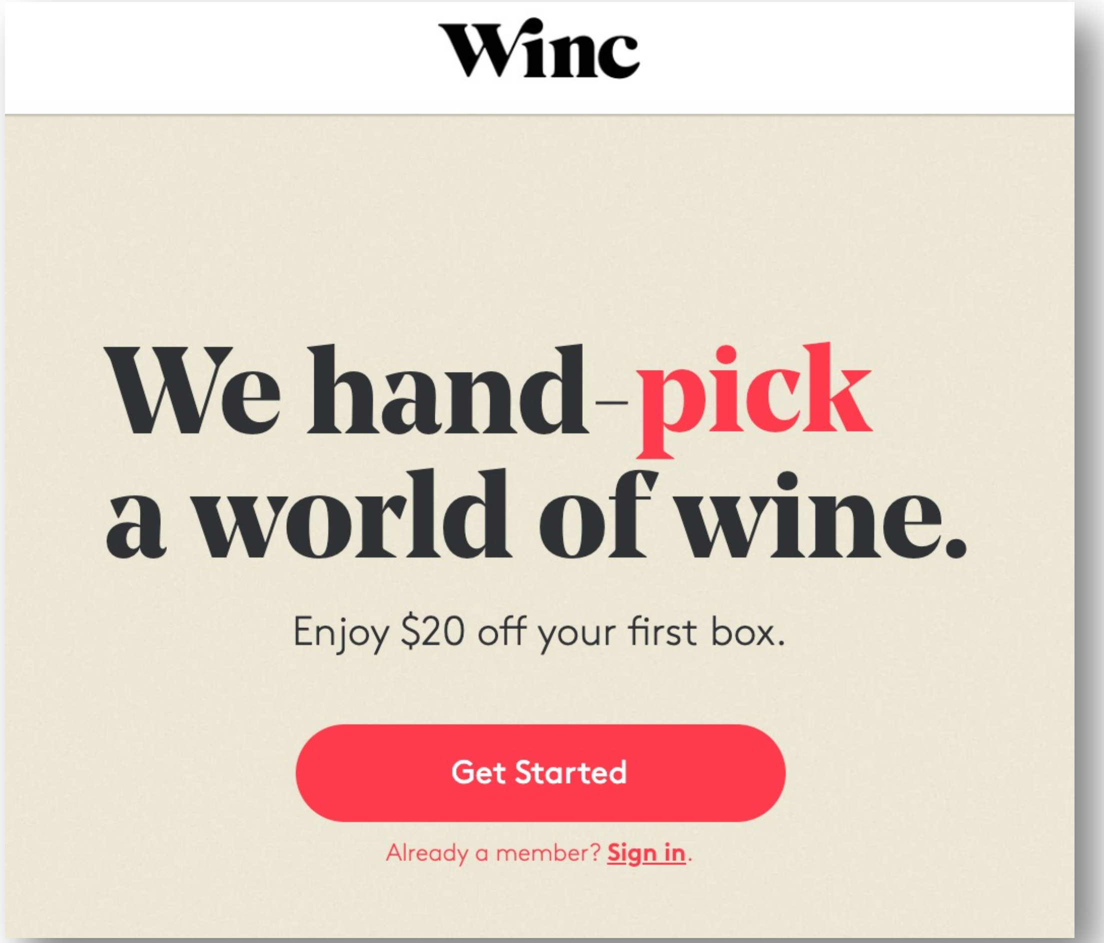
# **Adaptive Preference Measurement with Unstructured Data**, with Applications to Adaptive Onboarding Surveys

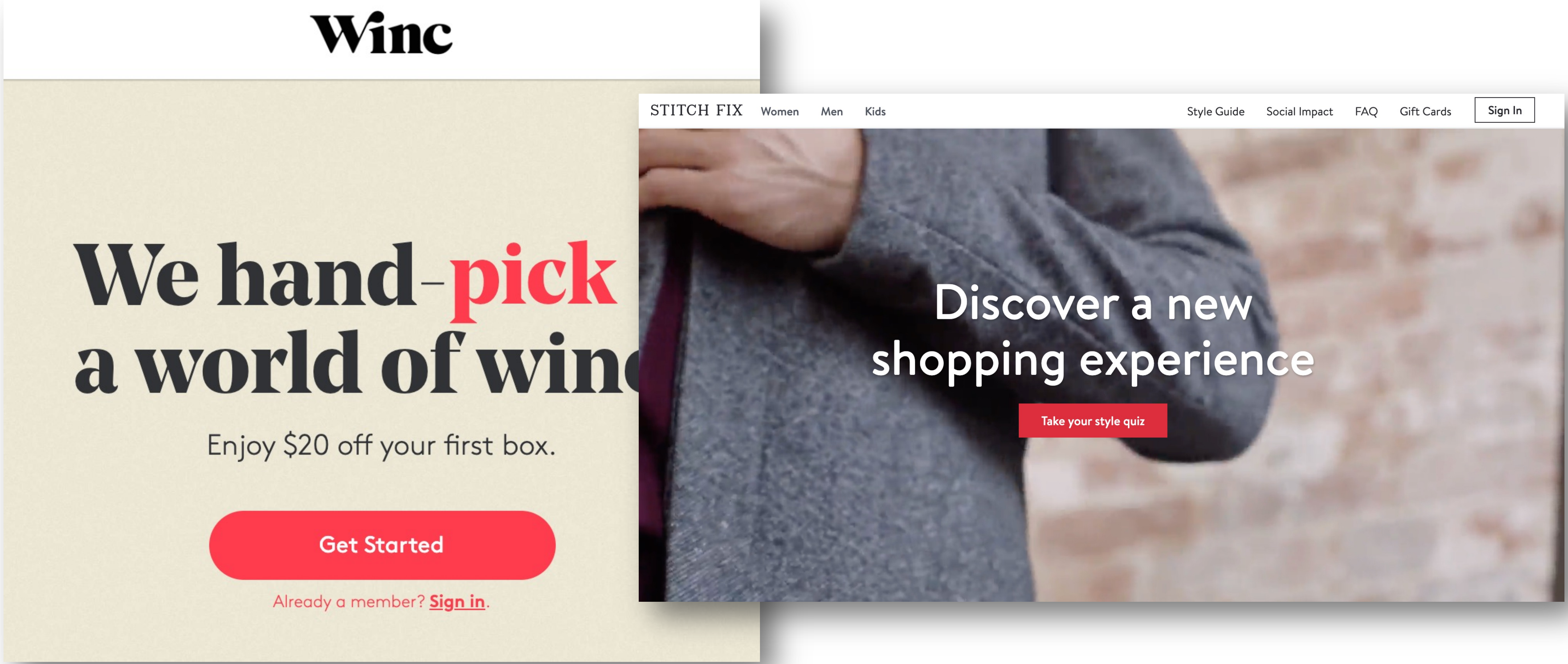
**Ryan Dew**
Assistant Professor of Marketing
The Wharton School
University of Pennsylvania

