Agenda ครับ ------
1. Meet guest developers from Silicon Valle
2. What you need to know to convince your prospective CTO or developer?
3. How can you communicate your ideas to your CTO or developer?

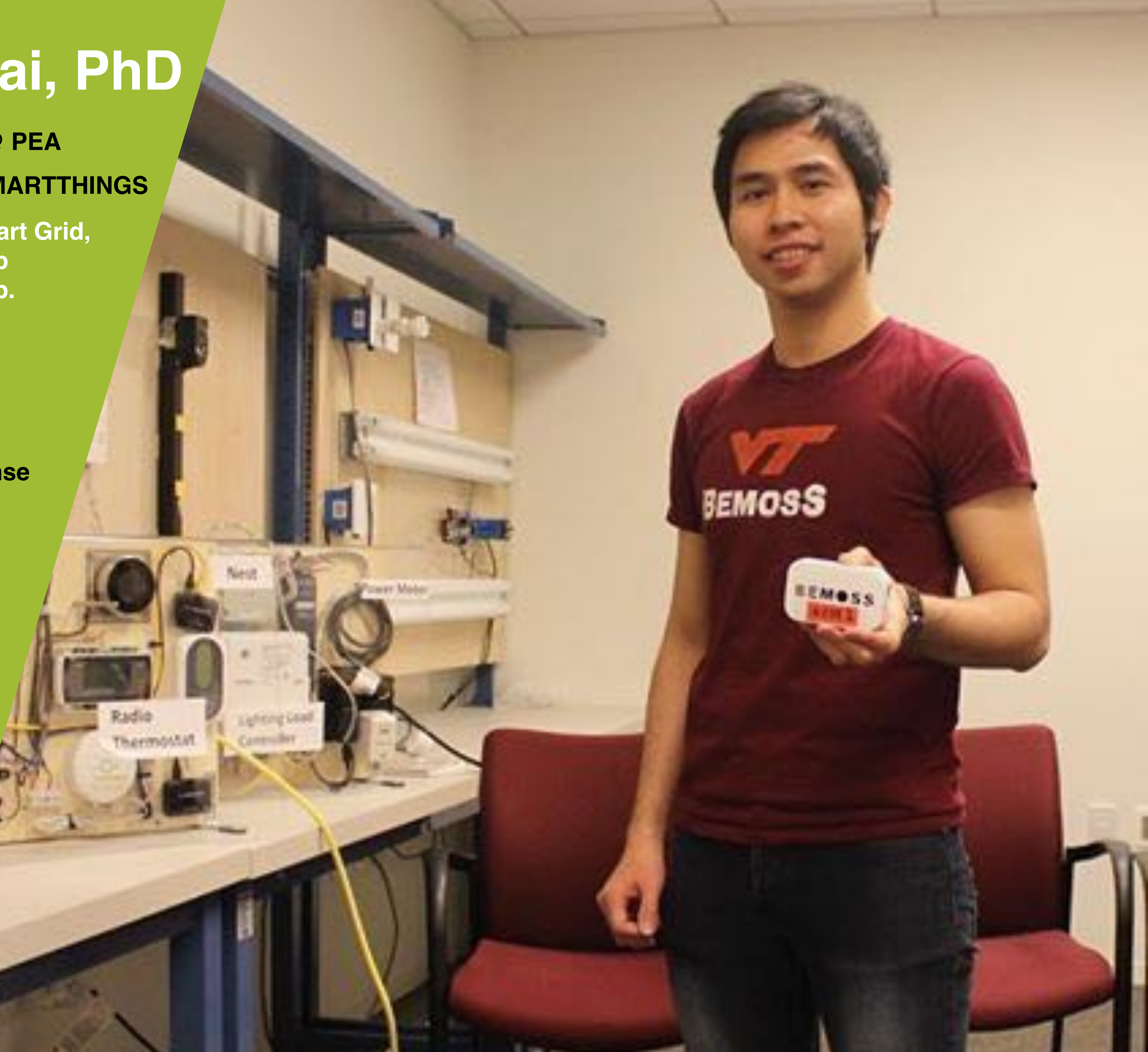
Warodom Khamphanchai, PhD

- LEAD SMART HOME/BUILDING PLATFORM DEVELOPER @ PEA
- EX-SOFTWARE DEVELOMENT ENGINEER @ SAMSUNG SMARTTHINGS

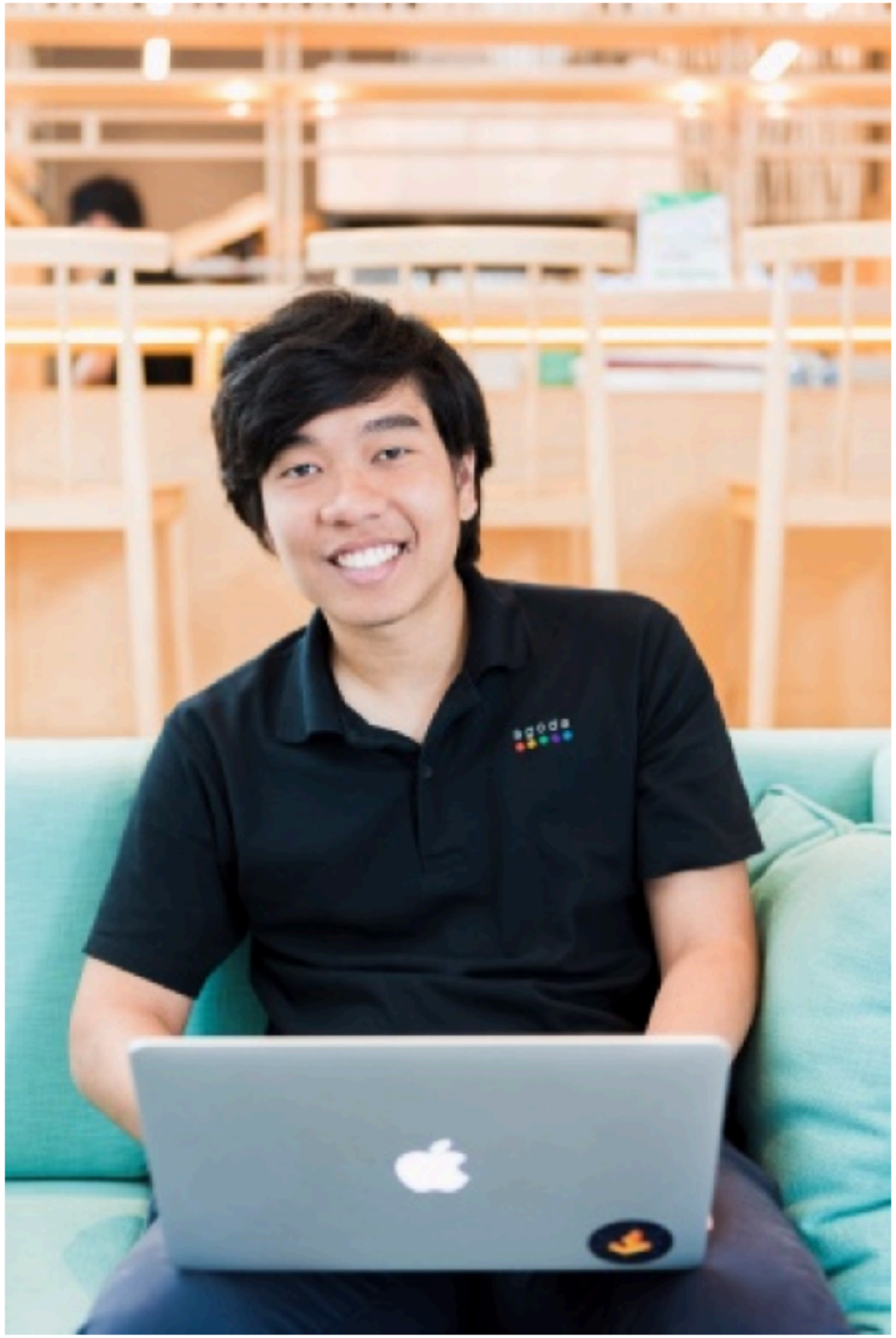
Interests: Home/Building Automation, Internet of Things, Smart Grid, Multi-Agent systems, Data Analytics, Machine Learning, Deep Learning, AI, Energy Audit, and Technology Entrepreneurship.

