Movie Recommendation System using MovieLens Dataset

Tim Chu and Elliot Frost

MovieLens Dataset Info

- 100,000 ratings and 3,600 tag applications applied to 9,000 movies by 600 users
- ratings.csv
  - userId’s, movielens movieId’s, and ratings
- movies.csv
  - movielens movieId’s, titles, and genres
- links.csv
  - movielens movieid’s, imdbId’s, and tmdbId’s
- tags.csv
  - userId’s, movielens movieId’s, and user submitted tags

The Problem

- We wanted to see if we would be able to predict if a user would like a movie
    - We extended that idea by predicting a user’s star rating of a movie

- Lots of data was provided, but what would be relevant?
    - tags.csv - Using natural language processing to generalize tags would’ve been possible
        - Outside of the scope of our project
    - links.csv - We didn’t care too much about imdb and tmdb’s movie id’s

The Solution

- Collaborative Filtering!
    - Average the reviews from other likeminded people
    - Check our user’s reviews from movies with similar genres

Challenges

- Which distance function to use?
    - We tested the following:
        - Pearson Correlation
        - Cosine Similarity
        - Euclidean Distance
- What language to use?
    - We both weren’t too familiar with any of the programming languages suitable for data mining
        - Python!

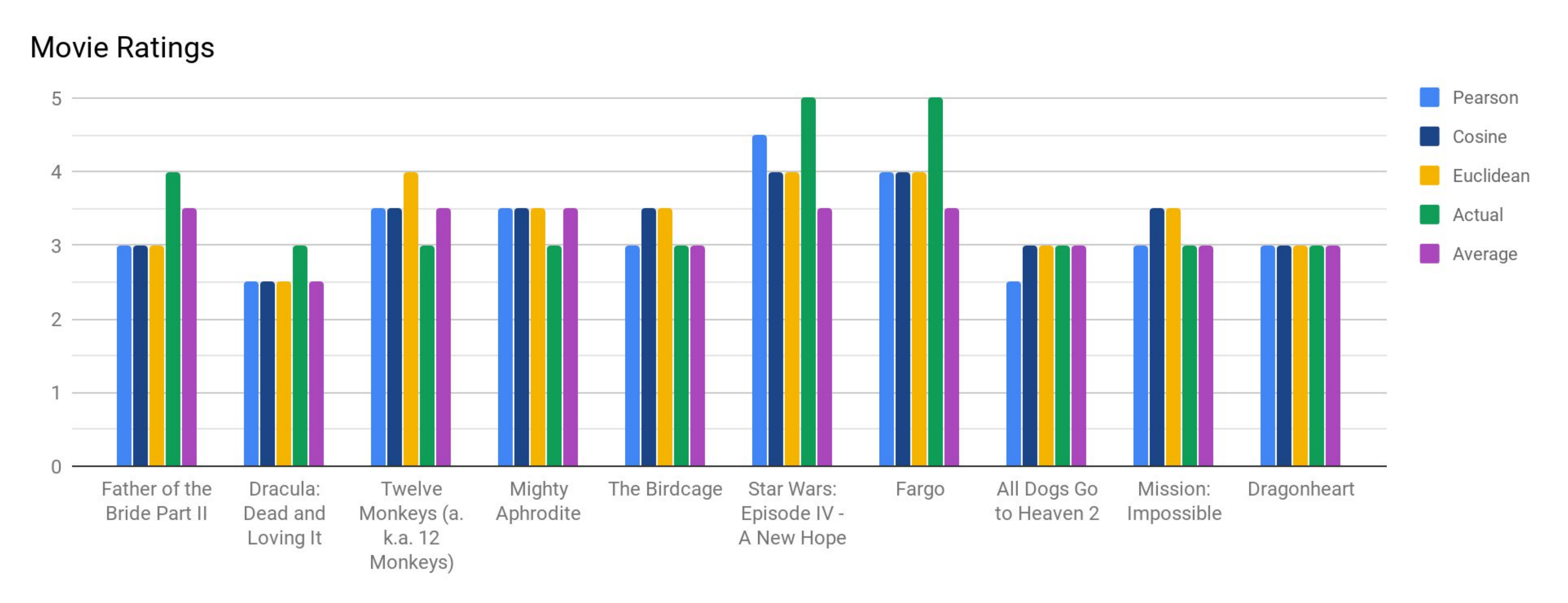
Results

- Ran a cross-validation test routine to calculate the Root Mean Square Error for each distance measure

