Recommender Workshop

Part 3: Matrix Factorization

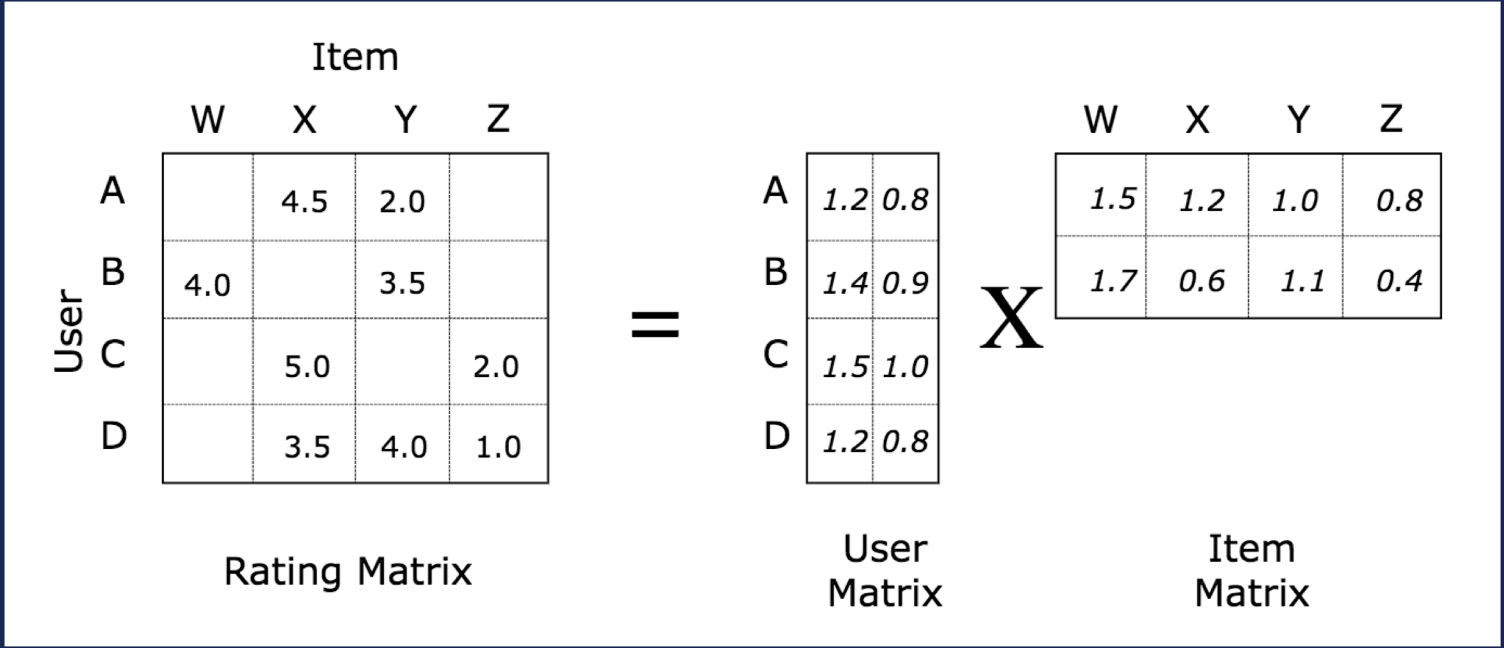
© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recommender Workshop Agenda

- Part 1: Introduction
    - Overview of Machine Learning Process, Amazon SageMaker
    - Hands-on: Data Exploration
- Part 2: Collaborative Filtering
    - Core Concepts for Recommendations
    - Hands-on: K-Means Clustering
- Part 3: Matrix Factorization (You Are Here)
    - Refining Recommendations
    - Hands-on: Factorization Machine
- Part 4: Hyperparameter Tuning
    - Key Concepts
    - Hands-on: Hyperparameter Tuning

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Recommender: Matrix Factorization



