

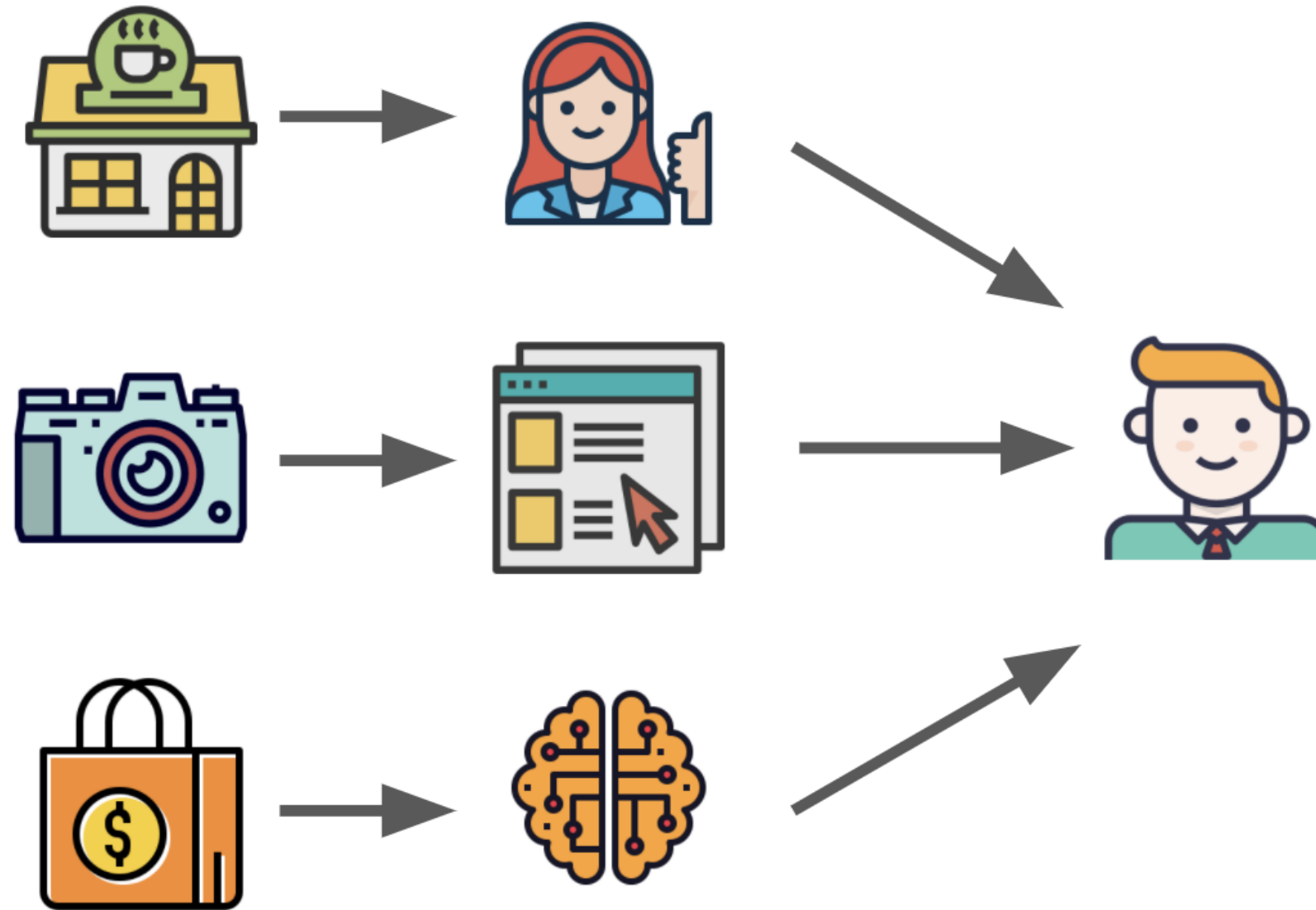
What are recommendation engines?

