

Instacart Basket Prediction

Eitan Angel

June 27, 2019



1 Introduction

2 Exploration

3 Partition

4 Feature Design

5 Random Forest Classifier

6 TopN

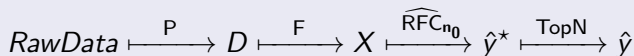
7 Prediction Explorer

8 References

- Instacart is a grocery-on-demand start-up which, in 2017, released a dataset containing 3 million orders from 200,000 (anonymized) users.
- A now-completed Kaggle competition asked entrants to predict: Which previously purchased products would be in a consumer's next order?
- In addition to returning predictions to optimize traditional metrics, our model includes product application variants for
 - more precise predictions to, for example, populate a user's cart
 - top- N most-likely products a particular user will purchase to, for example, display on a web page of fixed results

- For each user, Instacart provides between 4 and 100 of their orders, including the
 - intraorder sequence in which products were purchased,
 - week and hour of the day the order was placed,
 - relative time between orders, and
 - grocery store department to which each product belongs.
- All data obtained from the Kaggle competition website. A relational set of .csv files which describe customer orders over relative times. Each entity (customer, order, department, etc.) has a unique id.
- A blog post by Instacart provides some information about the dataset and a Kaggle competition provides additional details.

Structure



RawData: Exploratory Data Analysis

P: Partition *RawData* into D_{train} , D_{test} , ...

F: Feature Design builds design matrix X from D

\widehat{RFC}_{n_0} : Random Forest Classifier makes probabilistic predictions \hat{y}^* from X

n_0 : Hyperparameters found using OOB classifier

TopN: TopN Variants make binary predictions \hat{y} from \hat{y}^*

\hat{y} : Prediction Explorer inspects binary predictions \hat{y}

Table: order_products*.csv

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0
1	49683	4	0
1	43633	5	1

Dictionaries `products.csv`, `aisles.csv`, and `departments.csv` provide a map from `product_id` to `product_name`, `aisle_id`, `aisle_name`, `department_id`, and `department_name`.

Table: orders.csv

order_id	user_id	order_num	dow	hour	days
2539329	1	1	2	8	NaN
2398795	1	2	3	7	15.0
473747	1	3	3	12	21.0
2254736	1	4	4	7	29.0
431534	1	5	4	15	28.0

order_num, dow, hour, and days are abbreviations of order_number, order_dow, order_hour_of_day, and days_since_prior_order.

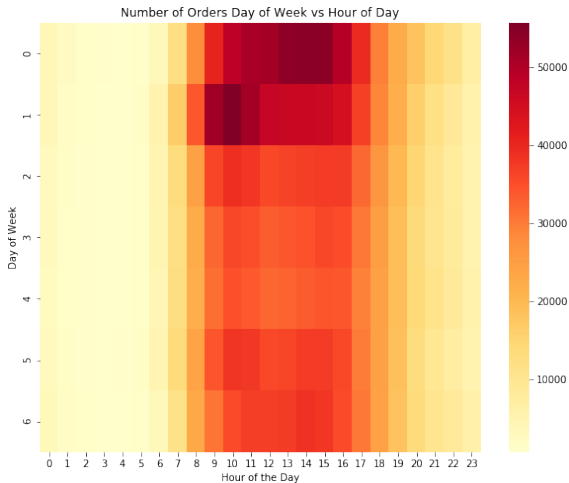


Figure: Saturday afternoon and Sunday morning are the most popular time to make orders.