Item

User

$$\begin{array}{c|cccc} & W & X & Y & Z \\ \hline A & & 4.5 & 2.0 & \\ B & 4.0 & & 3.5 & \\ C & & 5.0 & & 2.0 \\ D & & 3.5 & 4.0 & 1.0 \end{array} = \begin{array}{c|cc} A & 1.2 & 0.8 \\ B & 1.4 & 0.9 \\ C & 1.5 & 1.0 \\ D & 1.2 & 0.8 \end{array} \times \begin{array}{cccc} W & X & Y & Z \\ \hline 1.5 & 1.2 & 1.0 & 0.8 \\ 1.7 & 0.6 & 1.1 & 0.4 \end{array}$$

Rating Matrix

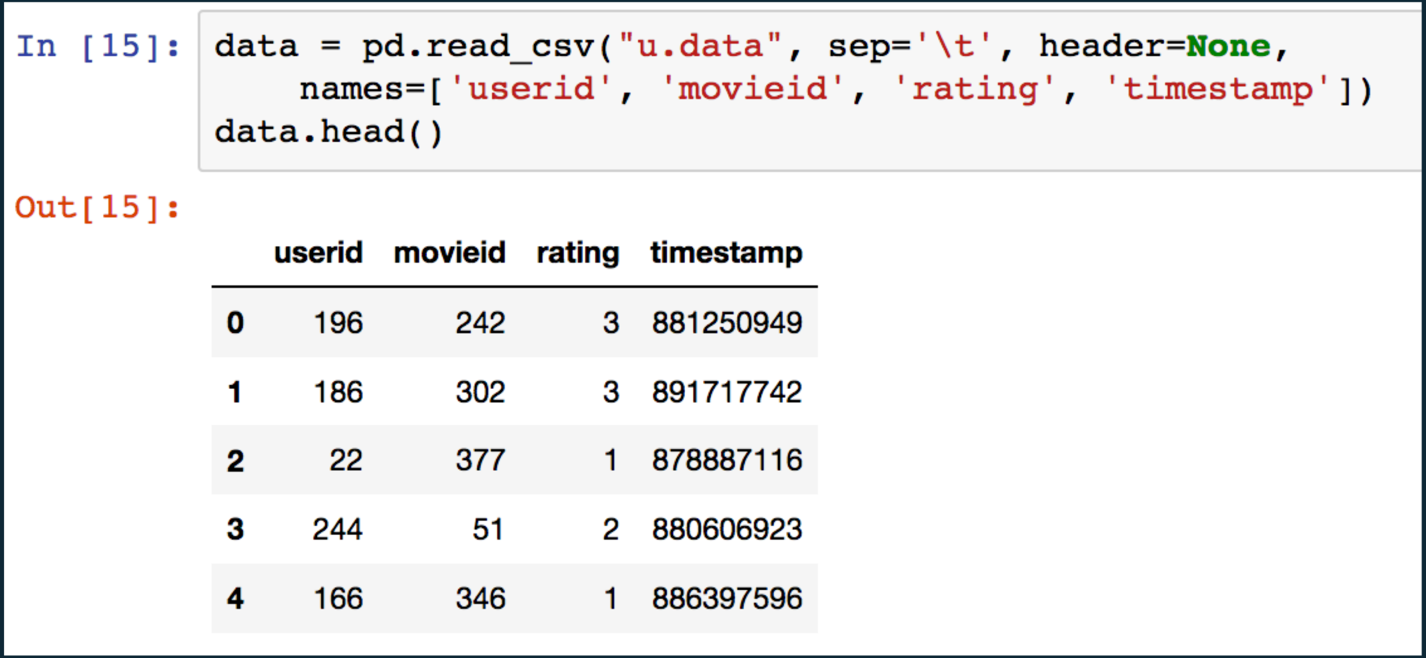
User Matrix

Item Matrix

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Our Data Set: Movielens

- Public Data Set produced by **GroupLens Research**
- https://grouplens.org/datasets/movielens/

