BUILDING A RECOMMENDATION ENGINE USING DIVERSE FEATURES

Divyanshu Vats
Sailthru

SPARK SUMMIT EAST
DATA SCIENCE AND ENGINEERING AT SCALE
FEBRUARY 16-18, 2016 NEW YORK CITY

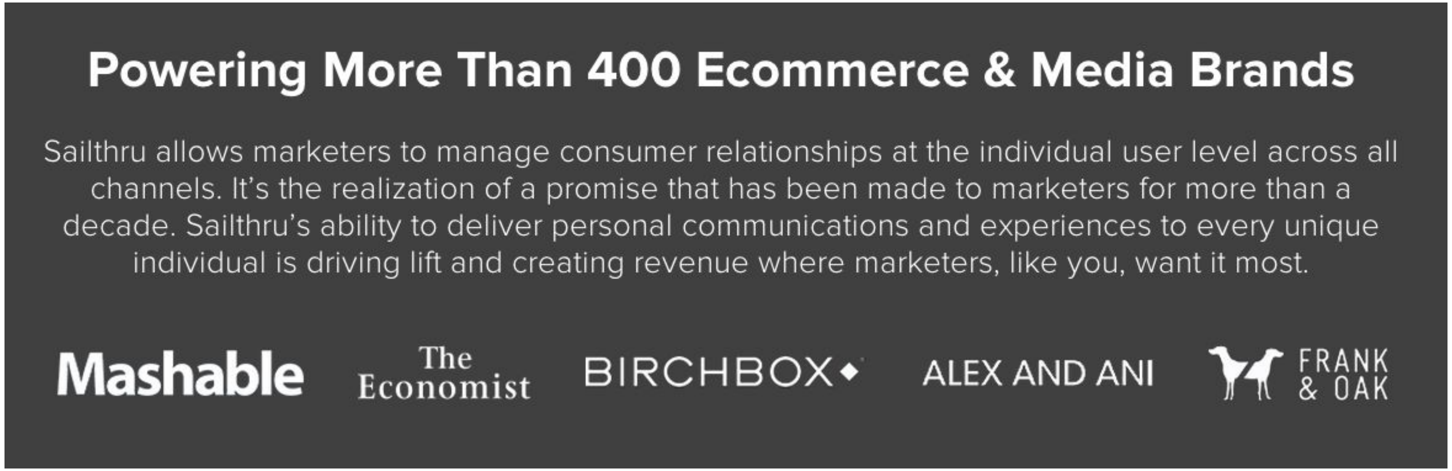
Joint work with

Max Sperlich
(Sailthru)

David Glueck
(Sailthru)

Alex Gaudio
(Alluvium)

Jeremy Stanley
(Instacart)



Powering More Than 400 Ecommerce & Media Brands

Sailthru allows marketers to manage consumer relationships at the individual user level across all channels. It's the realization of a promise that has been made to marketers for more than a decade. Sailthru's ability to deliver personal communications and experiences to every unique individual is driving lift and creating revenue where marketers, like you, want it most.

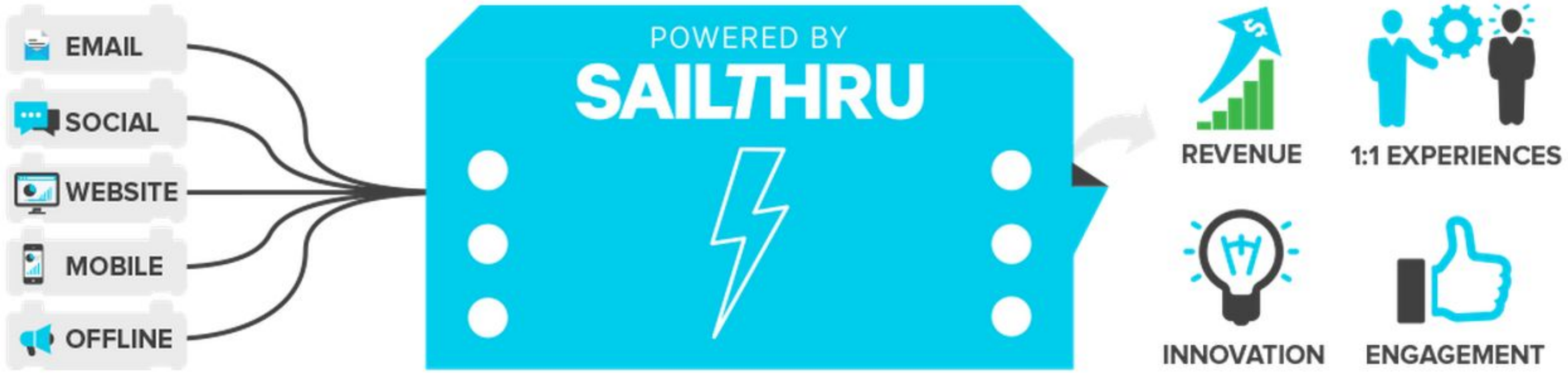
Mashable The Economist BIRCHBOX ALEX AND ANI FRANK & OAK

