Interaction Based Feature Extraction

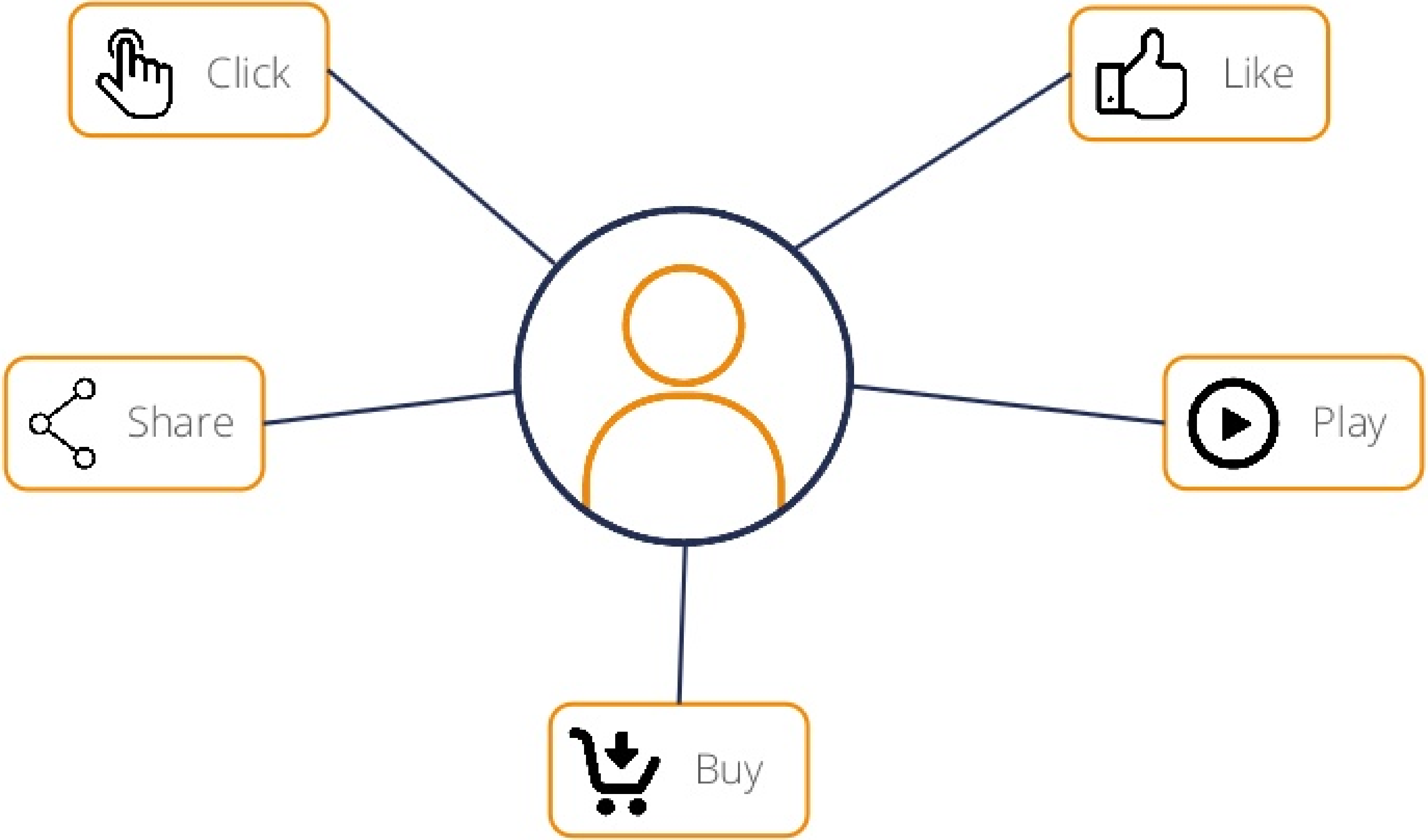
How to Convert User Activity into Valuable Features

