Recommending Movies

● ● ●

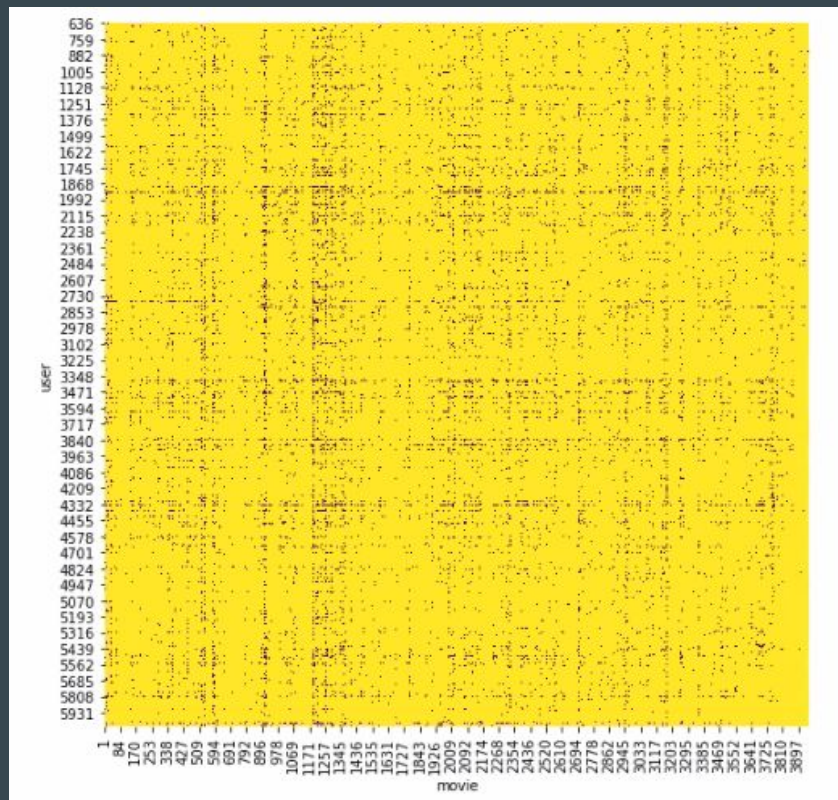By: Chelsea, Morgan, Tom, & Alex

The Data

- Movie Ratings dataset from MovieLens
- 'User' attributes: Gender, Age, Occupation and Zip-code of **6,040 users**
- 'Movies' attributes: Title and Genre of **3,883 movies**
  - 301 unique genre combinations
- **Training dataset had ratings from 5,399 users & 3,662 movies**
- **# of observations:**
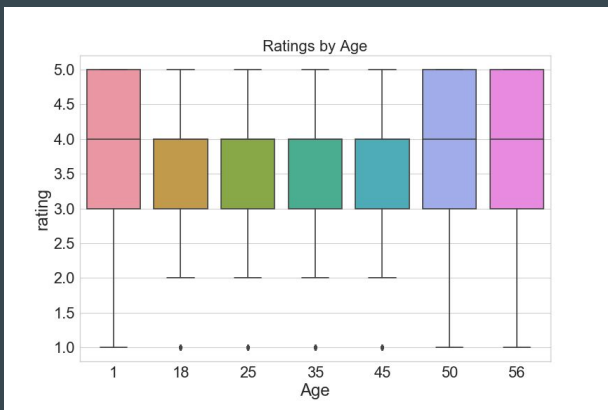 - ~200k test
  - ~800k train

