Capstone 3 : Movie Recommendation System

Jennifer Markham

Problem Identification

- In today's world, users have a plethora of choices in terms of what content to watch and where - with streaming services, Youtube, standard cable/dish TV- it can be overwhelming

- Users need a simple tool that can help guide them in terms of which content they would be most likely to enjoy to help sift through everything that is available

- The goal is to leverage 3 datasets from MovieLens that contain a variety of movies with plot overview, genre information, and user ratings to create a recommendation system for users to determine what to watch

Recommendations/Key Findings

- Item-based and user-based recommendation approaches both have merit, however user-based recommendations can create more unique recommendations

- K-Nearest Neighbors Regression and SVD/Matrix Factorization methods to predict movie ratings for a user based on ratings of other users were trialled compared to actual ratings

- SVD/Matrix Factorization addresses sparsity issue of original user ratings dataframe & produced user rating predictions closest to actual ratings of users
    - K-Nearest Neighbors Regressor vs Actual Ratings : RMSE = 2.0
    - SVD/Matrix Factorization vs Actual Ratings : RMSE = 1.898

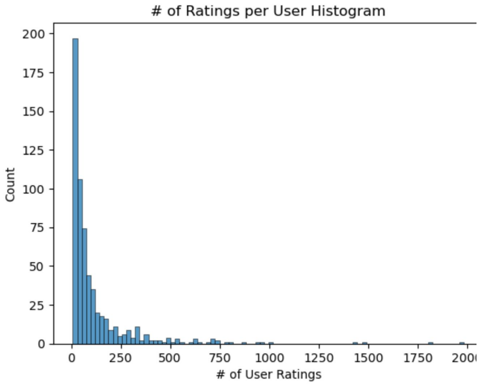
Model Results/Analysis

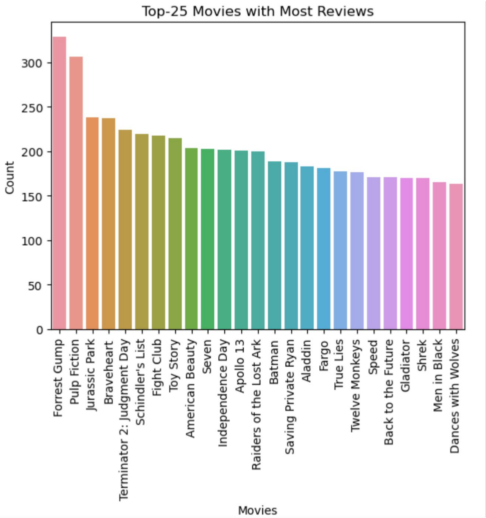
- Exploratory data analysis was performed on user ratings data

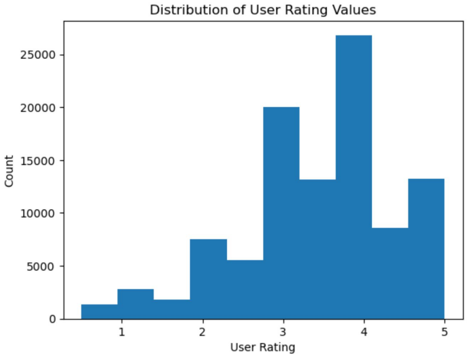
- Histograms were used to investigate:
  - Number of movies reviewed per user
  - Top-25 movies with the highest number of ratings
  - Distribution of user rating values

- Most users reviewed between 0-250 movies and gave them ratings between 3-5

- The most reviewed movies in the dataset included Forrest Gump, Pulp Fiction, Jurassic Park, Braveheart, and Terminator 2: Judgement Day







Model Results/Analysis

- Concept of sparsity is important for quantifying the amount of data in a matrix

- It will be difficult to compare ratings other users had for the same movie if no other users have rated this movie

- Sparsity/density was calculated for the user ratings of various movies dataframe, values are listed below
    - Sparsity: 0.98258
    - Density: 0.01742

Sparsity = (Empty Values)/(Total Cells)

Density = 1 - Sparsity

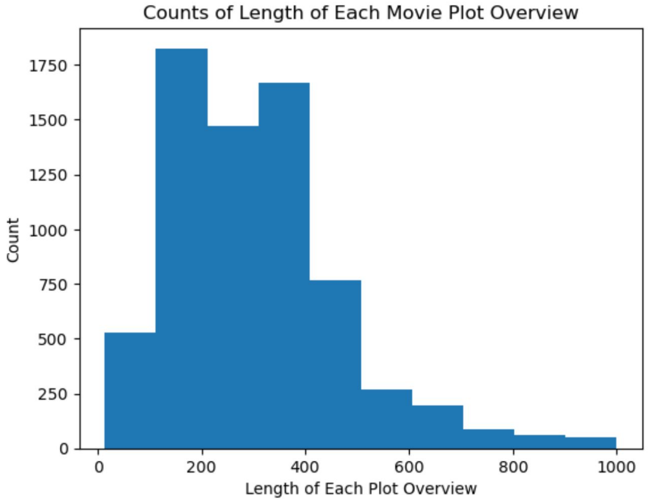
Model Results/Analysis

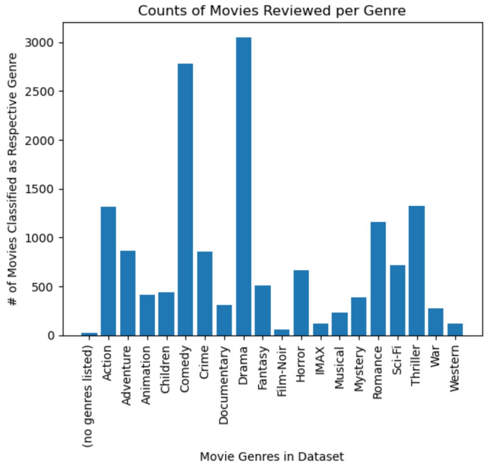
- Exploratory data analysis was also performed on the movie plot overview & genres data