Education:

- [2011-2016] Ph.D. in Electrical and Computer Engineering, Virginia Tech
  Dissertation: An Agent-based Platform for Demand Response Implementation in Smart Buildings
- [2009-2011] M.E. in Energy (Area of Specialization: Electric Power System Management), Asian Institute of Technology
  Thesis: A Multi-Agent Based Power System Restoration Approach in Distributed Smart Grid
- [2005-2009] B.E. in Electrical Engineering, Chulalongkorn University



Line: kwarodom
LinkedIn: www.linkedin.com/in/kwarodom
Web: kwarodom.wordpress.com
Email: kwardom@vt.edu
Tel: +6695-161-5011 Github: kwarodom

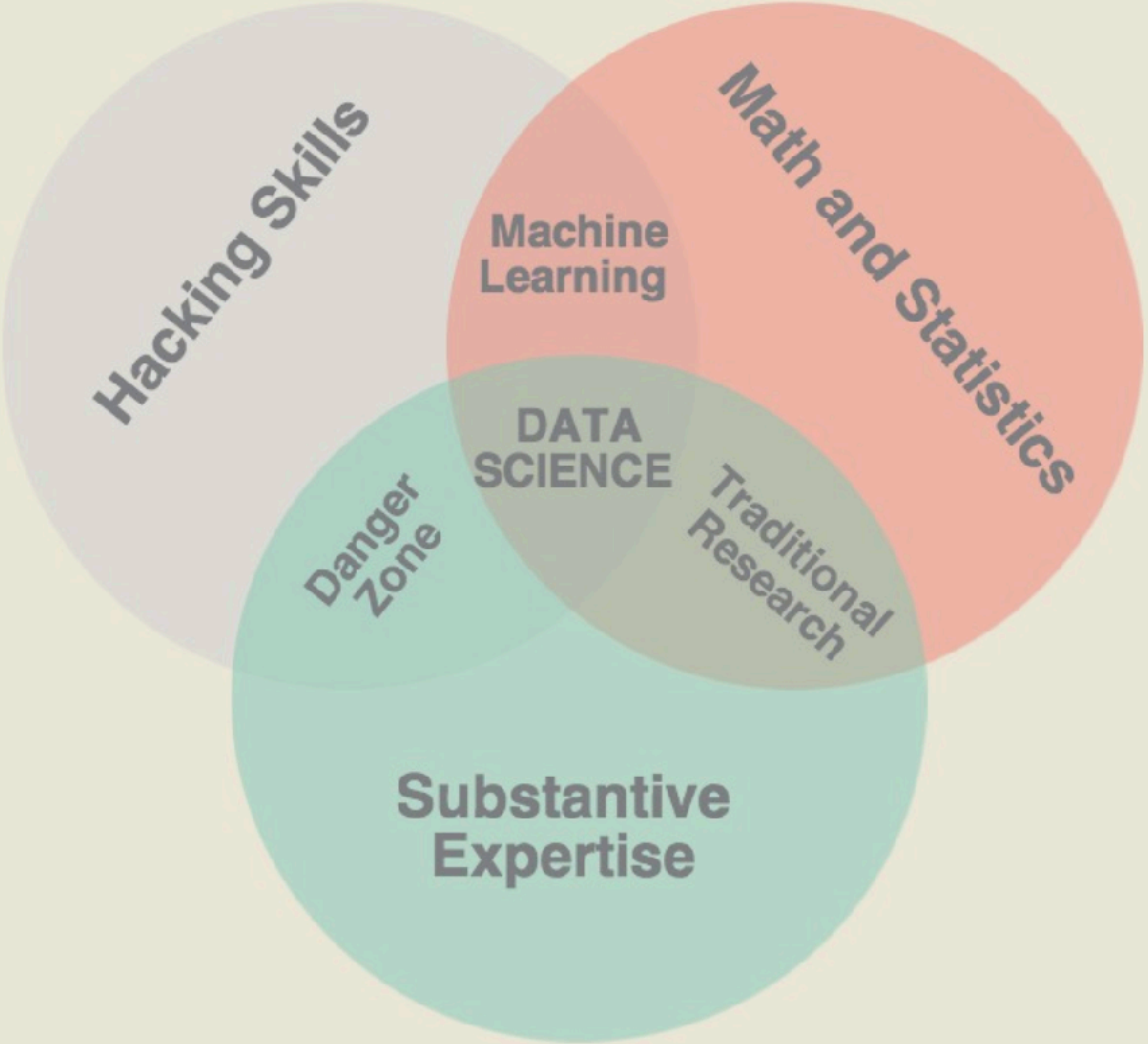


Profile

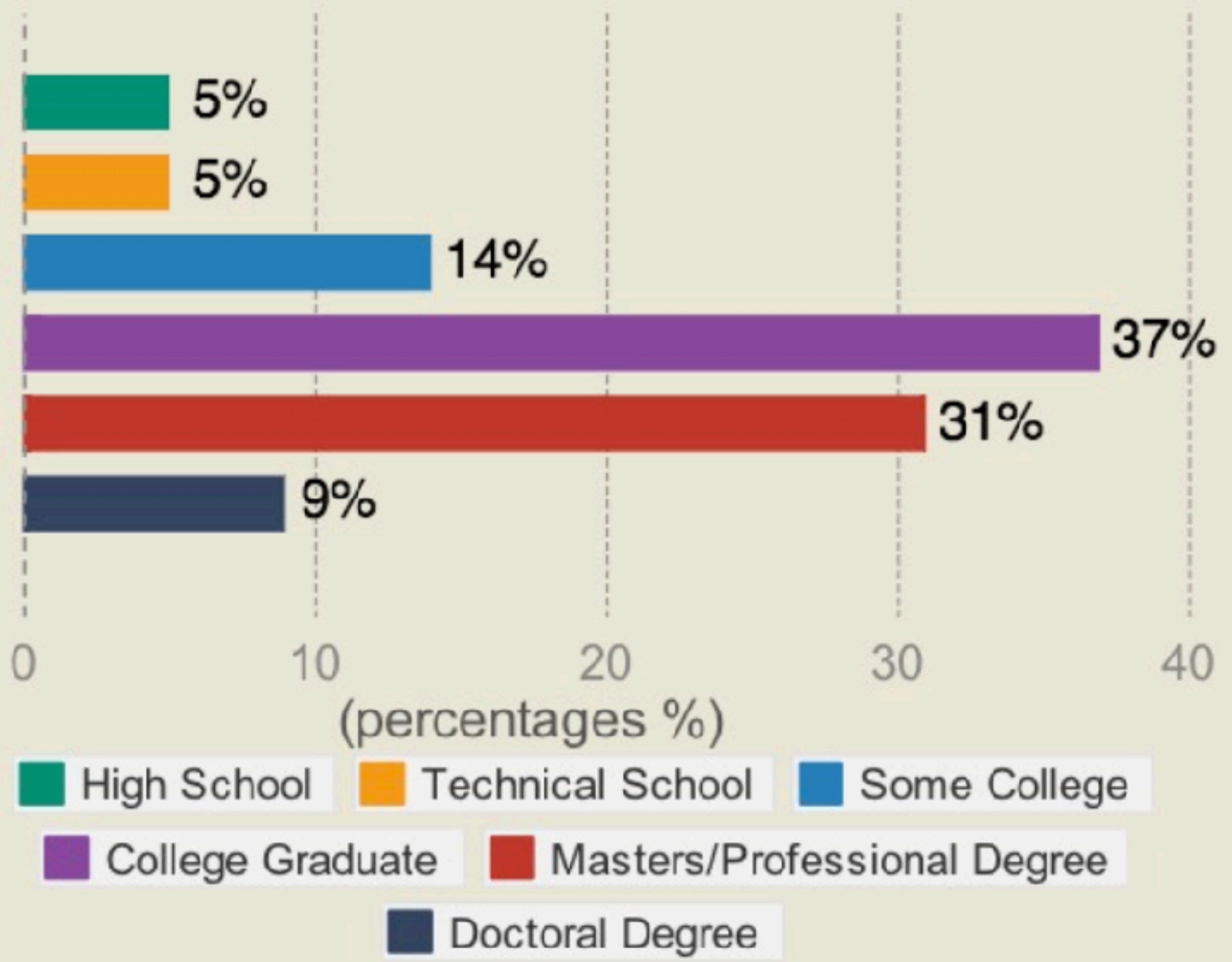
Sorawit Saengkyongam (**James**)

- Data Scientist at Agoda where he works on research and development of personalization algorithms and recommendation systems
- Google Developer Expert in Machine Learning.
- Organize Bangkok Machine Learning Meetup
- Graduated with a major in Mathematical Statistics from Chulalongkorn Unviersity (with first class honours)