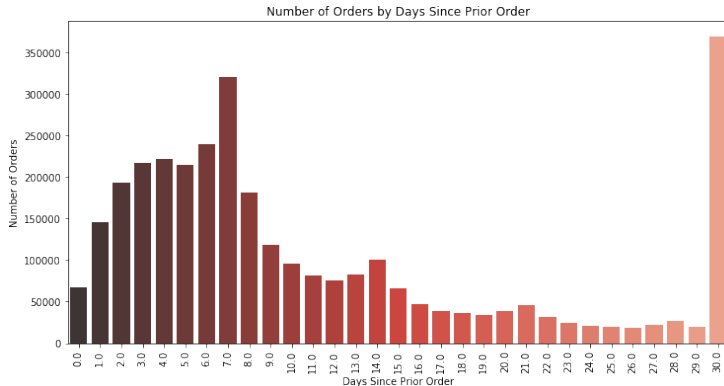


Figure: The most popular relative time between orders is monthly (30 days), but there are “local maxima” at weekly (7 days), biweekly (14 days), triweekly (21 days), and quadriweekly (28 days).

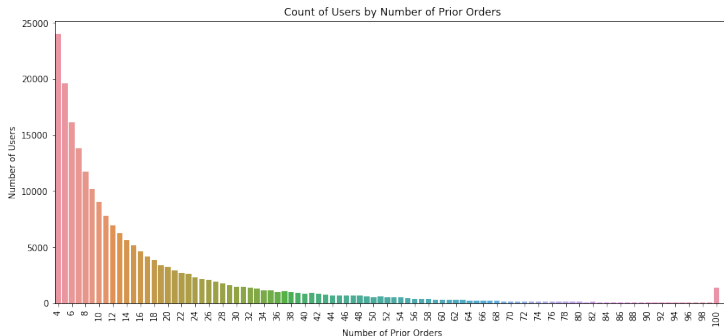


Figure: Users have between 4 – 100 orders. Those users in the dataset with 100 orders seem to have at least 100 orders and we have only their most recent 100 orders.

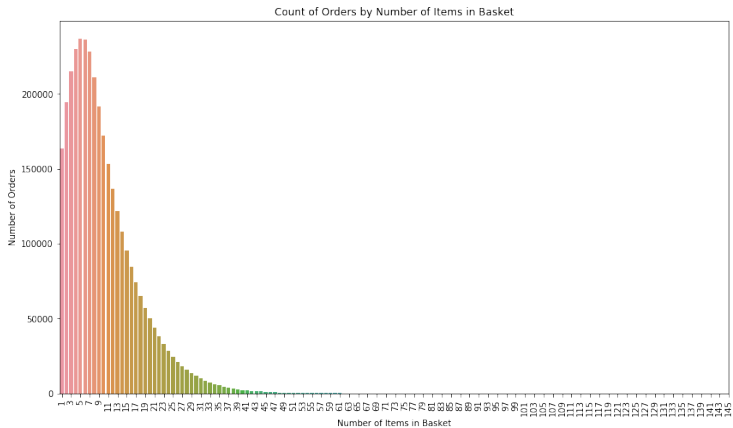


Figure: As one should expect, this distribution is right-skew. The mode basket size is 5.

product_name	count
Banana	491291
Bag of Organic Bananas	394930
Organic Strawberries	275577
Organic Baby Spinach	251705
Organic Hass Avocado	220877
Organic Avocado	184224
Large Lemon	160792
Strawberries	149445
Limes	146660
Organic Whole Milk	142813

Table: The most popular products are organic fruits or vegetables.

product_name	reorder
Raw Veggie Wrappers	0.9420
Serenity Ultimate Extrema ...	0.9333
Chocolate Love Bar	0.9215
Bars Peanut Butter	0.8985
Soy Crisps Lightly Salted	0.8955
Maca Buttercups	0.8942
Benchbreak Chardonnay	0.8918
Organic Blueberry B Mega	0.8888
Sparkling Water	0.8870
Fragrance Free Clay ...	0.8702

Table: The most reordered products with at least 50 orders.

aisle	department	reorder
milk	dairy eggs	0.7818
water seltzer...	beverages	0.7299
fresh fruits	produce	0.7188
eggs	dairy eggs	0.7063
soy lactosefree	dairy eggs	0.6923
...
beauty	personal care	0.2128
first aid	personal care	0.1958
kitchen supplies	household	0.1948
baking supplies ...	pantry	0.1675
spices seasonings	pantry	0.1529

Table: The most and least reordered from aisles.