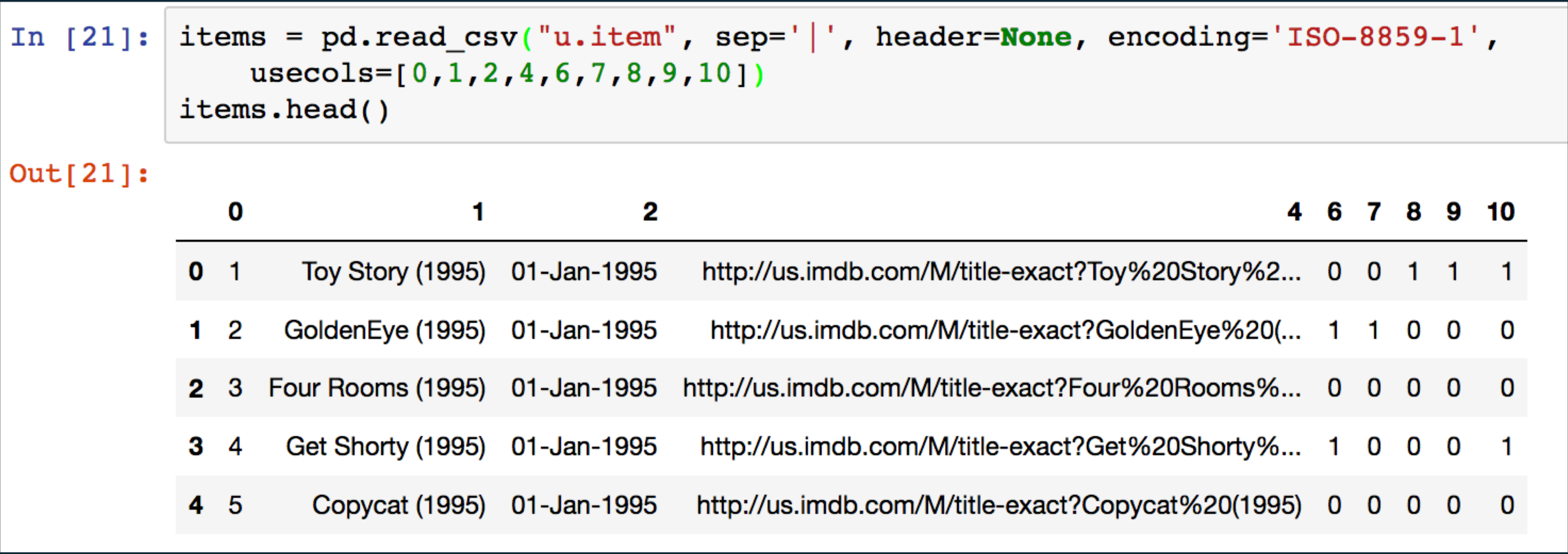


```
In [15]: data = pd.read_csv("u.data", sep='\t', header=None,
              names=['userid', 'movieid', 'rating', 'timestamp'])
         data.head()
```

Out[15]:

	userid	movieid	rating	timestamp
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596