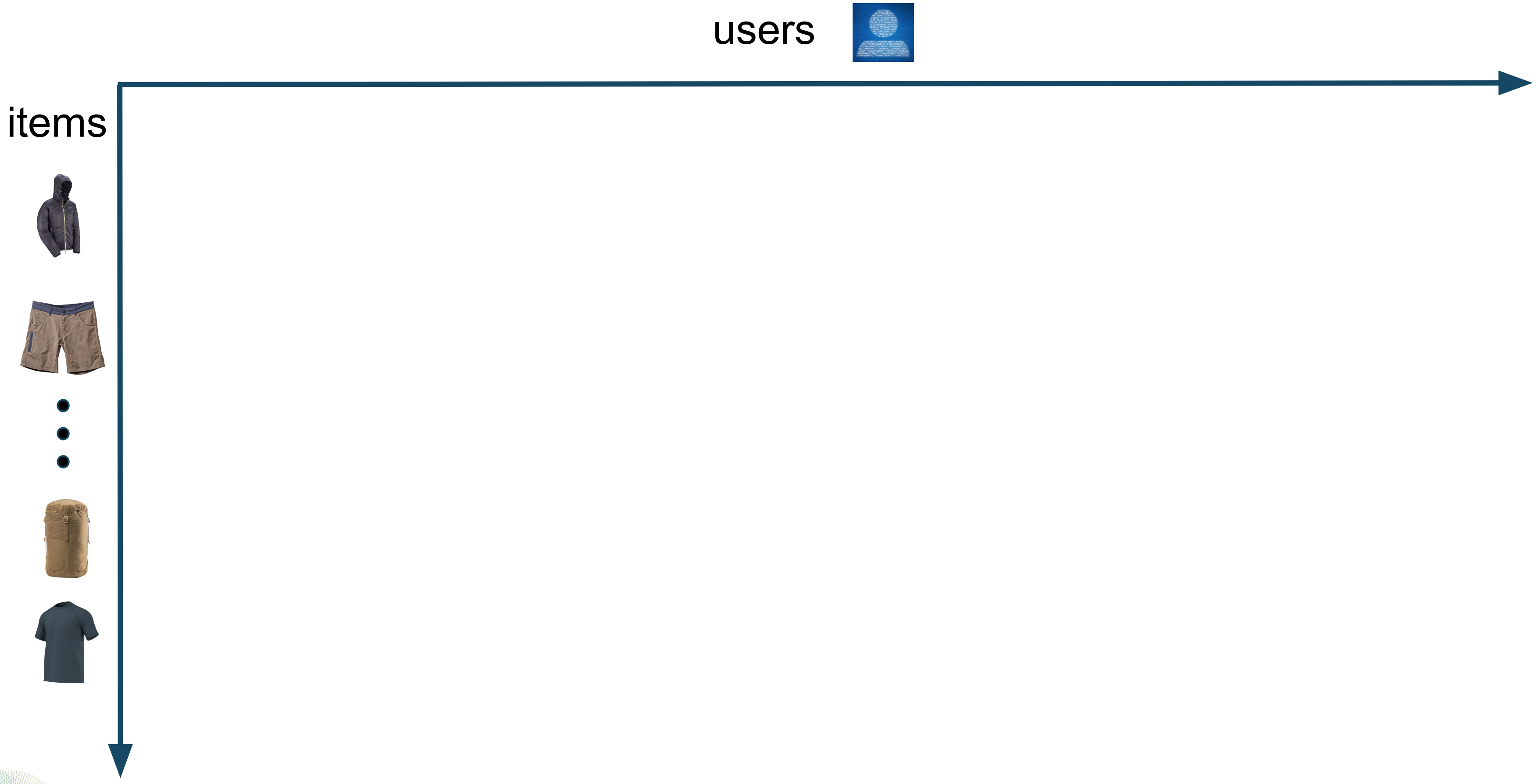
Customer Retention Cloud

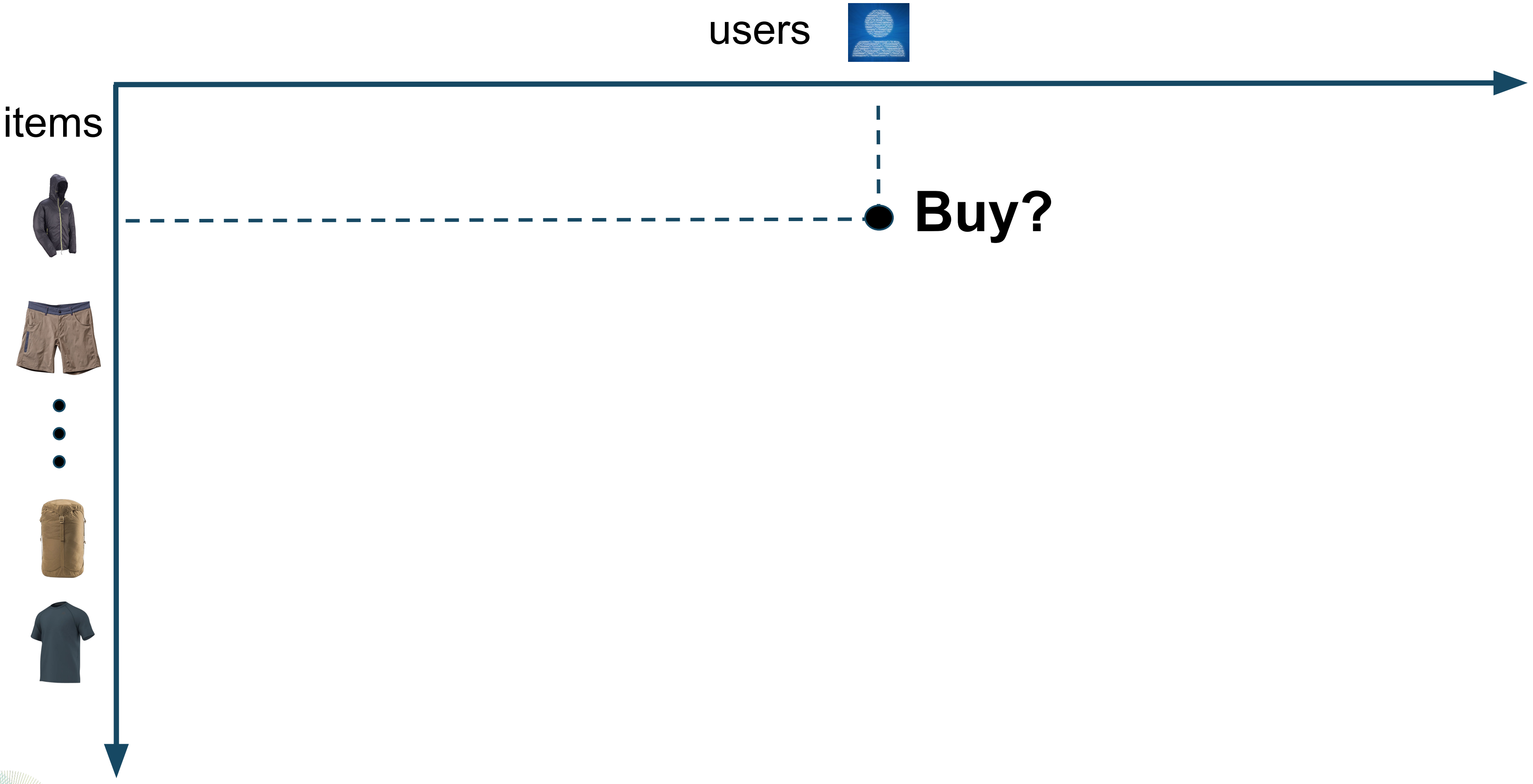
Powering More Than 400 Ecommerce & Media Brands

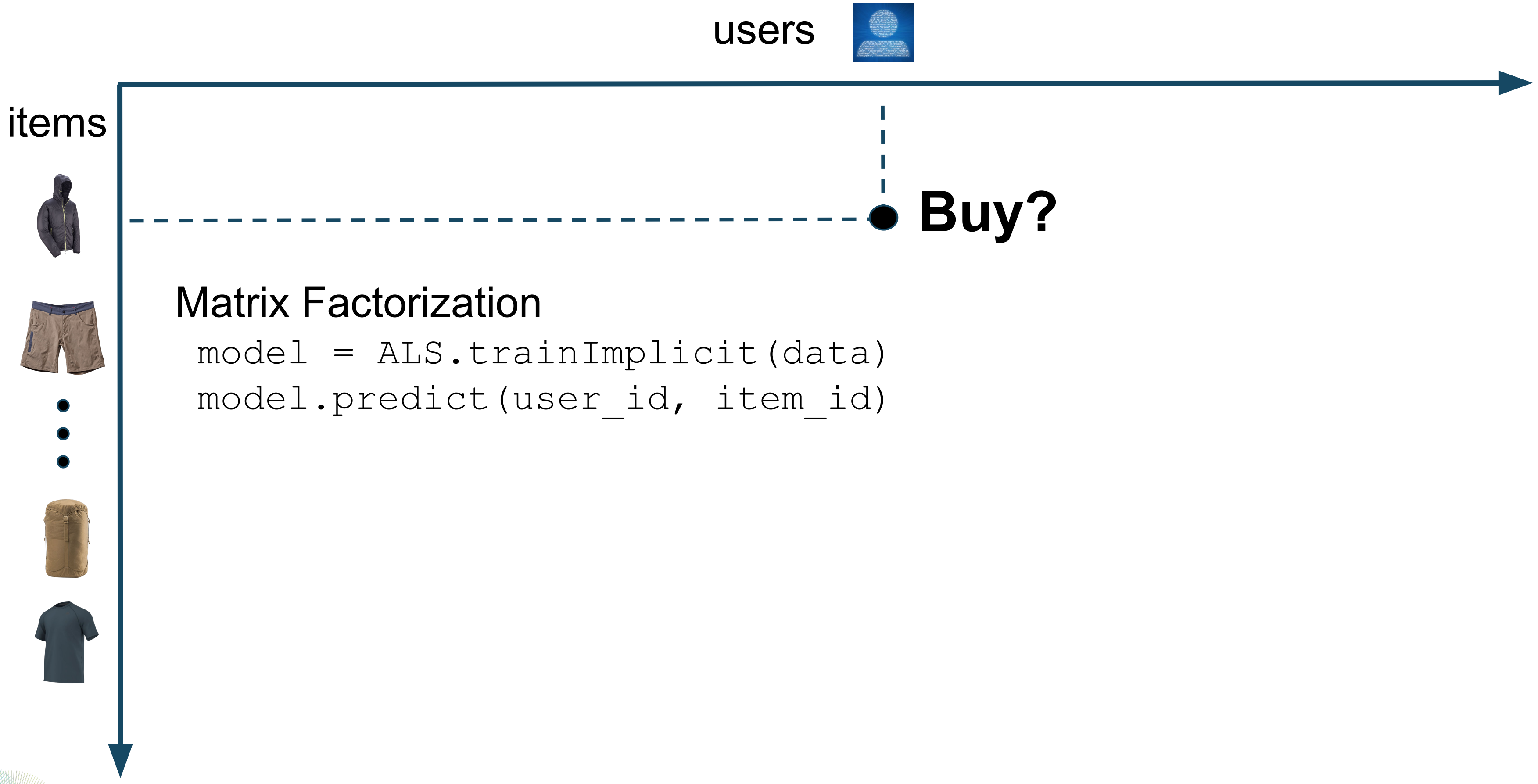
Sailthru allows marketers to manage consumer relationships at the individual user level across all channels. It's the realization of a promise that has been made to marketers for more than a decade. Sailthru's ability to deliver personal communications and experiences to every unique individual is driving lift and creating revenue where marketers, like you, want it most.

Mashable The Economist BIRCHBOX ALEX AND ANI FRANK & OAK



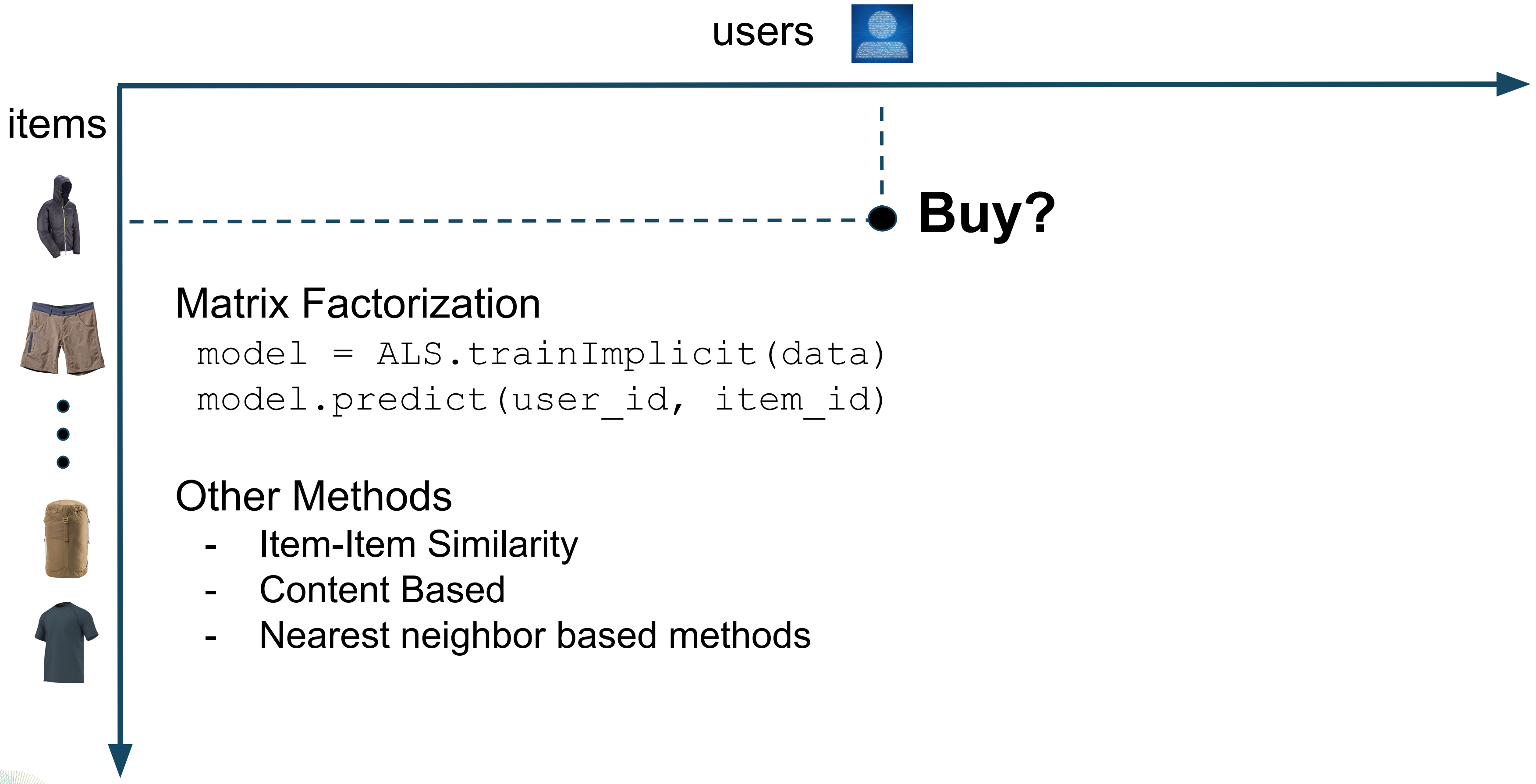

users
items


users
items
Buy?