Partition RawData

$$\text{RawData} \xrightarrow{P} \mathcal{D}^{\text{DSets}}$$

$$\text{RawData} \mapsto \{D_s\}_{s \in \text{DSets}}$$

eval_set	user_id
train	131209
test	75000

- $\text{DSets} = \{\text{train}, \text{test}, \text{kaggle}\},$
- P is the unique partition defined by a partition of the set of users, U , ordered by, say, `user_id`, into $\{U_s\}_{s \in \text{DSets}}$

U_{train} : 80% of 131,209 users with available ultimate orders.

U_{test} : 20% of 131,209 users with available ultimate orders.

U_{kaggle} : 75,000 users whose ultimate orders are withheld by Kaggle.

- $\{D_s\}_{s \in \text{DSets}}$ is the image of RawData under P .

Feature Design

$$F : D \rightarrow M_{n \times m}(\mathbb{R})$$

$$D \mapsto X$$

- $F = (f_j)_1^m$ is an m -tuple of functions $f_j : D \rightarrow \mathbb{R}^n$
 - a *feature* $f_j(D)$ is a column of X
 - f_j are compositions of aggregations, filtrations, arithmetic ...
 - compute some f_j via an unsupervised learning technique, Latent Dirichlet Allocation (LDA), introduced in [BNJ03].
- $Prod(u)$ are all products purchased by $u \in U$
- Rows of X are user-product pairs

$$((u, p) \mid u \in U \ \& \ p \in Prod(u))$$

- $X \in M_{n \times m}(\mathbb{R})$ with shape values

n_{train}	n_{test}	n_{kaggle}	m
6760791	1713870	4833292	47

- Encode labels $y_{\text{true}} \in \{0, 1\}^n$ as

$$y_{\text{true}}^{(u,p)} = \begin{cases} 1 & u \text{ purchased } p \text{ in ultimate order} \\ 0 & u \text{ did not purchase } p \text{ in ultimate order} \end{cases}$$

- Inspired by [LNZ⁺16], group features into a few Profiles:

U	User Profile
P	Product Profile
UP	User-Product Profile
LDA	Latent Dirichlet Allocation User Features





feature	dtype	description
U_items_total	uint16	number of total items a given user has purchased
U_reordered_ratio	float16	proportion of items a given user has purchased which are reorders
P_unique_users	uint16	number of purchasers
UP_order_ratio	float16	fraction of baskets in which a given product appears for a given user (count of orders in which product appears divided by total orders)
UP_penultimate	bool	product in user's penultimate (previous) order

Random Forest Classifier Map

$$\widehat{\text{RFC}}_{\mathbf{n}_0} : M_{n \times m}(\mathbb{R}) \rightarrow [0, 1]^n$$

$$X \mapsto \hat{y}^*$$

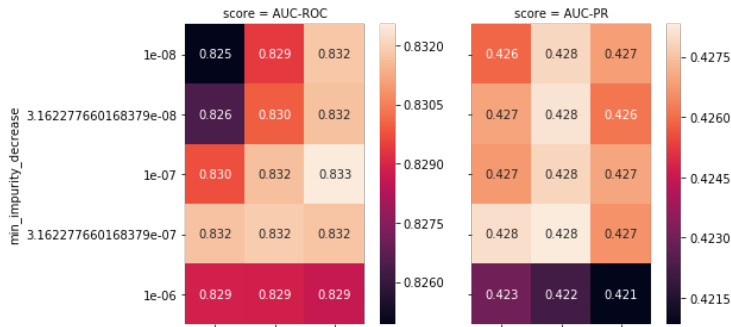
for a(n optimal) choice of hyperparameters \mathbf{n}_0 ; number of trees B .

