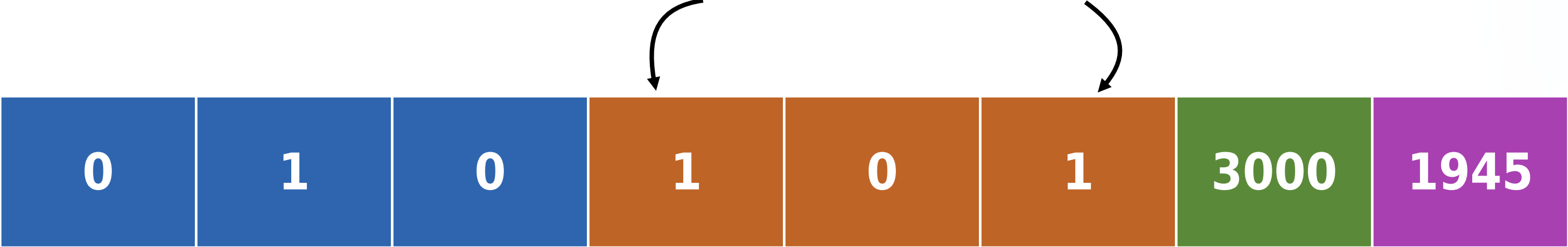


Feature Vectors for Books

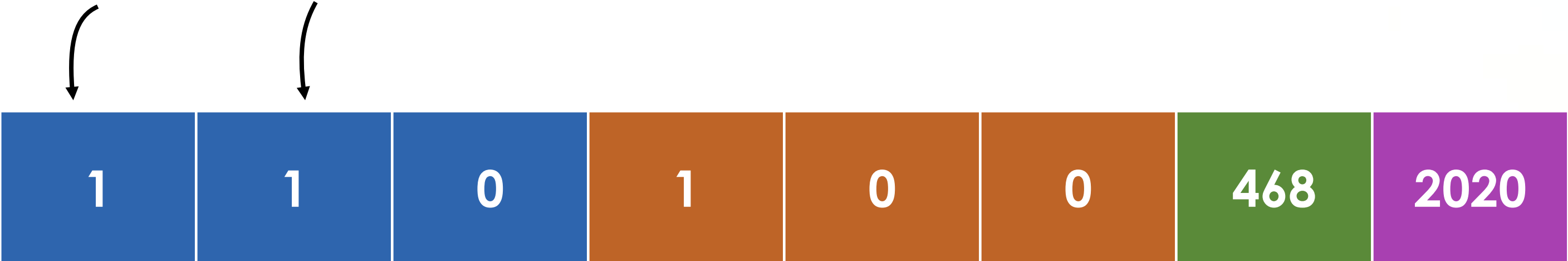
- Encoding all our books into their respective **Feature Vectors**



A book that is classified under "Computing" and "History"