Item Information

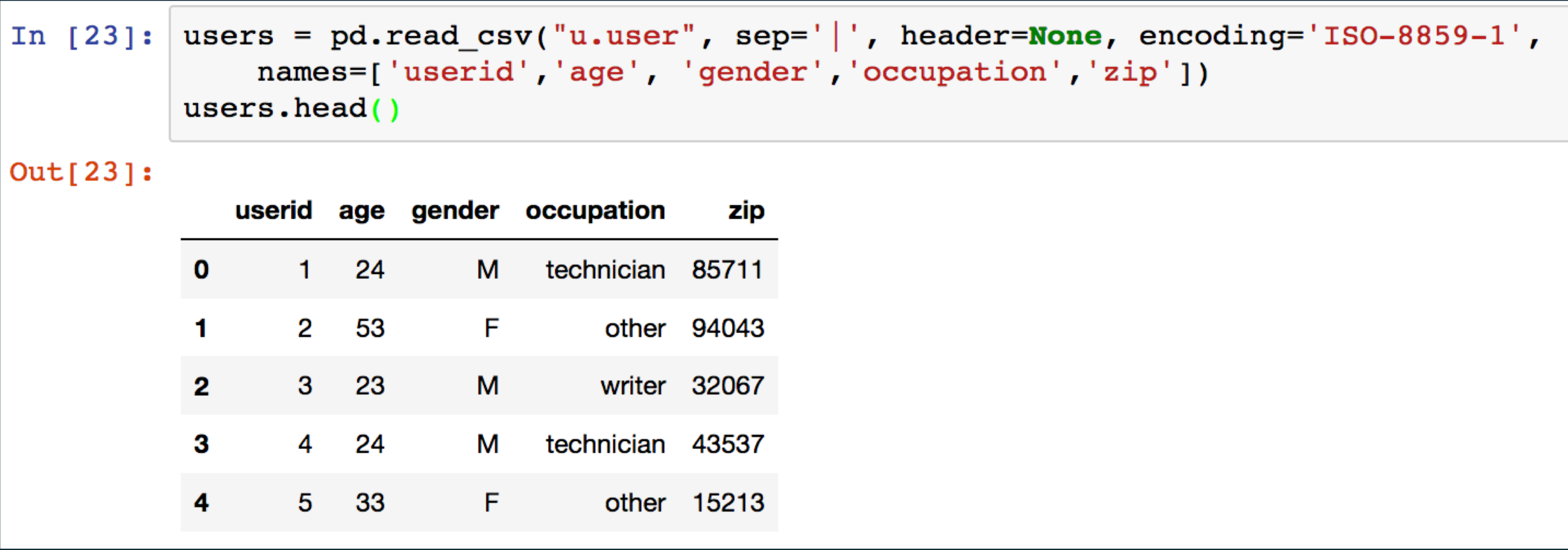


```
In [21]: items = pd.read_csv("u.item", sep='|', header=None, encoding='ISO-8859-1',
             usecols=[0,1,2,4,6,7,8,9,10])
         items.head()
```

Out[21]:

	0	1	2	4	6	7	8	9	10
0	1	Toy Story (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	1	1	1
1	2	GoldenEye (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?GoldenEye%20(...	1	1	0	0	0
2	3	Four Rooms (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Four%20Rooms%...	0	0	0	0	0
3	4	Get Shorty (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Get%20Shorty%...	1	0	0	0	1
4	5	Copycat (1995)	01-Jan-1995	http://us.imdb.com/M/title-exact?Copycat%20(1995)	0	0	0	0	0

User Information

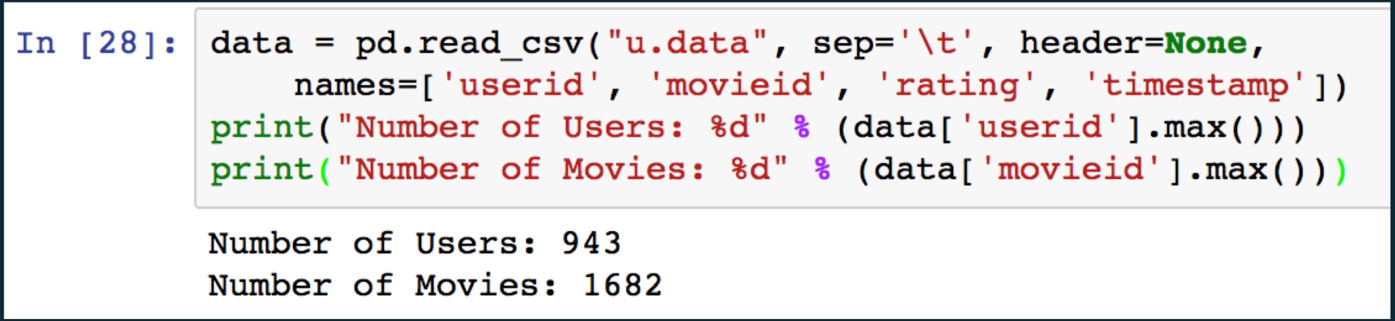


```
In [23]: users = pd.read_csv("u.user", sep='|', header=None, encoding='ISO-8859-1',
              names=['userid','age', 'gender','occupation','zip'])
         users.head()
```