BUILDING RECOMMENDATION ENGINES IN PYTHON

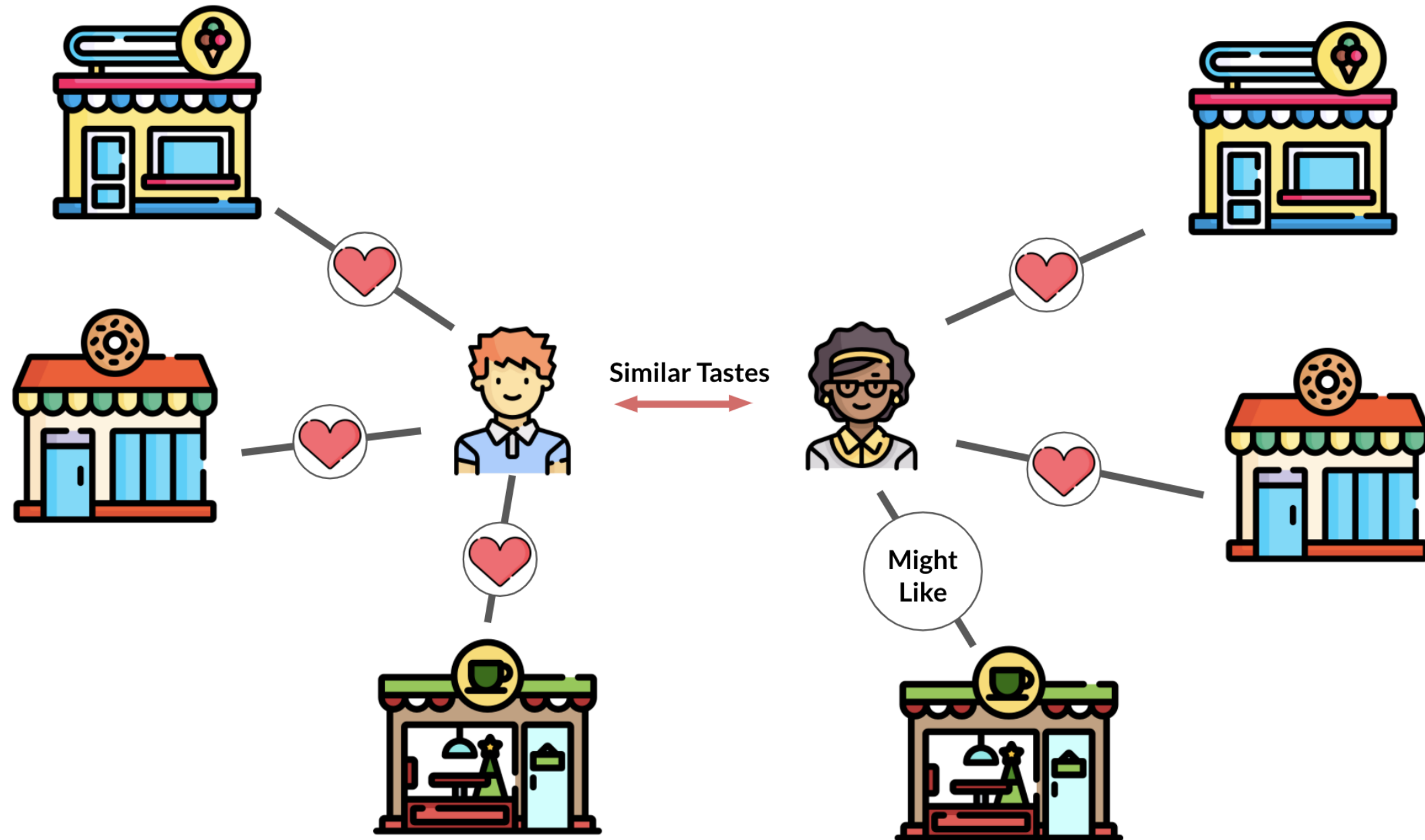


Rob O'Callaghan
Director of Data

What are recommendation engines?



What kind of data do I need?



What kind of data do I need?

User ID	Item ID	Review
User 1	Item 1	😞
User 1	Item 4	😄
User 2	Item 2	😐
User 2	Item 4	😞
...

	Item 1	Item 2	Item 3	Item 4	Item 5
User 1	😞			😄	😞
User 2		😐	😞		
User 3	😄		😐		
User 4		😄		😄	😐
User 5	😞		😄		😐

What are recommendation engines useful for?

Recommendation Engines

What movie should a viewer watch?

Will a diner enjoy a restaurant?

Other Statistical Models

Will a movie sell a lot of tickets?

How much is a house worth?

Implicit vs. explicit data

Implicit Feedback

A user often selecting the same genre of music.

An item is repeatedly purchased by the same consumer.

A viewer gives up halfway on a movie they were watching.

Explicit Feedback

Ratings given on movie sites like Rotten Tomatoes.

A user telling a social media platform that an ad is not relevant to their interests.

A user gives a thumbs up or thumbs down on YouTube.