Shlomi Babluki, Principal Data Scientist

October, 2018

SimilarWeb

User Activity



SimilarWeb

Provides Digital Insights for 190+ countries

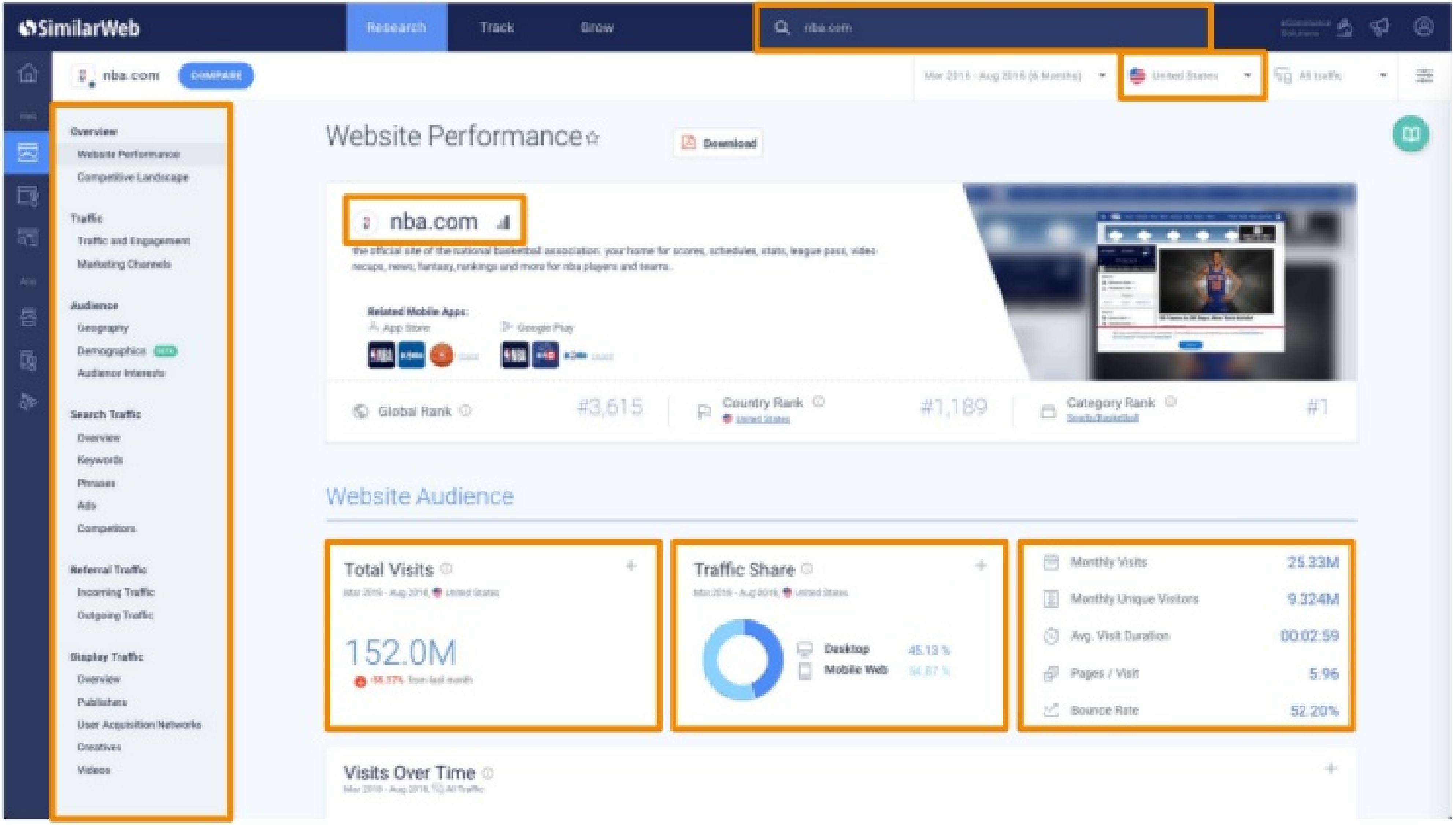
Every Website

- ✓ Traffic Metrics
- ✓ Traffic Sources
- ✓ Audience
- ✓ Industry
- ✓ Content

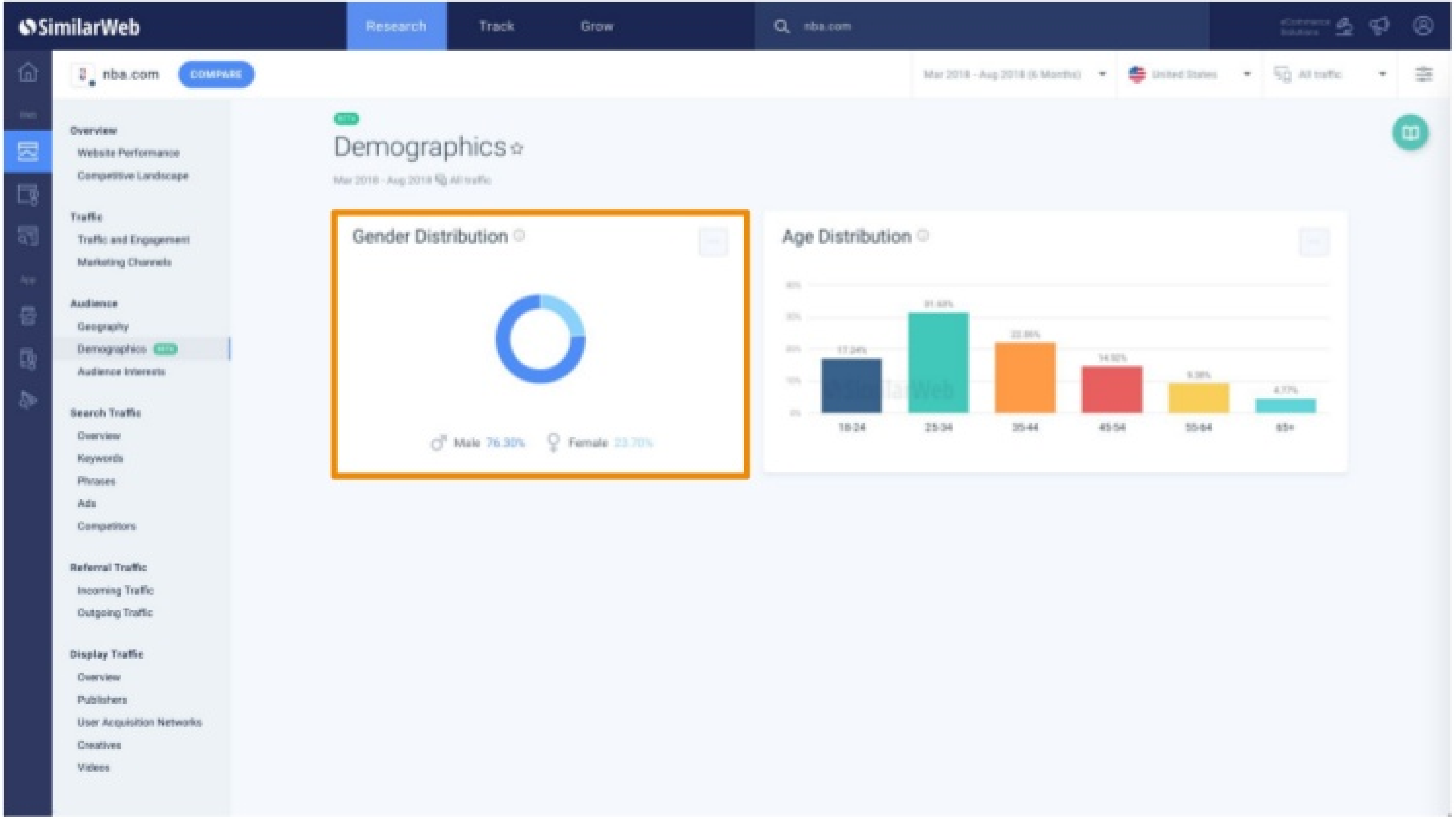
Every Mobile App

- ✓ Ranking
- ✓ Engagement
- ✓ App Store
- ✓ Category
- ✓ Keywords

SimilarWeb PRO



Website Demographics



Our Data

International Panel

Millions of user in almost every country.

Learning Set

Direct measurement data (like Google Analytics) for **~50,000** Websites.