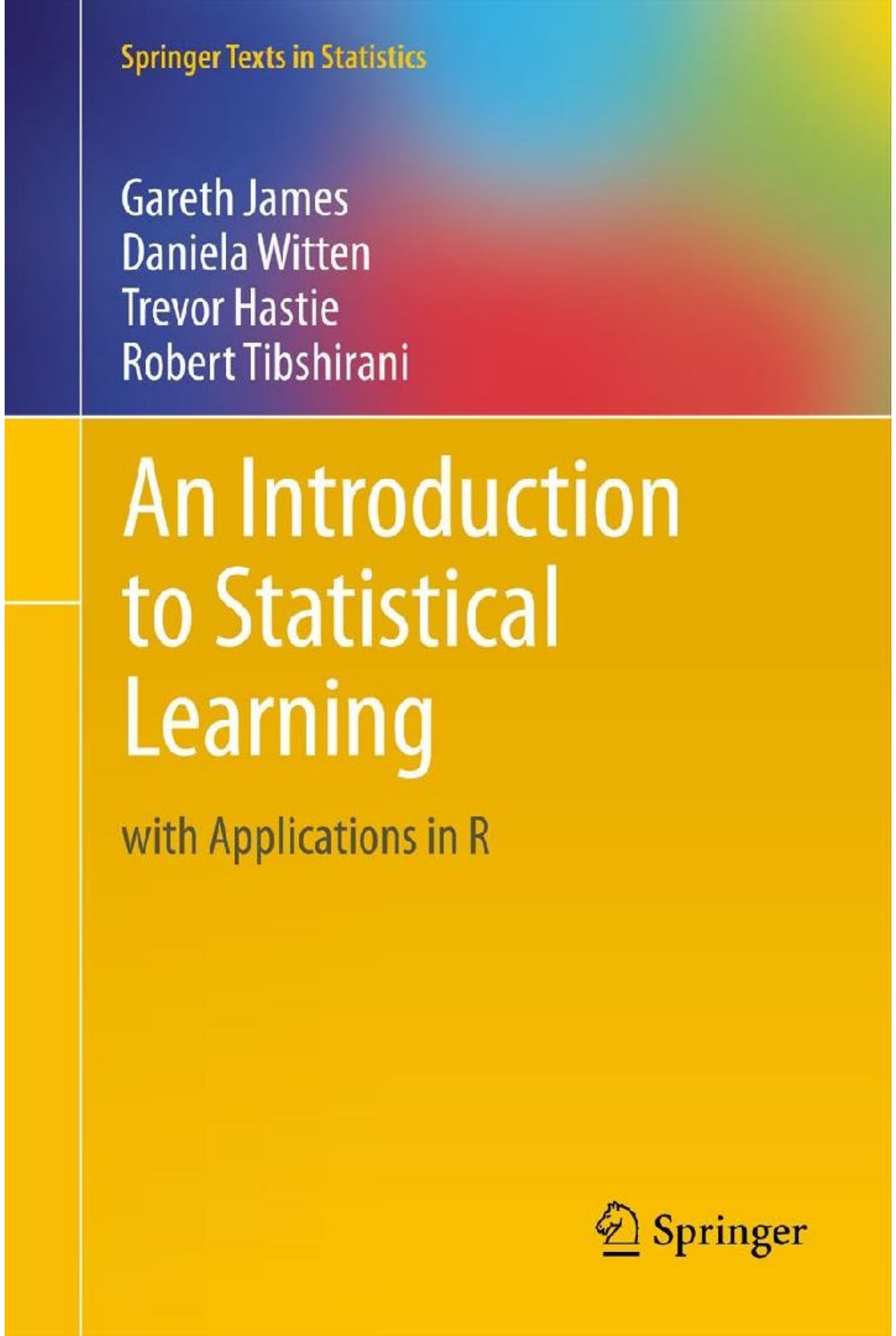
What's a data scientist?



Typical Background

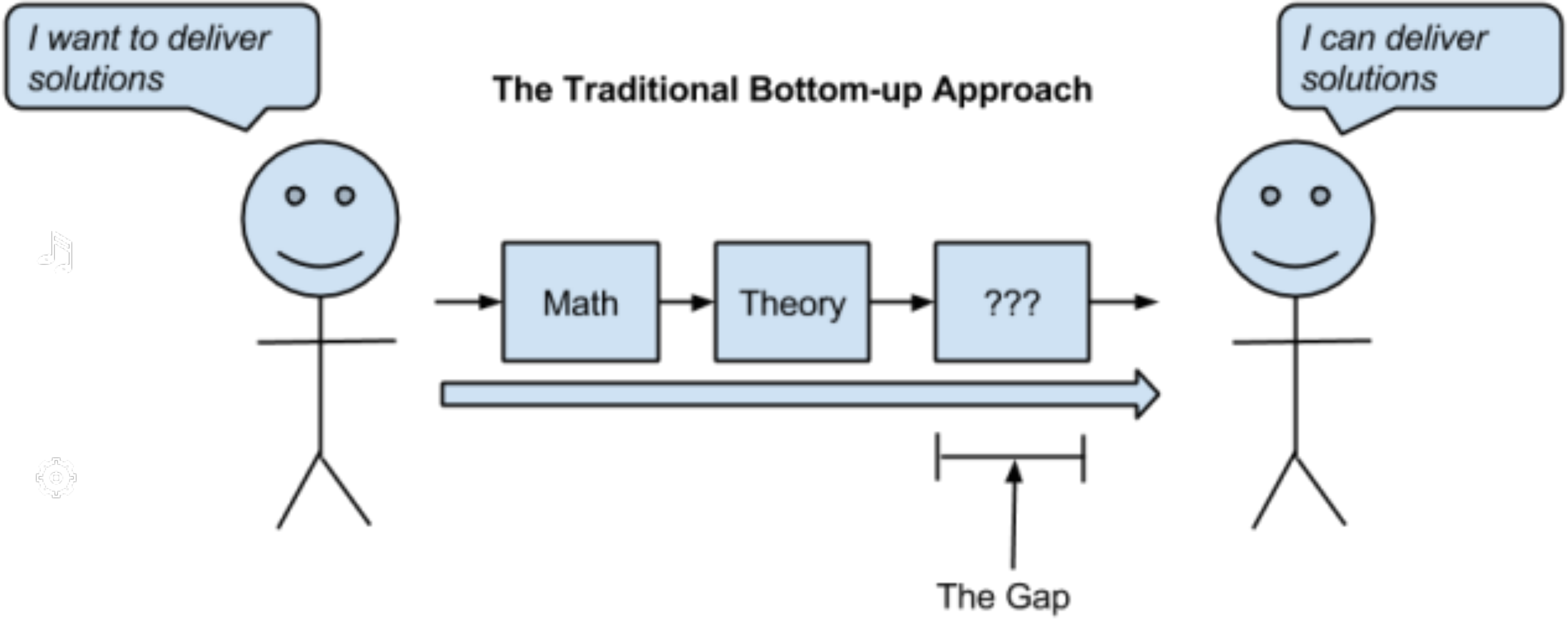


i A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician.



- ☐ **Programming** 05
  - ☐ R programming language
  - ☐ Python programming language
  - ☐ Spreadsheet tools (like Excel)
  - ☐ JavaScript and HTML
  - ☐ C/C++
- ☐ **Statistics** 07
  - ☐ Descriptive and Inferential statistics
  - ☐ Experimental design
- ☐ **Mathematics** 09
  - ☐ College Algebra
  - ☐ Functions and Graphing
  - ☐ Multivariable Calculus
  - ☐ Linear Algebra
- ☐ **Machine Learning** 10
  - ☐ Supervised Learning
  - ☐ Unsupervised Learning
  - ☐ Reinforcement Learning
- ☐ **Data Wrangling** 12
  - ☐ Python
  - ☐ Database Systems
  - ☐ SQL
- ☐ **Communication and Data Visualization** 13
  - ☐ Visual Encoding
  - ☐ Data Presentation
  - ☐ Knowing Your Audience
- ☐ **Data Intuition (Thinking like a data scientist)** 14
  - ☐ Project Management
  - ☐ Industry Knowledge

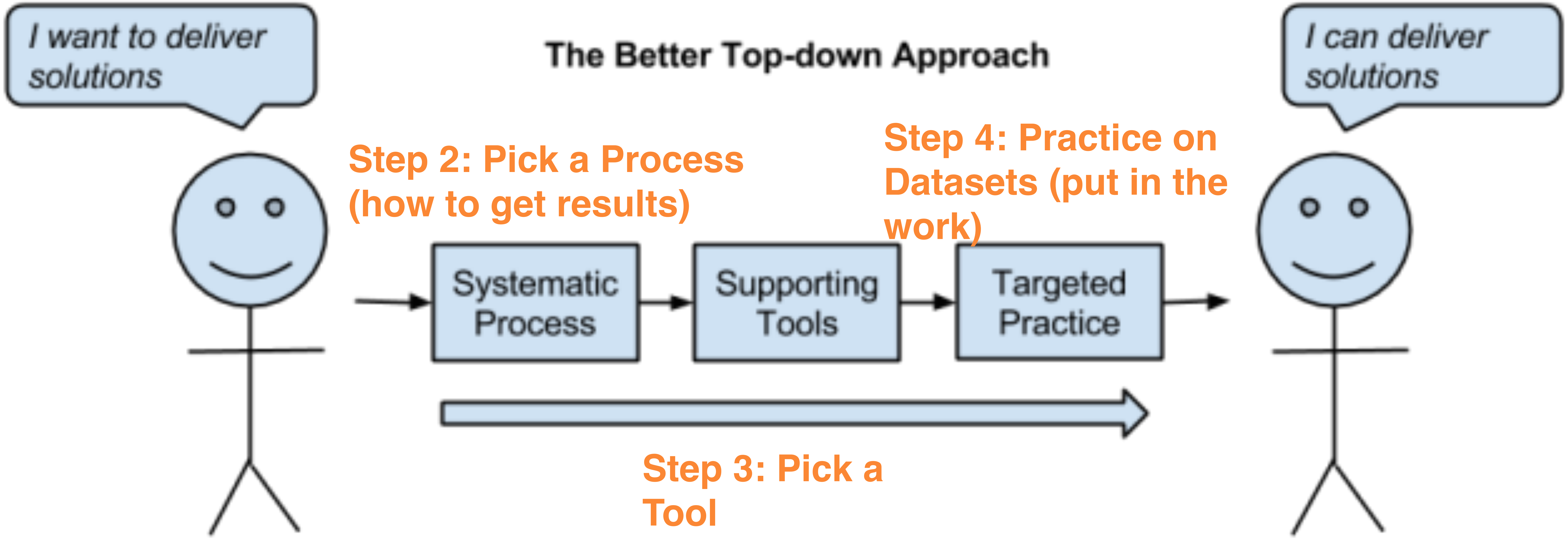
ACHINE LEARNING PRACTIONER



REF: http://machinelearningmastery.com/machine-learning-mastery-method/

MACHINE LEARNING PRACTIONER

5-Steps To Get Started and Get Good at Machine Learning



Step 1: Adjust Mindset (believe!)