-  s: information criterion determines recursive binary splittings by hyperplanes \perp axes
-  classifier: \mathbf{n} determines distribution on  topology; empirical risk minimization with respect to some loss L
- $\widehat{\text{RFC}}_{\mathbf{n}}$: average  estimators on bootstrap samples $(\mathbf{Z}_b^*)_{b \leq B}$ but limit dimensions considered in splittings to `max_features`
- \hat{y}^* is *probabilistic prediction* – interpret as probability ranking

- \mathbf{Z}_b^* only includes 2/3 of samples – use the remaining 1/3 to obtain the corresponding out-of-bag (OOB) classifier

$$\widehat{\text{OOB}}_{\mathbf{n}} : M_{n \times m}(\mathbb{R}) \rightarrow [0, 1]^n$$

- OOB error \gtrapprox test error
- Use OOB error (on train!) instead of N -fold cross-validation
 - Simplifies phases of project
 - N -fold cross-validation is memory-intensive in splitcopy data
- OOB rank metric scores for $\{\widehat{\text{OOB}}_{\mathbf{n}}\}_{\mathbf{n} \in \mathbf{G}}$, \mathbf{G} a *parameter grid*,
 - `min_samples_leaf` $\in \{12, 24, 48\}$,
 - `min_impurity_decrease` $\in [10^{-8}, 10^{-7.5}, 10^{-7}, 10^{-6.5}, 10^{-6}]$,
 - + `sklearn.ensemble.RandomForestClassifier` defaults



- AUC-PR may be more relevant than AUC-ROC since

$$\text{skew} = \frac{\text{Negative Classes}}{\text{Positive Classes}} \approx 10$$

- AUC-PR argmax occurs close to AUC-ROC argmax:
 $n_0 = (\text{min_impurity_decrease} = 10^{-6.5},$
 $\text{min_samples_leaf} = 24, \dots)$

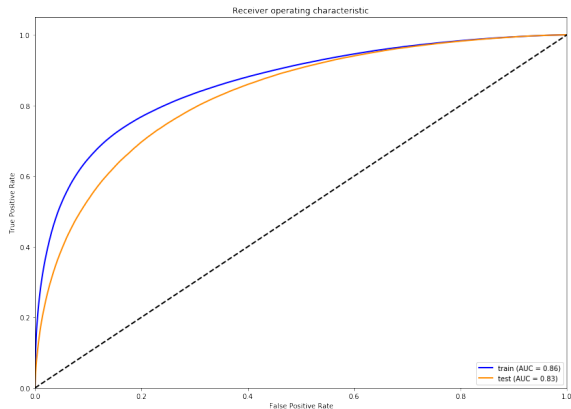


Figure: Receiver Operating Characteristic (ROC) curve for \widehat{RFC}_{n_0} ; AUC-ROC = 0.83.

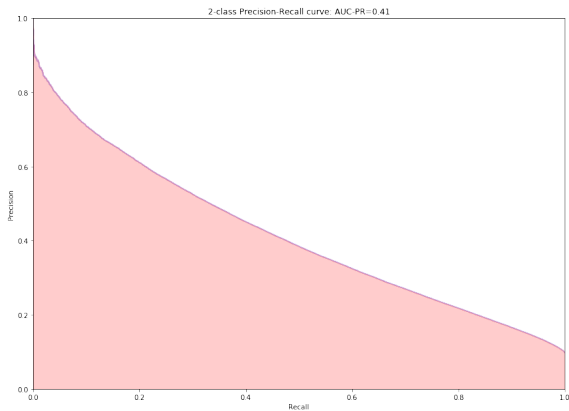


Figure: Precision-Recall Curve for \widehat{RFC}_{n_0} ; AUC-PR = 0.41.