*TPM 2023*

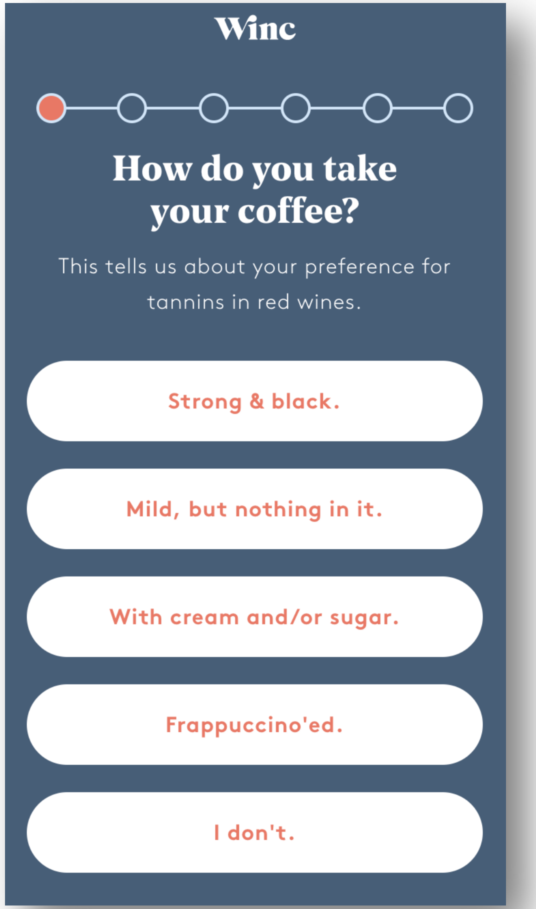
# Subscription Boxes

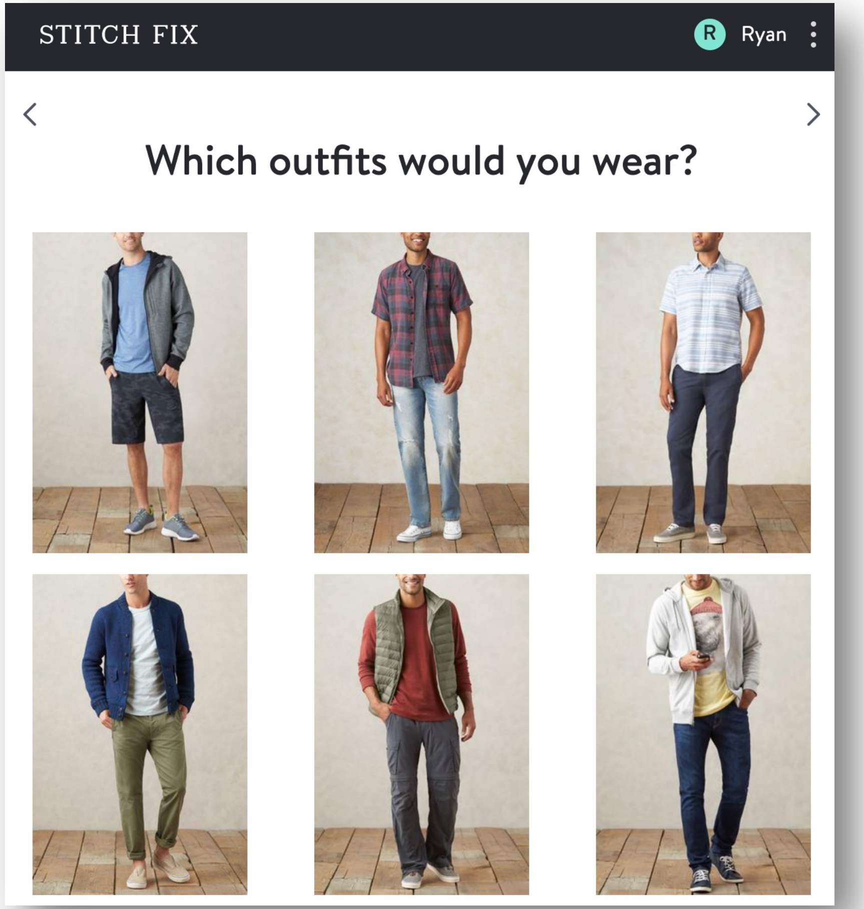


# Subscription Boxes

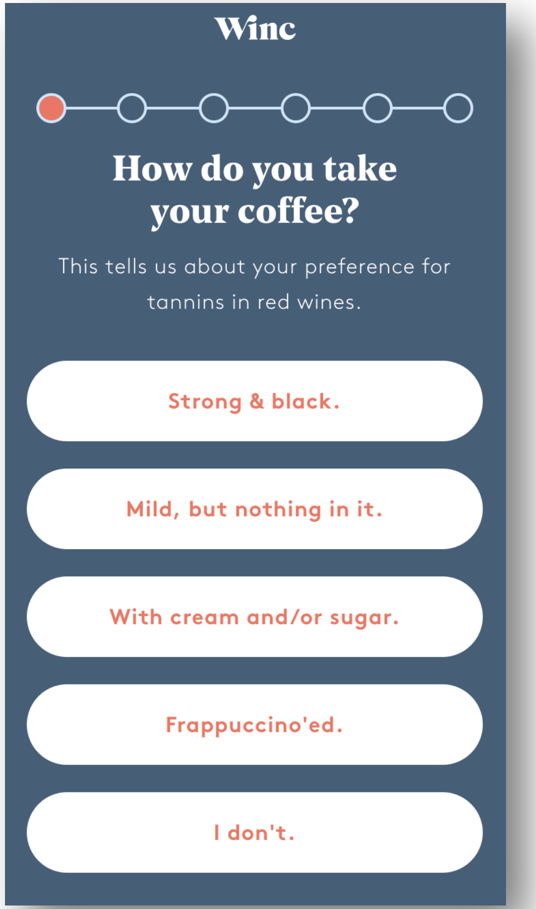


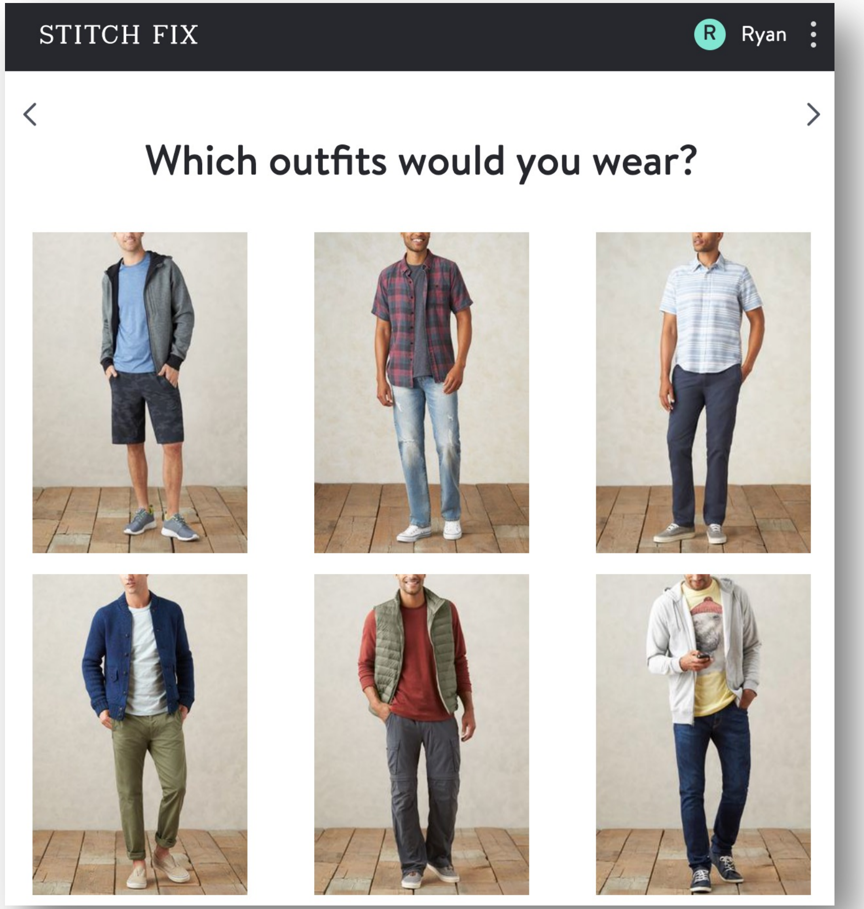
# On-boarding Surveys



# On-boarding Surveys




Different formats, same goal: **efficiently learn a user's preferences**

# This Project

Representation Learning → 
- Convert unstructured product data (images) to (numeric) vector representations

+

Bayesian Optimization → 
- Nonparametrically model person's utility function over representation space
- Adaptively select query points to reduce uncertainty

=

Preference Measurement → 
- Initializing recommendations and the on-boarding process
- Nonlinear preference prediction without resorting to pre-specified attributes/levels

# Step 1: Representation Learning

### Recommender-based

- Matrix factorization or modern neural collaborative filtering algorithms
- Produce representations of products as a “by-product” of learning: for person i, item j,