Step 2: Pick a Process (how to get results)

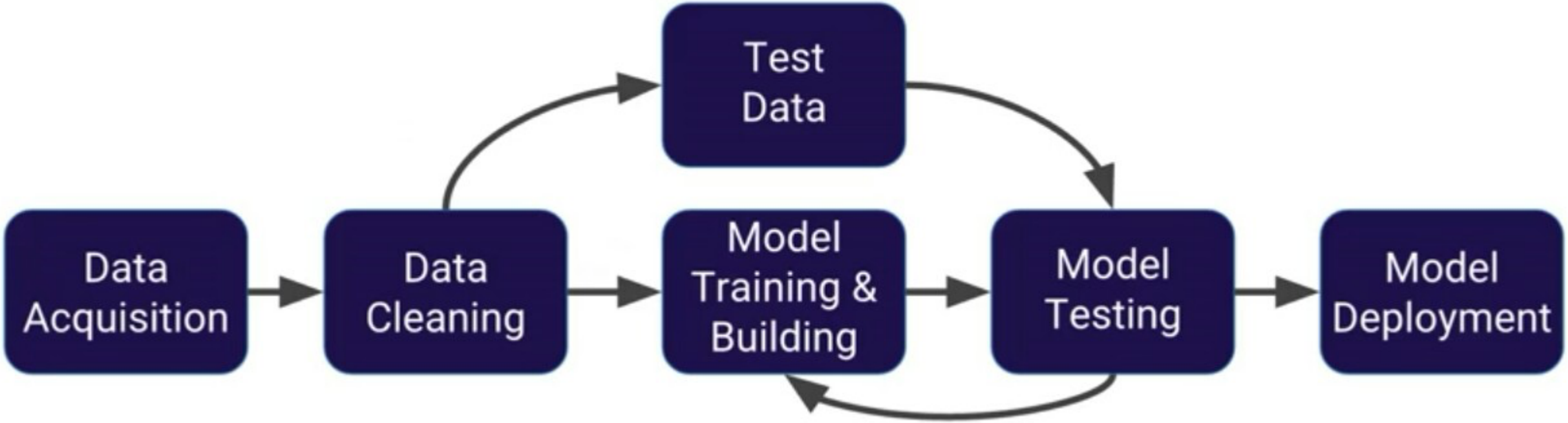
Step 3: Pick a Tool (implementation)

Step 4: Practice on Datasets (put in the work)

Step 5: Build a Portfolio (show your skills)

REF: http://machinelearningmastery.com/machine-learning-mastery-method/

MACHINE LEARNING PROCESS

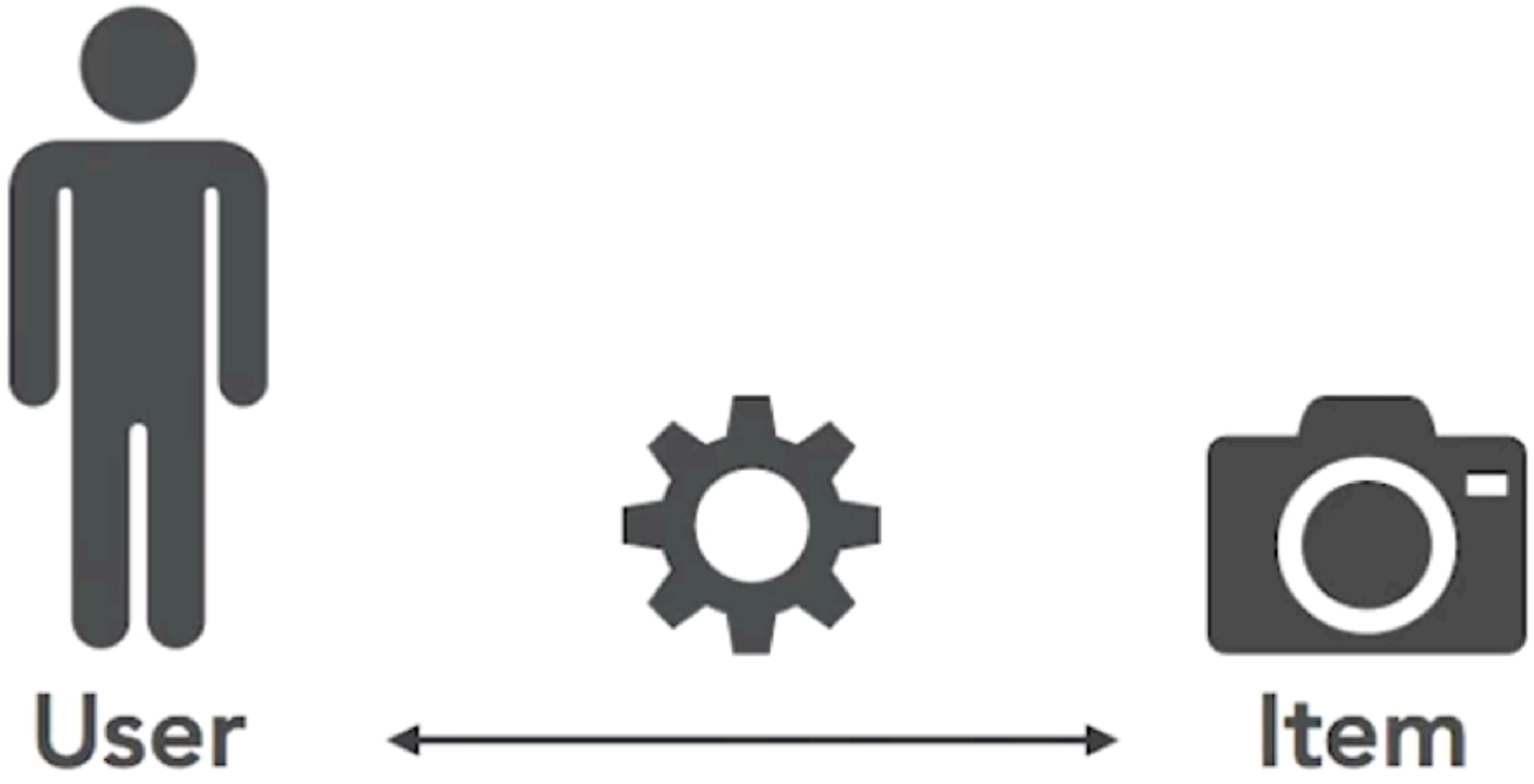


Source: PIERIAN DATA

MACHINE LEARNING PROCESS

Introduction to Python Recommendation System

Purpose: to find and recommend items that a user is most likely to be interested in



Introduction to Python Recommendation System

1. Popularity-Based Recommender Systems
2. Correlation-Based Recommender Systems
3. Classification-Based Collaborative Filtering
4. Model-Based Collaborative Filtering
5. Content-Based Recommender Systems
6. Evaluating of Recommender Systems

Examples of Recommendation Engines

- Product recommendations: Amazon and Etsy
- Movie recommendation: Netflix
- Music recommendation: Apple Music
- Social connection recommendations: Facebook, LinkedIn, and Instagram