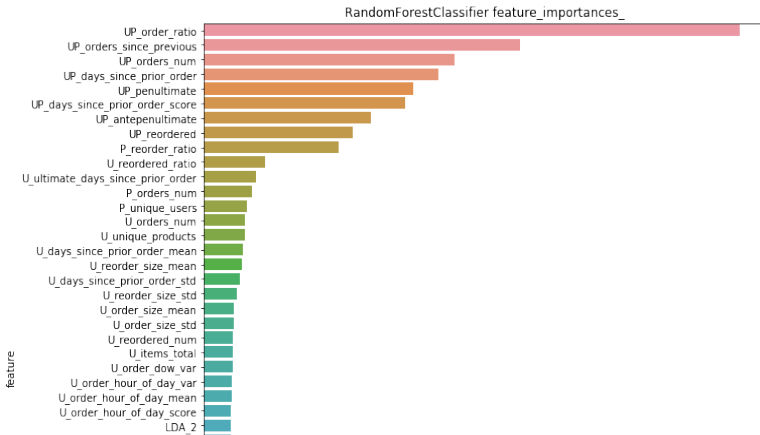


Figure: The RandomForestClassifier uses an information criterion to determine variable importances. There's still gold to mine in UP !

TopN Variants

$$\text{TopN} : [0, 1]^n \rightarrow \{0, 1\}^n$$

$$\hat{y}^* \mapsto \hat{y}$$

- TopN maps make binary predictions from probability rankings

$\text{TopN}_{\text{threshold}}$ chooses a classification threshold p_0 ;
best for optimizing threshold metrics (F1-score);
not best for product applications

TopN_u uses user's mean basket size as N ;
set basket size by user reduces its variance;
higher precision versions to autopopulate carts?

TopN_N uses a constant value N for each user;
worse metrics but best for displaying on
fixed-width web page (zero basket size variance)

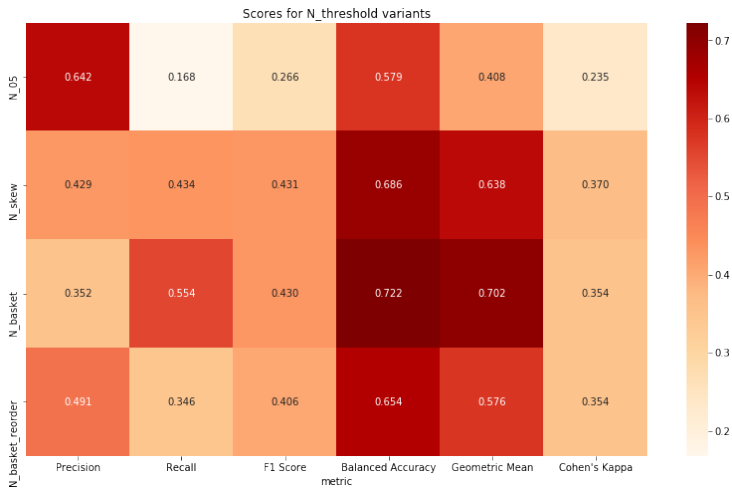


Figure: Scores for TopN_{threshold} variants.

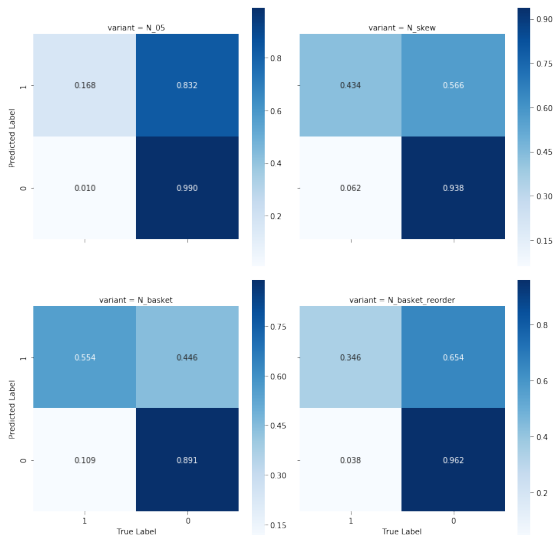


Figure: Normalized confusion matrices for TopN_{threshold} variants.