Out[23]:

	userid	age	gender	occupation	zip
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213

Visualising The Data



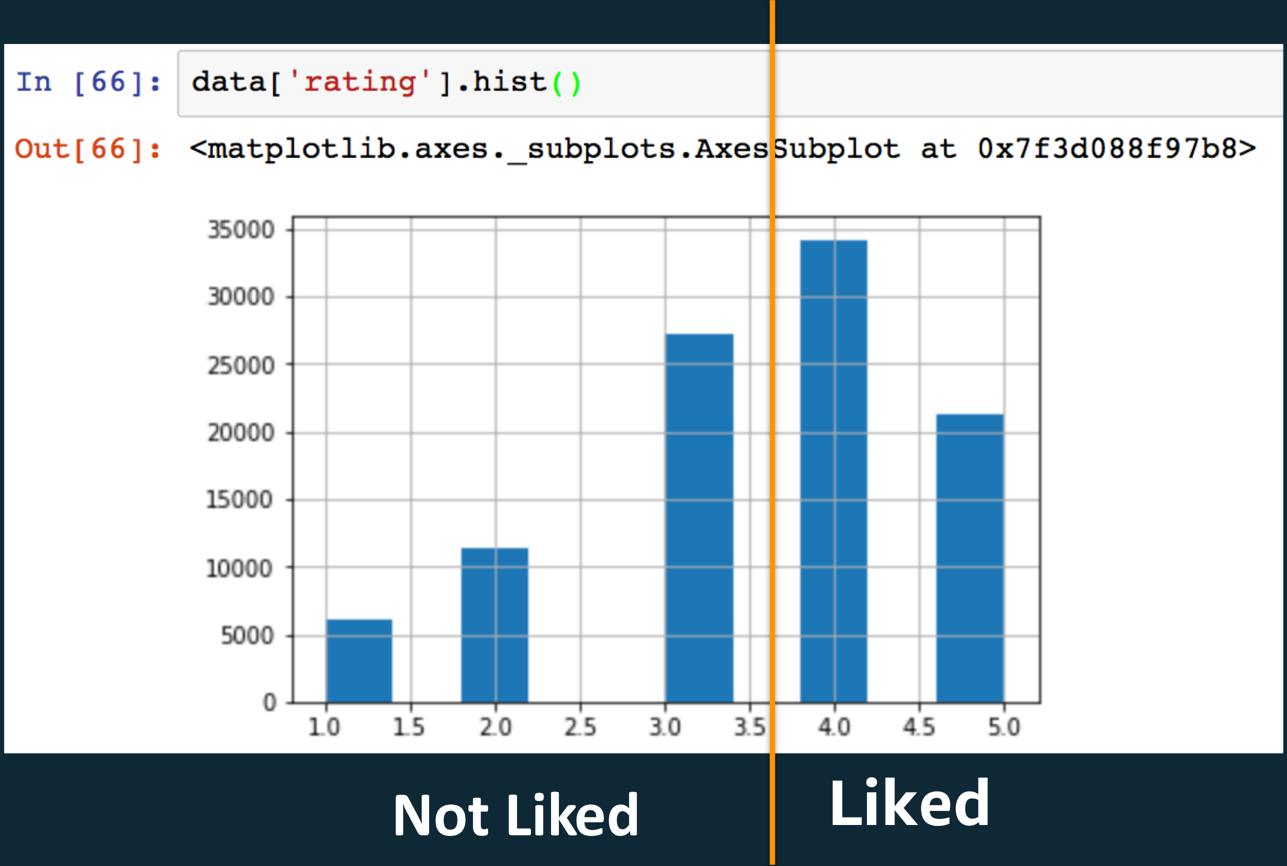
```
In [28]: data = pd.read_csv("u.data", sep='\t', header=None,
                 names=['userid', 'movieid', 'rating', 'timestamp'])
         print("Number of Users: %d" % (data['userid'].max()))
         print("Number of Movies: %d" % (data['movieid'].max()))

         Number of Users: 943
         Number of Movies: 1682
```