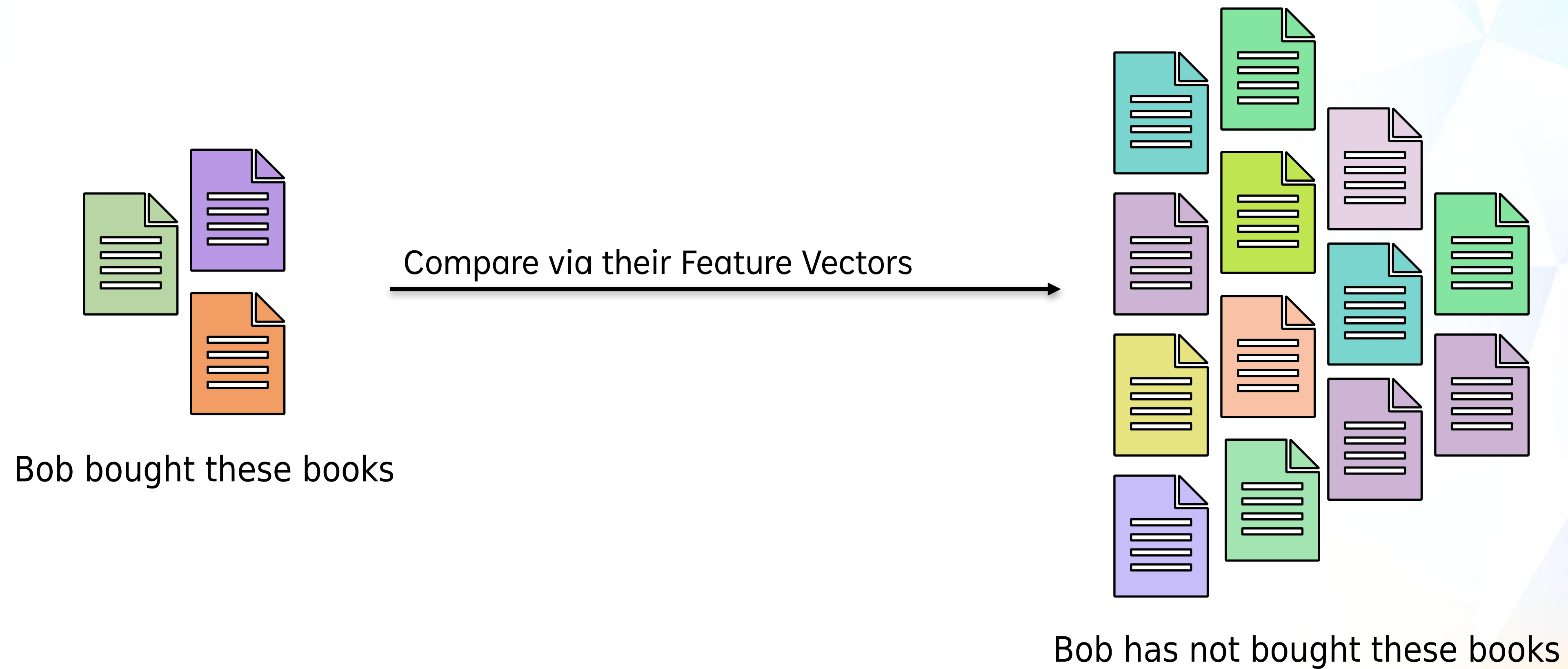


A book that has both "English" and "French" text

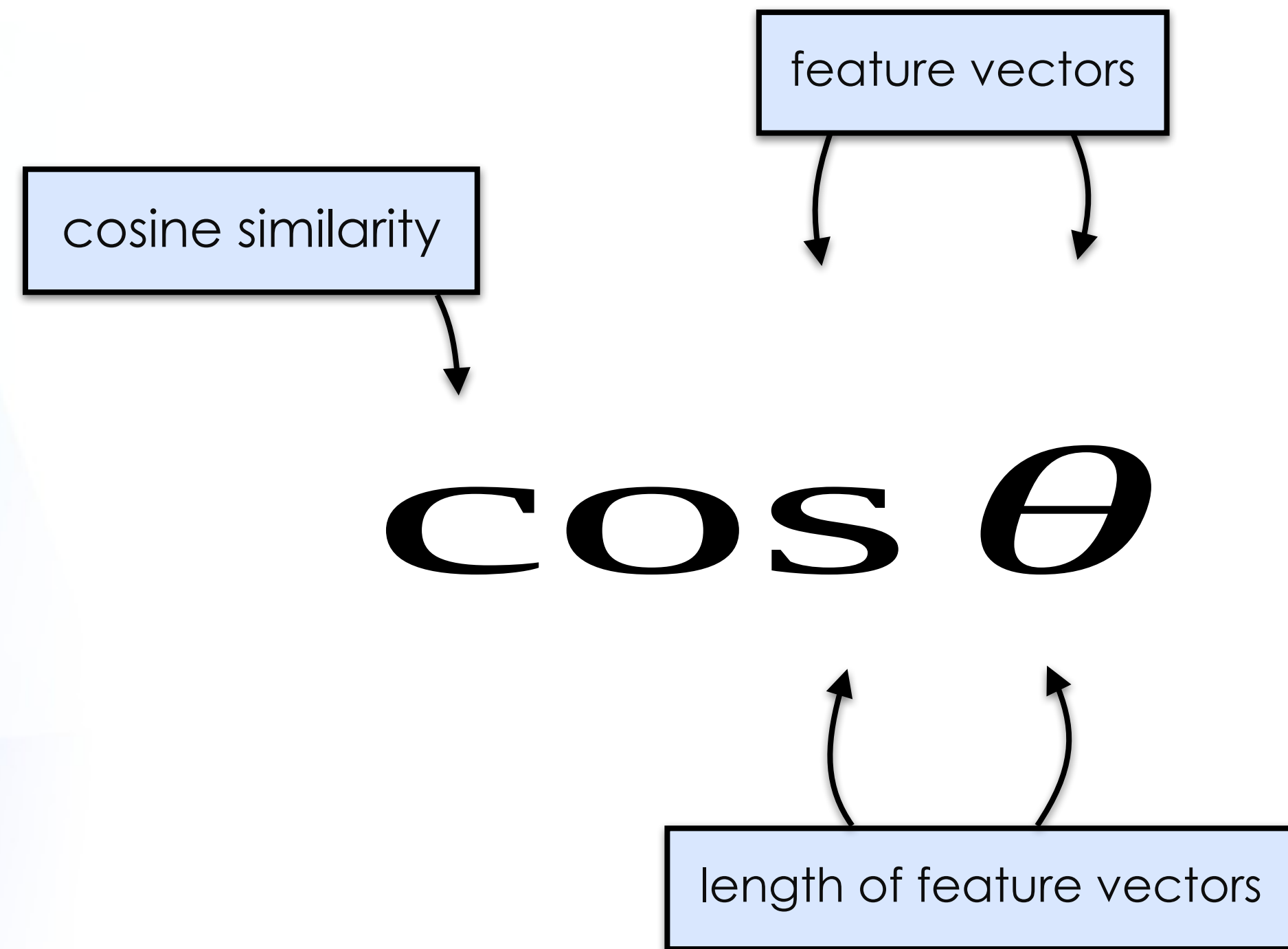


How to select Books to Recommend