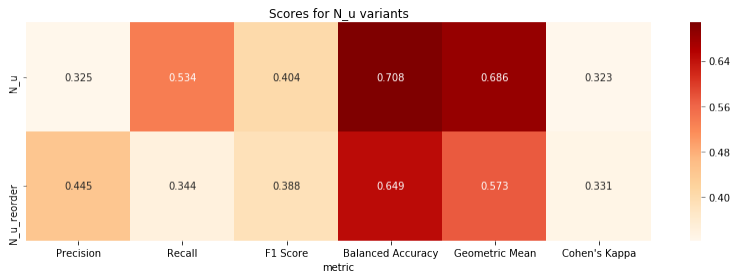


Figure: Scores for TopN_u variants.

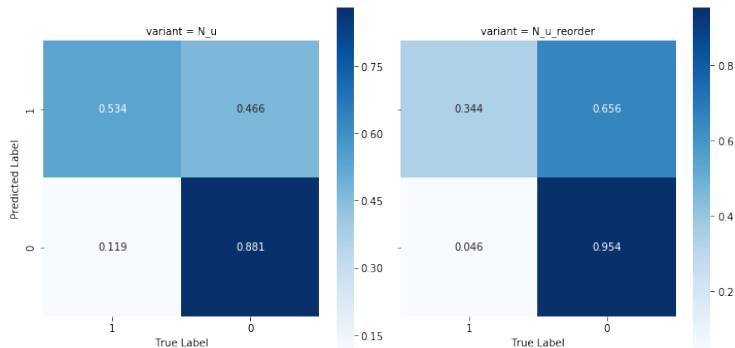


Figure: Normalized confusion matrices for TopN_u variants.

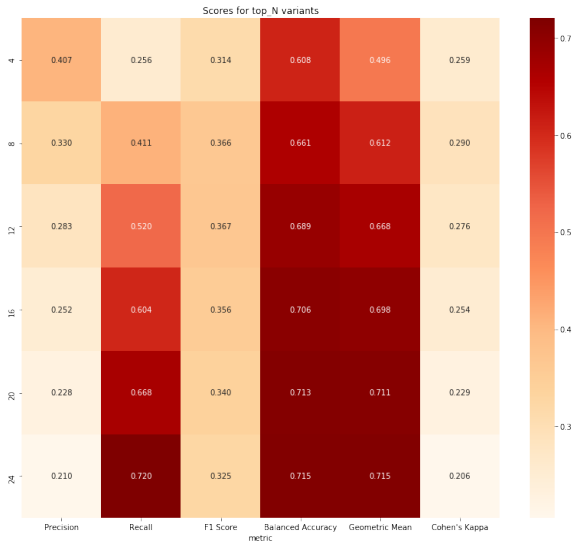


Figure: Scores for TopN_N variants.

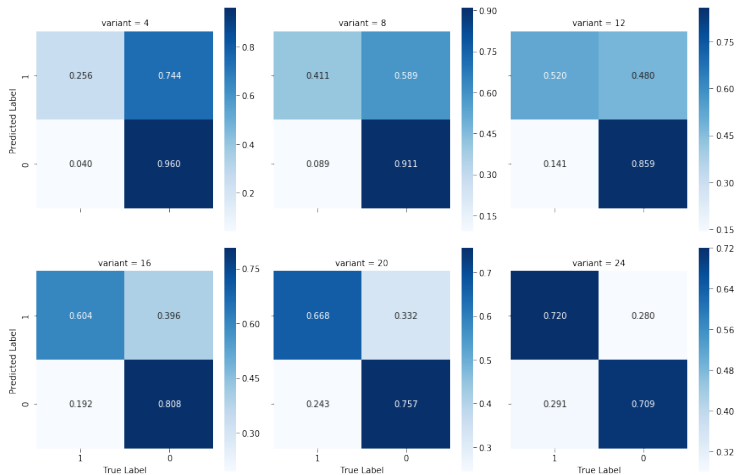


Figure: Normalized confusion matrices for TopN_N.

