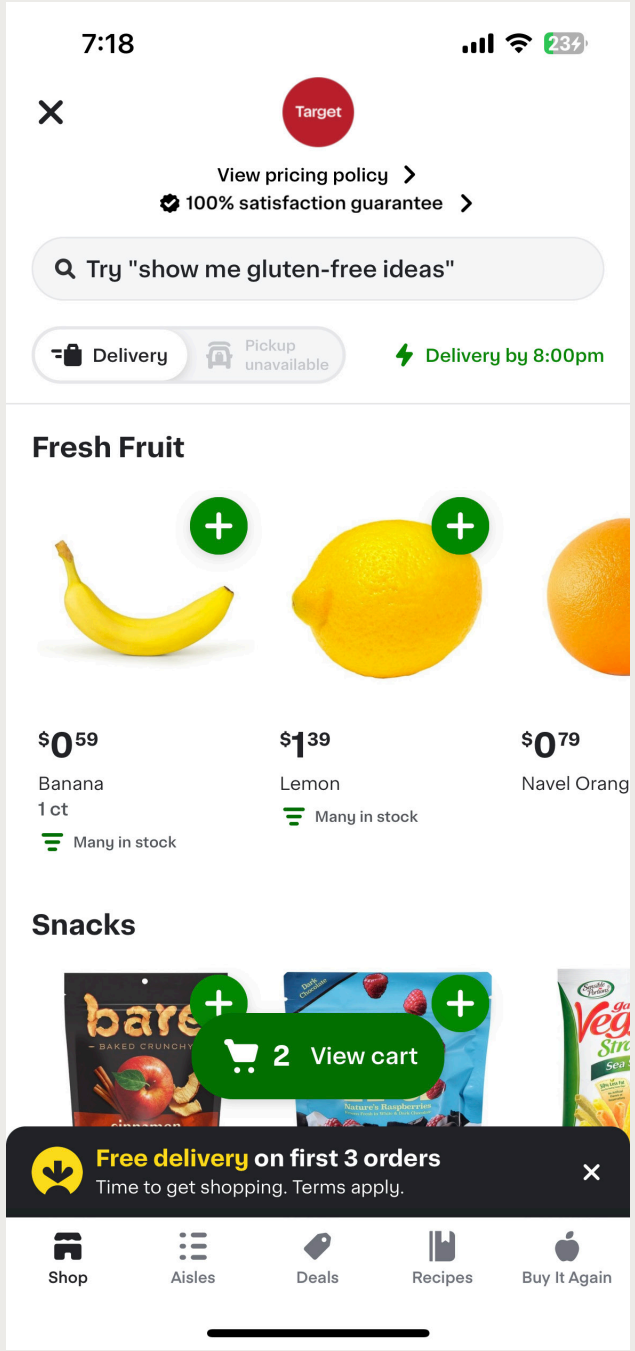
Instacart

Building a Recommendation Engine

Paige Landis | GA Data Science | 2023



What is Instacart?

Instacart is an online grocery / product delivery service. Users select a store, a pickup time, and the items they want to order. Those items picked up from the store and delivered to the user in a set amount of time by an instacart shopper.

The Problem

Instacart replaces the in store shopping experience. For user that like the novelty of walking through the aisles to find new items to try, some of the user experience may be lost in switching over to online shopping.

Defining the Research Project

The Question

How do we account for the novelty item discovery of in person shopping?

Assumptions

Recommendation engines will work best with regular users of Instacart

Hypothesis

Building a recommendation engine will introduce that novelty back in to online shopping by giving users new products to buy based on their purchase history

Metrics of Success

Make a recommendation engine that returns five reccomendations to a user

Data We Have

Aisles

- aisle_id
- aisle

Departments

- department_id
- department

Orders

- order_id
- user_id
- eval_set
- order_number
- order_dow
- order_hour_of_day
- days_since_prior_order

Products

- product_id
- product_name
- aisle_id
- department_id

Ordered Products

- order_id
- product_id
- add_to_cart_order
- reordered

Data We Want

1. **User ID**
   from Orders data set
2. **Product Name**
   from Products data set
3. **Product ID**
   from products data / ordered products
4. **Order ID**
   from ordered products