Let's practice!

BUILDING RECOMMENDATION ENGINES IN PYTHON

Non-personalized recommendations

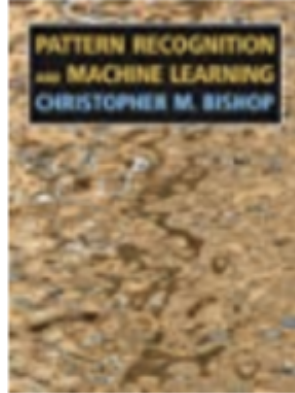

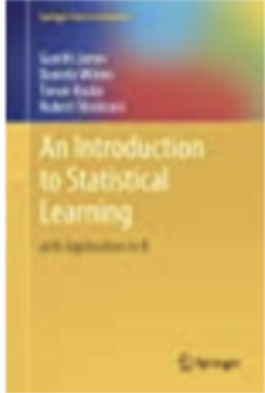

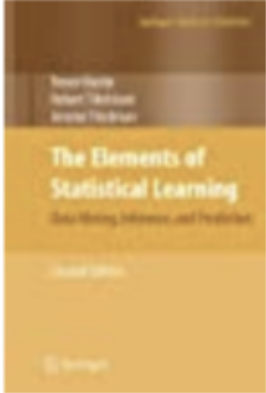
BUILDING RECOMMENDATION ENGINES IN PYTHON



Rob O'Callaghan
Director of Data

Non-personalized ratings


Frequently bought together



Total price: **\$209.02**

Add all three to Cart

Add all three to List

 These items are shipped from and sold by different sellers. [Show details](#)

- ☒ **This item:** The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition... by Trevor Hastie Hardcover **\$71.84**
- ☒ [An Introduction to Statistical Learning: with Applications in R \(Springer Texts in Statistics\)](#) by Gareth James Hardcover **\$65.39**
- ☒ [Pattern Recognition and Machine Learning \(Information Science and Statistics\)](#) by Christopher M. Bishop Hardcover **\$71.79**

Finding the most popular items

`book_df` DataFrame:

User	Book
User_233	The Great Gatsby
User_651	The Catcher in the Rye
User_131	The Lord of the Rings
User_965	Little Women
User_651	Fifty Shades of Grey
...	...

Finding the most popular items

```
book_df['book'].value_counts()
```

```
40 Shades of Grey          524
Harry Potter and the Sorcerer's Stone  487
The Da Vinci Code          455
The Twilight Saga          401
Lord of the Rings           278
...
```

Finding the most popular items

```
print(book_df.value_counts().index)
```

```
Index(['40 Shades of Grey', 'Harry Potter and the Sorcerer's Stone',  
      'The Da Vinci Code', 'The Twilight Saga',  
      'The Lord of the Rings'],  
      dtype='object')
```

Finding the most liked items

`user_ratings` DataFrame:

User	Book	Rating
User_233	The Great Gatsby	3.0
User_651	The Catcher in the Rye	5.0
User_131	The Lord of the Rings	3.0
User_965	Little Women	4.0
User_651	Fifty Shades of Grey	2.0
...

Finding the most liked items

```
avg_rating_df = user_ratings[["book", "rating"]].groupby(['book']).mean()  
avg_rating_df.head()
```

	rating
title	
Hamlet	4.1
The Da Vinci Code	2.1
Gone with the Wind	4.2
Fifty Shades of Grey	1.2
Wuthering Heights	3.9
	...

Finding the most liked items

```
sorted_avg_rating_df = avg_rating_df.sort_values(by="rating", ascending=False)
sorted_avg_rating_df.head()
```

	rating
title	
The Girl in the Fog	5.0
Behind the Bell	5.0
Across the River and into the Trees	5.0
The Complete McGonagall	5.0
What Is to Be Done?	5.0
	...

Finding the most liked items

```
(user_ratings['title']=='The Girl in the Fog').sum()
```

1

```
(user_ratings['title']=='Valley of the Dolls').sum()
```

1

```
(user_ratings['title']=='Across the River and into the Trees').sum()
```

1

Finding the most liked popular items

```
book_frequency = user_ratings["book"].value_counts()  
print(book_frequency)
```

```
40 Shades of Grey          524  
Harry Potter and the Sorcerer's Stone  487  
...
```

```
frequently_reviewed_books = book_frequency[book_frequency > 100].index  
print(frequently_reviewed_books)
```

```
Index([u'The Lord of the Rings', u'To Kill a Mockingbird', u'Of Mice and Men',  
       u'1984', u'Hamlet'])
```

Finding the most liked popular items

```
frequent_books_df = user_ratings_df[user_ratings_df["book"].isin(frequently_reviewed_books)]
```

```
frequent_books_avgs = frequently_reviewed_books[["title", "rating"]].groupby('title').mean()  
print(frequent_books_avgs.sort_values(by="rating", ascending=False).head())
```

	rating
title	
To Kill a Mockingbird	4.7
1984.	4.7
Harry Potter and the Sorcerer's Stone	4.6
	...