Matrix Factorization

```
model = ALS.trainImplicit(data)
model.predict(user_id, item_id)
```

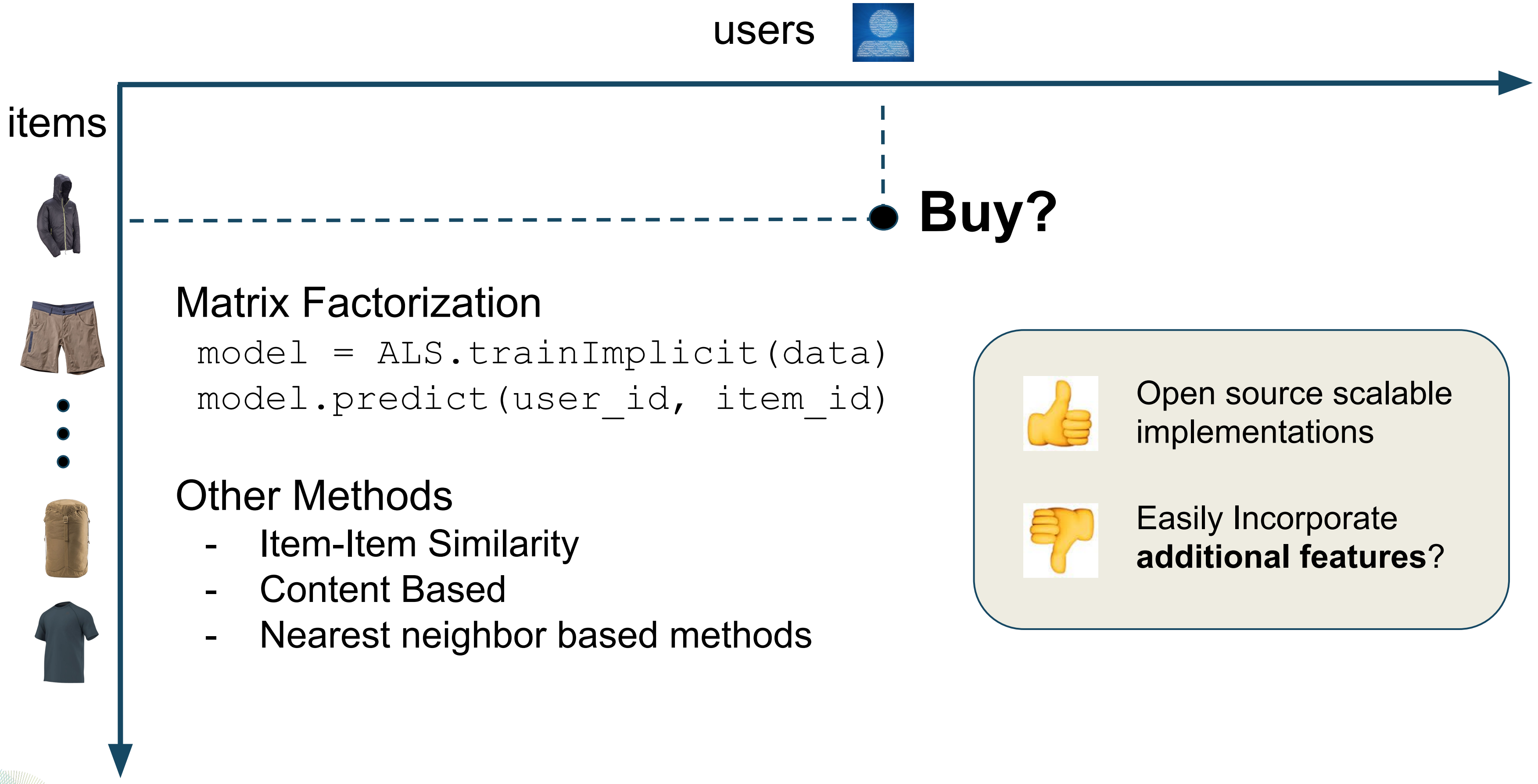


Matrix Factorization

```
model = ALS.trainImplicit(data)
model.predict(user_id, item_id)
```

Other Methods

- Item-Item Similarity
- Content Based
- Nearest neighbor based methods



Matrix Factorization

```
model = ALS.trainImplicit(data)
model.predict(user_id, item_id)
```

Other Methods

- Item-Item Similarity
- Content Based
- Nearest neighbor based methods

👍 Open source scalable implementations

👎 Easily Incorporate **additional features**?

Additional Features?

Additional Features?

user

location

# clicks

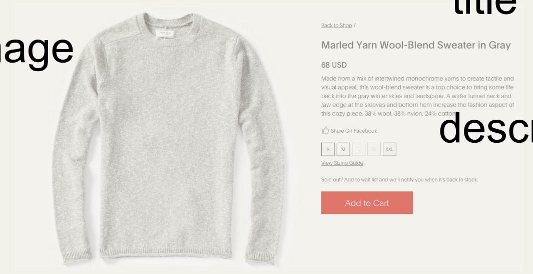
browser

item

title

image

description



Back to Shop /

Marled Yarn Wool-Blend Sweater in Gray

68 USD

Made from a mix of intertwined monochrome yarns to create tactile and visual appeal, this wool-blend sweater is a top choice to bring some life back into the gray winter skies and landscape. A wider tunnel neck and raw edge at the sleeves and bottom hem increase the fashion aspect of this cozy piece. 38% wool, 38% nylon, 24% cotton

Sold out? Add to wait list and we'll notify you when it's back in stock.

Additional Features?

user

location

# clicks

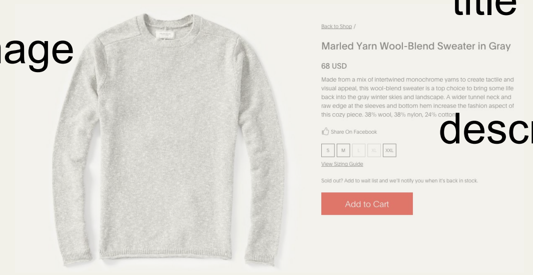
browser

item

image

title

description



Back to Shop /

Marled Yarn Wool-Blend Sweater in Gray

68 USD

Made from a mix of intertwined monochrome yarns to create tactile and visual appeal, this wool-blend sweater is a top choice to bring some life back into the gray winter skies and landscape. A wider tunnel neck and raw edge at the sleeves and bottom hem increase the fashion aspect of this cozy piece. 38% wool, 38% nylon, 24% cotton.

Share On Facebook

S M L XL XXL

View Sizing Guide

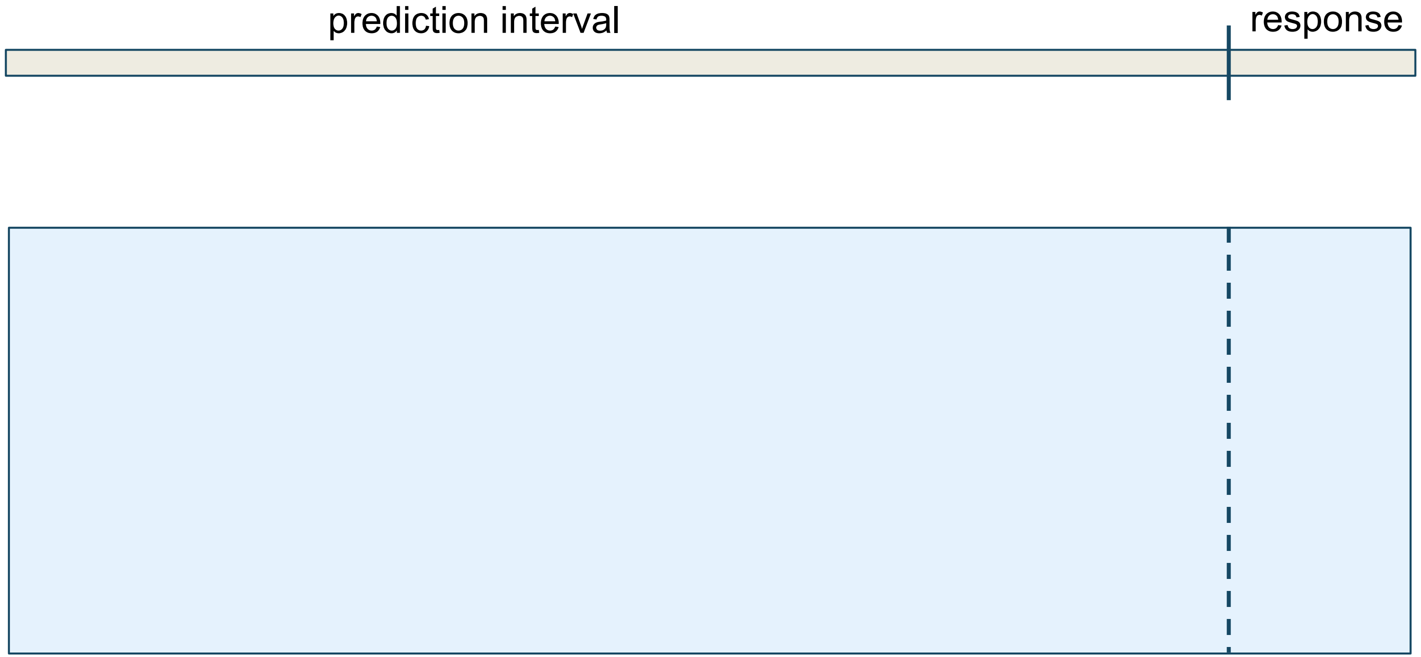
Sold out? Add to wait list and we'll notify you when it's back in stock.