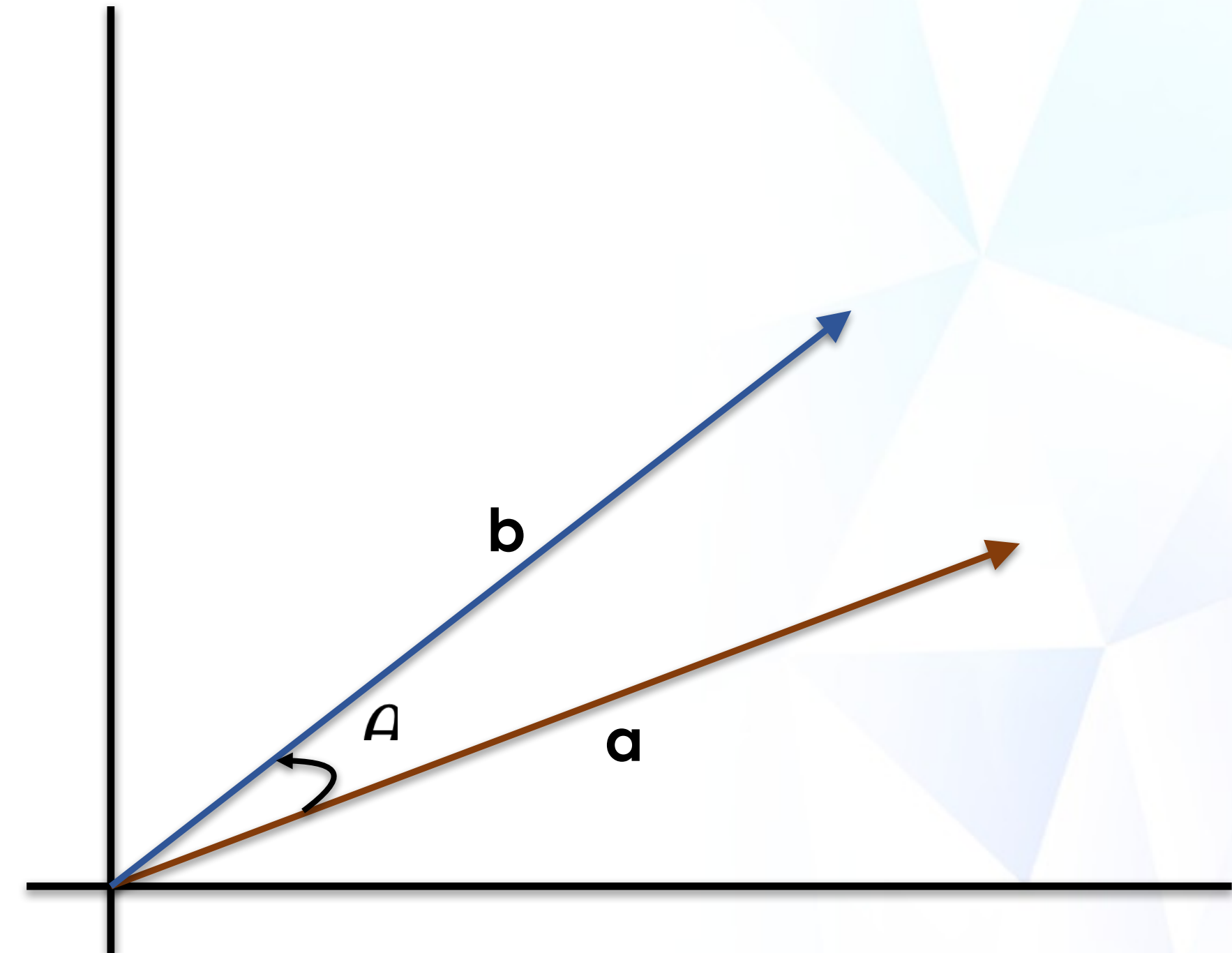
- Based on Bob's purchased books, look for **similar** books within our book repository; use the **feature vectors** for **comparison**