New Data View

	user_id	product_id	user_count	product_name	overall_popularity
0	1	196	10	Soda	35791
1	1	12427	10	Original Beef Jerky	6476
2	1	10258	9	Pistachios	1946
3	1	25133	8	Organic String Cheese	6196
4	1	13032	3	Cinnamon Toast Crunch	3751
5	1	46149	3	Zero Calorie Cola	8558
6	1	13176	2	Bag of Organic Bananas	379450
7	1	26088	2	Aged White Cheddar Popcorn	2523
8	1	26405	2	XL Pick-A-Size Paper Towel Rolls	1214
9	1	49235	2	Organic Half & Half	76360
10	1	10326	1	Organic Fuji Apples	5526
11	1	14084	1	Organic Unsweetened Vanilla Almond Milk	15935
12	1	17122	1	Honeycrisp Apples	13880
13	1	30450	1	Creamy Almond Butter	21490
14	1	35951	1	Organic Unsweetened Almond Milk	57895

1. Who is our user?
2. What is the ID of the product they're ordering?
3. How many times has our user ordered that product?
4. What is the name of the product?
5. How popular is the product on at that store according to all Instacart orders in our data set?

Using Natural Language Processing (NLP)

Natural Language Processing (NLP) is a set of techniques that help turn human languages in to something that is digestible and usable for a computer.

Term Frequency–Inverse Document Frequency

An NLP technique that splits words in to tokens and counts how many times they appear in a document. It then measures how important the word is to the document. It’s often used in user modeling

Using TF-IDF to Build User Shopping Profiles

We'll use TF-IDF to understand the importance of different food items in a user's document (order history). The more they've purchased an item, the more important the item is going to be within their order history. This becomes their shopping profile that we can make recommendations off of.

User 1 Order History

Soda , Soda , Soda , Soda , Soda , Soda , Soda , Soda, Soda, Soda , Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Original Beef Jerky, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Pistachios, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Organic String Cheese, Cinnamon Toast Crunch, Cinnamon Toast Crunch, Cinnamon Toast Crunch, Zero Calorie Cola, Zero Calorie Cola, Zero Calorie Cola, Bag of Organic Bananas...

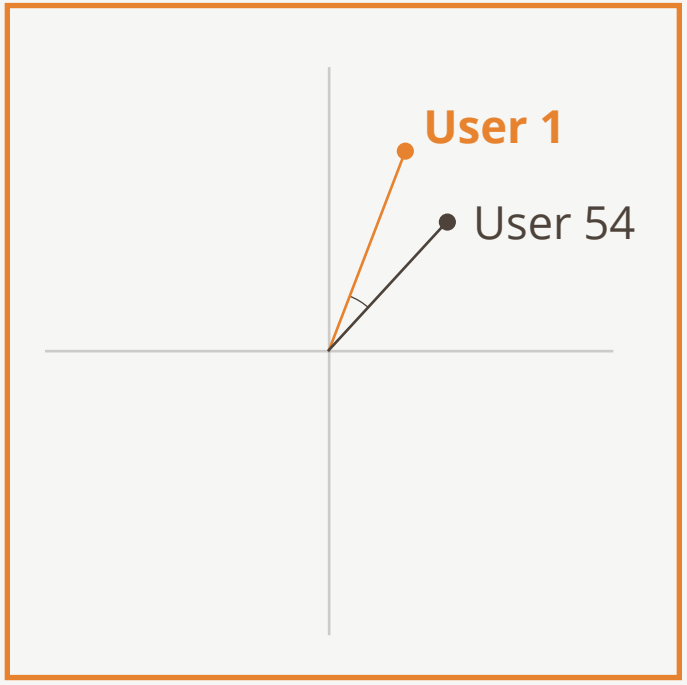
User 1 Order History – TF-IDF

Soda 10	Original Beef Jerky 10	Pistachios 9
Organic String Cheese 8	Cinnamon Toast Crunch 3	
Zero Calorie Cola 2	Bag of Bananas 2	
Aged White Cheddar Popcorn 2	Organic Half & Half 2	