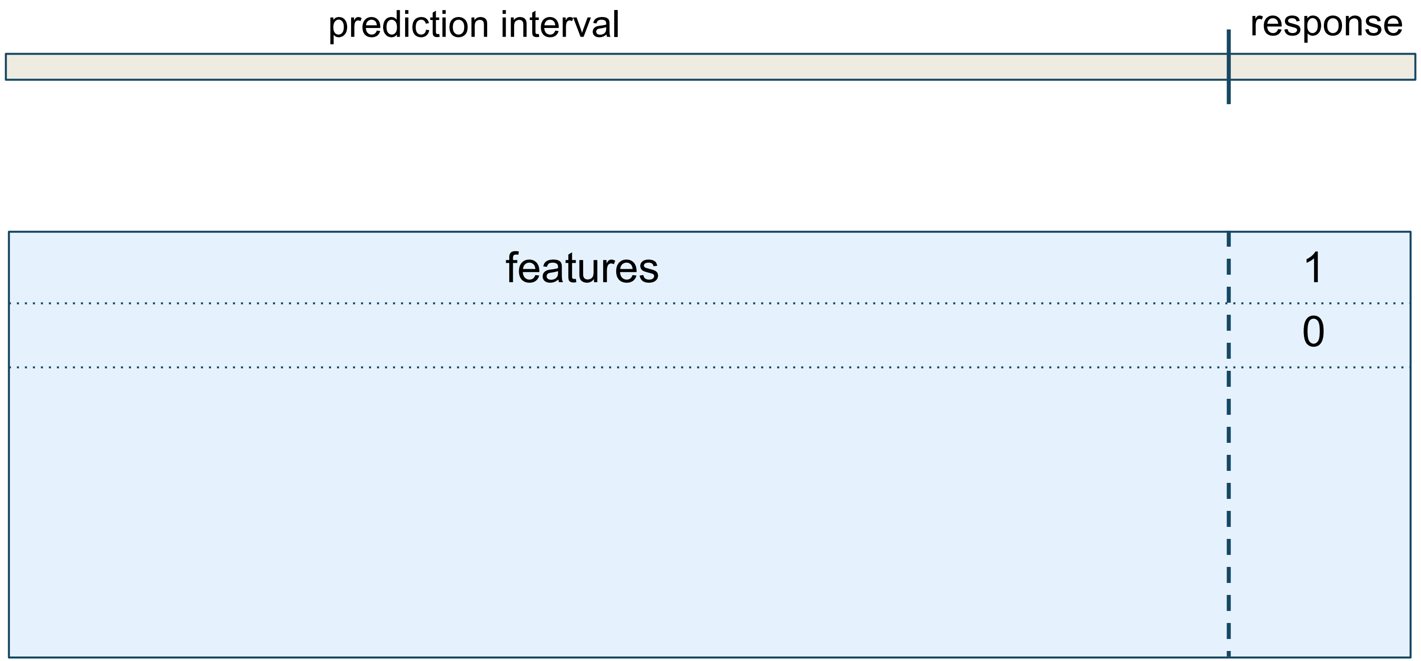
Add to Cart

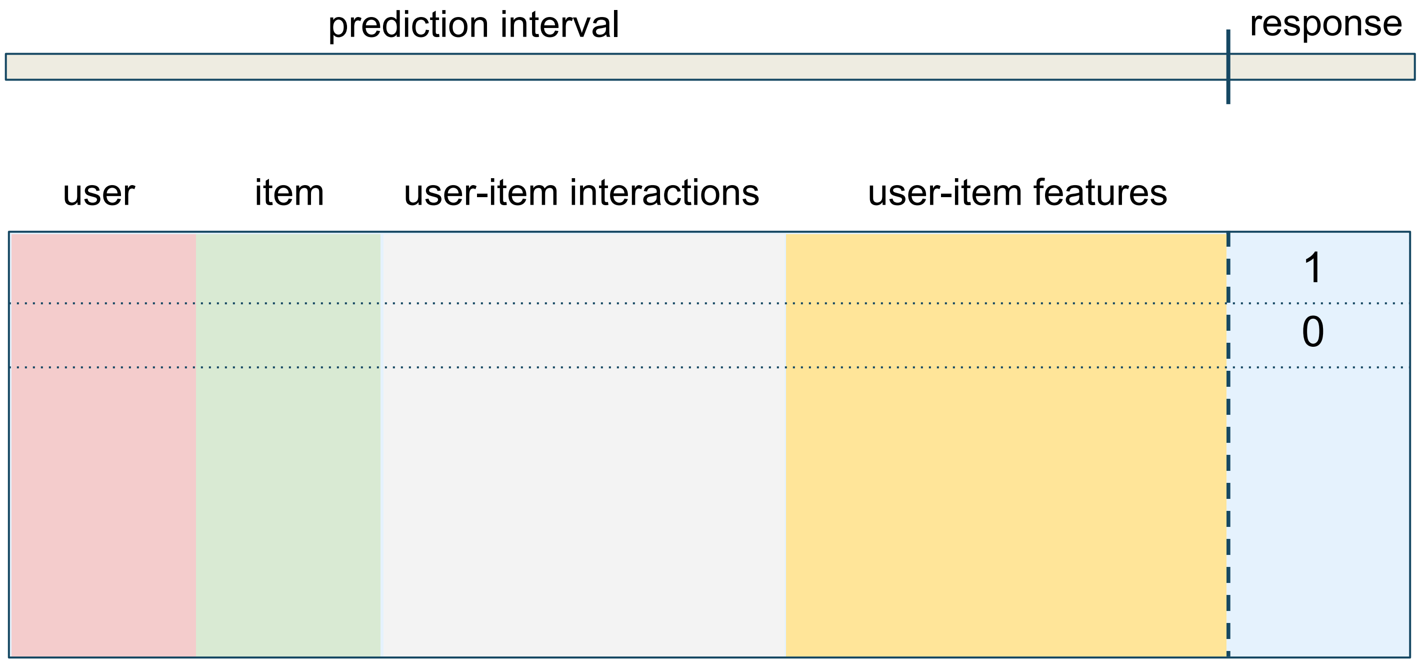
user-item

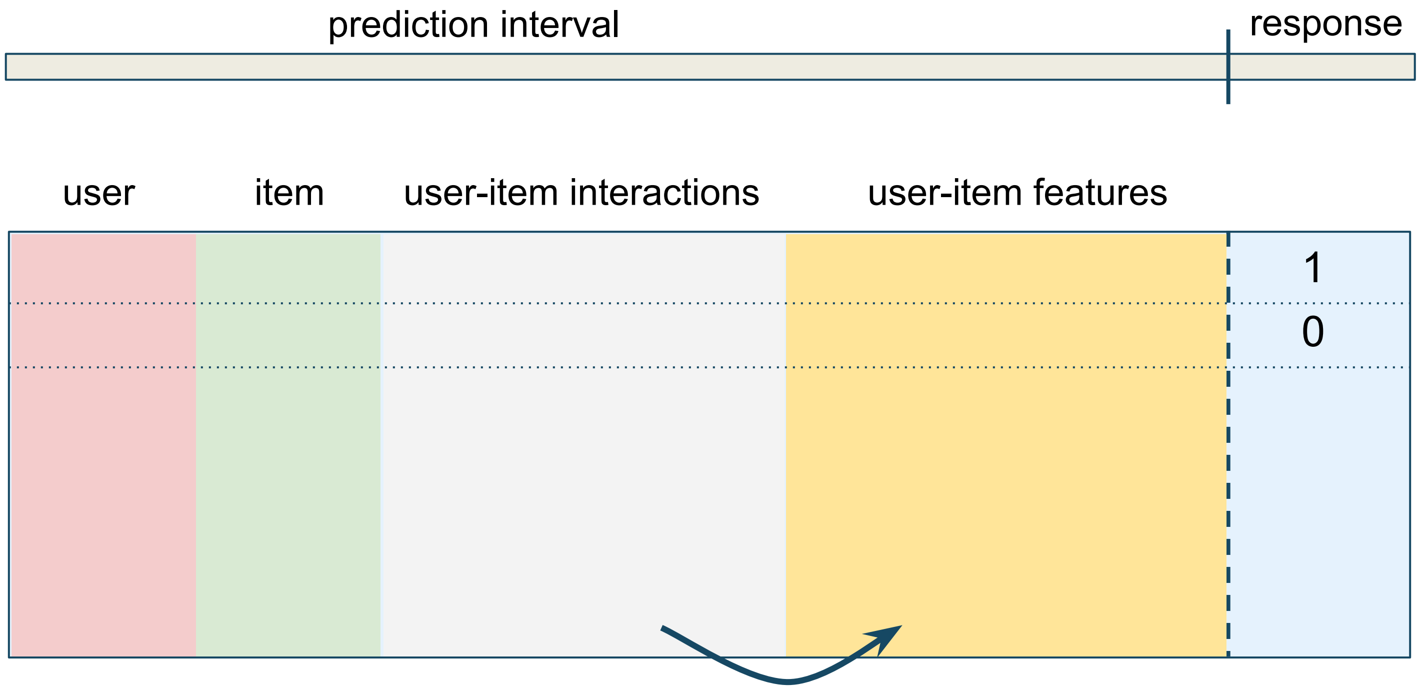
- purchases
- item views

Goal: Framework to **easily** integrate **user**, **item**, and **user-item** interactions for meaningful recommendations