Prediction Explorer

Given a user and a model variant, visualize the model prediction, true order, and basket history, with colorings for

- True Positives: Ordered and Predicted
- False Positives: Predicted but not Ordered
- False Negatives: Ordered but not Predicted
- True Negatives: neither Ordered nor Predicted

A model with greater **precision** has fewer False Positives;

A model with greater **recall** has fewer False Negatives.

Note: 'add_to_cart_order' is not meaningful for the top rows of predictions and true orders.

add_to_cart_order	1	2	3	4	5	6	7	8
order_number								
prediction	Hampshire 100% Natural Sour Cream	Cut & Peeled Baby Carrots	Bistro Bowl Chicken Caesar Salad	Salisbury Steak with macaroni and cheese Salisbury Steak with macaroni and cheese	Multi-Grain Club Crackers	Sliced Sourdough Bread	Organic Fat-Free Milk	Blueberry on the Bottom Nonfat Greek Yogurt
true	Chicken Thighs	Cut & Peeled Baby Carrots	Bistro Bowl Chicken Caesar Salad	Deluxe Plain Bagels	Ritz Crackers	Sliced Sourdough Bread	Organic Fat-Free Milk	French Onion Dip
13	Strawberry on the Bottom Nonfat Greek Yogurt	Blueberry on the Bottom Nonfat Greek Yogurt	Non Fat Black Cherry on the Bottom Greek Yogurt	Colgate Total Whitening Toothpaste	Cut & Peeled Baby Carrots	Sliced Sourdough Bread	Poppycock Cashew Lovers	Butter Toffee Peanuts
12	Cut & Peeled Baby Carrots	French Onion Dip	Sliced Sourdough Bread	Original Cream Cheese	Deluxe Plain Bagels	Ultra Plush® 3 Ply Double Toilet Paper Rolls	Select-a-Size Rolls Paper Towels Tissue	Coke
11	Roma Tomato	French Bread	Organic Navel Orange	Whole Grains Oatnut Bread	Hampshire 100% Natural Sour Cream	Broccoli Crown	Cooking Beef Stock	Horseradish
10	Spiced Rum	French Onion Dip	Bistro Bowl Chicken Caesar Salad	Complete Clean Power Toilet Bowl Cleaner Value Pack	Assorted Chocolate Miniatures Chocolate Candy Bars	Multi-Grain Club Crackers	Cut & Peeled Baby Carrots	Classic Mix Variety
9	Coke	Spiced Rum	Sliced Sourdough Bread	Cut & Peeled Baby Carrots	French Onion Dip	Rigatoni Pasta	Crackers	Chicken Thighs
8	Spiced Rum	Butter	Coke	Original Citrus Sparkling Flavored Soda	Meatloaf and Mashed Potatoes	Salisbury Steak with macaroni and cheese Salisbury Steak with macaroni and cheese	360 Dusters Refills Unscented	French Onion Dip

Figure: user_id == 125; model == ('N_threshold', 'N_basket')

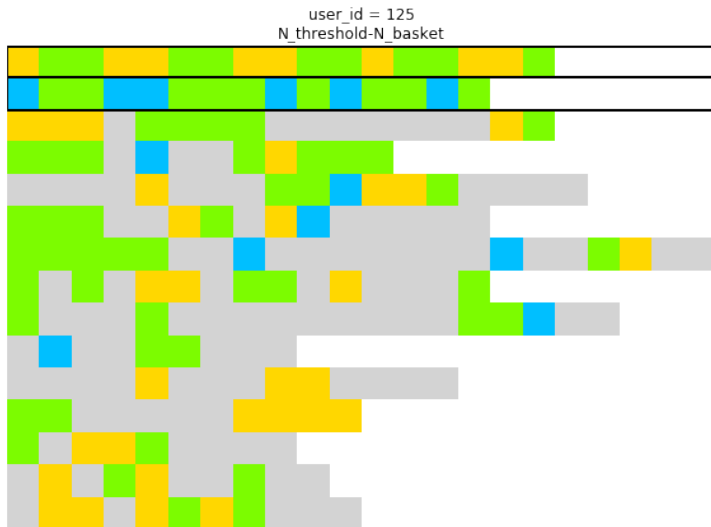


Figure: “Zoom out” by ignoring product names.