$$r_{ij} = u_i \cdot z_j$$

- Product representations are implicit: $z_1 = z_2$ if people tend to buy / like both products 1 and 2

# Step 1: Representation Learning

### Recommender-based

- Matrix factorization or modern neural collaborative filtering algorithms
- Produce representations of products as a "by-product" of learning: for person i, item j,

$$r_{ij} = u_i \cdot z_j$$

- Product representations are implicit: $z_1 = z_2$ if people tend to buy / like both products 1 and 2

### Feature-based

- Given features $x_j$, find a representation $z_j \in \mathbb{R}^K$ such that $x_j = f(z_j)$
- Numeric features: PCA
- Textual features:
  - Latent Dirichlet allocation
  - Embedding-based methods (doc2vec)
- Image features:
  - Variational autoencoders (VAE)
  - **Pre-trained / transfer learning**

# Step 2: Bayesian Optimization

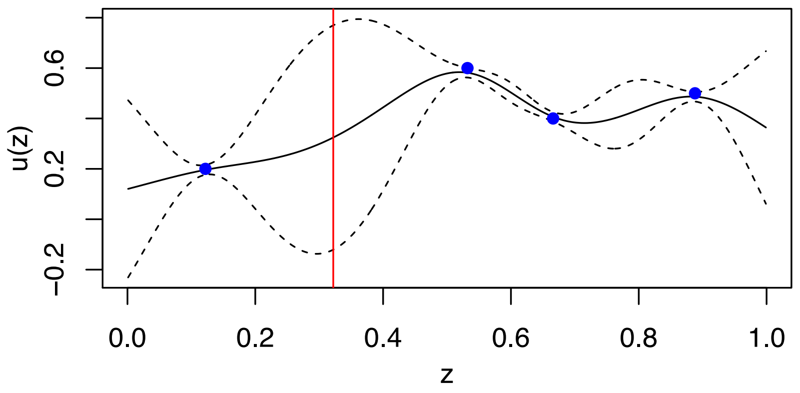
Quickly learn a user's utility over representations, u(z),
by having the user **rate** products