prediction interval
response

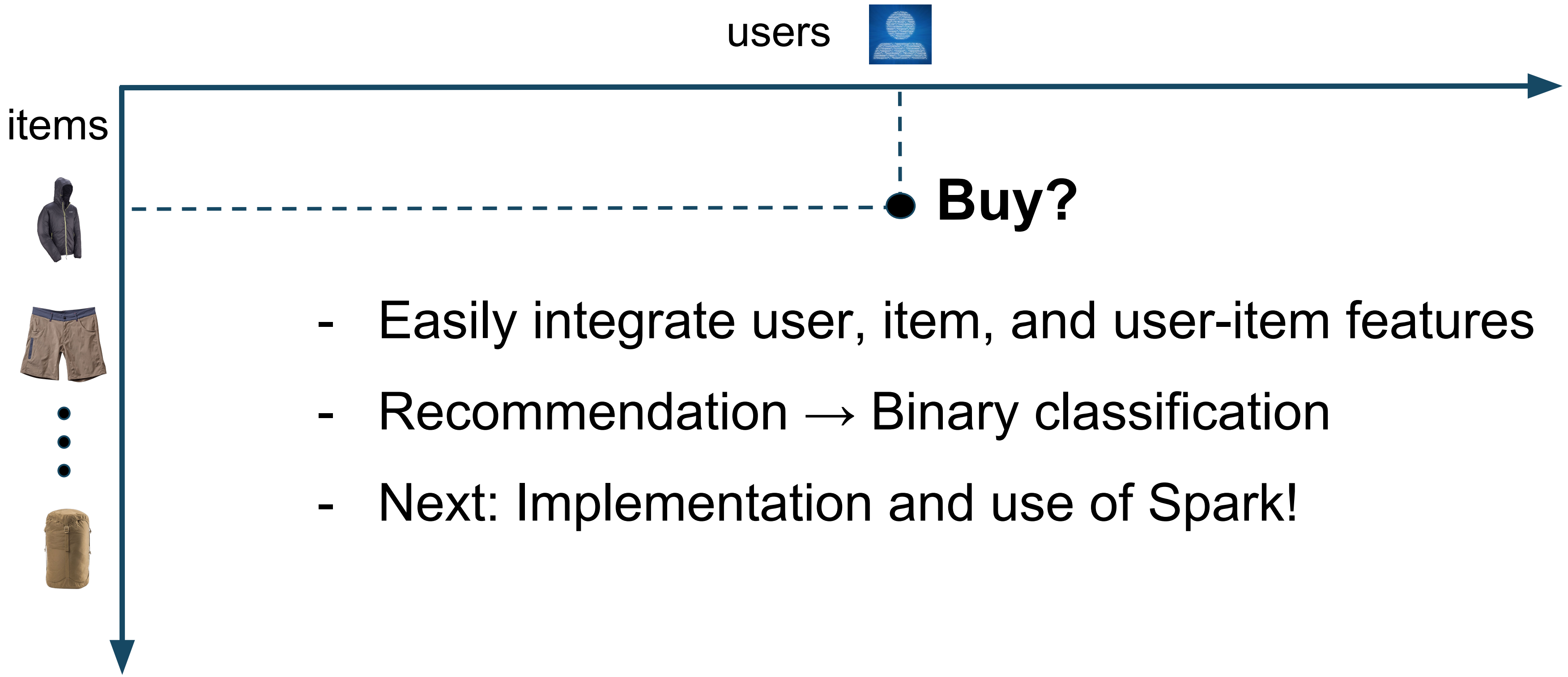

prediction interval
response
features
1
0


prediction interval
response
user
item
user-item interactions
user-item features
1
0



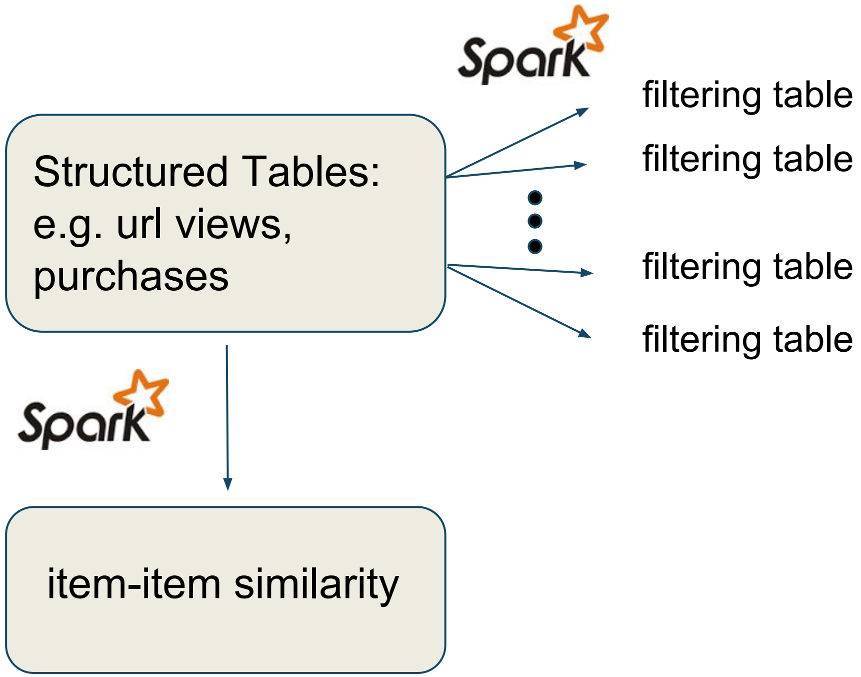
Can use a combination of collaborative filtering algorithms!

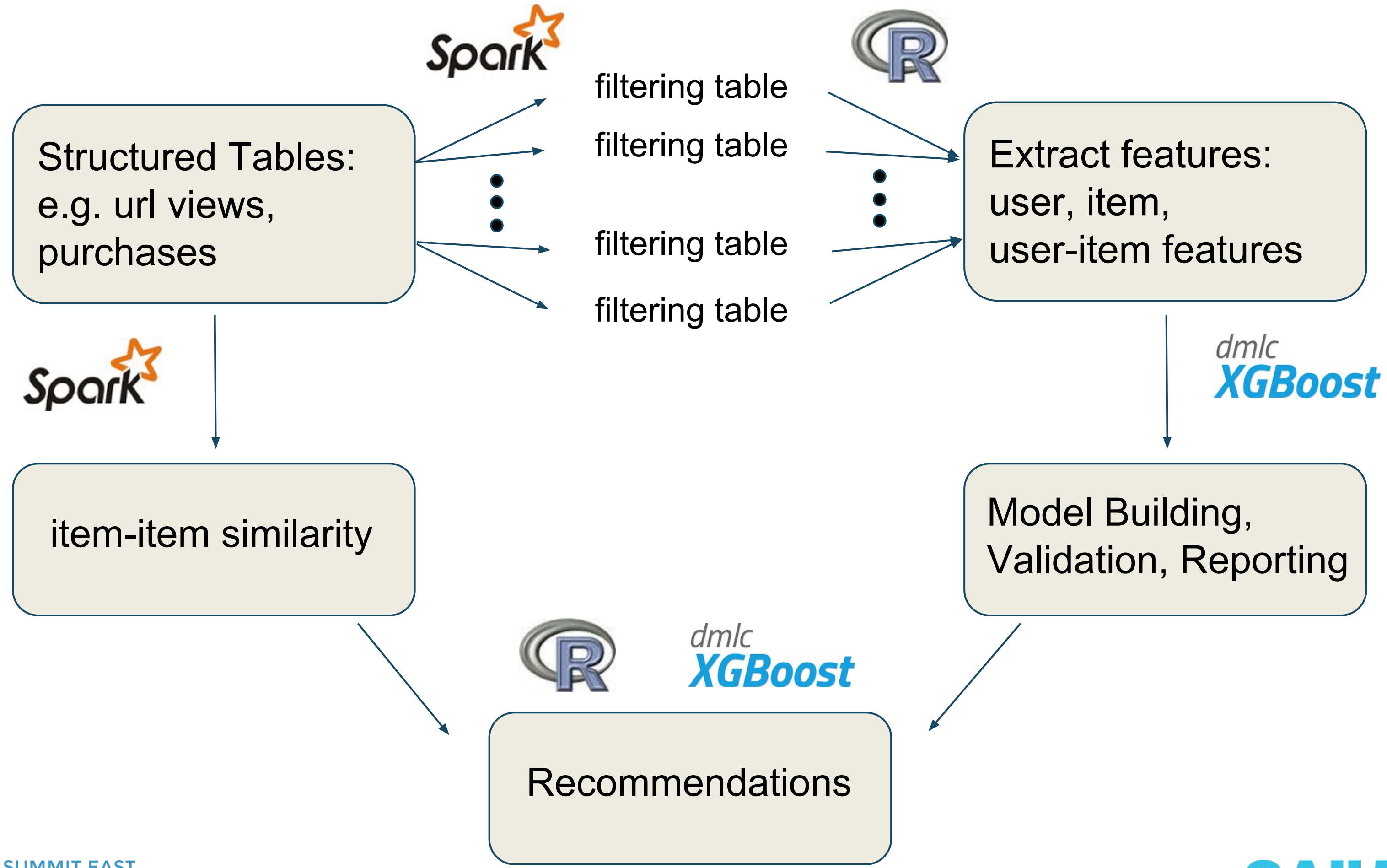
So far….