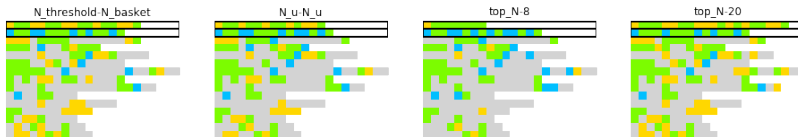


Figure: “Zoom out” further by displaying predictions for `user_id == 125` using a few model choices. By fixing a user, we can inspect the behavior of different models.

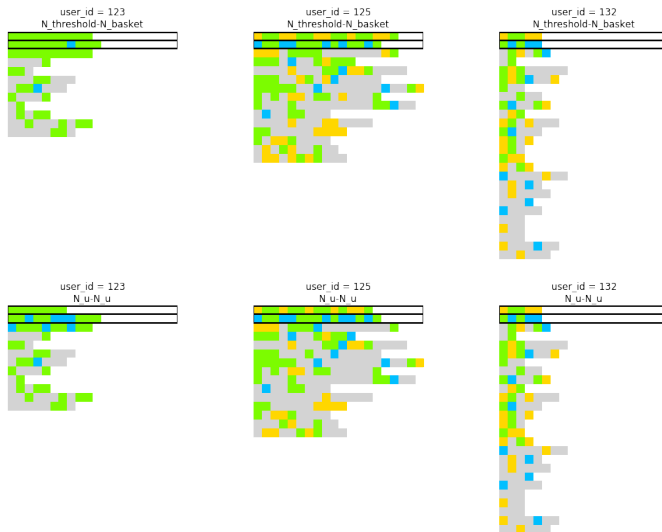


Figure: Zoom out on Figure 16 to include more users.

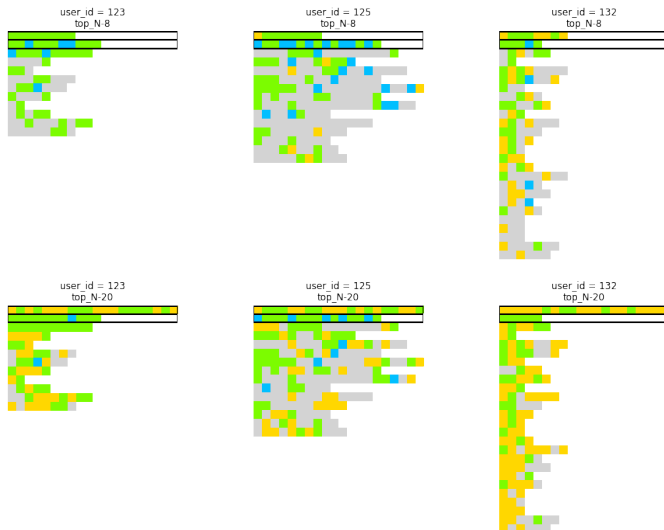


Figure: Zoom out on Figure 16 to include more users.



David M. Blei, Andrew Y. Ng, and Michael I. Jordan.

Latent Dirichlet Allocation.

Journal of Machine Learning Research, 3(Jan):993–1022, 2003.



Trevor Hastie, Robert Tibshirani, and Jerome Friedman.

The Elements of Statistical Learning.

Springer Series in Statistics. Springer New York, New York, NY, 2009.



Guimei Liu, Tam T. Nguyen, Gang Zhao, Wei Zha, Jianbo

Yang, Jianneng Cao, Min Wu, Peilin Zhao, and Wei Chen.

Repeat Buyer Prediction for E-Commerce.

In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 155–164, San Francisco, California, USA, 2016. ACM Press.