# Movie Recommender

## Case Study

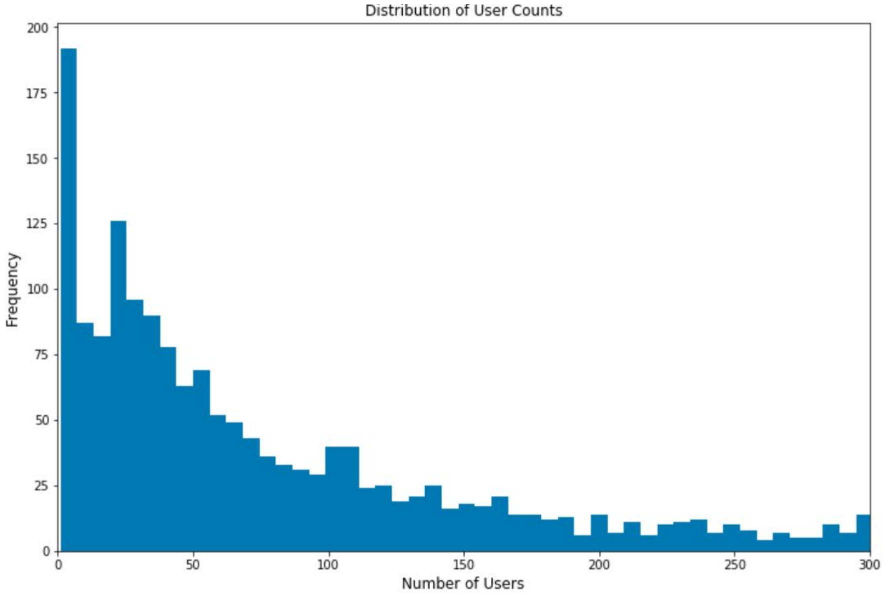
Group: Aaron Burr
Members: Hamilton, Nabor, Sonia, Match

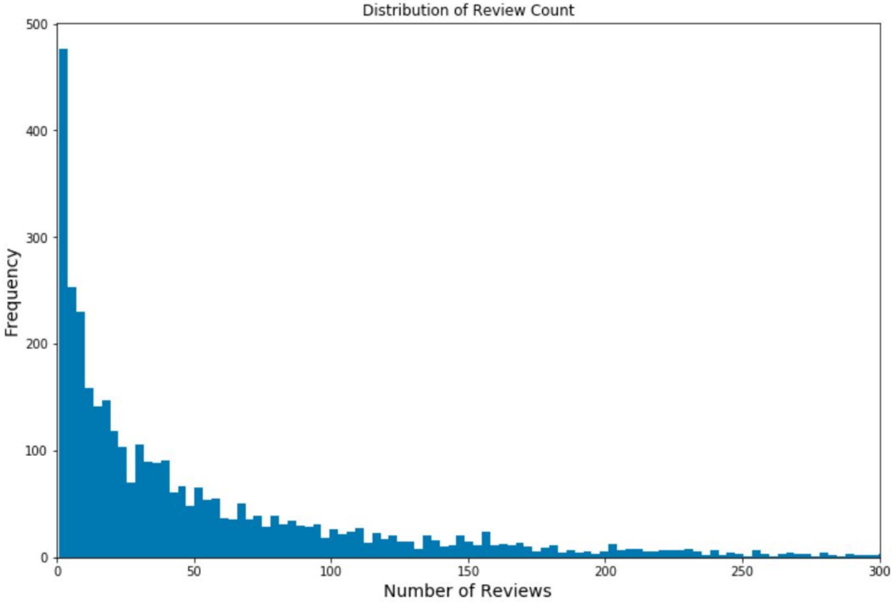
# Initial Observations

- Most movies are reviewed by less than ~50 people.
- Most users review less than ~10 movies
- Data is very sparse
- ALS will generate a lot of NaNs

(insert photoshopped movie poster here)

# Initial Observations



# Model Implementation

- Train/test split on time series
- UVD + ALS
- Tested various hyperparams
    - Rank
    - Max iterations
    - Regularization
- Imputed NaNs with movie average
    - Popular movies have higher mean ranking
    - Better than just global mean ranking

(insert photoshopped movie poster here)

# Results

Best evaluation metric:

**Moses score = 3.54**

| | | user | rating |
|---|---|---|---|
| **movie** | **title** | | |
| **2858** | **American Beauty (1999)** | 2901 | 4.328852 |
| **1196** | **Star Wars: Episode V - The Empire Strikes Back (1980)** | 2516 | 4.290143 |
| **260** | **Star Wars: Episode IV - A New Hope (1977)** | 2515 | 4.452087 |
| **1210** | **Star Wars: Episode VI - Return of the Jedi (1983)** | 2456 | 4.019544 |
| **589** | **Terminator 2: Judgment Day (1991)** | 2284 | 4.067863 |
| **2028** | **Saving Private Ryan (1998)** | 2245 | 4.326503 |
| **480** | **Jurassic Park (1993)** | 2232 | 3.755376 |
| **1270** | **Back to the Future (1985)** | 2183 | 3.983509 |
| **2571** | **Matrix, The (1999)** | 2172 | 4.316298 |
| **1580** | **Men in Black (1997)** | 2156 | 3.730519 |

# Conclusions

- Very large number of NaNs
- Scoring metric heavily dependent on imputed NaN values
- Need better imputation method or build model to reduce NaN count
- Alternatively, gather A LOT more data
- ALS would not work efficiently if new data was introduced
    - SDG is better

# Next Steps

- Improve impute NaN strategy to include mean user score using weighted mean
    - Weighted Mean = alpha x movie mean ranking + (1-alpha) x user mean ranking
- Adjust model based on reviews per user/reviews per movie
- Implement SVD model
- Combine content-based filtering
- Apply clustering methods

(insert photoshopped movie poster here)

# Thank you

## Questions?

(...and here)