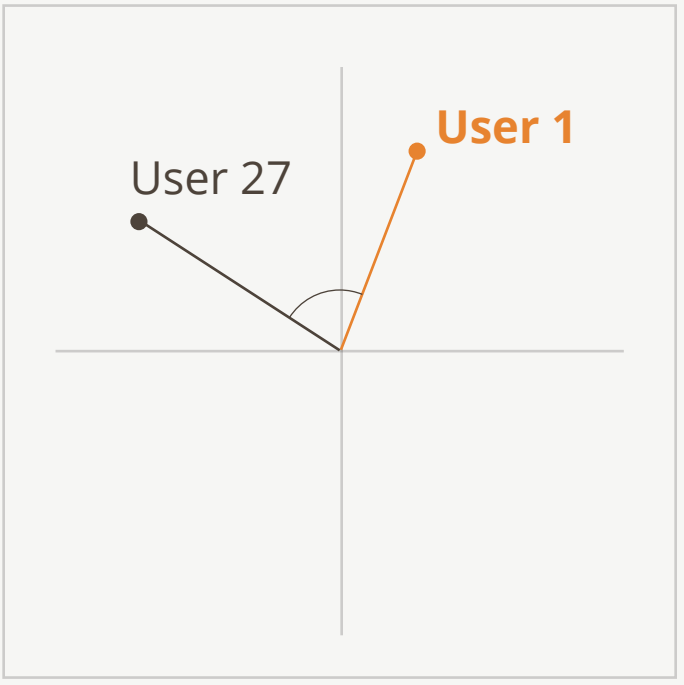
Finding Recommendations for Our User

We can use cosine similarity to find the user with the closest user profile to User 1. The smaller the angle between two users, the closer their documents (order history) are.

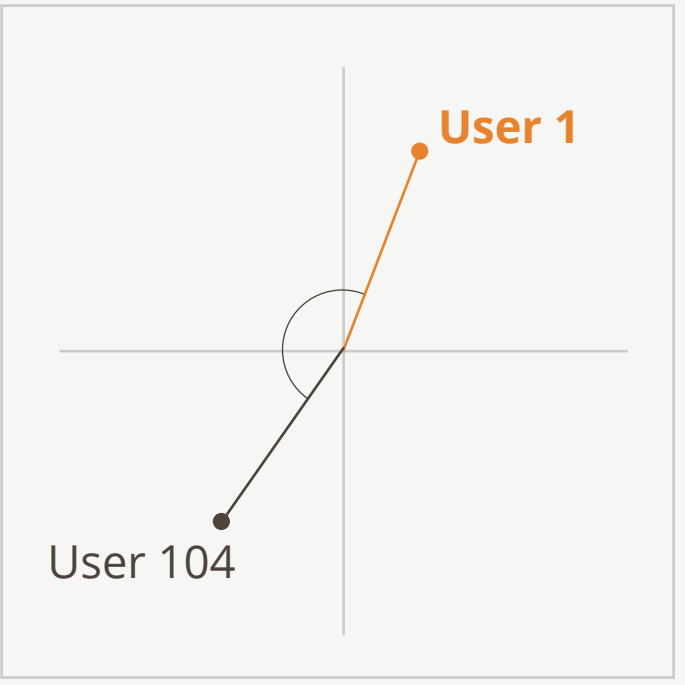
Most Similar to User 1



Less Similar to User 1



Almost opposite of User 1



Recommending New Products From User 1's Closest Profile

User 1 Shopping Profile

- Soda 10
- Original Beef Jerky 10
- Organic String Cheese 8
- Zero Calorie Cola 2
- Cinnamon Toast Crunch 3
- Aged White Cheddar Popcorn 2
- Bag of Bananas 2
- Organic Half & Half 2