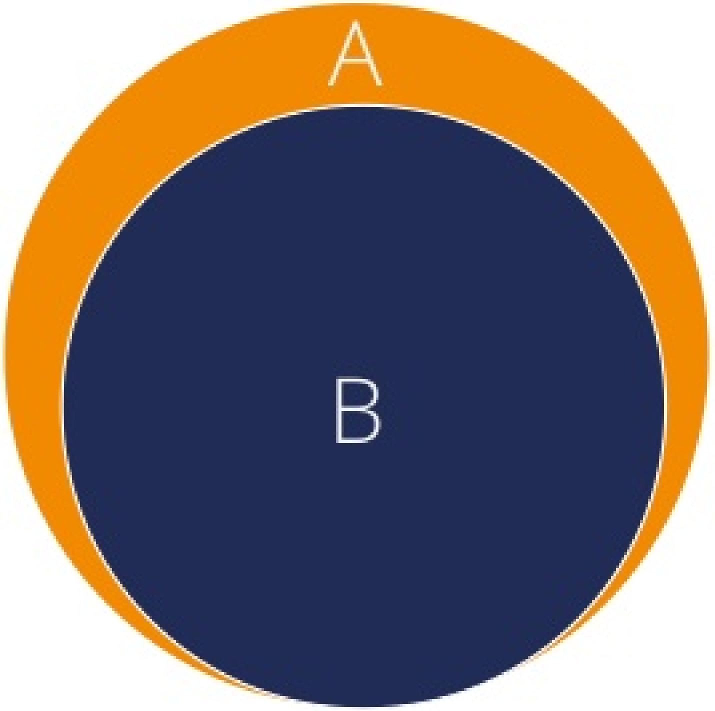
Gender Distribution - Standard Solution

- For each website in our panel:
    - Count the number of males
    - Count the number of females
    - Calculate the gender distribution

- Use the learning set to improve the estimation

Our panel is completely anonymous!

Our Idea - Example



Website A:

- Panel: **100** Users
- Learning set: **80%** Females / **20%** Males

Website A - **80** Females and **20** Males

Website B:

- Panel: **90** Users
- Learning set: **N/A**

↓

Website B Gender Distribution

77% - 88% Females

Our Idea

Estimate website gender distribution

based on its user engagement with the other sites in the learning set.

Our Panel Matrix (**P**)

Convert Our Panel Into An Interaction Matrix

- An indicator matrix of websites (**S**) and users (**U**):

  P(i, j) = 1 if user j visited website i

  P(i, j) = 0 Otherwise

- |S| = Millions, |U| = Millions

  dim(P) = |S| x |U| → **<u>P is a very large sparse matrix!</u>**

"The Curse of Dimensionality"

We need to reduce the dimension of the panel matrix (P):

$$\dim(P) = |S| \times \mathbf{|U|}$$
$$\downarrow$$
$$\dim(\mathbf{F}) = |S| \times \mathbf{K}$$

Feature Extraction / Dimension Reduction Algorithms:

- Principal Component Analysis
- Matrix Factorization (ALS Model)
- Word2Vec
- ...

The standard algorithms didn't solve our problem

Dimension Reduction - Conclusion

The Problem

The standard algorithms reduce the dimension **without taking into account our problem.**

↓

The Solution

An algorithm that reduces the dimension in a way that is **optimized to solve our problem.**

Interaction Based Feature Extraction

Interaction Based Feature Extraction - Step 1/3

- Convert our learning set (**L**) into a matrix (**D1**):
  - Split the gender percentiles into **K** (=10) “buckets”:

    [0.0-0.1, 0.1-0.2, 0.2-0.3, ... 0.9-1.0]

  - Map each value from the learning set into an indicator vector:

    Website A, **0.73** → [0, 0, 0, 0, 0, 0, 0, **1**, 0, 0]
    Website B, **0.26** → [0, 0, **1**, 0, 0, 0, 0, 0, 0, 0]
    ...

  dim(D1) = |L| x K

Interaction Based Feature Extraction - Step 1/3

Learning Set Vector (K)

Learning Set Websites

0.0	0.0	0.0	0.0	...
0.0	0.0	**1.0**	0.0	...
...	...	...	...	...

D1

Interaction Based Feature Extraction - Step 2/3

- Create another matrix (**D2**) of users (**U**) and <u>only the learning set</u> websites (**L**):

  D2(i, j) = 1 if user i visited learning set website j
  D2(i, j) = 0 Otherwise

  dim(D2) = |U| x |L|

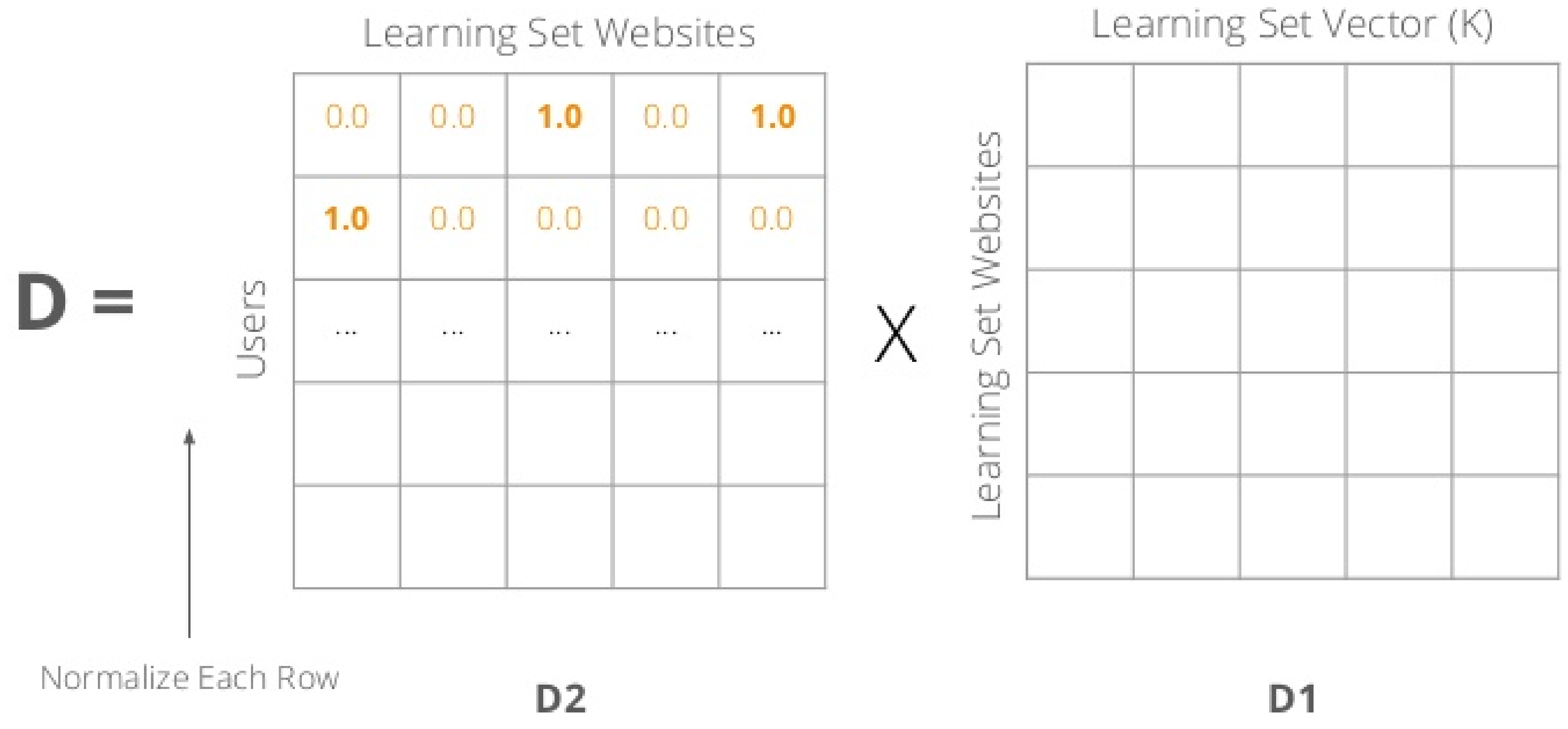
- Multiply the D2 matrix by D1:

  D = D2 * D1

  dim(D) = (|U| x |L|) * (|L| x K) → **|U|x K**

- Normalize each row (user) in the matrix **D** to **1.0**

Interaction Based Feature Extraction - Step 2/3

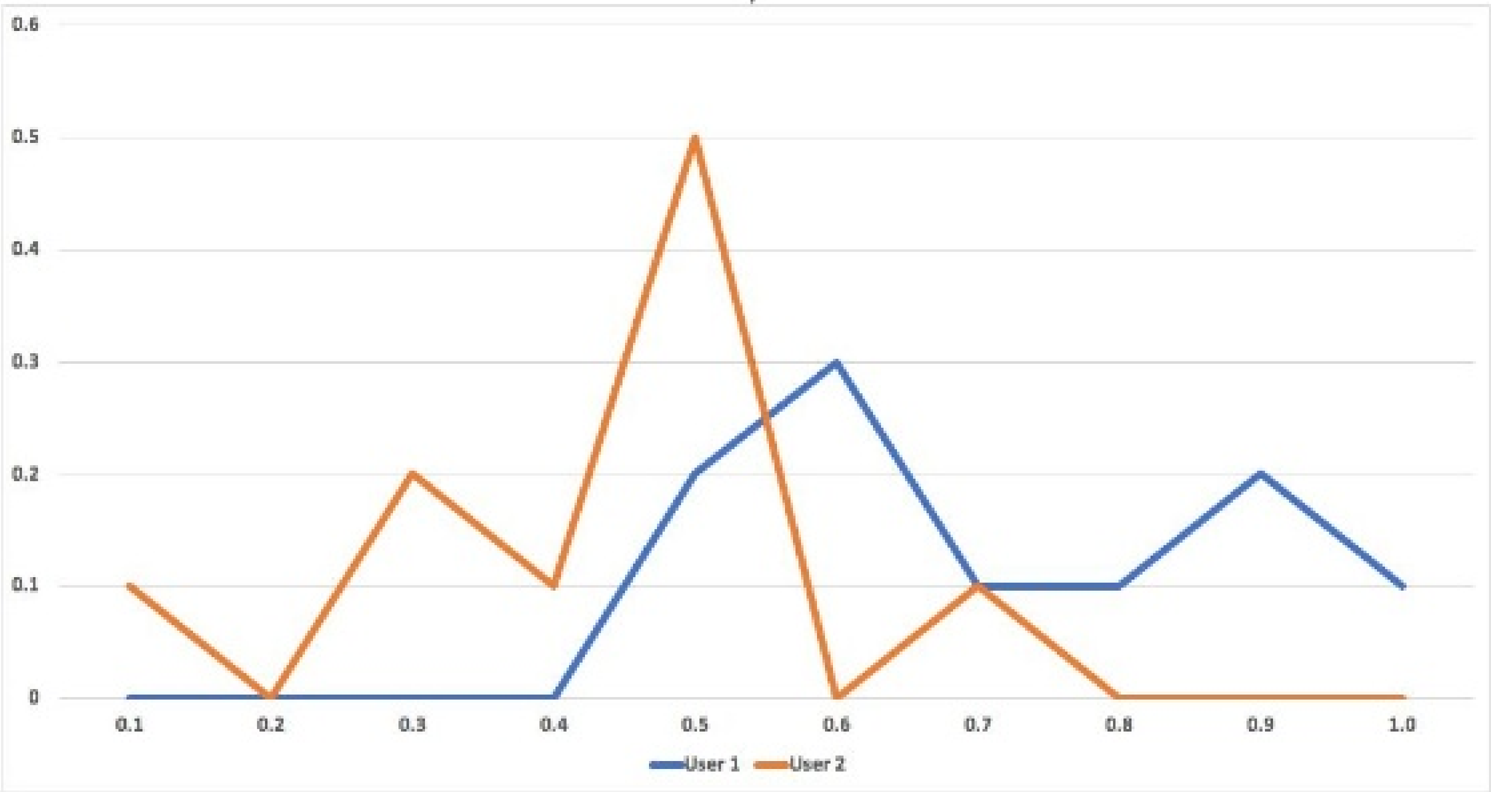


Interaction Based Feature Extraction - Step 2/3

- **D** Matrix - Example:

User 1: [0.0, 0.0, 0.0, 0.0, **0.2, 0.3, 0.1 0.1, 0.2, 0.1**]

User 2: [**0.1,** 0.0, **0.2, 0.1, 0.5**, 0.0, **0.1**, 0.0, 0.0, 0.0]