Total Feature Count = Users + Movies

= **2625 Features**

Data Preparation: Binary Classification



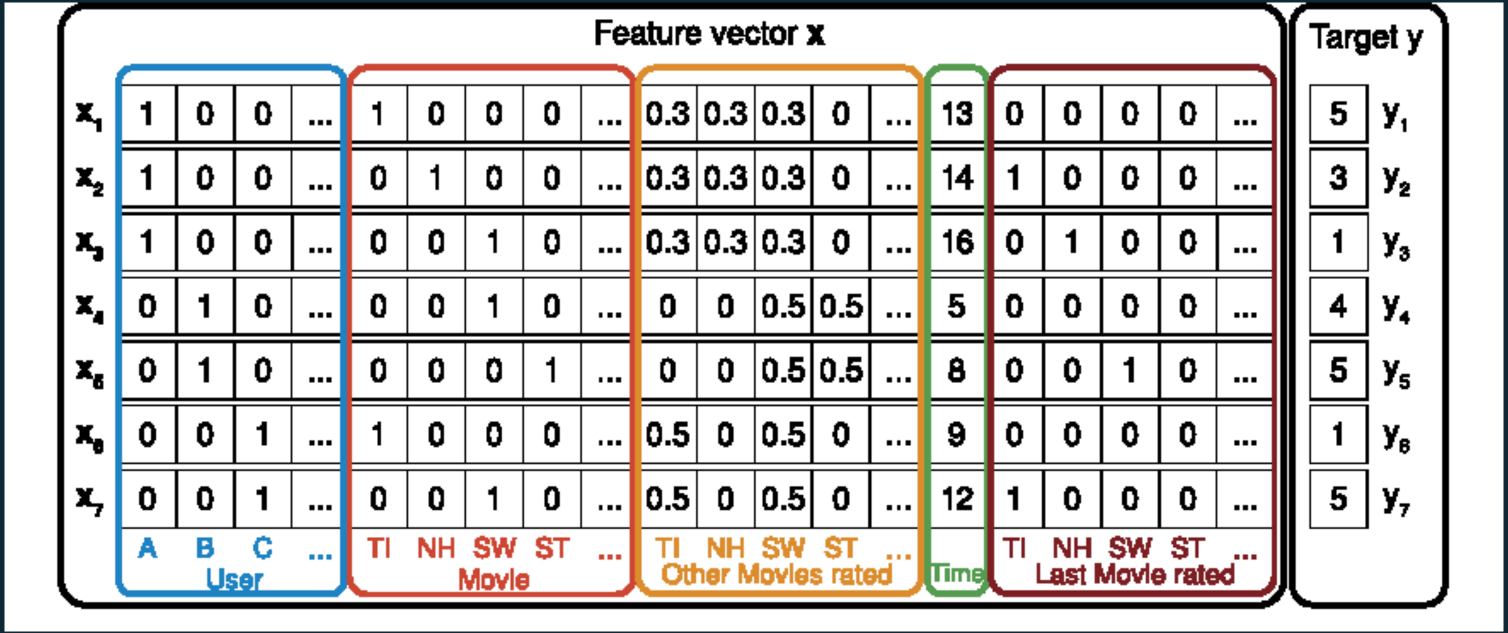
```
In [66]: data['rating'].hist()

Out[66]: <matplotlib.axes._subplots.AxesSubplot at 0x7f3d088f97b8>
```