- Easily integrate user, item, and user-item features
- Recommendation → Binary classification
- Next: Implementation and use of Spark!

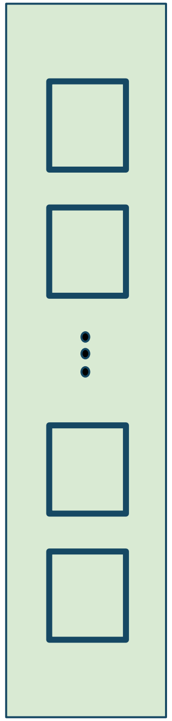
Structured Tables: e.g. url views, purchases


Spark
Structured Tables: e.g. url views, purchases
filtering table
filtering table
filtering table
filtering table
Spark
item-item similarity


Spark
Structured Tables: e.g. url views, purchases
filtering table
filtering table
filtering table
filtering table
Extract features: user, item, user-item features
dmlc XGBoost
Model Building, Validation, Reporting
Spark
item-item similarity
dmlc XGBoost
Recommendations

Know Thy Partitions!

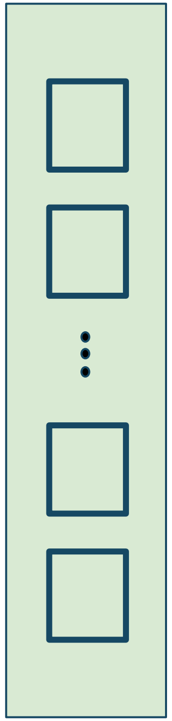
```
data.select('user_id', 'url')
     .groupBy('user_id')
     .count()
     .filter(count > 10)
```



Know Thy Partitions!

```
data.select('user_id', 'url')
     .groupBy('user_id')
     .count()
     .filter(count > 10)
```

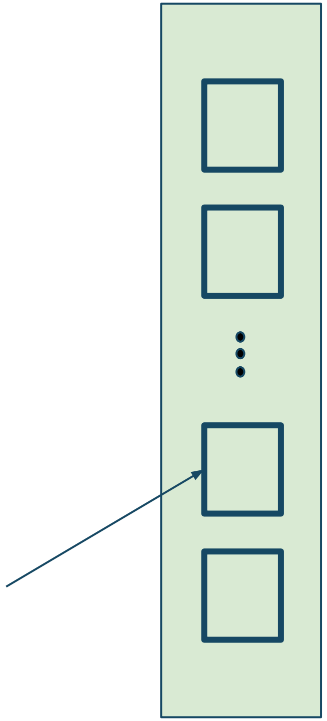
Inefficient!



Know Thy Partitions!

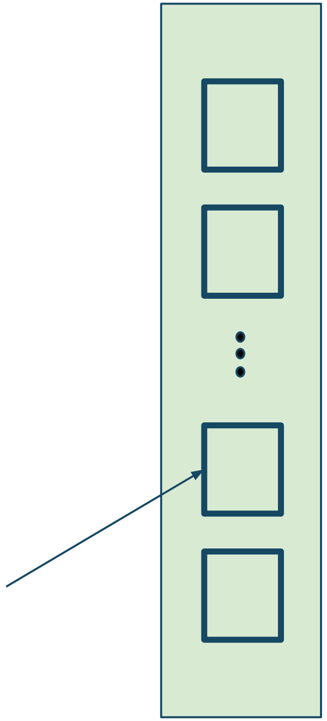
```
data.select('user_id', 'url')
     .groupBy('user_id')
     .count()
     .filter(count > 10)
```

partition by user_id



Know Thy Partitions!

```
data.select('user_id', 'url')
      .mapPartitions(filter_url)
```



Break up application into small chunks!

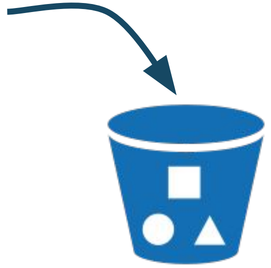
```
users = data.select('user_id', 'url')
            .mapPartitions(filter_url)
            .collect()




BigData.filter(lambda x: x.user_id in users)
```

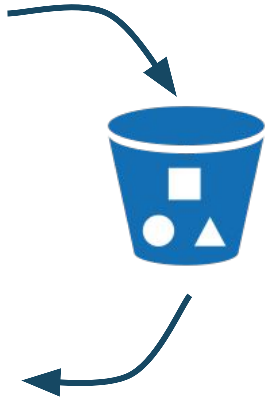
Break up application into small chunks!

```
users = data.select('user_id', 'url')
            .mapPartitions(filter_url)
            .collect()
```



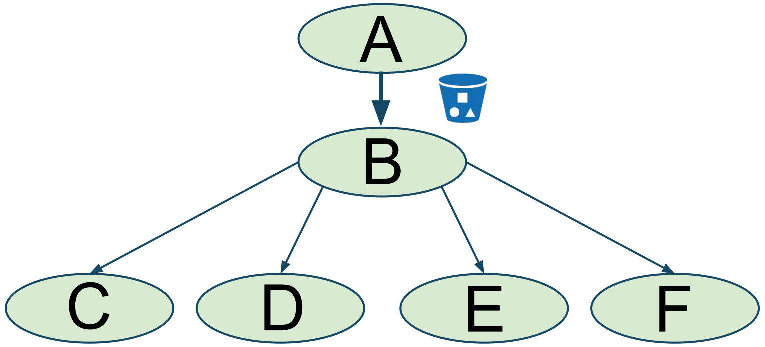
Break up application into small chunks!

```
users = data.select('user_id', 'url')
            .mapPartitions(filter_url)
            .collect()
```



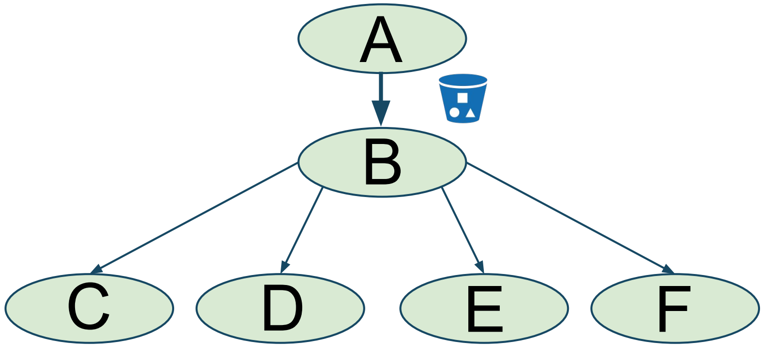
```
BigData.filter(lambda x: x.user_id in users)
```

Mesos + Scheduler + Docker + Spark



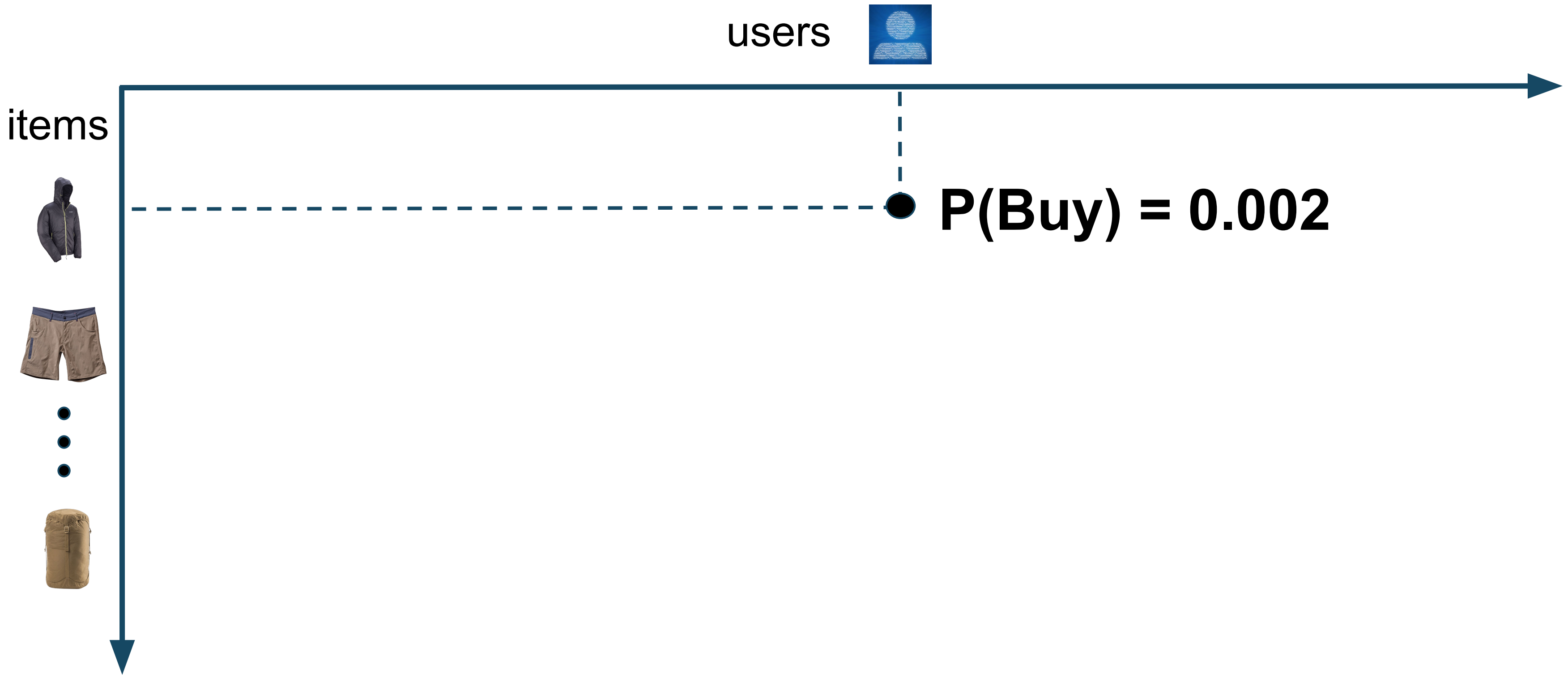
- Carefully define applications and state a dependency graph
- Manage graph using: github.com/sailthru/stolos