Interaction Based Feature Extraction - Step 3/3

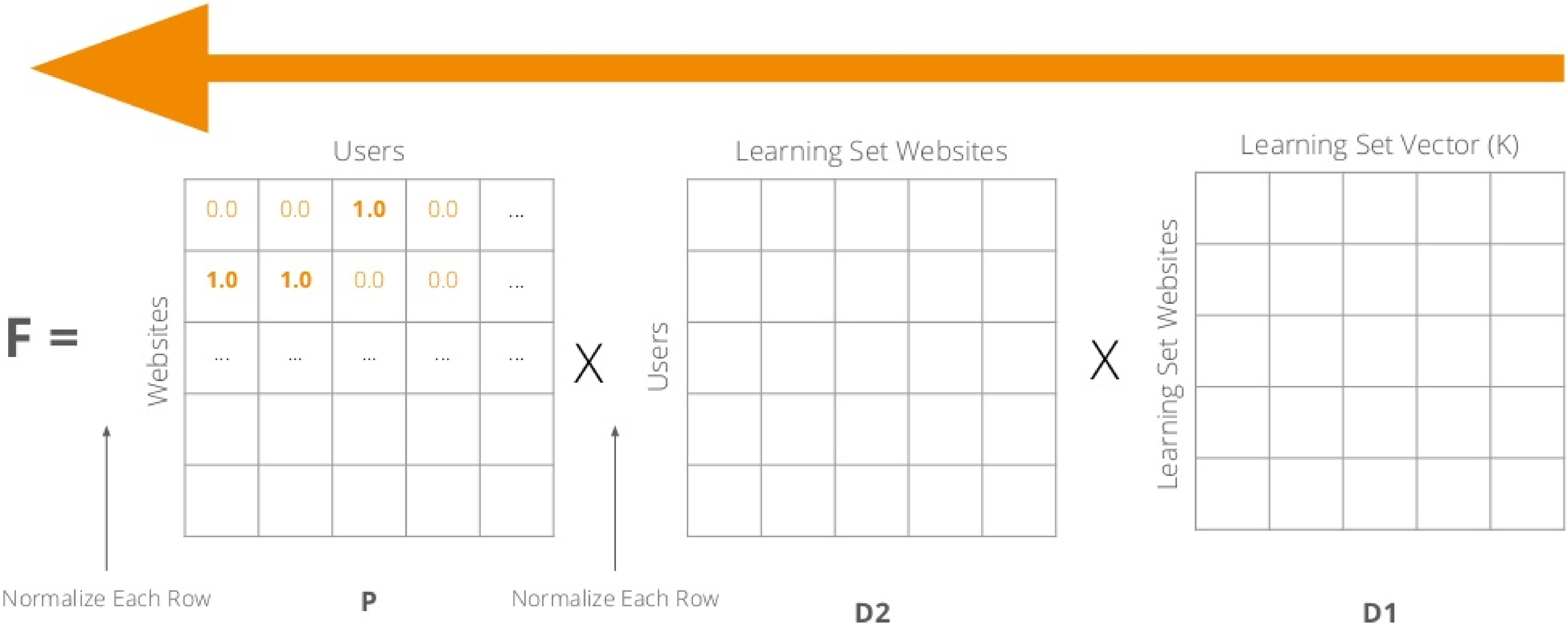
- Multiply the panel matrix (**P**) by **D**:

  F = P * D

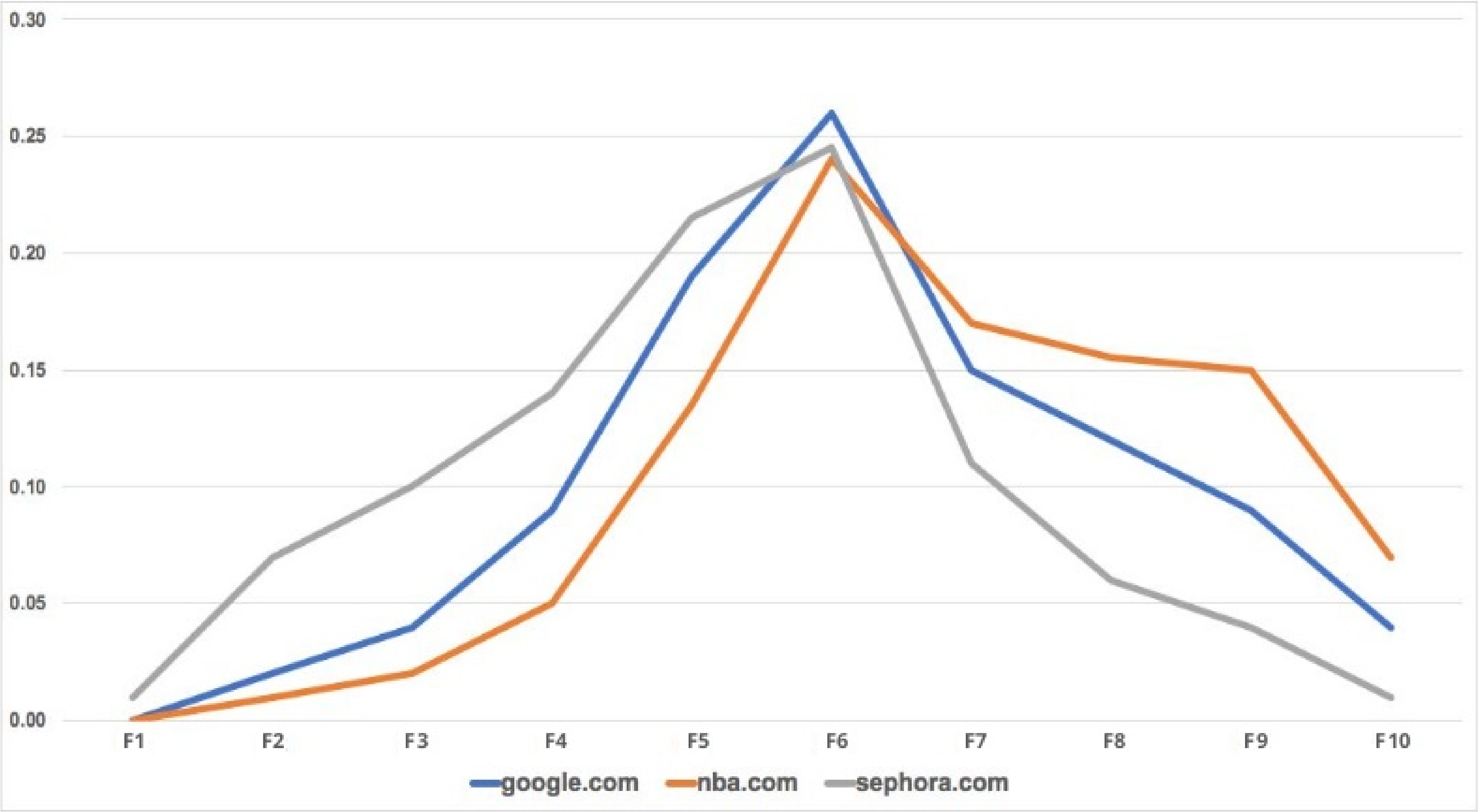
  dim(F) = (|S| x |U|) * (|U| x K) → **|S|x K**