Cosine RMSE:	Pearson RMSE:	Euclidean RMSE:
0.687	0.506	0.891

- Pearson Correlation gave us the best results
  - Our default for recommender system

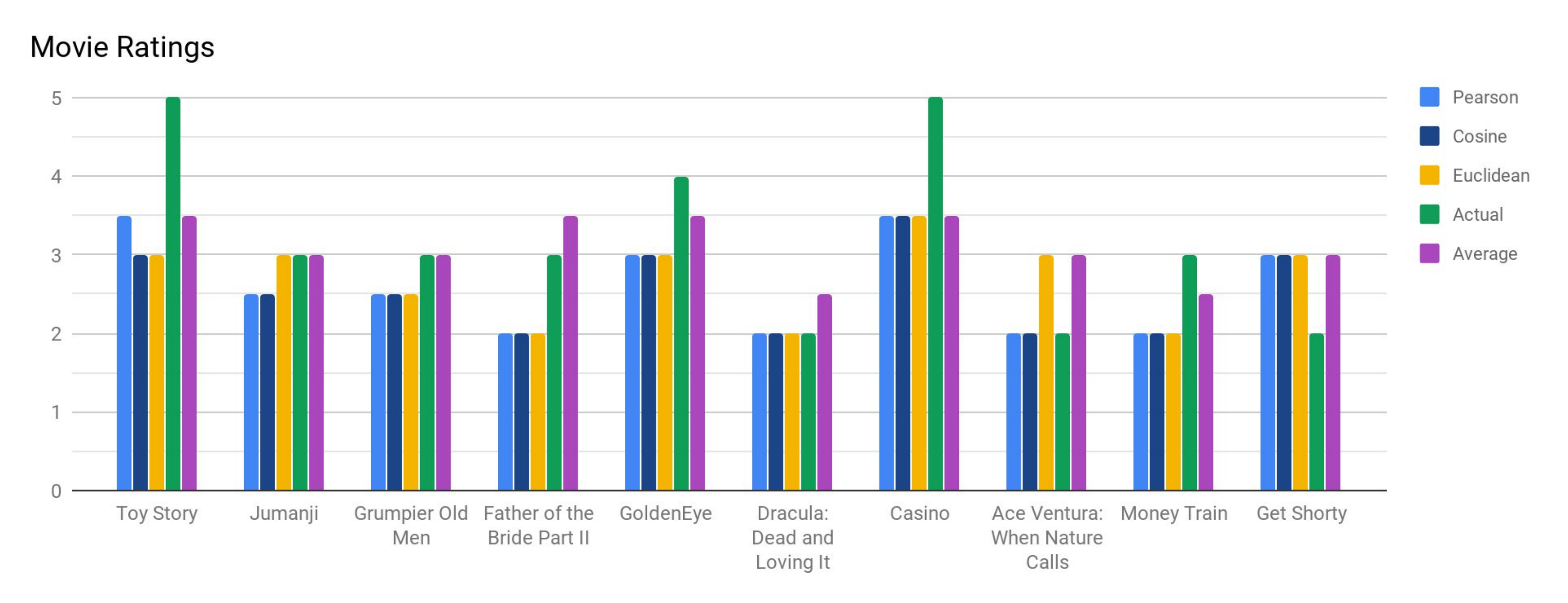
Results cont.



Rating Accuracy

Pearson: 87.8% Cosine: 85.2% Euclidean: 83.4%

Results cont.



Rating Accuracy

Pearson: 76.5% Cosine: 75.5% Euclidean: 72.2%

Thank You!