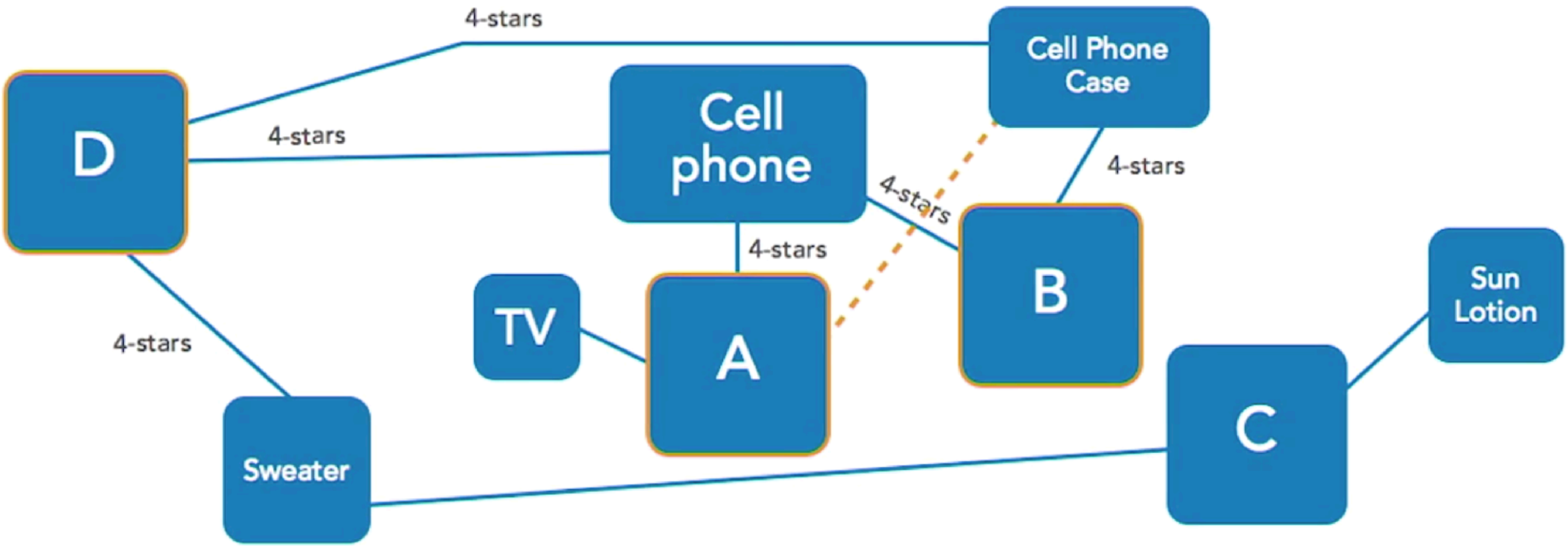
Collaborative Filtering Systems

Collaborative filtering systems recommend items based on crowdsourced information about users' preferences for items.

Two approaches:

- User based
- Item based

Conceptualizing Item-Based Filtering



User B and User D both gave high ratings to the cell phone and the cell phone case. Since User A also likes the cell phone, let's recommend to her the cell phone case also.

Conceptualizing User-Based System

User	Age	Net Worth ($)	Marital Status
User A	32	25,000	Divorced
User B	64	250,000	Married
User C	19	3,000	Single
User D	71	135,000	Married

Based on known user attributes, we know that User B is similar to User D. User D really likes his life insurance policy, so let's recommend it to User B also.

Content-Based Recommender System

Content-based recommenders recommend items based on similarities between features.

City Name	Average Temperature (F) in Winter	Average Cost of Living ($)	Average Wi-Fi Speed (MBPS)
Austin, Texas	65.1	2,147	165
Spokane, Washington	51.8	3,059	40
Miami, Florida	67	2,520	125

A user who loves Miami might also love Austin, based on the similarities between the temperature, cost of living, and Wi-Fi speeds at both places.

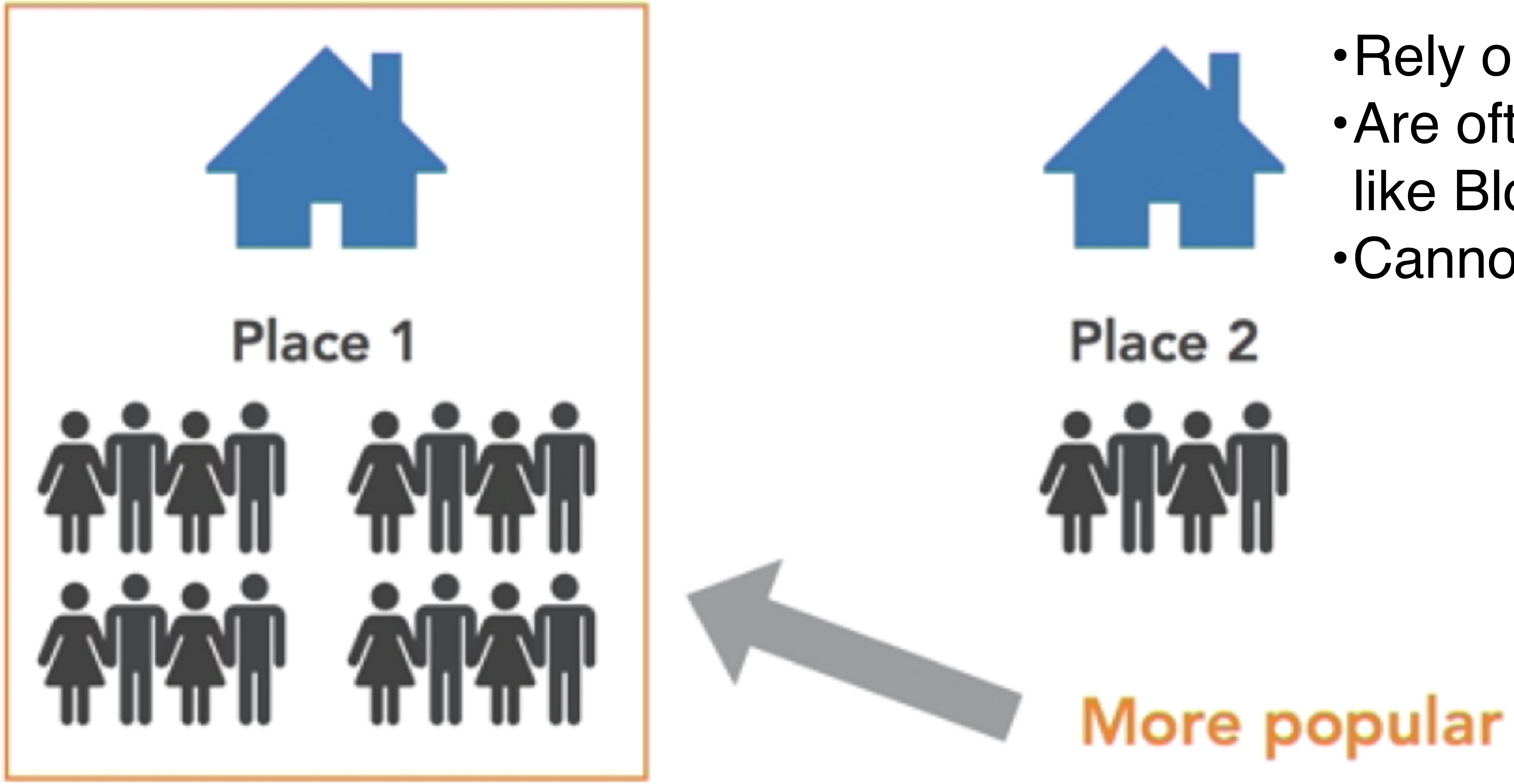
Popularity-Based Recommender Systems

Based on simple count statistics

User	Place	Rating
User A	Place 1	10
User B	Place 1	8
User C	Place 2	8
User D	Place 2	7
User E	Place 1	8
User F	Place 1	7
User G	Place 1	10

Place	Rating Count
Place 1	5
Place 2	2

Popularity-Based Recommender Systems



- Rely on purchase history data
- Are often used by online news site like Bloomberg
- Cannot produce personalized results

Correlation-Based Recommendation Systems

- Use Pearson's *r* correlation to recommend an item that is most similar to the item a user has already chosen
- Item-based similarity: How *correlated* are two items based on user ratings?