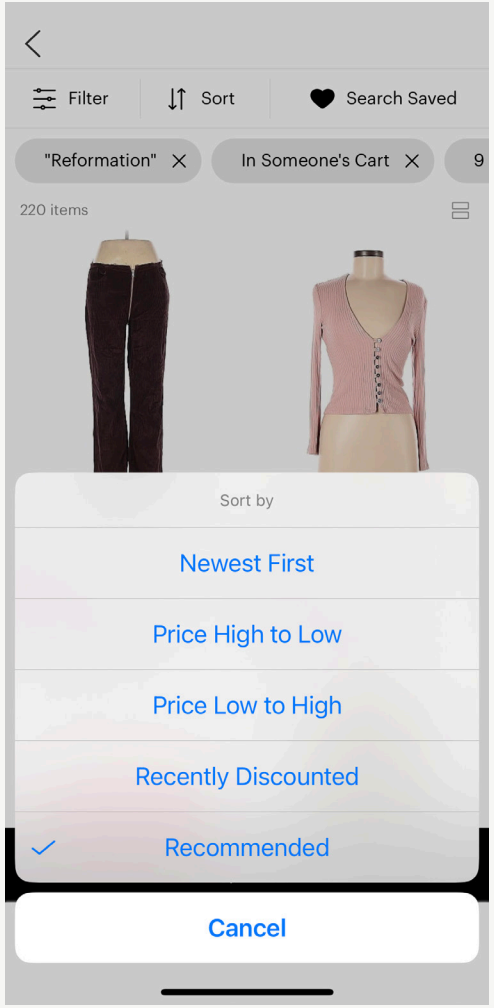
User 54 Shopping Profile

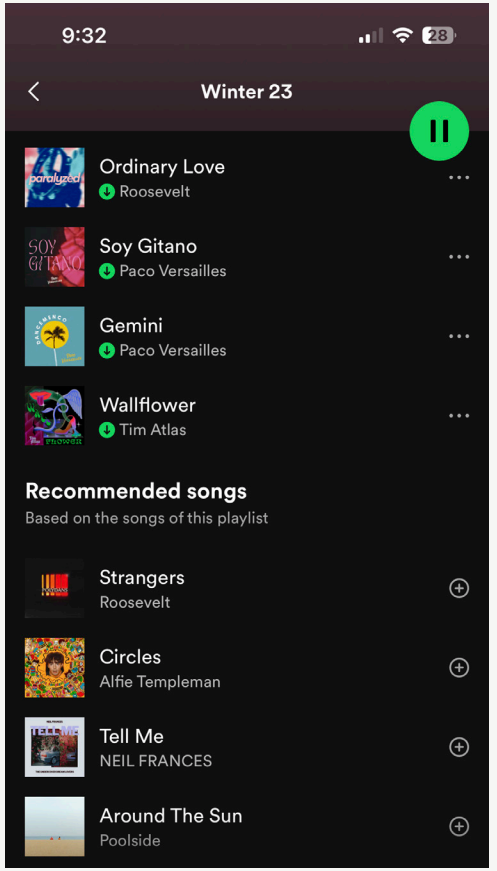
- Soda 13
- Original Beef Jerky 11
- Organic String Cheese 6
- Chocolate Almonds 5
- 0% Greek Strained Yogurt 3
- Aged White Cheddar Popcorn 3
- Bag of Bananas 3
- Organic Hass Avocado 2

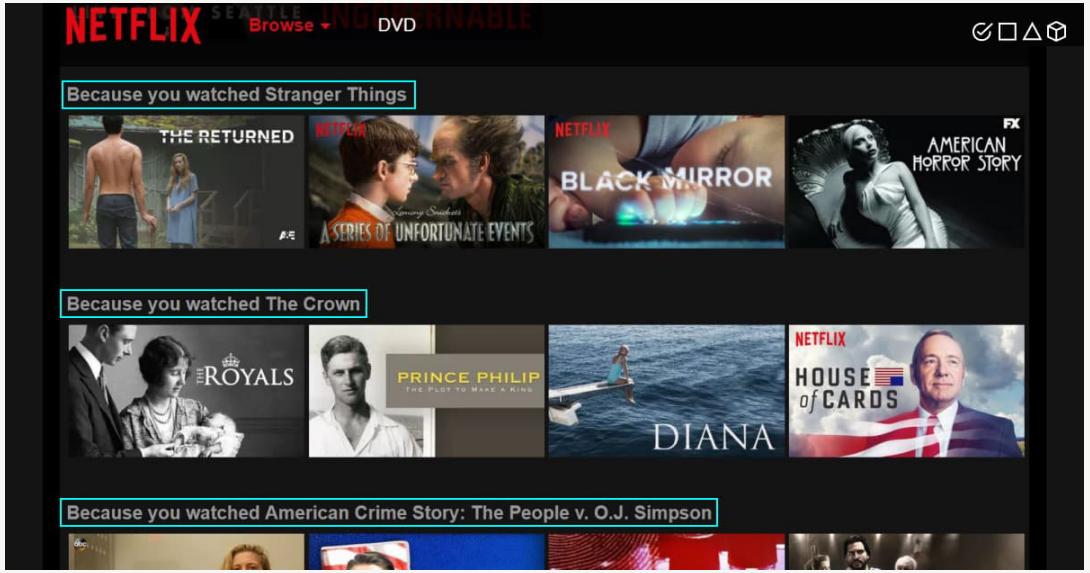
User 1 Recommendations

- Chocolate Almonds
- 0% Greek Strained Yogurt
- Organic Hass Avocado

Applying the Model to a User Interface







Applied Recommendation Engines

1. ThreadUp – Can sort clothing on what they would recommend for a user based on purchase history
2. Spotify – Has recommended songs based on playlists
3. Netflix – Recommends shows based on watch history

Next Steps

Model Return for User 1

1. Boneless Skinless Chiken Breasts
2. Total 2% Lowfat Greek Strained Yogurt with Peach
3. Total 2% Greek Strained Yogurt with Cherry 5.3 oz
4. Total 2% with Strawberry Lowfat Greek Strained Yogurt
5. Bag of Organic Bananas

Model Optimization

1. The model only uses part of the full data set
2. The model does not account for items currently in the users cart
3. It does not account for time – the model should account for changes in behavior in user shopping patterns
4. Some of the model's recommendations are repetitive.

Implement in to User Interface, Test With Google Analytics

Click-through rates : How often is a user clicking on a recommended item?

Questions?