Model of the utility function:

$$u(z) \sim \mathcal{GP}(m(z), k(z, z'))$$

Likelihood of product ratings:

$$r_m \sim \mathcal{N}(u(z_{j_m}), \sigma)$$



# Step 2: Bayesian Optimization

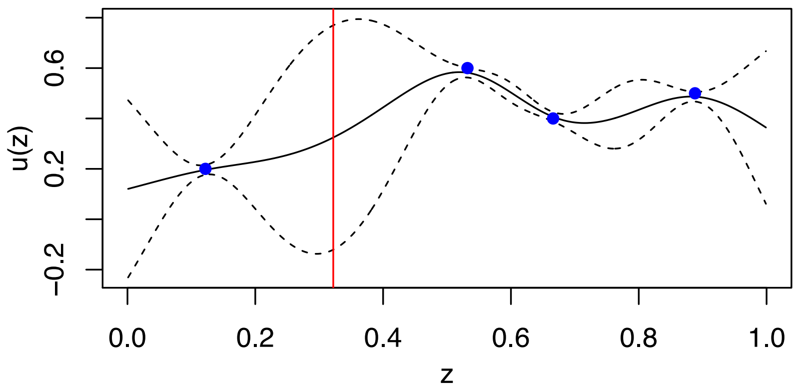
Quickly learn a user's utility over representations, u(z),
by having the user **rate** products

Model of the utility function:

$$u(z) \sim \mathcal{GP}(m(z), k(z, z'))$$

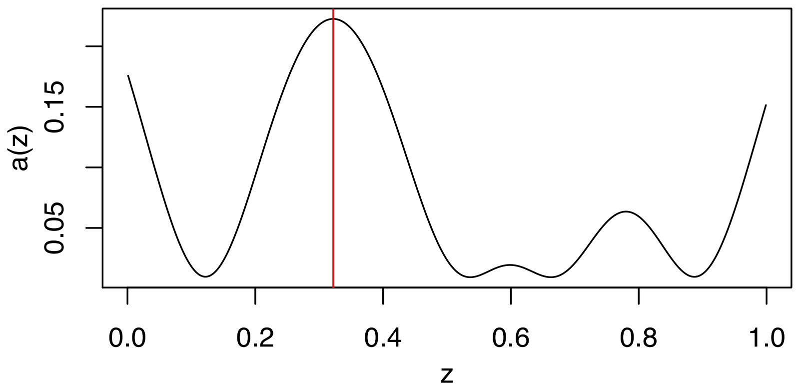
Likelihood of product ratings:

$$r_m \sim \mathcal{N}(u(z_{j_m}), \sigma)$$




**Acquisition function**: select next user query based on uncertainty over u(x)

- Maximum Uncertainty (MU)
- Upper Confidence Bound (UCB)
- *Global Uncertainty Reduction (GUR)*



# Experiments: Live Deployment of Method

Please rate the following dress
on a scale from 1-10.

Enter your rating for this dress

Submit

Which of these two dresses do you prefer?

or

Enter your preference for the dresses

| Strongly Prefer Left | Slightly Prefer left | Indifferent | Slightly Prefer Right | Strongly Prefer Right |
|---|---|---|---|---|
| | | | | |

Submit

# Results

In live tests, we can identify meaningful utility functions **even with as few as 20 questions**, using just unstructured data

# Results

In live tests, we can identify meaningful utility functions **even with as few as 20 questions**, using just unstructured data

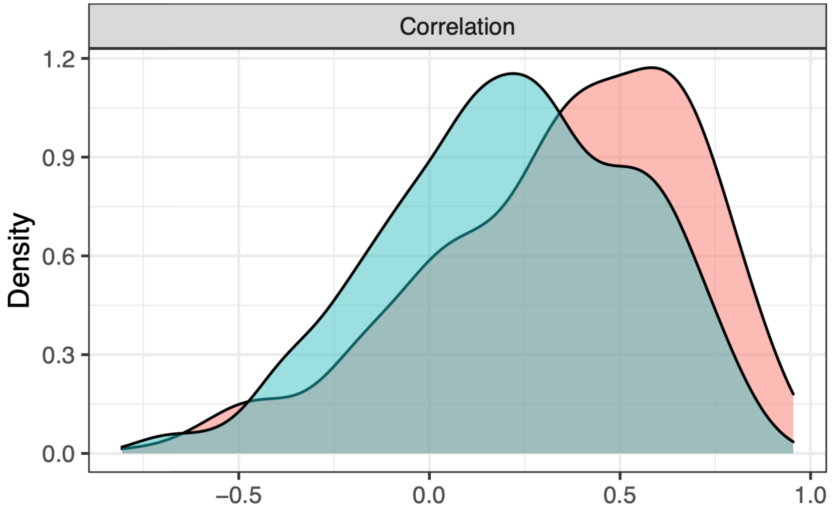



**Figure 1:** Distribution of individual-level correlations between predicted and actual item ratings for our method (red) versus a simple benchmark (blue)

# Results

In live tests, we can identify meaningful utility functions **even with as few as 20 questions**, using just unstructured data

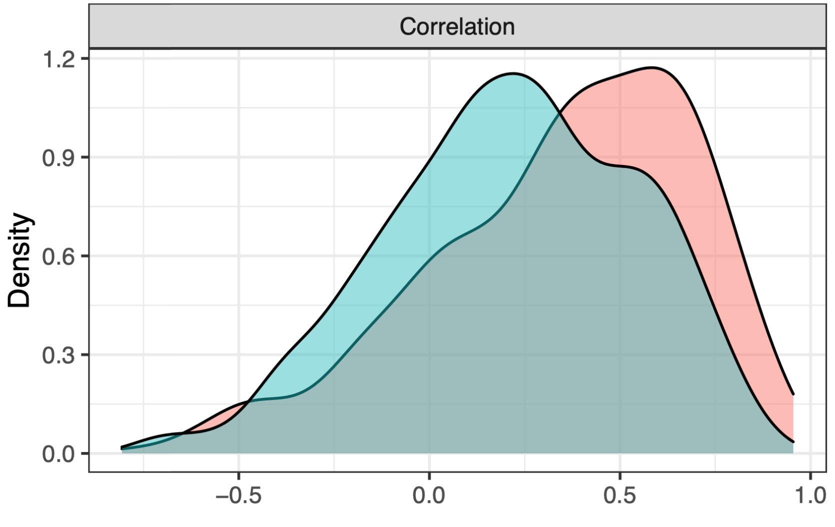



**Figure 1:** Distribution of individual-level correlations between predicted and actual item ratings for our method (red) versus a simple benchmark (blue)