movielens

EDA

Utility Matrix Density = 0.040

EDA
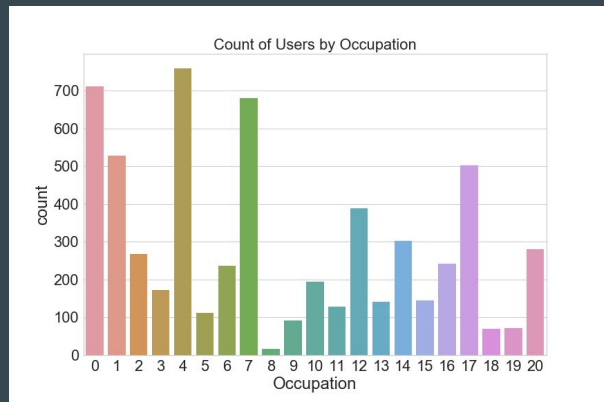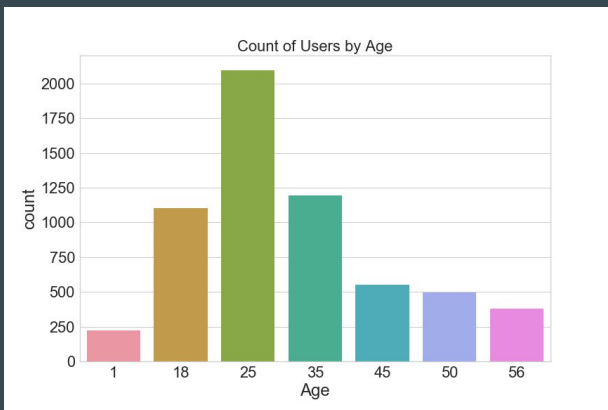
EDA
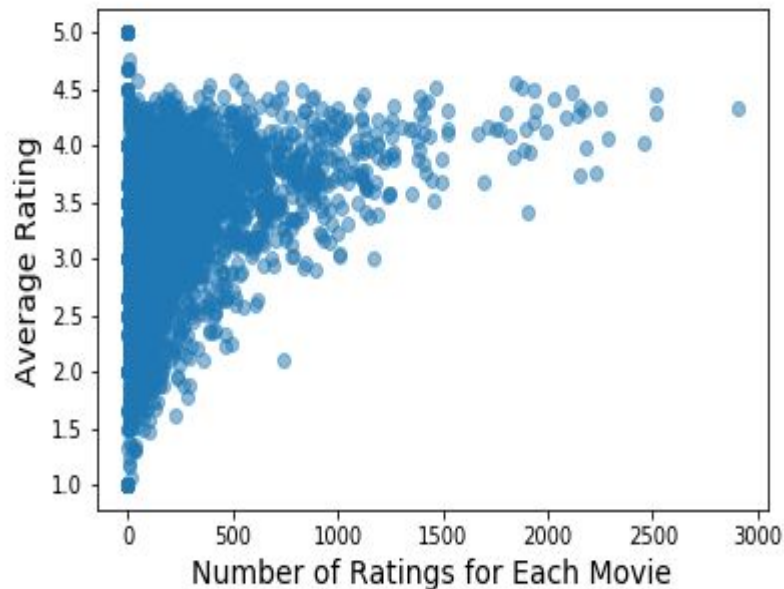
| Title	# Ratings	Rating
American Beauty (1999)	2,901	4.3289
Star Wars: Episode V - The Empire Strikes Back (1980)	2,516	4.2901
Star Wars: Episode IV - A New Hope (1977)	2,515	4.4521
Star Wars: Episode VI - Return of the Jedi (1983)	2,456	4.0195
Terminator 2: Judgment Day (1991)	2,284	4.0679
Saving Private Ryan (1998)	2,245	4.3265
Jurassic Park (1993)	2,232	3.7554
Back to the Future (1985)	2,183	3.9835
Matrix, The (1999)	2,172	4.3163
Men in Black (1997)	2,156	3.7305
Silence of the Lambs, The (1991)	2,151	4.3598
Fargo (1996)	2,144	4.2607
Raiders of the Lost Ark (1981)	2,110	4.4787
Braveheart (1995)	2,081	4.2403
Sixth Sense, The (1999)	2,024	4.4175

Cold Start

- Three scenarios
    1. Unknown user
        a. Use the average rating of the movie
    2. Unknown movie
        a. Use the average rating of the user
    3. Unknown user and unknown movie
        a.

Clustering

- Kmeans for clustering
  - Movies by genre
  - Users by age group, occupation, gender
- Elbow method to identify K
  - Movie = 4
  - User = 6

K cluster elbow search - Movie
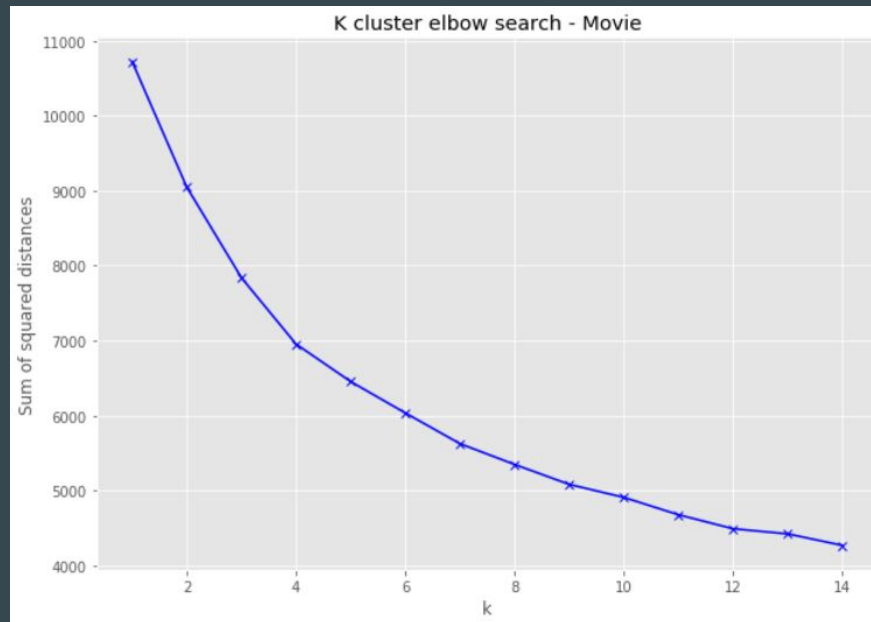
Clustering

- Create utility matrix with crossings
  - 'Tom'   User cluster 2
  - 'Memento'   Movie cluster 0
  - Avg rating (U-2, M-0) = 3.6
- This becomes our baseline rating
  - Test score = 3.774

	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
					U2 M0 3.6
		U0 M4 3.3			

Final Recommender

- Most instances are warm
  - We have either avg movie rating or avg user rating or both
  - Blend with the baseline
- True cold start
  - Stick with the baseline
- Final tweaks
  - Dampened the weight of the average movie ratings if movies had little reviews (Noise)
- Final submission score = 4.227

Conclusion & Next Steps

Conclusion

- Clustering movies & users helped improve recommender performance
  - Clusters provided more fine-tuned rating averages per user/movie group
- Adding weights based on # ratings per movie also helped slightly

Next Steps

- Add/tune more weights to scale predictions for certain movies/users based on EDA
  - Example: For user groups that tend to rate higher, bump their predicted rating up proportionally

Thank You