Cosine Similarity



Cosine Similarity for feature vectors **a** and **b**



a and **b** are similar if cosine similarity is close to 1 (ranges from -1 to 1)

Cosine Similarity

Feature Vector of Book A

1	0	0	1	0	0	236	1978
---	---	---	---	---	---	-----	------

Feature Vector of Book B

0	1	0	1	0	1	3000	1945
---	---	---	---	---	---	------	------

$$\cos \theta = .$$

$$= \frac{1 \cdot 0 + 0 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 0 \cdot 1}{\sqrt{1^2 + 0^2 + 0^2 + 1^2 + 0^2 + 0^2} \sqrt{0^2 + 1^2 + 0^2 + 1^2 + 0^2 + 1^2}}$$

$$= 0.63958$$

Content-Based Recommendation

- Compute the **pairwise cosine similarity scores** of each item against the rest
- Extract those unpurchased items that has high similarity scores with respect to that user's purchased items
- Recommend those **unpurchased** and **high similarity scores items** to that user
- The key idea about **content-based** recommendation is that it only focuses on **item features** and recommend **similar items** that share those features (it does not care if other users like those items or not)

COLLABORATIVE FILTERING

Collaborative Filtering focuses on **finding patterns** from **user-interactions** with the items

Explicit Interactions

- Gives a Rating
- Writes a Review
- Likes/Dislikes
- Adds to Favorites
- Makes a Purchase

Implicit Interactions

- No. of Views or Clicks
- Search History
- Time Spent browsing an item

Types of Collaborative Filtering

Item-Based Filtering

- A recommendation algorithm that analyzes **item similarities** based on co-occurrence patterns in **user-item interactions**
- Uses the **relationships between items** to generate recommendations

User-Based Filtering

- A recommendation algorithm that analyzes **user similarities** based on co-occurrence patterns in **user-item interactions**
- Uses the **relationships between users** to generate recommendations

Item-Based Collaborative Filtering

- **Item-Based** Filtering recommends items to a user based on the preferences or interactions of similar items from other users

	Bob	Alice	James	Larry
Book 1	4	5	5	
Book 2	2	4	3	4
Book 3	5	4	5	4
Book 4	3	3	3	
Book 5	4	3	3	

Table shows ratings by users.

We want to recommend books to Larry using this table.

Item-Based Collaborative Filtering

- Based on Larry's ratings, identify the books that he likes

	Bob	Alice	James	Larry
Book 1	4	5	5	
Book 2	2	4	3	4
Book 3	5	4	5	4
Book 4	3	3	3	
Book 5	4	3	3	

Larry likes Book2 and Book3 as he has given high ratings to them.