- Normalize each row (website) in the matrix **F** to **1.0**

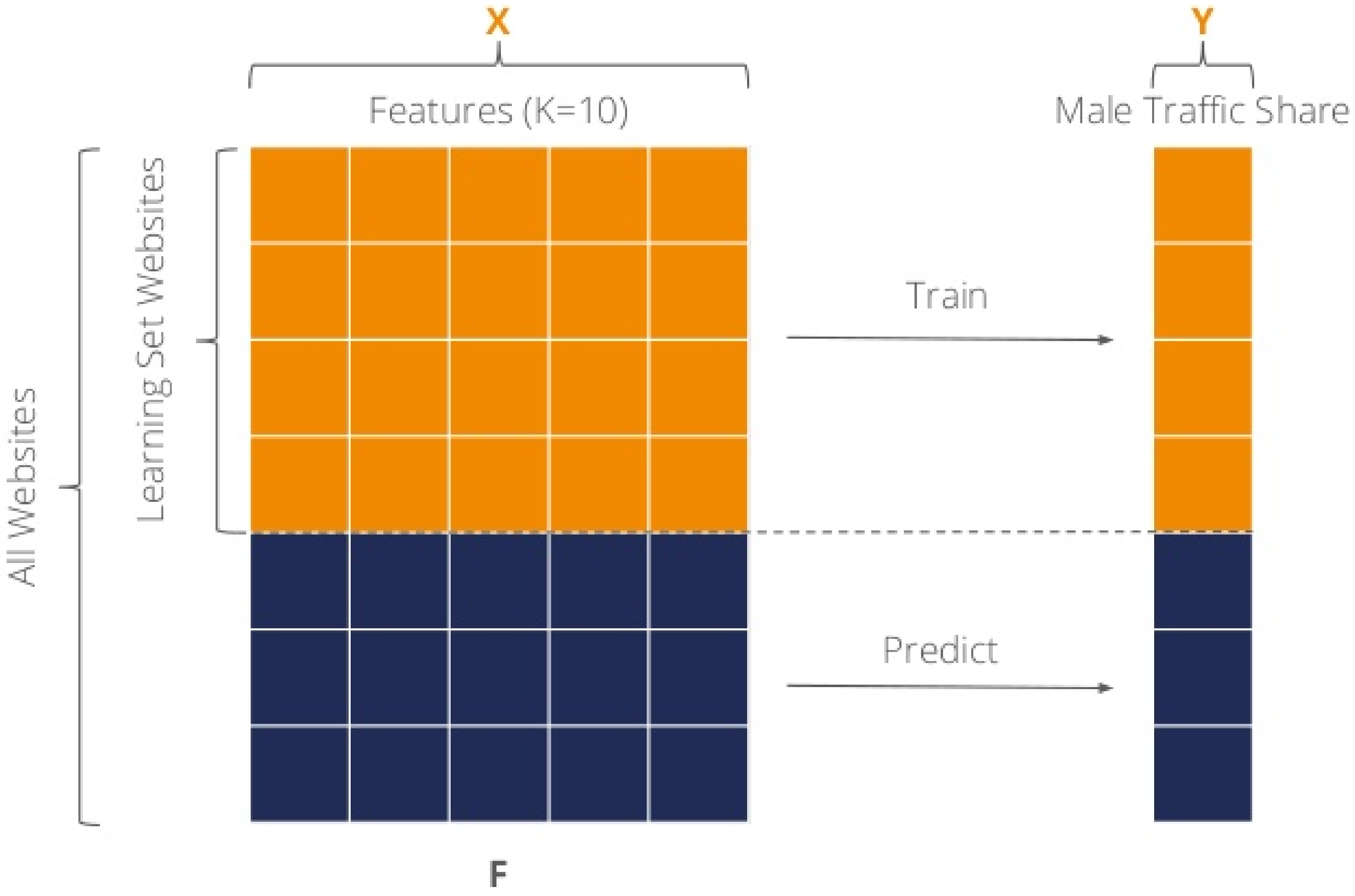
Interaction Based Feature Extraction - Step 3/3



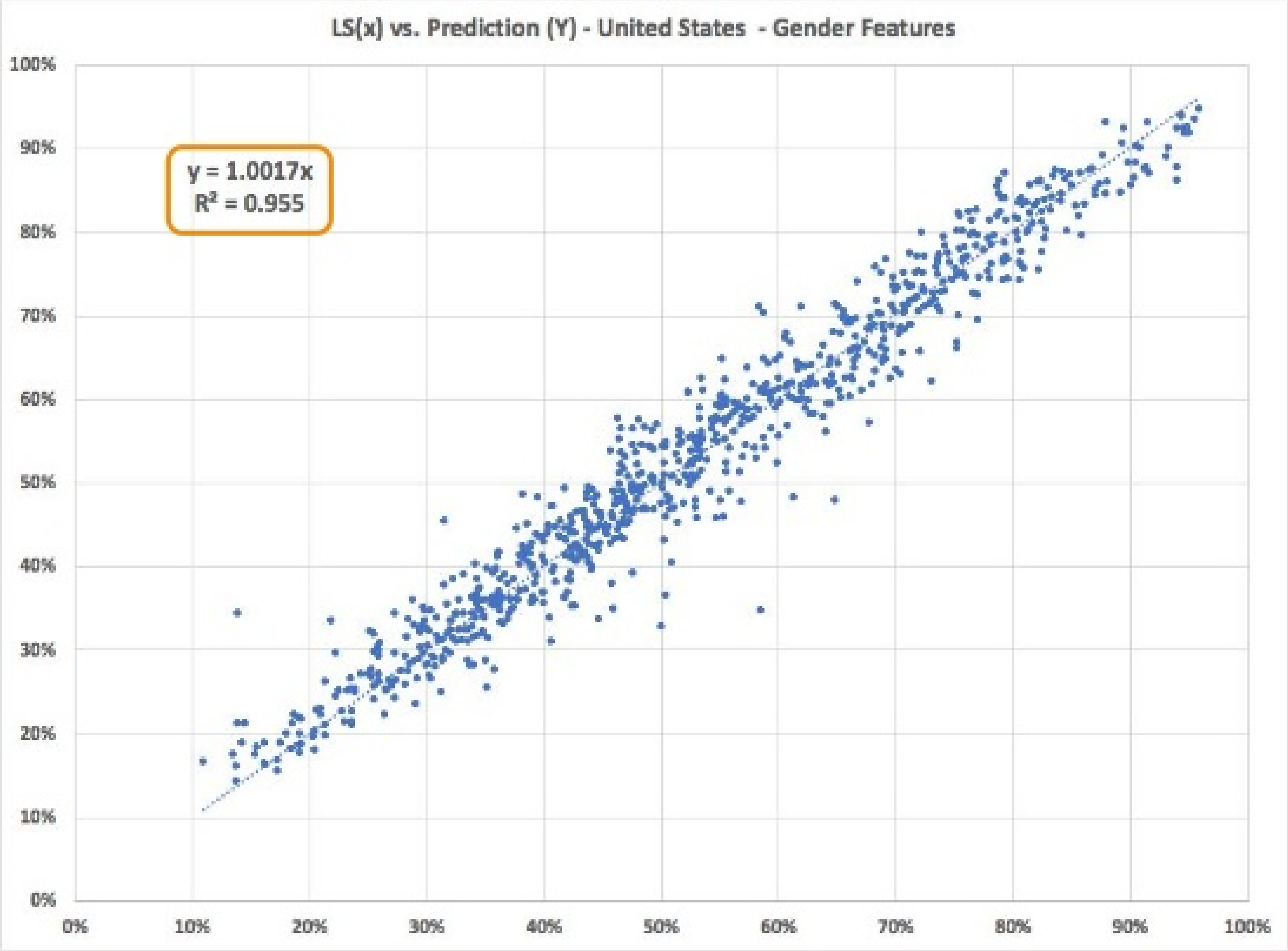
Interaction Based Feature Extraction - Features