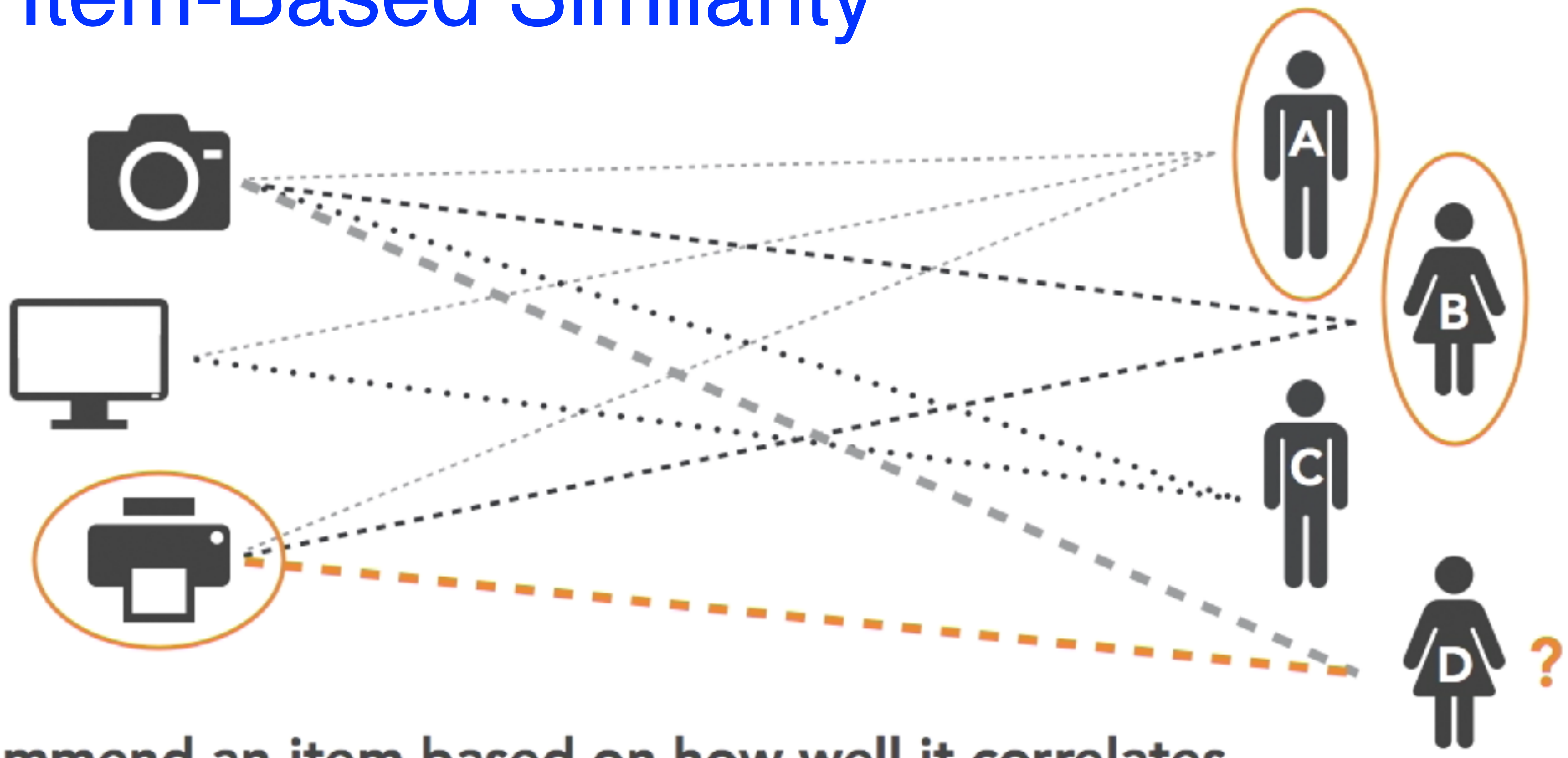
Correlation-Based Recommendation Systems

Pearson correlation coefficient (r)

- $r = 1$ → Strong positive *linear* relationship
- $r = 0$ → Not linearly correlated
- $r = -1$ → Strong negative *linear* relationship

Correlation-Based Recommendation Systems

Item-Based Similarity



Recommend an item based on how well it correlates with other items with respect to user ratings

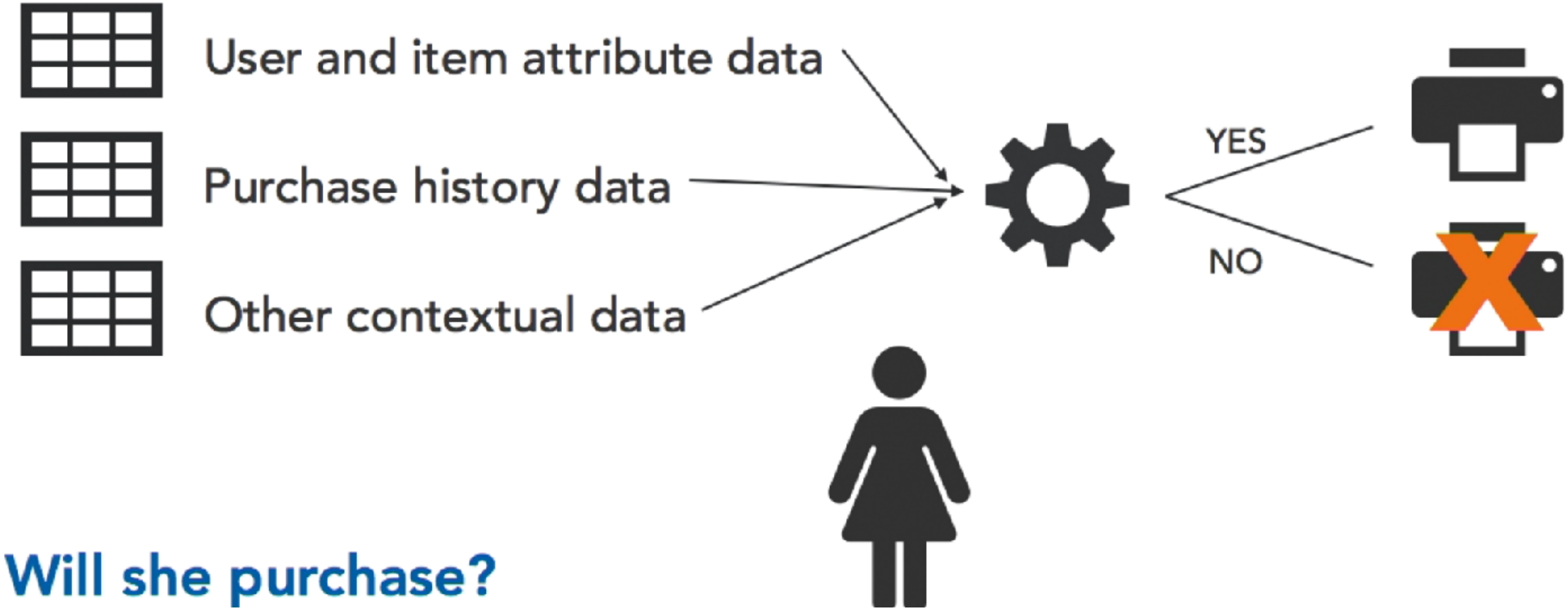
Classification-based Collaborative Filtering

- Naive Bayes classification
- Logistic regression

A simple machine learning method you can use to predict the value of a numeric categorical variable based on its relationship with predictor variables

Classification-based Collaborative Filtering

Provides personalization by accepting



Will she purchase?

A Logistic Regression Recommender

Will he accept the offer?

A Logistic Regression Recommender

"I am a single, divorced man who works in management and is not in credit default. I don't have a home loan or any other personal loans... Also, no one from your marketing department has ever solicited me before."

housing_loan	credit_in_default	personal_loans	prev_failed_to_subscribe	prev_subscribed	job_management	job_tech	job_entrepreneur
0	0	0	0	0	1	0	0

job_bluecollar	job_unknown	job_retired	job_services	job_self_employed	job_unemployed	job_maid	job_student
0	0	0	0	0	0	0	0

married	single	divorced
0	1	1

↓

[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1]

Model-Based Collaborative Filtering

- With model-based collaborative filtering systems, you build a recommender model from user ratings, and then make recommendations based on that model.

Model-Based Collaborative Filtering

Utility Matrix

Items	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6
Users						
User A						
User B						
User C						
User D						
User E						

Sparse matrices!

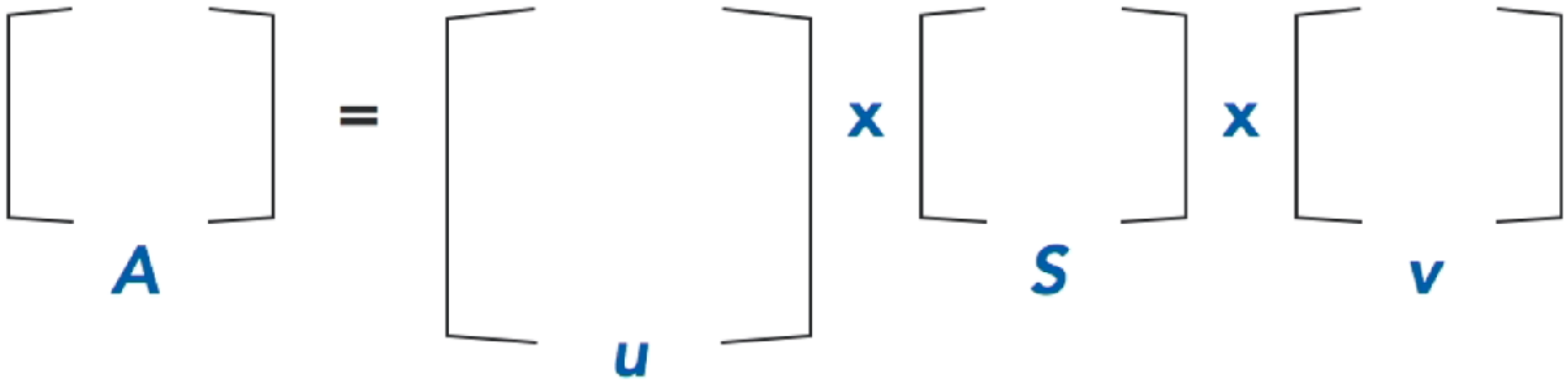
Model-Based Collaborative Filtering

Singular Value Decomposition (SVD)