Mesos + Scheduler + Docker + Spark

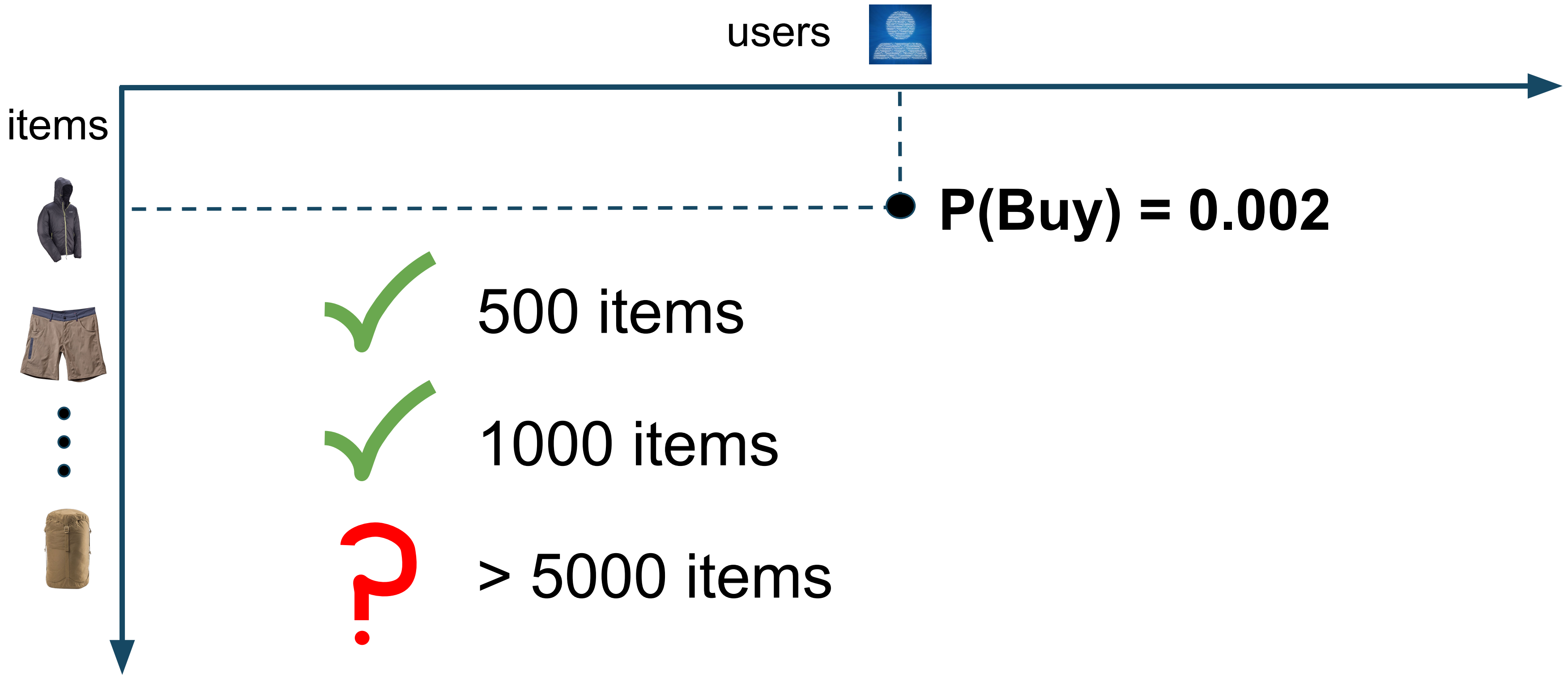

$$\begin{tikzcd} & & A \arrow[d] & \\ & & B \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] & \\ C & D & E & F \end{tikzcd}$$

- Carefully define applications and state a dependency graph
- Manage graph using: github.com/sailthru/stolos