Item-Based Collaborative Filtering

- Locate similar books that Larry has liked based on other user's ratings
- Of those similar books, only interested in books that Larry has not read

	Bob	Alice	James	Larry
Book 1	4	5	5	
Book 2	2	4	3	4
Book 3	5	4	5	4
Book 4	3	3	3	
Book 5	4	3	3	

Each row is a feature-vector.
Found Book 4.

Item-Based Collaborative Filtering

- Likewise, based on their feature-vectors, Book 1 and Book 3 are similar

	Bob	Alice	James	Larry
Book 1	4	5	5	
Book 2	2	4	3	4
Book 3	5	4	5	4
Book 4	3	3	3	
Book 5	4	3	3	

Found Book 1.

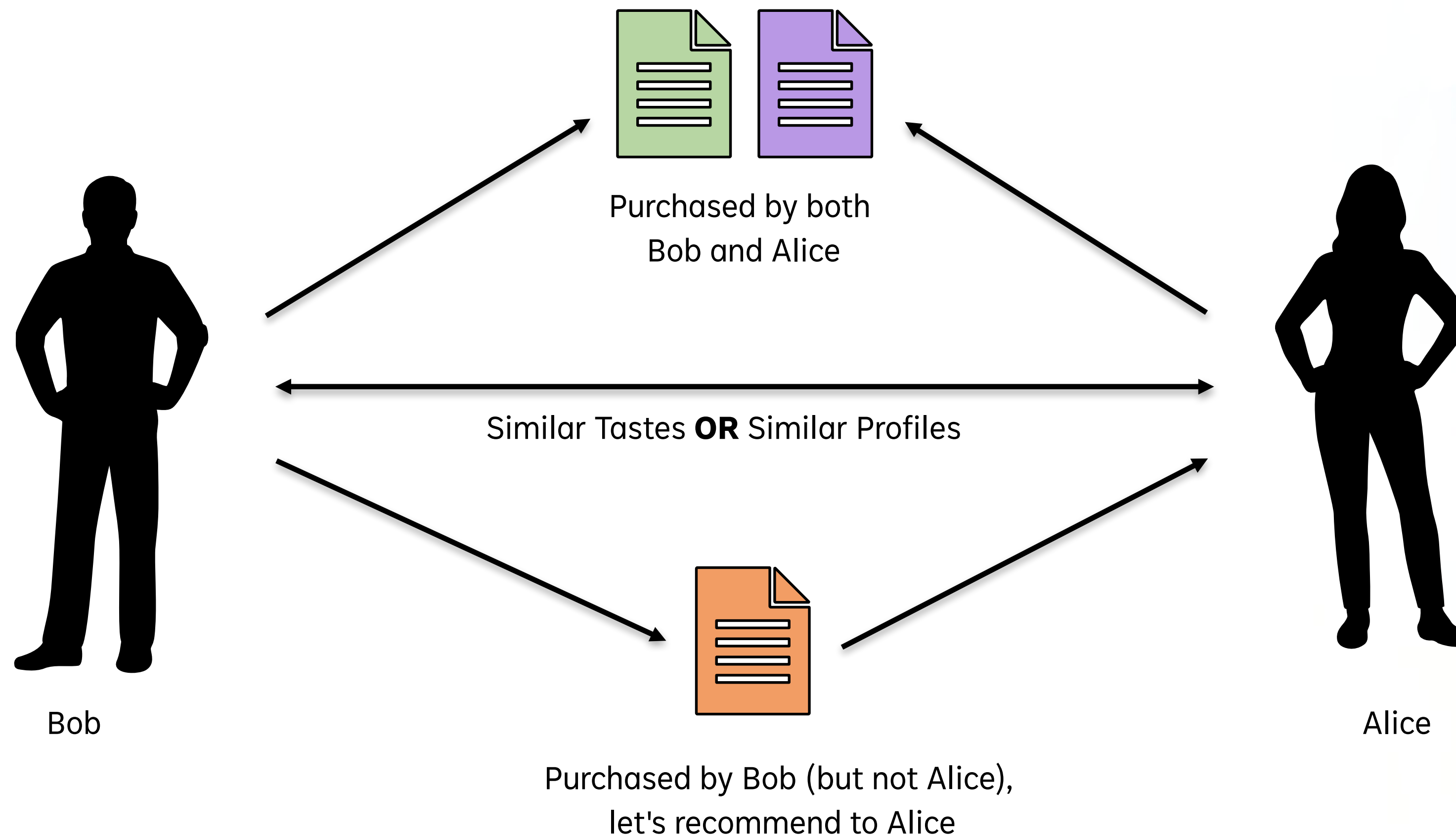
Item-Based Collaborative Filtering

- Recommend Books 1 and 3 to Larry as he has yet to read them

	Bob	Alice	James	Larry
Book 1	4	5	5	
Book 2	2	4	3	4
Book 3	5	4	5	4
Book 4	3	3	3	
Book 5	4	3	3	