- A linear algebra method that can decompose a utility matrix into three compressed matrices
- Model-based recommender – use these compressed matrices to make recommendations without having to refer back to the complete data set
- Latent variables – inferred, nonobservable variables that are present within, and affect the behavior of a data set

Model-Based Collaborative Filtering

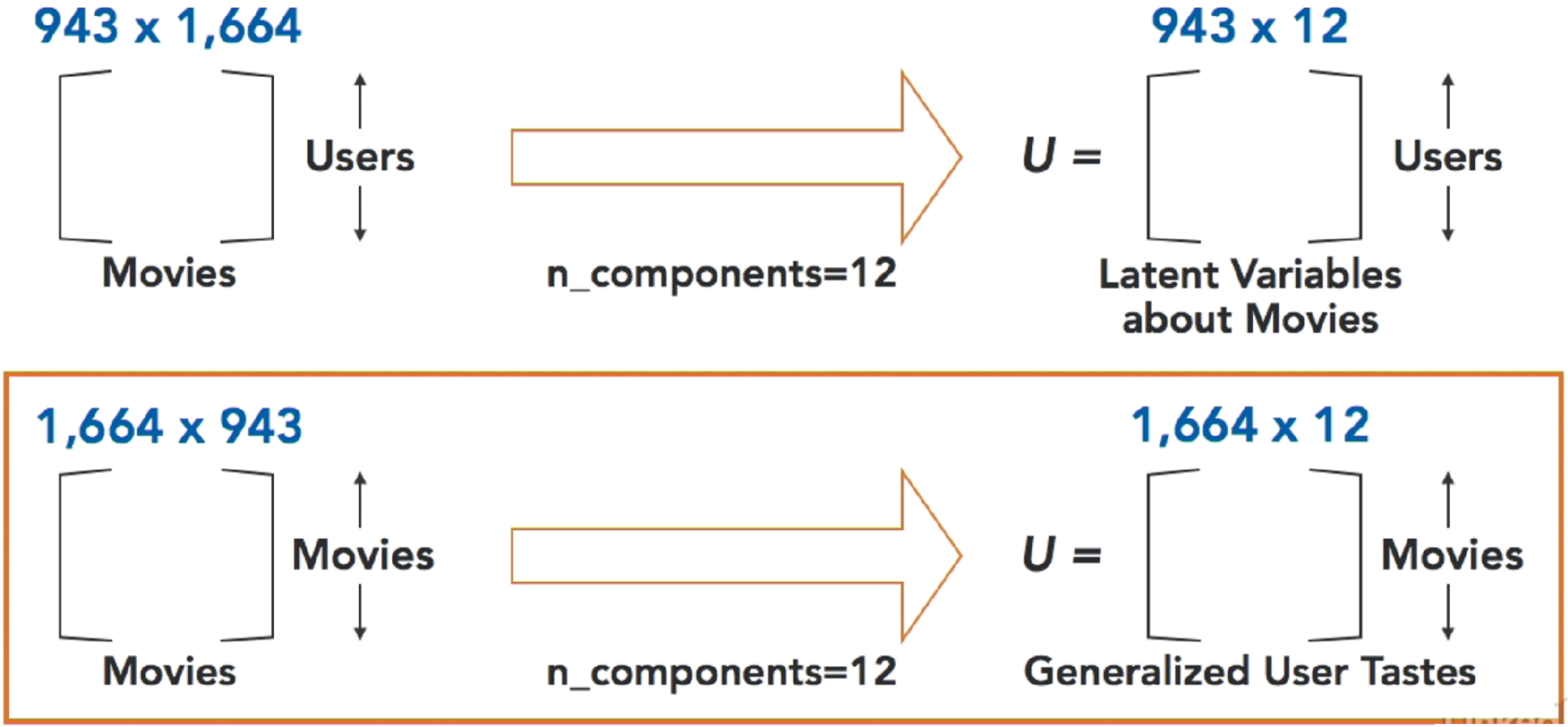
The Anatomy of SVD



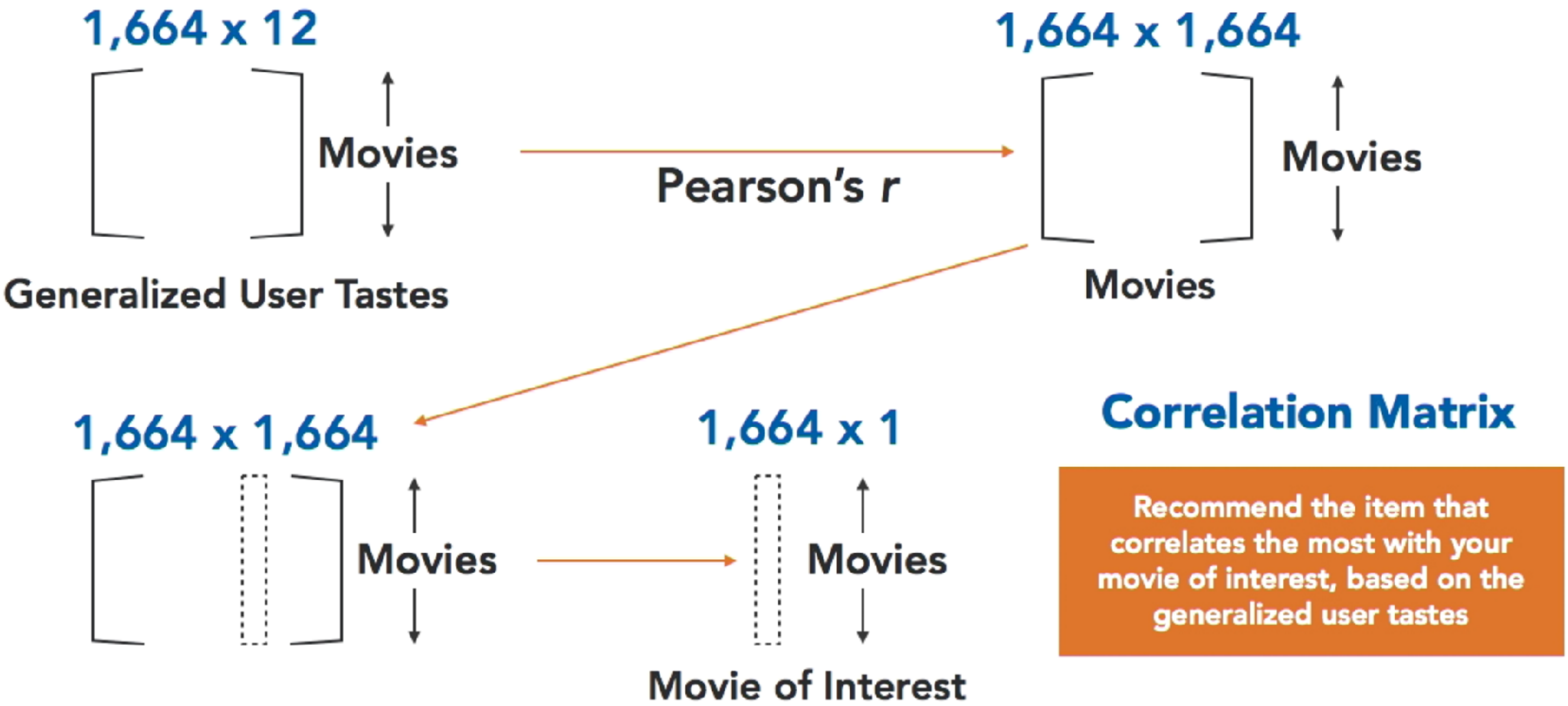
$$A = u \times S \times v$$

- **A** = Original matrix (utility matrix)
- **u** = Left orthogonal matrix – holds important, nonredundant information about users
- **v** = Right orthogonal matrix - holds important, non-redundant information on items.
- **S** = Diagonal matrix – contains all of the information about the decomposition processes performed during the compression

Model-Based Collaborative Filtering



Model-Based Collaborative Filtering



Content-Based Recommender Systems

- Recommends an item based on its features and how similar they are to features of other items in the data set

