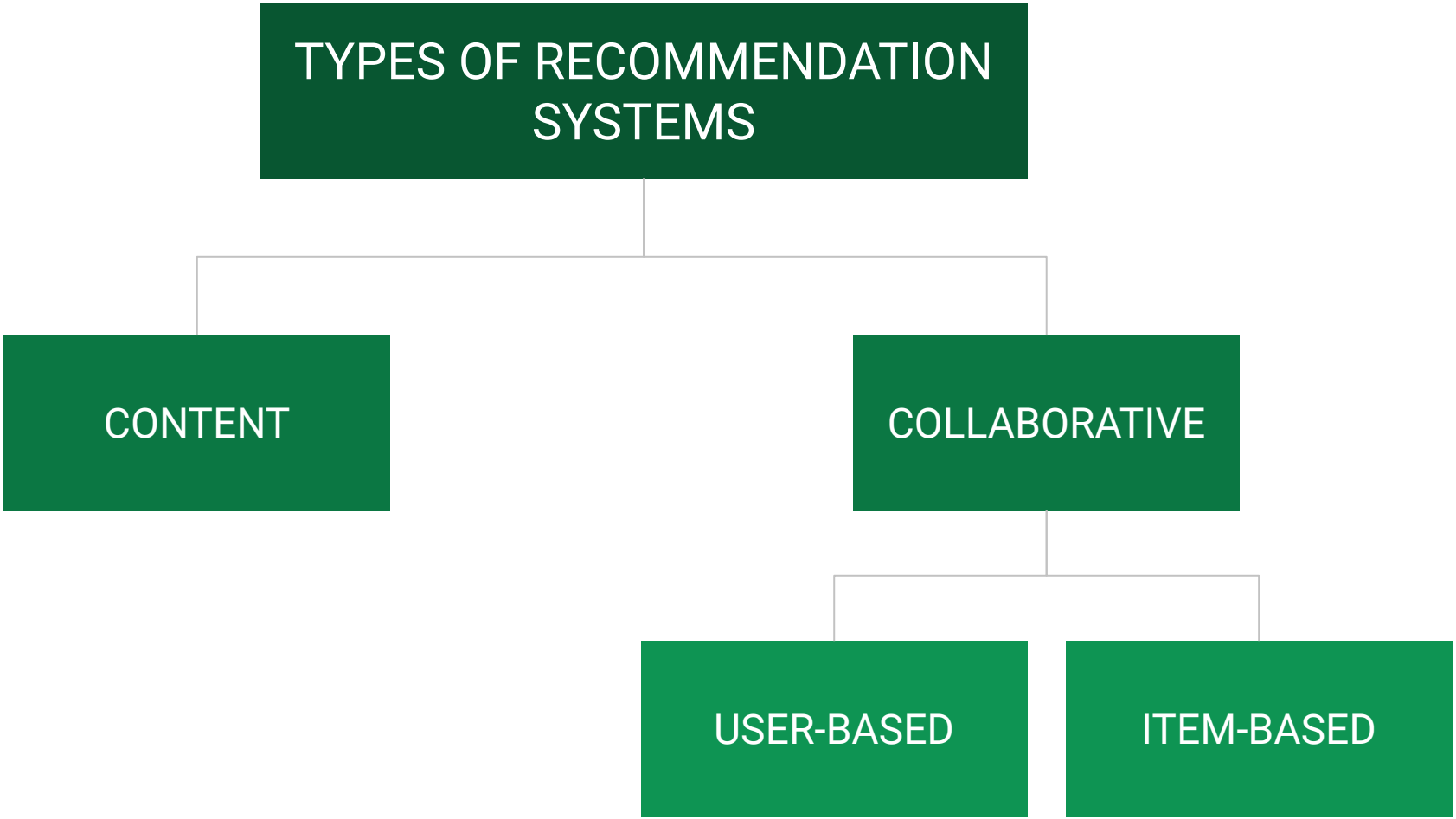
Recommendation Systems

Sydney Herndon
29 August 2019


TYPES OF RECOMMENDATION SYSTEMS
CONTENT
COLLABORATIVE
USER-BASED
ITEM-BASED

Content-Based

Movie 1	1999	Action	Director 1
Movie 2	2017	Comedy	Director 2
Movie 3	2019	Drama	Director 3

- Makes recommendations based on product features
- Need to one-hot encode categorical features
- Likely to recommend sequels because they have similar features (cast, director, genre, ets)
- Other examples?