Let's practice!

BUILDING RECOMMENDATION ENGINES IN PYTHON

Non-personalized suggestions

BUILDING RECOMMENDATION ENGINES IN PYTHON



Robert O'Callaghan
Director of Data

Identifying pairs

```
0      User_223          The Great Gatsby <---| Read by the same user
1      User_223      The Catcher in the Rye <---|
2      User_131          The Lord of the Rings
3      User_965          Little Women <---| Read by the same user
4      User_965      Fifty Shades of Grey <---|
... ..
```

Permutations versus combinations

User	book_title
User_233	The Great Gatsby
User_233	The Catcher in the Rye

To:

	Book A	Book B
0	The Great Gatsby	The Catcher in the Rye
1	The Catcher in the Rye	The Great Gatsby

Books seen with The Great Gatsby -> The Catcher in the Rye

Books seen with The Catcher in the Rye -> The Great Gatsby

Creating the pairing function

```
from itertools import permutations

def create_pairs(x):

    return pairs
```

Creating the pairing function

```
from itertools import permutations

def create_pairs(x):
    pairs = permutations(x.values, 2)

    return pairs
```

- `permutations(list, length_of_permutations)` Generates iterable object containing all permutations

Creating the pairing function

```
from itertools import permutations

def create_pairs(x):
    pairs = list(permutations(x.values, 2))

    return pairs
```

- `permutations(list, length_of_permutations)` Generates iterable object containing all permutations
- `list()` Converts this object to a usable list

Creating the pairing function

```
from itertools import permutations

def create_pairs(x):
    pairs = pd.DataFrame(list(permutations(x.values, 2)),
                          columns=['book_a', 'book_b'])

    return pairs
```

- `permutations(list, length_of_permutations))` Generates iterable object containing all permutations
- `list()` Converts this object to a usable list
- `pd.DataFrame()` Converts the list to a DataFrame containing the columns `book_a` and `book_b`

Applying the function to the data

```
book_pairs = book_df.groupby('userId')['book_title'].apply(perm_function)
print(book_pairs.head())
```

		book_a	book_b
userId			
User_223	0	The Great Gatsby	The Catcher in the Rye
	1	The Catcher in the Rye	The Great Gatsby
User_965	0	Little Women	40 Shades of Grey
	1	40 Shades of Grey	Little Women
User_773	0	The Twilight Saga	Harry Potter and the Sorcerer's Stone
			...

Cleaning up the results

```
book_pairs = book_pairs.reset_index(drop=True)  
print(book_pairs.head())
```

```
      book_a      book_b  
0  The Great Gatsby  The Catcher in the Rye  
1  The Catcher in the Rye  The Great Gatsby  
3      Little Women  40 Shades of Grey  
4  40 Shades of Grey      Little Women  
5  The Twilight Saga  Harry Potter and the Sorcerer's Stone  
                        ...
```

Counting the pairings

```
pair_counts = book_pairs.groupby(['book_a', 'book_b']).size()
```

book_a	book_b	
The Twilight Saga	Fifty Shades of Grey	16
	Pride and Prejudice	12
	...	

```
pair_counts_df = pair_counts.to_frame(name = 'size').reset_index()
print(pair_counts_df.head())
```

	book_a	book_b	size
1	The Twilight Saga	Fifty Shades of Grey	16
2	The Twilight Saga	Pride and Prejudice	12
		...	

Looking up recommendations

```
pair_counts_sorted = pair_counts_df.sort_values('size', ascending=False)
```

```
pair_counts_sorted[pair_counts_sorted['book_a'] == 'Lord of the Rings']
```

	book_a	book_b	size
137	Lord of the Rings	The Hobbit	12
147	Lord of the Rings	Harry Potter and the Sorcerer's Stone	10
143	Lord of the Rings	The Colour of Magic	9
		...	

Let's practice!

BUILDING RECOMMENDATION ENGINES IN PYTHON