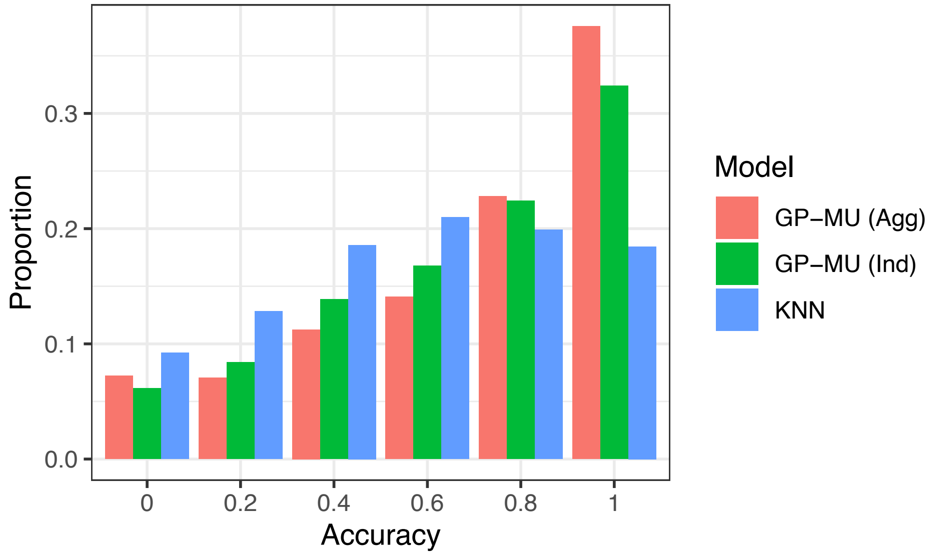
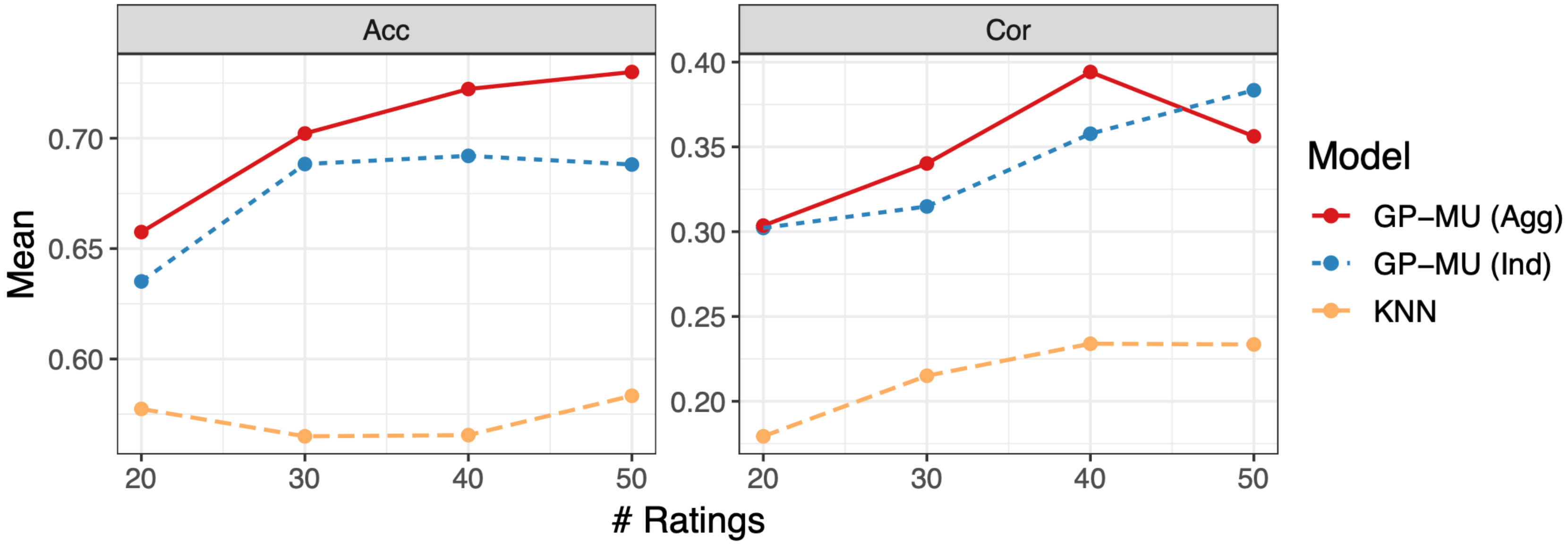
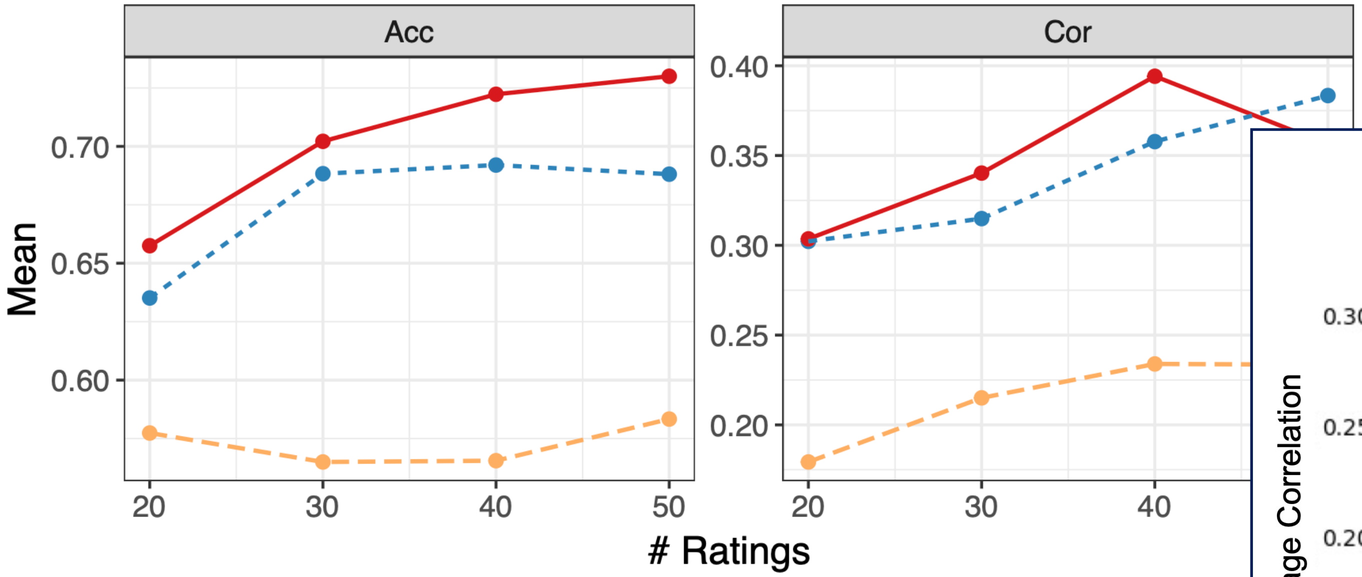