Collaborative **User-Based**

	Movie 1	Movie 2	Movie 3
User 1	5	1	??
User 2	4	1	4
User 2	1	5	1

- Makes recommendations based on users with similar ratings/purchases/etc.
- Who is similar to user 1?
  - Based on that, is User 1 likely to enjoy Movie 3?
- How might this look for purchases?
- Real-life applications might add weights to some features

Collaborative **Item-Based**

	User 1	User 2	User 3
Movie 1	4	5	2
Movie 2	2	1	5
Movie 3	5	4	2

- Makes recommendations based on products with similar ratings/purchases
- These tend to show up on item-pages (think amazon)
- If you like this product, here are other similar products
- Rabbit hole (we've all been there)
- What is the difference between this and a content-based recommender?

Let’s Talk Linear Algebra

- Each Row of the data frame becomes a Vector:
- Using vector directions/magniture to understand which row-items are similar

	Harry Potter	**Die Hard**
Sydney (S)	3	4
Amber (A)	4	3

Let’s Talk Linear Algebra

	Harry Potter	**Die Hard**
Sydney (S)	3	4
Amber (A)	4	3

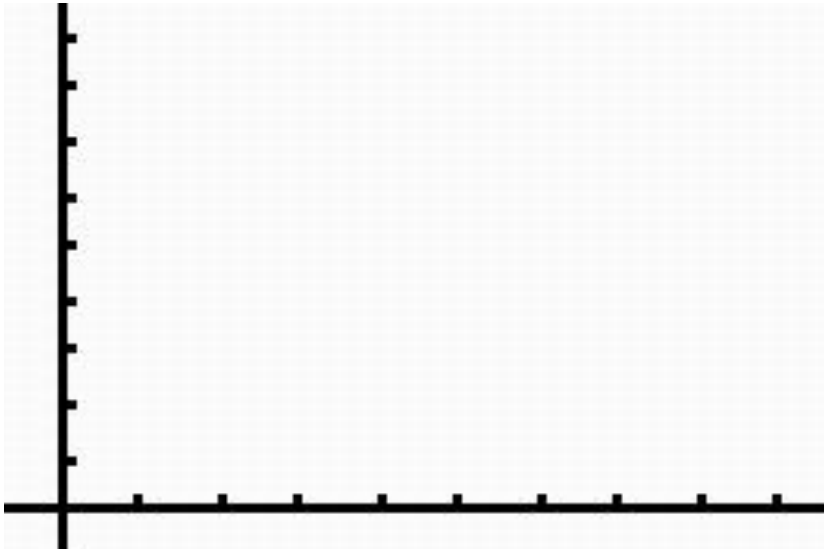
Vector 1=

Vector 2 =

Let’s Talk Linear Algebra

	Harry Potter	Die Hard
Sydney (S)	3	4
Amber (A)	4	3

Die Hard



Harry Potter

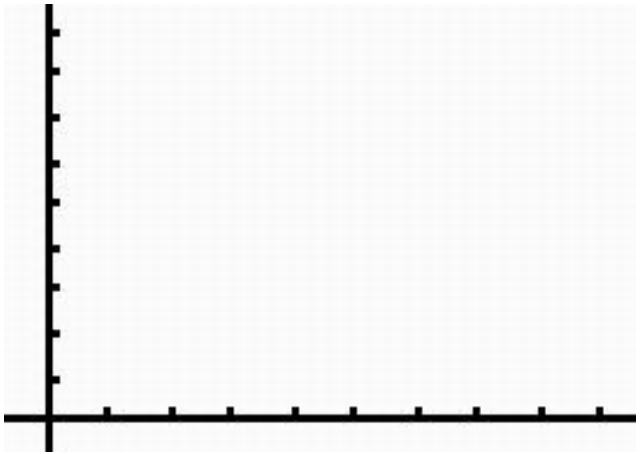
Cosine Similarity

Quantifies the degree of similarity of two Vectors

Formula:

Calculating Magnitude:

Die Hard



Harry Potter

- Magnitude = length
- Pythagorean Theorem
- Dividing by the Magnitude normalizes each vector (length becomes 1)
  - Focusing on Direction
- Calculate for S vector and for A vector

Calculating Dot Product

Interpreting Cosine Similarity (CS):

- **CS = 1**
  - Vectors are same direction
  - Exact same preferences

- **CS = -1**
  - Vectors are going in the exact opposite direction
  - Exact opposite preferences

- **CS = 0**
  - Vectors are perpendicular
  - Can’t infer anything about the preference relationship