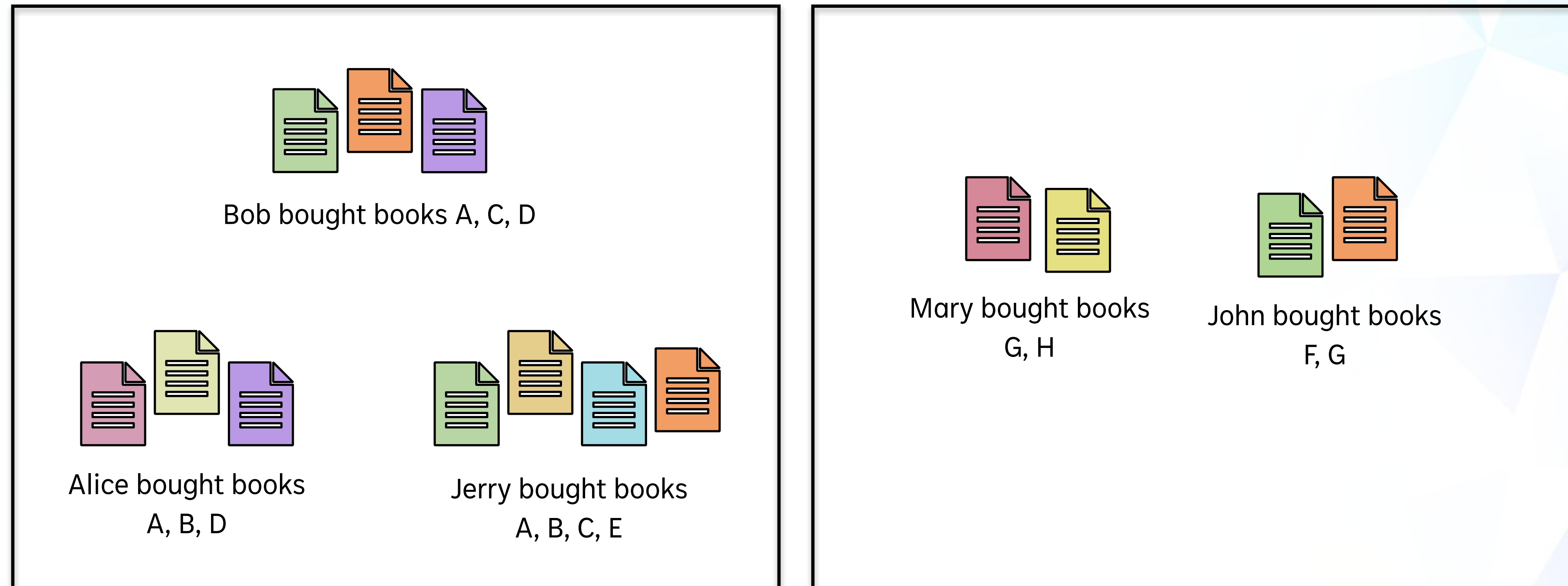
User-Based Collaborative Filtering

- **User-Based** Filtering recommends items to a user based on the preferences or interactions of **similar users**



Which users are similar?