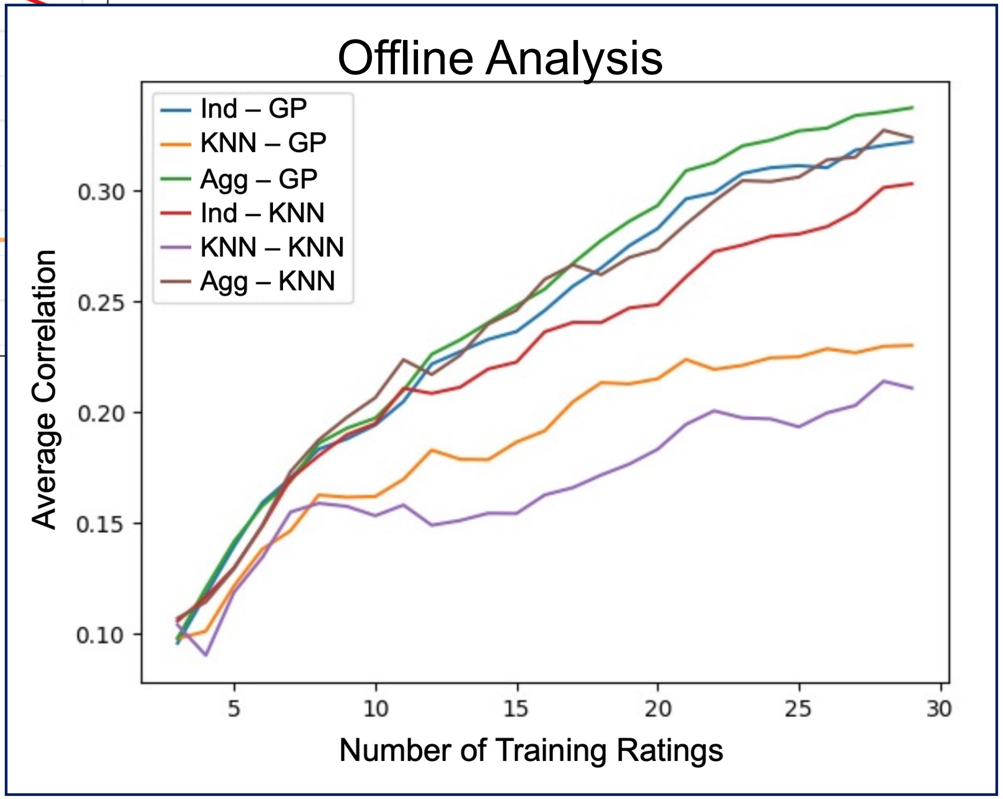

**Figure 2:** Comparison of three models in terms of correctly classifying whether a consumer would like 5 items. KNN is a k-nearest neighbors benchmark.