- Histograms were constructed to show:
  - Distributions of movie plot overview length data
  - Distributions of genres of movies included

- Movie plot overviews were typically between 200-400 characters

- Movies of classified as comedies and dramas were most common in the dataset

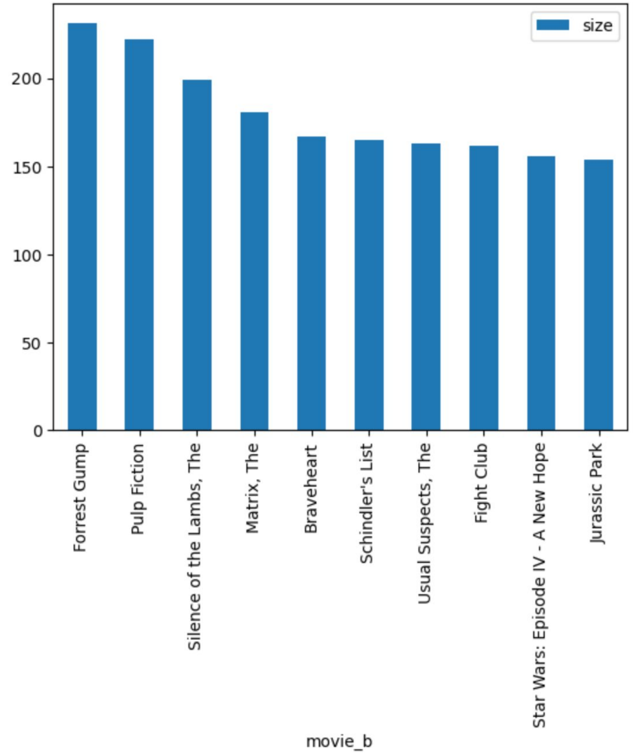




Model Results/Analysis

- Non-Personalized Recommendation Methods
    - Movies with Greatest Number of Reviews
    - Movies with Highest Average Rating
    - Movies with Greater than 50 Reviews with Highest Rating
    - Movies Commonly Viewed Together by the Same User

- These methods can yield useful results however recommendations derived from movie or user rating attributes are typically more powerful

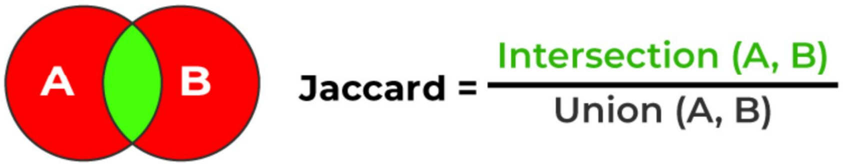


Top-10 Most Commonly Viewed Movies with The Shawshank Redemption

Model Results/Analysis

- Recommendations Based on Movie Attributes
  - Jaccard Similarity Scores Based on Movie Genres
  - Term frequency inverse document frequency & cosine similarity on Movie Plot Overviews

- These methods produce more tailored results than generalized methods use previously however, user-based recommendations can generate more unique results

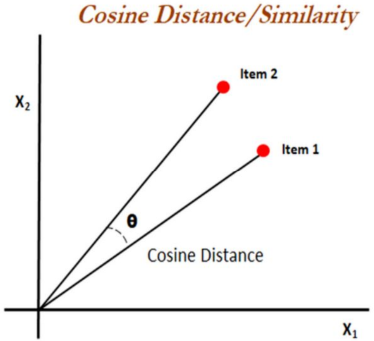


TFIDF

For a term i in document j:

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

tf_{ij} = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents



Model Results/Analysis

- Recommendations Based on User-Preferences
    - Term frequency inverse document frequency to Generate User Profiles Based on Movie Plot Overviews
        - Determine words associated with plot overviews of movies reviewed by each user
    - Cosine Similarities of User Review Data
    - K-Nearest Neighbors Regression to Predict User's Movie Ratings
    - Singular Vector Decomposition/Matrix Factorization to Predict User's Movie Ratings

- User-Preference based recommendations tend to generate more unique recommendations than item-based recommendations

- SVD/Matrix Factorization method yielded predicted user ratings closest to actual ratings using RMSE metric

Summary/Conclusion

- Numerous methods can be used to generate recommendation systems

- User-based recommendations are preferred as they typically generate more unique recommendations than item-based recommendations

- SVD/Matrix Factorization Method predicting user movie ratings based on other user's rating info yielded rating predictions closest to actual rating values
  - K-Nearest Neighbors Regressor vs Actual Ratings : RMSE = 2.0
  - SVD/Matrix Factorization vs Actual Ratings : RMSE = 1.898