- It is up to use to define; for us, let consider Customers are **similar** if they purchased the same books (**items**)



Based on their purchases, Bob, Alice and Jerry are similar users; same applies to Mary and John

User-Based Collaborative Filtering

- Users' purchase-histories can be used to form our feature vectors; a 1 denotes a purchase, and a 0 otherwise

	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	1	0	1	1	0	0	0	0
Alice	1	1	0	1	0	0	0	0
Jerry	1	1	1	0	1	0	0	0
Mary	0	0	0	0	0	0	1	1
...
John	0	0	0	0	0	1	1	0

User-based
feature vectors

Feature Vectors based on item-purchases by users

User-Based Collaborative Filtering

- We are free to extend our feature vectors for our needs. Here, let us add users' age as another feature (assume that users of similar age like the same thing)

	DOB	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	2001	1	0	1	1	0	0	0	0
Alice	2005	1	1	0	1	0	0	0	0
Jerry	1991	1	1	1	0	1	0	0	0
Mary	1995	0	0	0	0	0	0	1	1
...	
John	1994	0	0	0	0	0	1	1	0

User-based
feature vectors

Feature Vectors based on item-purchases and age

Normalizing a Feature

	Bob	Alice	Jerry	Mary	John
Year of Birth	2001	2005	1991	1995	1994

min(X): 1994
max(X): 2005

- The Year of Birth values can be normalized to a range of 0 and 1 with:

$$Norm(X) = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Code to Normalize a Feature

- Normalizing users' DOB to scale their year-values from 0 to 1

```
import numpy as np  
dobs = np.array([2001, 2005, 1991, 1995, 1994])  
norm_dobs = (dobs - np.min(dobs)) / (np.max(dobs) - np.min(dobs))  
print(dob_norms)
```



0.71428571	1	0	0.28571429	0.21428571
------------	---	---	------------	------------

- After normalizing the DOB feature, our feature vectors look like this

	DOB	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	0.714	1	0	1	1	0	0	0	0
Alice	1	1	1	0	1	0	0	0	0
Jerry	0	1	1	1	0	1	0	0	0
Mary	0.286	0	0	0	0	0	0	1	1
...	
John	0.214	0	0	0	0	0	1	1	0

User-based
feature vectors

Feature Vectors based on item-purchases and age

Find users similar to Bob

- **Pairwise** Cosine Similarity will give us the **similarity scores** of **each user** with respect to the **rest of the users**

Bob	0.714	1	0	1	1	0	0	0	0
Alice	1	1	1	0	1	0	0	0	0

Cosine Similarity (Bob, Alice) = 0.724

Bob	0.714	1	0	1	1	0	0	1	0
Jerry	0	1	1	1	0	1	0	0	0

Cosine Similarity (Bob, Jerry) = 0.534

Code to compute Cosine Similarity

- The code generates **Pairwise** Cosine Similarity values

```
1 import numpy as np
2 import pandas as pd
3 from sklearn.metrics.pairwise import cosine_similarity
4
5 users = ["bob", "alice", "jerry"]
6
7 bob = np.array([[0.714, 1, 0, 1, 1, 0, 0, 0, 0]])
8 alice = np.array([[1, 1, 1, 0, 1, 0, 0, 0, 0]])
9 jerry = np.array([[0, 1, 1, 1, 0, 1, 0, 0, 0]])
10
11 # stack the feature vectors
12 x = np.concatenate((bob, alice, jerry), axis=0)
13
14 # pairwise cosine similarity
15 scores = cosine_similarity(x)
16 df_sim = pd.DataFrame(
17     data = scores,
18     columns = users,
19     index = users
20 )
21
22 print(f"\ndf_sim=\n{df_sim}")
```



```
df_sim=
      bob    alice    jerry
bob  1.000000  0.724334  0.533776
alice 0.724334  1.000000  0.500000
jerry 0.533776  0.500000  1.000000
```

Recommendable books for Bob

- Assuming Alice and Jerry are among the Top 2 Cosine Similarity scores, with respect to Bob's, then books that they have purchased which Bob has not, are targets for recommendation

	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	1	0	1	1	0	0	0	0
Alice	1	1	0	1	0	0	0	0

Based on Alice's past purchases, Book B can be recommended to Bob

	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	1	0	1	1	0	0	0	0
Jerry	1	1	1	0	1	0	0	0