# Where do the gains come from?

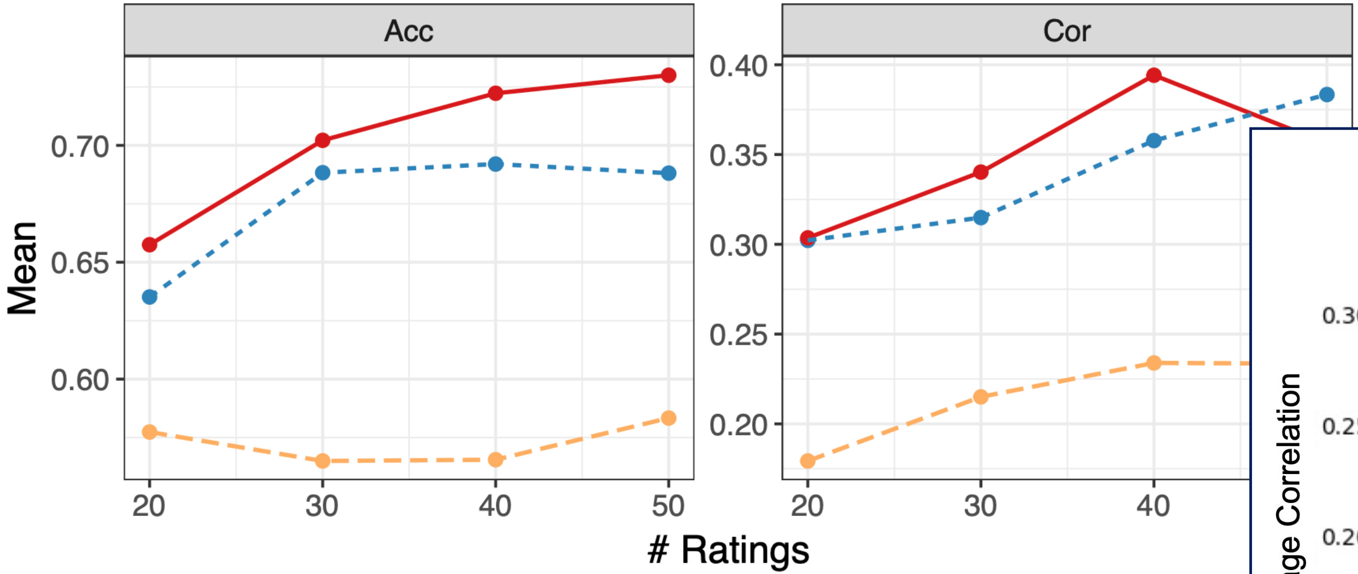


# Where do the gains come from?

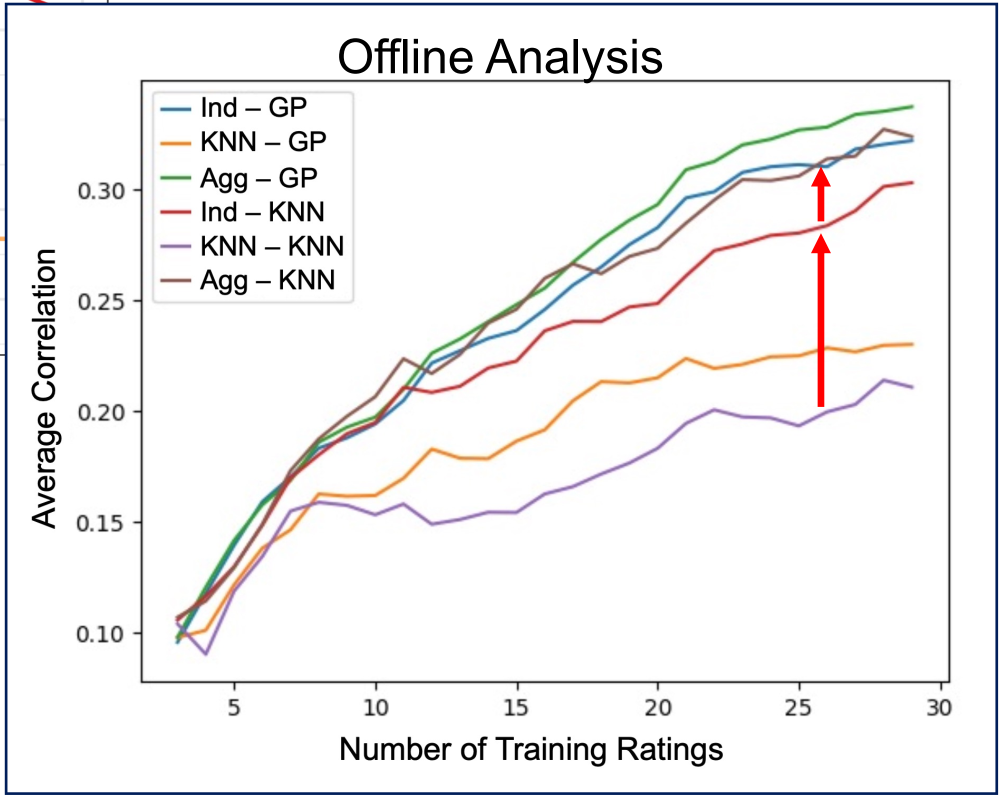


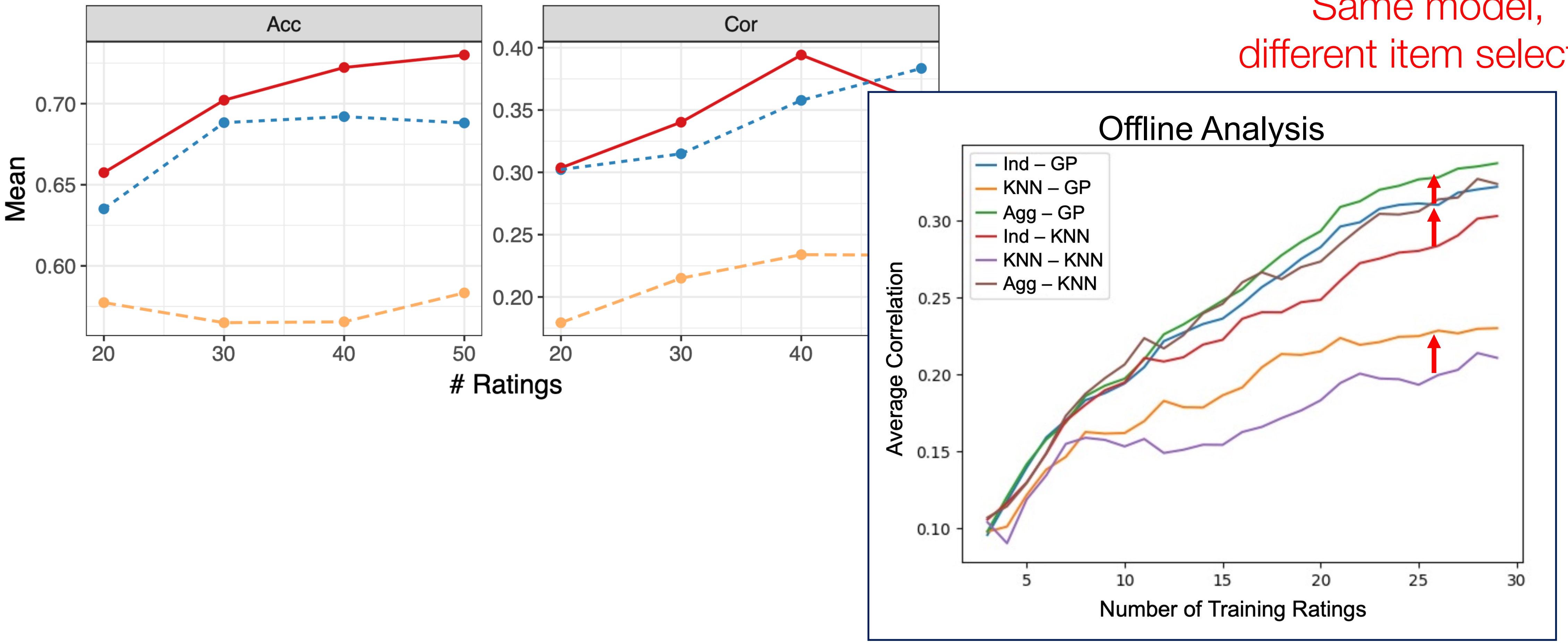
# Where do the gains come from?