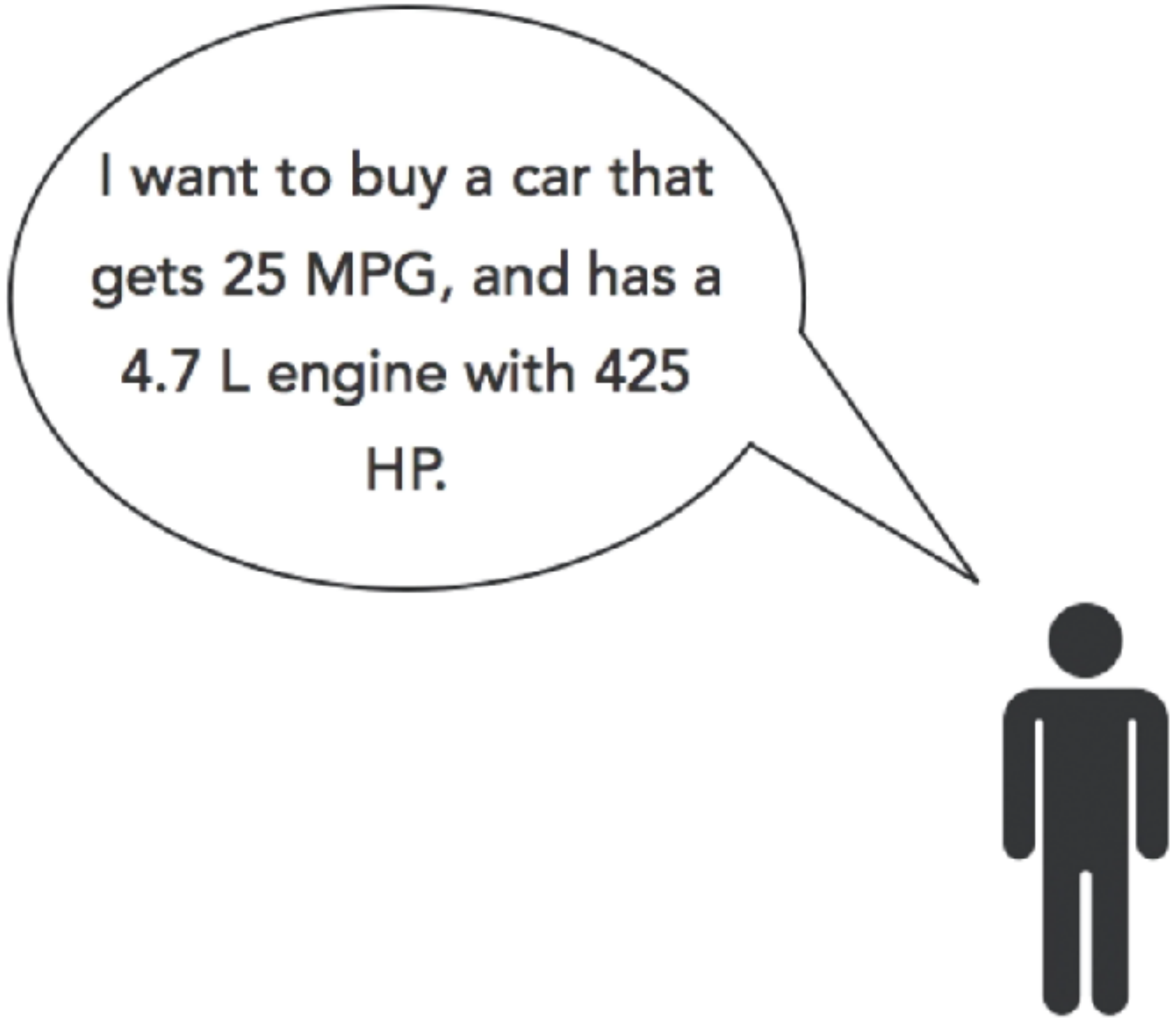
Content-Based Collaborative Filtering

Nearest neighbor algorithm

- Unsupervised classifier
- Also known as a memory-based system
- Memorizes instances and then recommends an item (a single instance) based on how quantitatively similar it is to a new, incoming instance

Content-Based Collaborative Filtering

car id no.	mpg	hp	engine size (L)
1	39	184	2.4
2	20	449	4.7
3	31	185	2.4
4	16	454	4.7



Evaluating Recommender Systems

	True Positive	True Negative
Predicted Positive	True Positive	False Positive
Predicted Negative	False Negative	True Negative

Evaluating Recommender Systems

$$\text{Precision} = \frac{\text{The number of items that I liked that were also recommended to me}}{\text{The number of items that were recommended}}$$

How relevant were the recommendations?

Evaluating Recommender Systems

$$\text{Recall} = \frac{\text{The number of items that I liked that were also recommended to me}}{\text{The number of items that I liked}}$$

How completely did the recommender system predict the items I liked?

Evaluating Recommender Systems

Will he accept the offer?

Evaluating Recommender Systems

	precision	recall	f1-score	support
0	0.90	0.99	0.94	39922
1	0.67	0.17	0.27	5289
avg / total	0.87	0.89	0.86	45211

***Precision*: a measure of the model's relevancy**

Of the entire data set, 87% of the recommended items were items that the user actually liked.

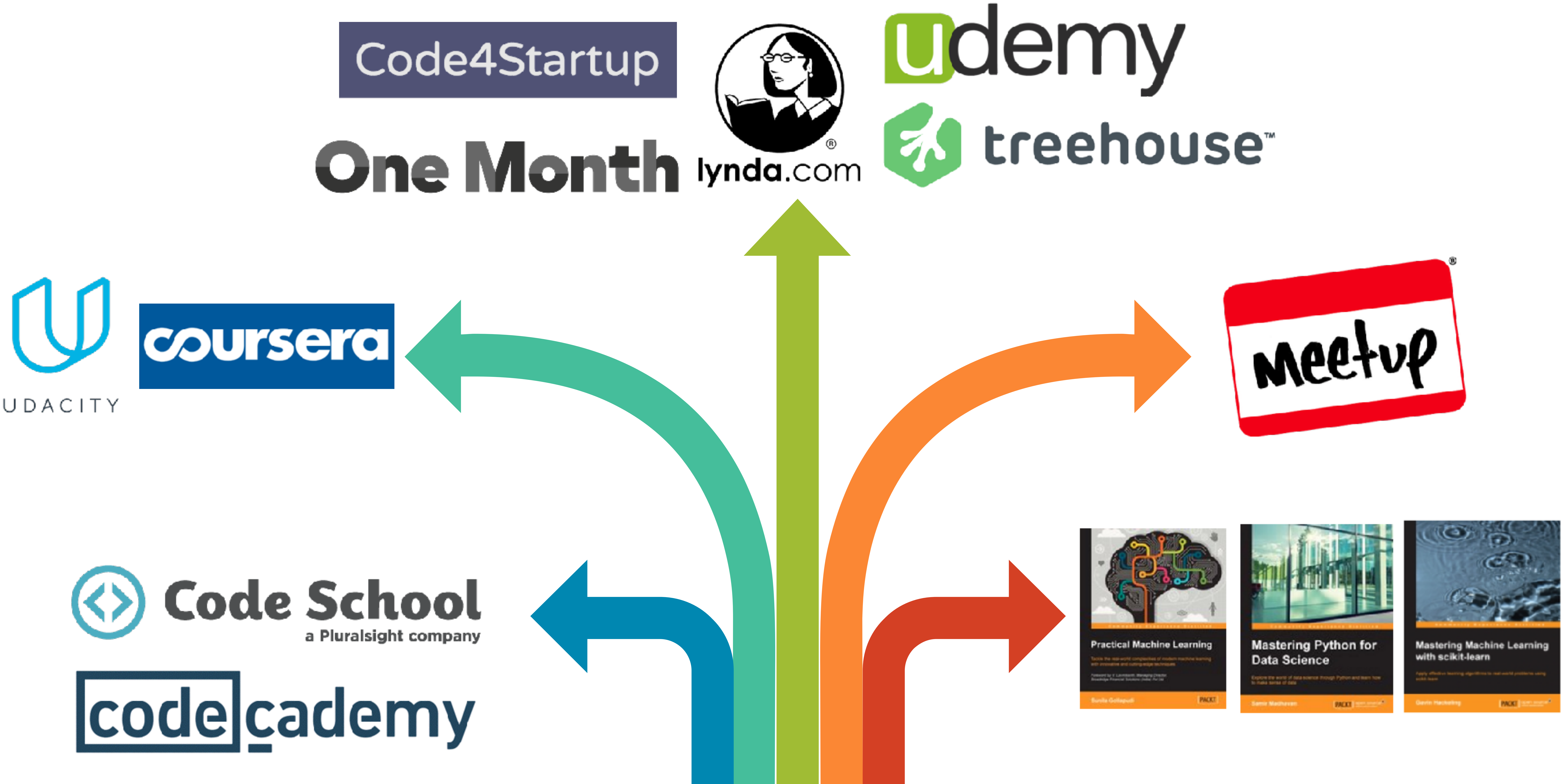
***Recall*: a measure of the model's completeness**

Of the entire data set, 89% of the user's preferred items were recommended.

High precision + High recall = High accurate model results

BURGER
Restaurants
Driver
Database
[Server]
Python/Django
RESTful API
RESTful API
Swift
Swift
Customers
Drivers

SO, WHAT IF I WANT TO LEARN HOW TO BUILD DATA



CHALLENGE YOURSELF WITH REAL-WORLD ML