Based on Jerry's past purchases, Books B and E can be recommended to Bob

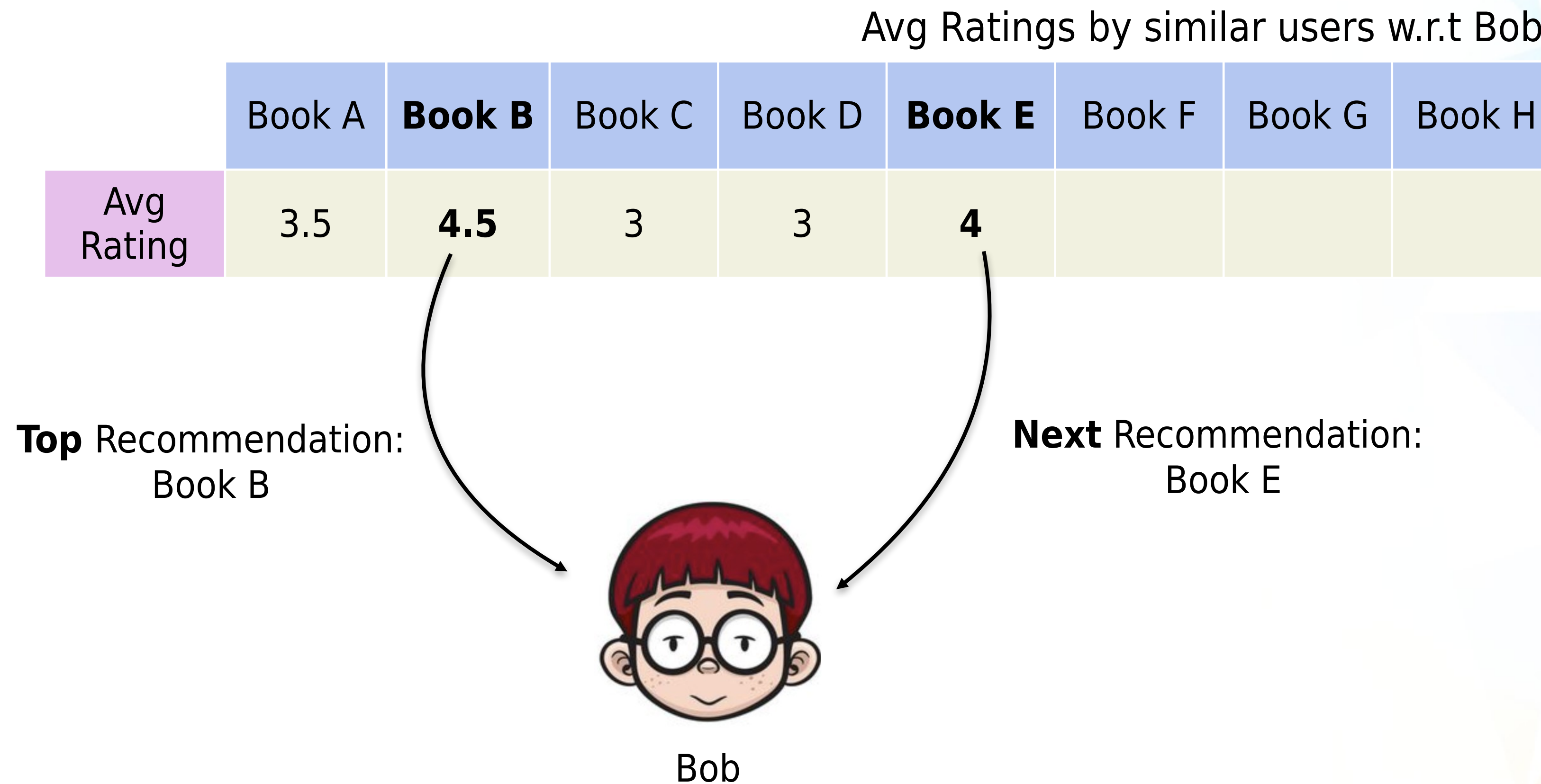
- Get the average ratings of books based on similar users

	Book A	Book B	Book C	Book D	Book E	Book F	Book G	Book H
Bob	3		3	4				
Alice	4	5		3				
Jerry	3	4	3		4			
Mary							3	4
...
John							1	
Avg Rating	3.5	4.5	3		4			

Ratings of purchased books by users

Ranking our Recommendations

- Due to a higher rating for Book B compared to Book E, we recommend Book B to Bob first, and Book E second



Cold Start Problem

- The **Cold Start** Problem is when there is not enough collected data to perform computations to generate good recommendations (e.g. a new system)
- Use **non-personalized** recommendations like Editorial and Simple Aggregation to overcome the cold start problem

THE END