Same data, different models



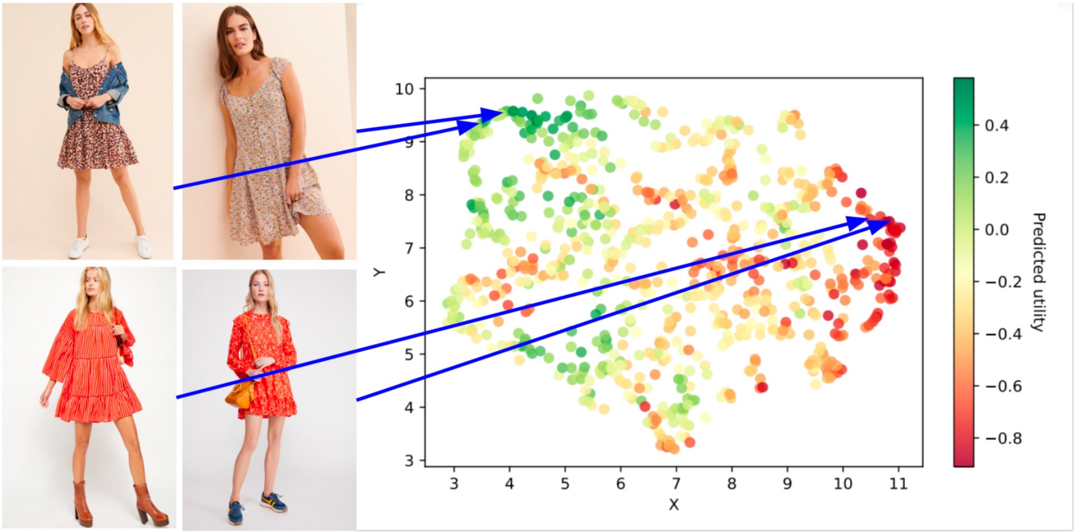
# Where do the gains come from?




Same model,
different item selection

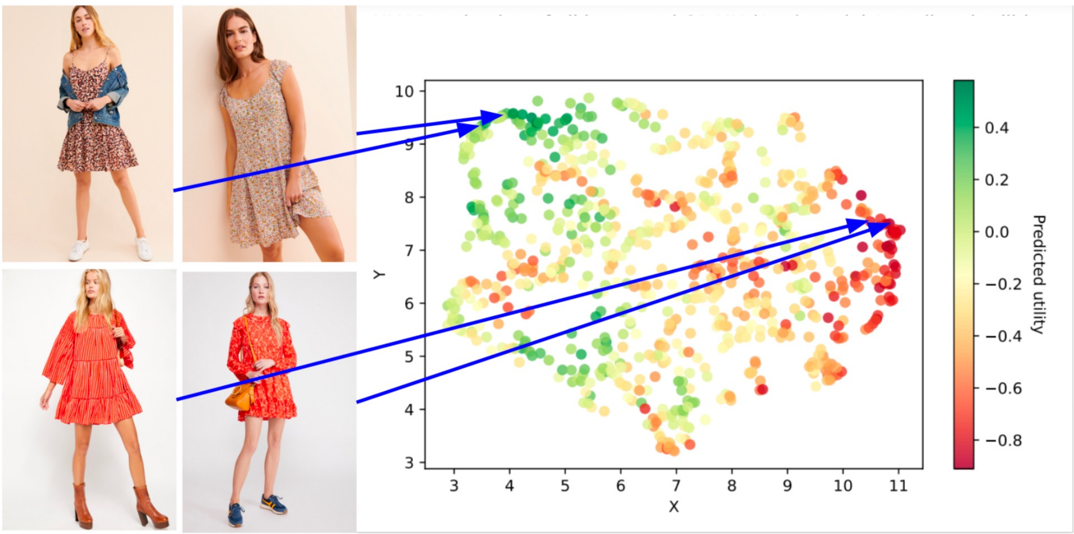
# Interpretation: Implications for Research

UMAP + Item Visualization



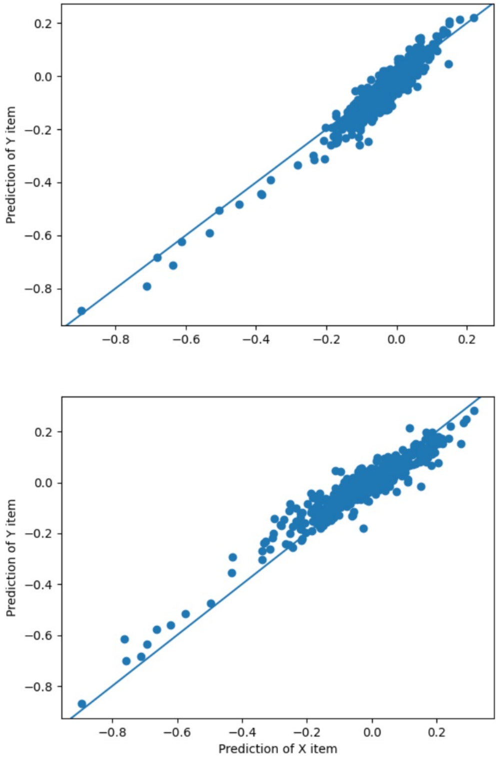
# Interpretation: Implications for Research

## UMAP + Item Visualization




## Paired Predictions

X Y



# Connecting Theory and Practice

- Our work leverages cutting edge methods for adaptive question design, built on decades of research on adaptive conjoint, and recent developments in the use of unstructured data

- We develop this methodology in a modern, applied context, to solve an increasingly prevalent problem in practice, especially for subscription industries

- The method works in real-time, using real world relevant data, without making unrealistic assumptions about firm data, or consumer preferences

- We introduce practical methodologies for interpreting the results, allowing firms to turn complex, messy data into actionable insights

# Thank you!

Questions / comments?

ryandew@wharton.upenn.edu

*Working paper available*

Special thanks to:

Wharton
University of Pennsylvania
ANALYTICS AT WHARTON

WHARTON BEHAVIORAL LAB

# Additional Slides

# Algorithm

**Algorithm 1:** Adaptive Preference Measurement

**Data:** Representations $z_j$ for products $j = 1, ..., J$, number of initial (random) ratings $M_0$, number of total ratings $M$

Ask user to rate $M_0$ initial products, $j_1, j_2, ..., j_{M_0}$

Optimize GP hyperparameters using Equation 7, and compute initial estimate estimate of $u(z)$ using Equation 4

**for** $m = 1, \ldots, M$ **do**

- Select next item for user to rate by $j_m = \arg\max_j a(z_j)$
- Show user item $j_m$, obtain rating $r_m$
- (Optional) Optimize GP hyperparameters using Equation 7 and the full history of ratings
- Update estimate of $u(z)$ using Equation 4 and the full history of ratings
- Update acquisition function $a(z)$ based on $u(z)$

**return** *User's utility function* $u(z)$

Step 1: Have the user rate some (random) initial products

Step 2: Initialize our model of u(z) using those initial product ratings

Repeat:

- Show next item based on the acquisition function
- Update posterior estimate of u(z) using full history