Train a Regressor



Random Forest Regression - Results



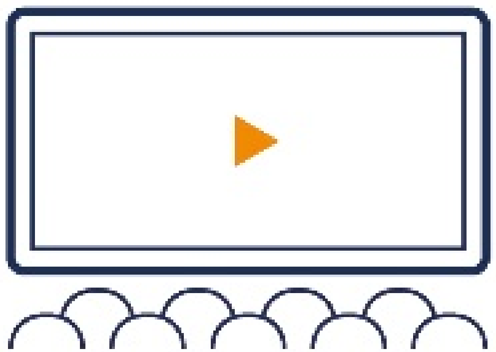
Expanding The Algorithm

Can We Expand It To Other Domains?

MovieLens Dataset:

- 27,000 Movies
- 138,000 Users
- ~20M Ratings (Explicit Feedback: 1-5)
- Multiple Genres per Movie

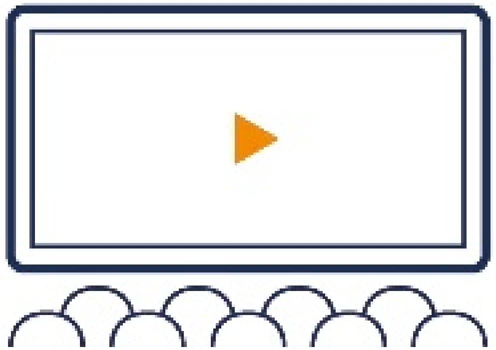
F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872



Predicting Movie Genres

MovieLens Movie Genres (**18**):

- Action
- Adventure
- Animation
- Children
- Comedy
- Crime
- Documentary
- Drama
- Fantasy
- Film-Noir
- Horror
- Musical
- Mystery
- Romance
- Sci-Fi
- Thriller
- War
- Western

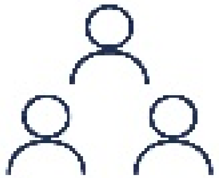


Predicting Movie Genres

ratings.csv

```
userId,movieId,rating,timestamp
1,2,3.5,1112486027
1,29,3.5,1112484676
1,32,3.5,1112484819
...
```

→



International Panel

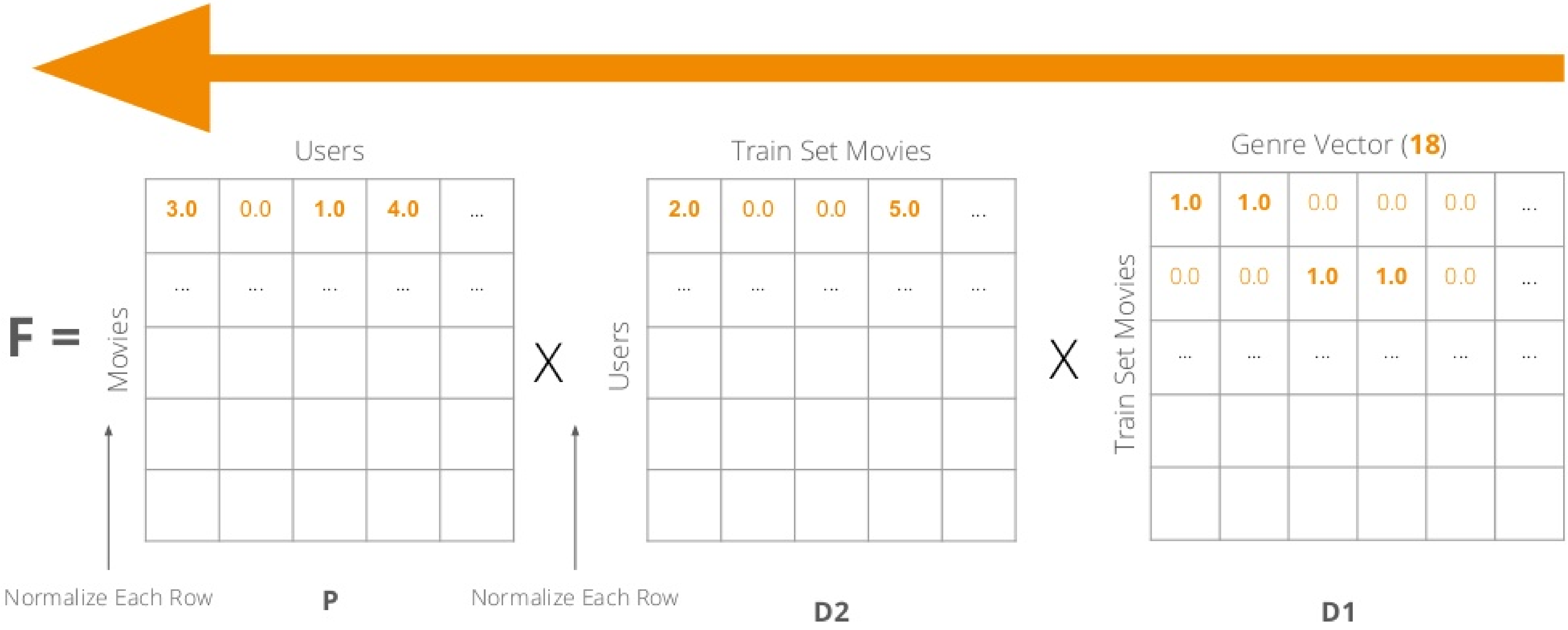
movies.csv

```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
...
```