Not Liked

Liked

Factorization Machines



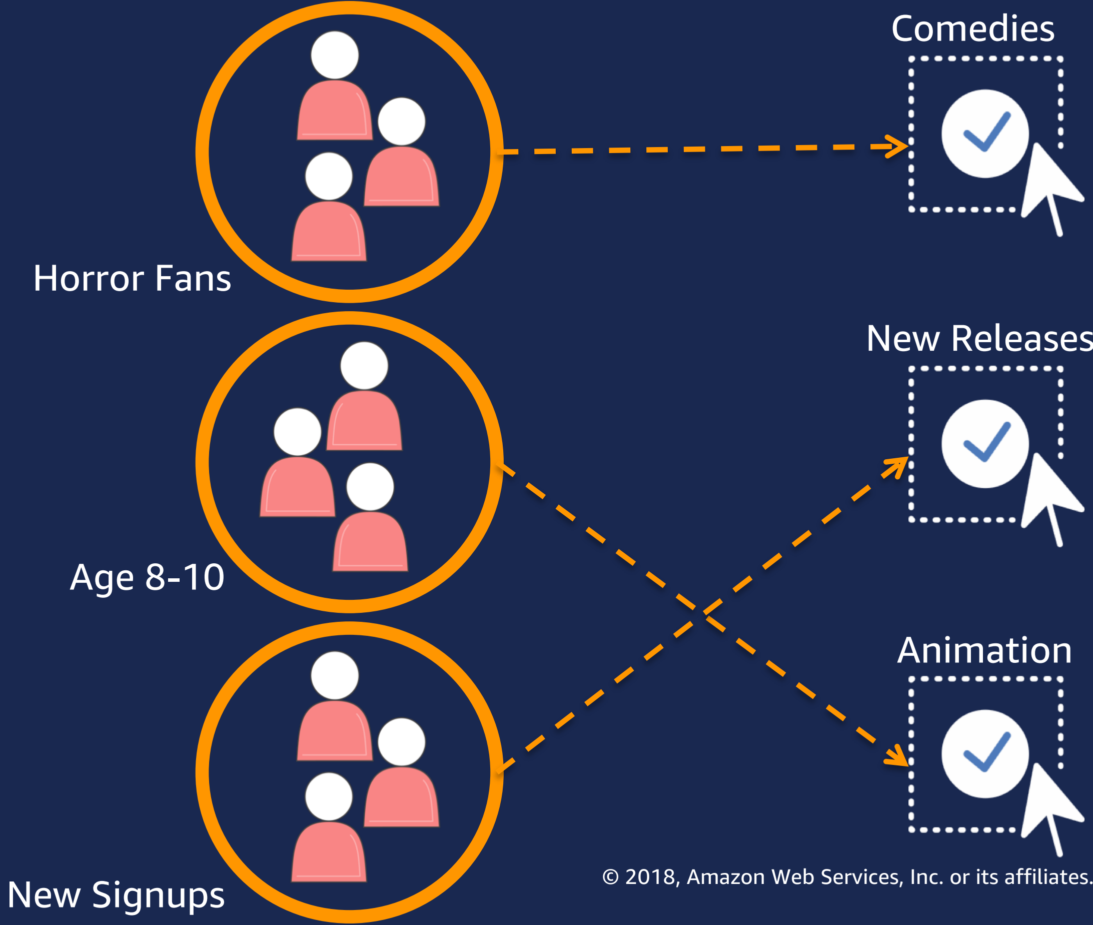
	Feature vector x																				Target y	
x_1	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	13	0	0	0	0	...	5	y_1
x_2	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	14	1	0	0	0	...	3	y_2
x_3	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	16	0	1	0	0	...	1	y_3
x_4	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	5	0	0	0	0	...	4	y_4
x_5	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	8	0	0	1	0	...	5	y_5
x_6	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	9	0	0	0	0	...	1	y_6
x_7	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	12	1	0	0	0	...	5	y_7
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		TI	NH	SW	ST	...		
	User				Movie					Other Movies rated					Time	Last Movie rated						

Recommender Workshop Activity

- Log into the AWS console
- Find the Amazon SageMaker service
- Find Notebooks
- Open the notebook instance and find within the repo path:
    - 03_factorization_machines.ipynb

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Putting it together



1. Cluster individual users into groups
2. Train models for each genre
3. Generate predictions using the model that aligns best to the application context

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Next: Part 4

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.