Item-Item Similarities



Item-Item Similarities

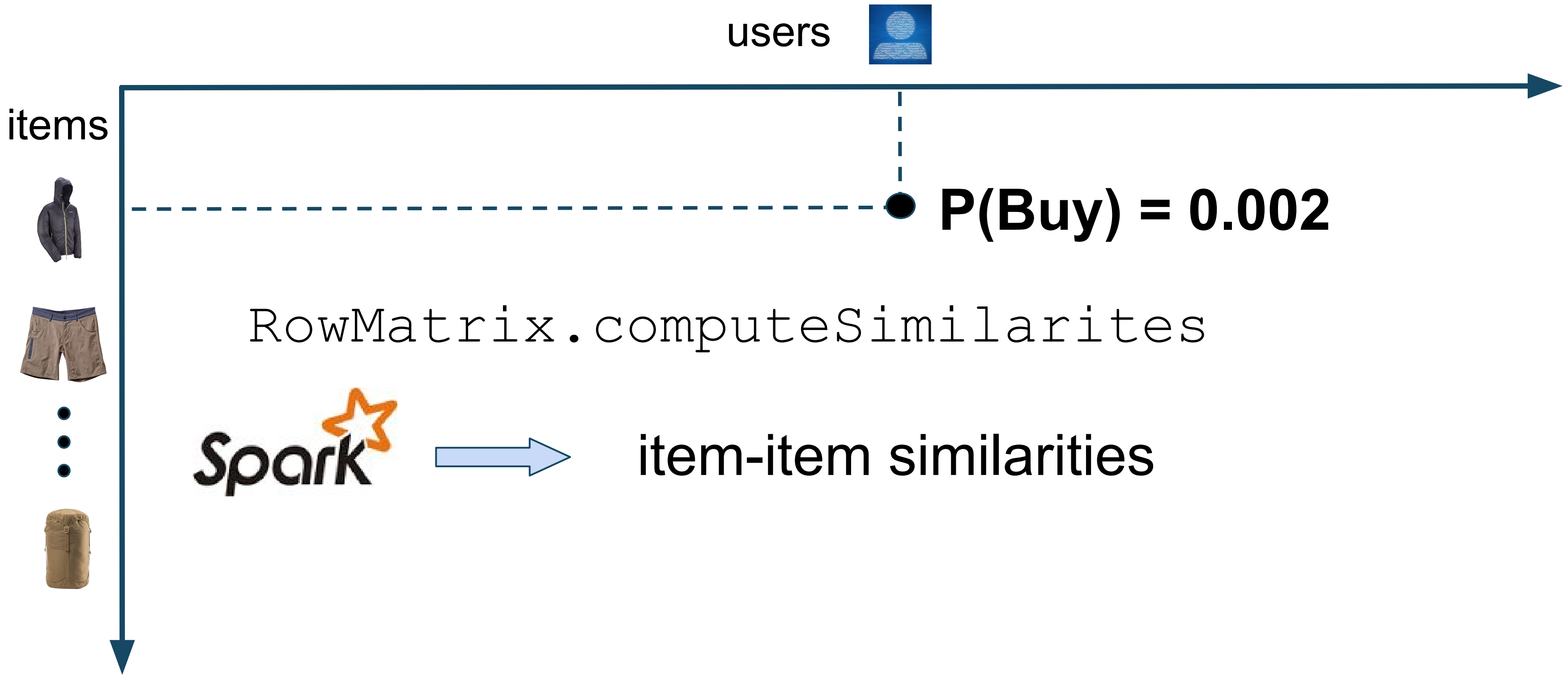


✓ 500 items

✓ 1000 items

? > 5000 items

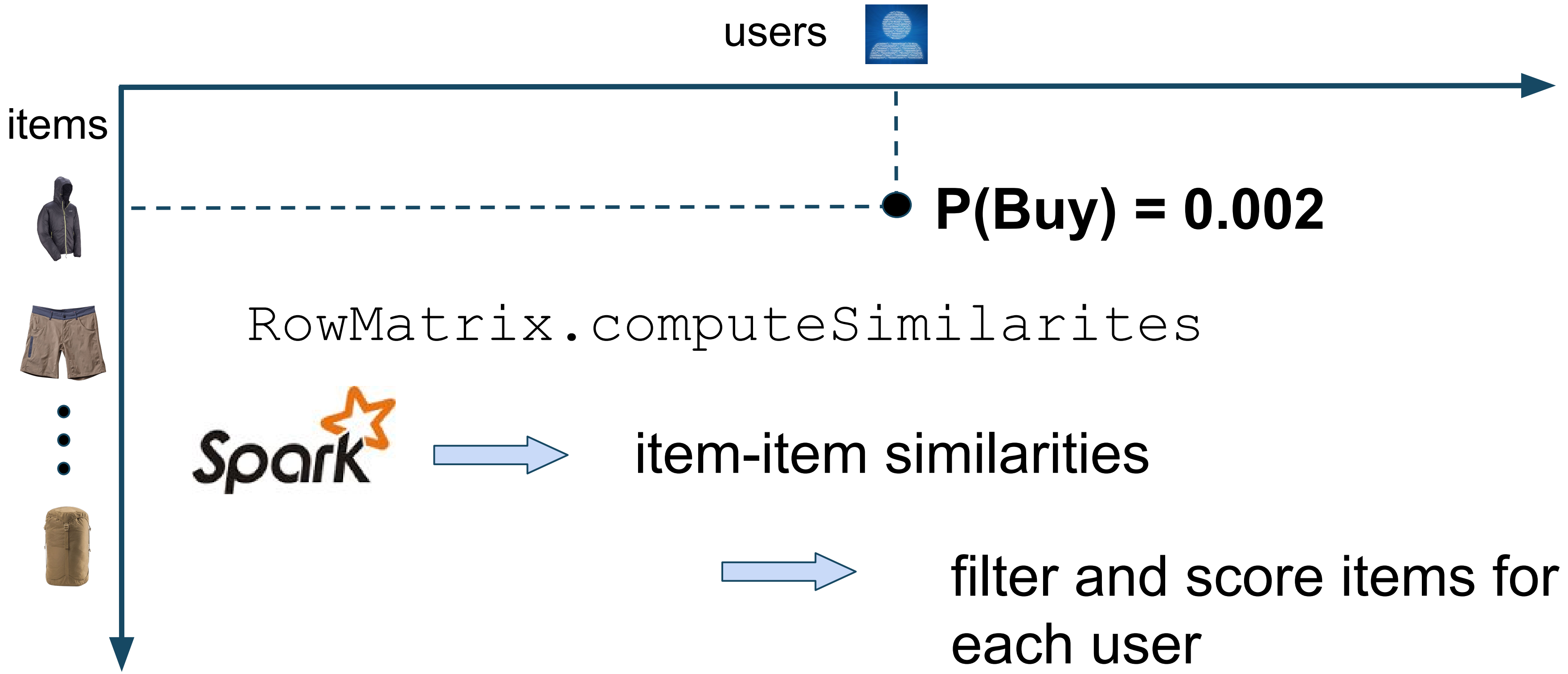
Item-Item Similarities



```
RowMatrix.computeSimilarites
```

Spark → item-item similarities

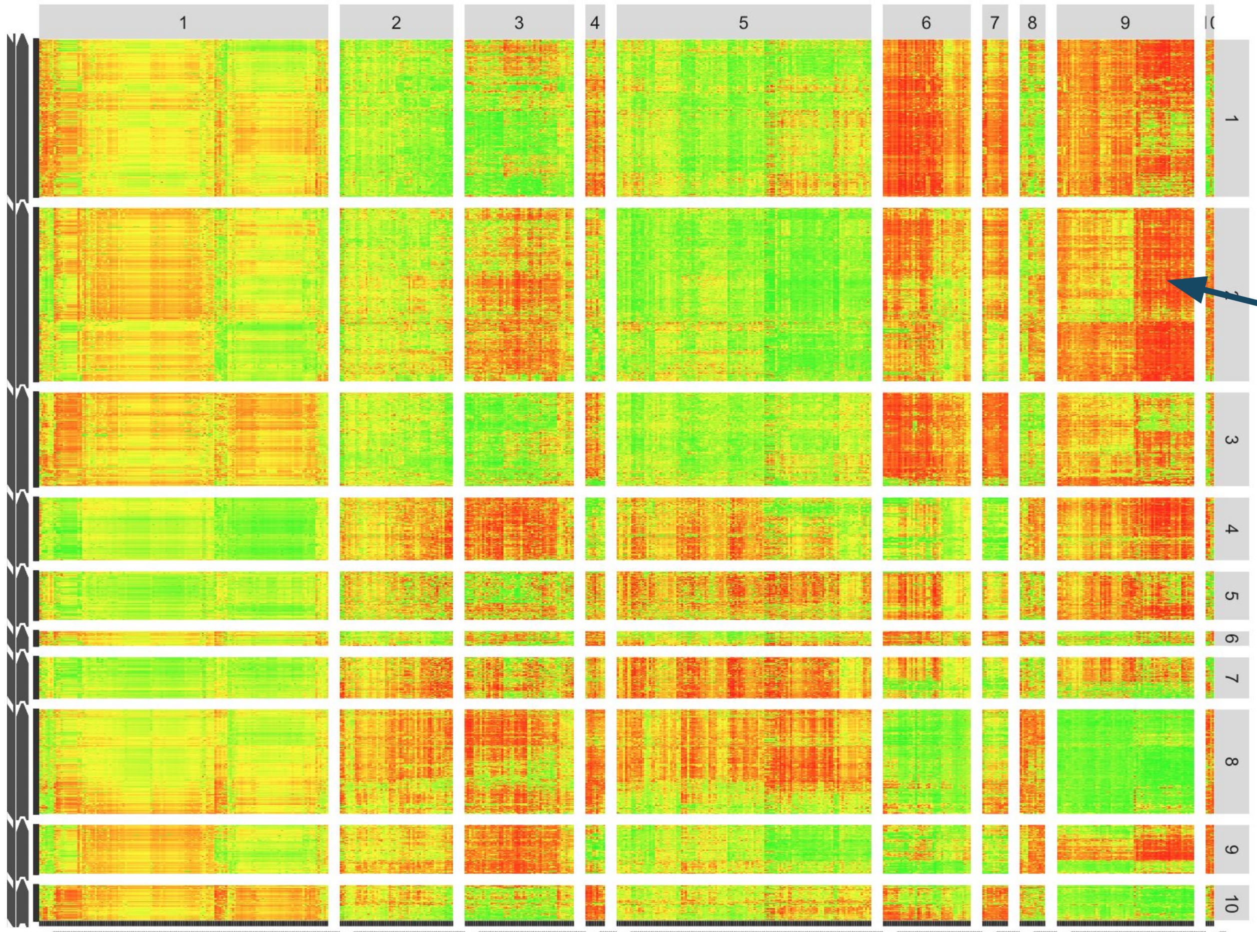
Item-Item Similarities



`RowMatrix.computeSimilarites`

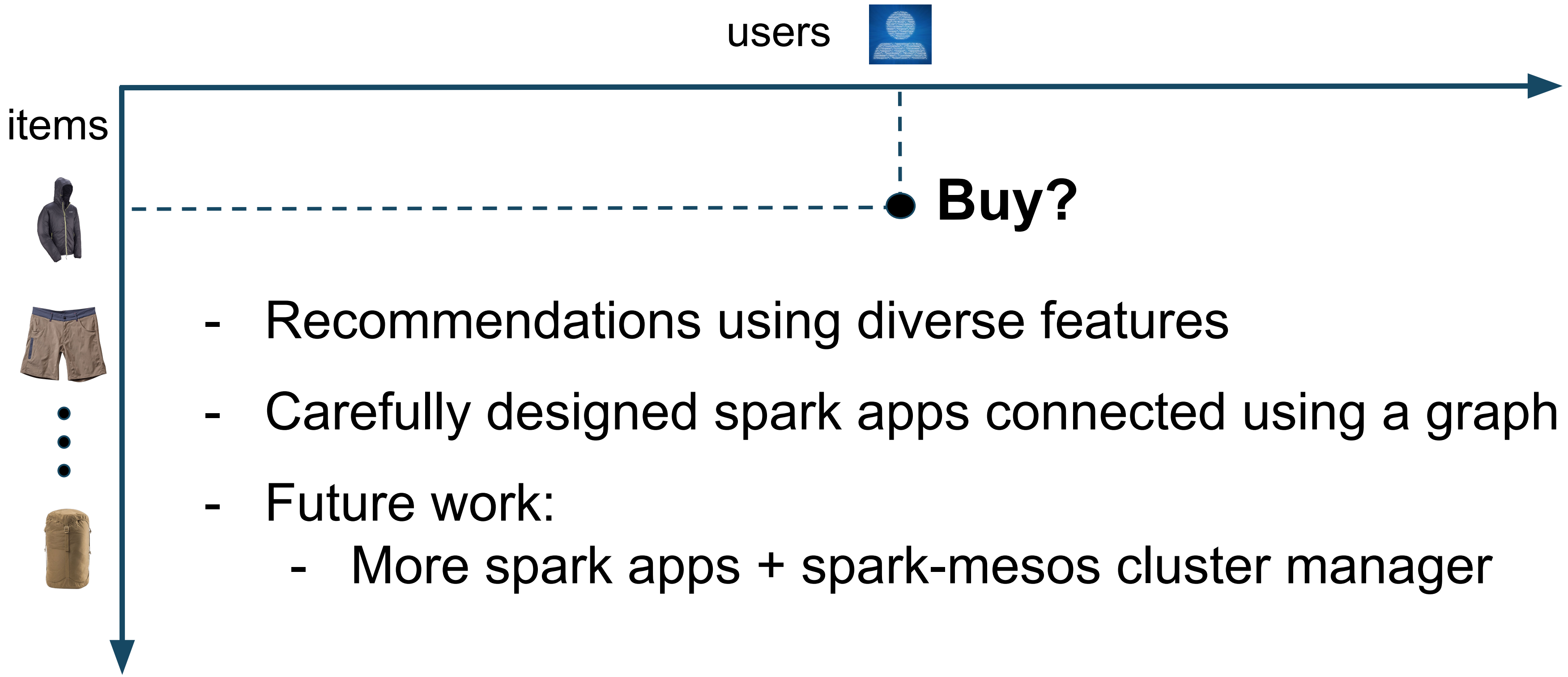
Spark → item-item similarities

→ filter and score items for each user



cluster of users and items

To Summarize..



- Recommendations using diverse features
- Carefully designed spark apps connected using a graph
- Future work:
  - More spark apps + spark-mesos cluster manager

THANK YOU.

email: dvats@sailthru.com