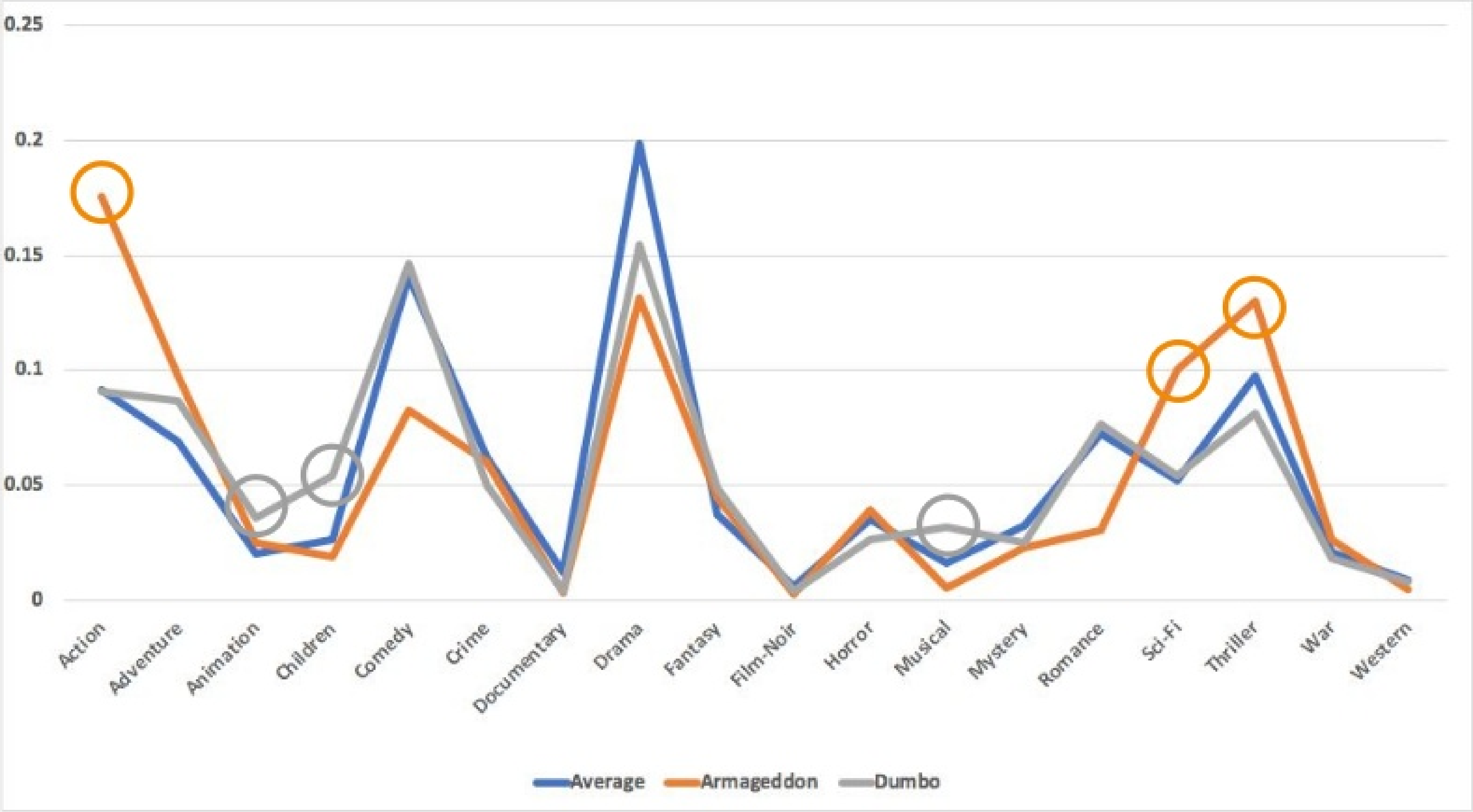
→

Learning Set

Predicting Movie Genres - Genre Features



Predicting Movie Genres - Genre Features



Predicting Movie Genres - Results

Genre: Animation

- Total Accuracy: 95.5% (6611)
- True Positive Rate: **93.4%** (273)
- True Negative Rate: 95.6% (6338)

Genre: Adventure

- Total Accuracy: 86.9% (6559)
- True Positive Rate: **70.0%** (682)
- True Negative Rate: 88.9% (5877)

Overall Accuracy: ~83%

* Movies with more than 20 users / votes

Interaction Based Feature Extraction

- High Accuracy
- Very Low Dimension
- Scalable
- Simple & Explainable (not a “black box” solution)

Questions?

Interaction Based Feature Extraction - SQL Implementation

F =

Normalize Each Row

P

Users (columns) × Websites (rows)

0.0	0.0	**1.0**	0.0	…
1.0	**1.0**	0.0	0.0	…
…	…	…	…	…

X

Normalize Each Row

D2

Learning Set Websites (columns) × Users (rows)

0.0	0.0	**1.0**	0.0	**1.0**
1.0	0.0	0.0	0.0	0.0
…	…	…	…	…

X

D1

Learning Set Vector (K) (columns) × Learning Set Websites (rows)

0.0	0.0	0.0	0.0	…
0.0	0.0	**1.0**	0.0	…
…	…	…	…	…

SimilarWeb

Interaction Based Feature Extraction - SQL Implementation

```
Table D1: <ls_site, k1, k2... k10>

Table D2: <user, ls_site>

Table P: <site, user>
```

Interaction Based Feature Extraction - SQL Implementation

```
SELECT
        site
        AVG(f1) as f1,
        ...
        AVG(f10) as f10
FROM (
        SELECT
                site,
                user,
                D.k1 / (D.k1 + ... + D.k10) as f1,
                ...
                D.k10 / (D.k1 + ... + D.k10) as f10
        FROM (
                SELECT P.site, D.user, D.ls_site, D.k1... D.k10
                FROM P
                JOIN (
                        SELECT D2.user, D1.ls_site, D1.k1... D1.k10
                        FROM D1
                        JOIN D2
                        ON D1.ls_site = D2.ls_site
                ) AS D
                ON P.user = D.user AND P.site <> D.ls_site
        ) as F1
        GROUP BY site, user
) as F
GROUP BY site
```

Interaction Based Feature Extraction - SQL Implementation

```
SELECT
        site
        AVG(f1) as f1,
        ...
        AVG(f10) as f10
FROM (
        SELECT
                site,
                user,
                D.k1 / (D.k1 + ... + D.k10) as f1,
                ...
                D.k10 / (D.k1 + ... + D.k10) as f10
        FROM (
                SELECT P.site, D.user, D.ls_site, D.k1... D.k10
                FROM P
                JOIN (
                        SELECT D2.user, D1.ls_site, D1.k1... D1.k10
                        FROM D1
                        JOIN D2
                        ON D1.ls_site = D2.ls_site
                ) AS D
                ON P.user = D.user AND P.site <> D.ls_site
        ) as F1
        GROUP BY site, user
) as F
GROUP BY site
```

Avoid Data Leakage

Thank You!

shlomib@similarweb.com

@shlomibabluki

SimilarWeb