ReelRx -Movie Recommendations

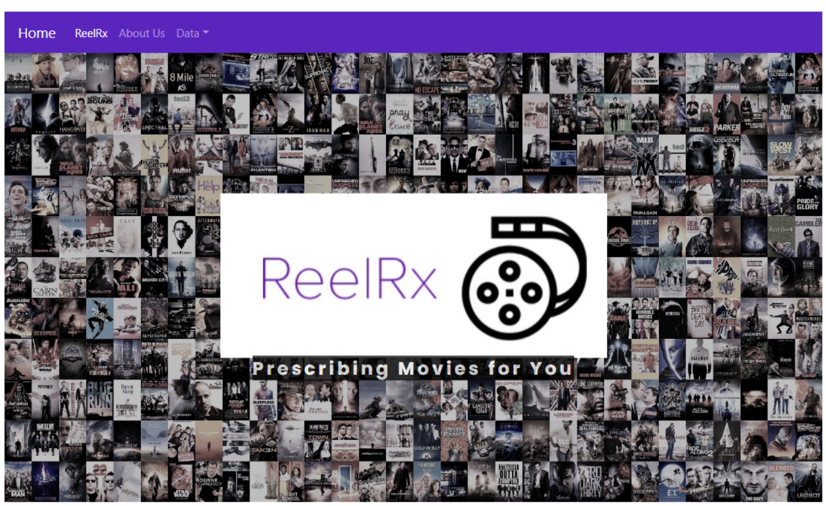
Group Members: Hoa Roach, Jammy Lo, Tim Schurmann, Tiffany Burns

1. Project Description/Outline:

There are so many movies to watch and we live in a world where we are increasingly busy, so we need recommendations and suggestions on what to watch. The purpose of this project is to create a better movie recommendation system from the MovieLens dataset. We developed a Flask web app that recommends to users movies to watch based on the user's favorite movies, using machine learning powered HTML/CSS and JavaScript. We also designed interactive Tableau Dashboard visualizations to provide the ability to explore the dataset's users' information as well as the dataset's movies, so users can draw actionable insights for their own recommendations. A clean dataset is uploaded to our website for users to explore with filter tools.

2. Site Design & Inspiration

Inspired by how Netflix, Amazon Prime and other movie websites recommend users TV shows and movies to Watch, but acutely aware that sometimes the recommendations are not relevant, our team built our own model for better movie recommendations. The web app is **ReelRx** – Prescribing Movies for You. **ReelRx** was designed to invite users to interact with it to explore their recommended movies and explore the universe of recommendable movies.



The color set of the app was derived by referencing the color palettes of top movie streaming websites: Netflix, Hulu, Amazon Prime, and Disney. Then we determined what primary/secondary/tertiary color palette was absent as well as identifying common traits of each website (white text over primary or secondary color, etc.). The end result is a clean, industry recognizable web application.

3. Data & Modeling Approach:

3.1. Data

This dataset contains a set of movie ratings from the MovieLens website, a movie recommendation service. This dataset was collected and maintained by GroupLens, a research group at the University of Minnesota. There are 5 versions included: "25m", "latest-small", "100k", "1m", "20m". In all datasets, the movies data and ratings data are joined on "movieId". The 25m dataset, latest-small dataset, and 20m dataset contain only movie data and rating data. The 1m dataset and 100k dataset contain demographic data in addition to movie and rating data.

For this reason, we chose to use the "1M" dataset as it is the largest MovieLens dataset that contains demographic data. We wanted more user data to be able to give more specific movie recommendations.

The dataset contains 1 million ratings of users to more than 3 thousand movies. This and other GroupLens data sets are publicly available for download at: https://grouplens.org/datasets/movielens/latest/

3.1.1 Data Limitations

Notably, the dataset only includes movies from the year 2000 and before. Additionally, the dataset mostly contains men in the age between 25 to 35 so the finding could be biased due to the limitation of our data.

3.1.2 Datasets

The raw data includes 3 tables of users, movies and ratings. We joined and cleaned the data using Python for our analysis and web app developing.

The backbone of **ReelRx** is a movie / user ratings matrix:

```
final_dataset = ratings.pivot(index='movie_id',columns='user_id',values='rating')
```

```
final_dataset.head()
```

user_id	1	2	3	4	5	6	7	8	9	10	...	6031	6032	6033	6034	6035	6036	6037	6038	6039	6040
movie_id																					
1	5.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	5.0	5.0	...	0.0	4.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	3.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0	...	0.0	0.0	0.0	0.0	2.0	2.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0

5 rows × 6040 columns

3.2. Movie Recommendation App **ReelRx** :

We built our recommendation app **ReelRx** using a Collaborative Filtering approach. The model is built from a user's current behavior (inputs) and similar decisions made by many other users and then predicts movies that the user may have an interest in using a cosine similarity k-nearest neighbors algorithm (k-NN).

3.2.1 Obscurity Filter:

One goal of **ReelRx** was to provide two forms of movie recommendations: “Popular” and “obscure” recommendations, with popularity being determined by number of ratings a movie has (more ratings means more popular).

An analysis was done to derive a threshold of 20th percentile to distinguish between obscure and not obscure movies. On this dataset, this means any movie with between 1 and 23 total ratings is considered “obscure.”

```
no_user_voted = ratings.groupby('movie_id')['rating'].agg('count')
```

```
no_user_voted.describe()
```

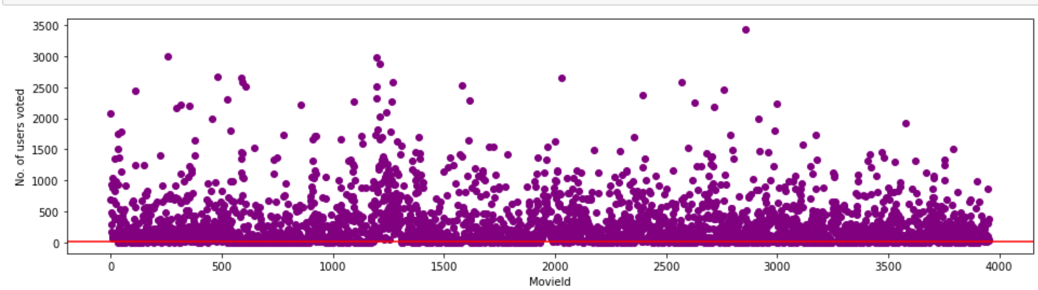
```
count    3706.000000
mean      269.889099
std       384.047838
min         1.000000
25%        33.000000
50%       123.500000
75%       350.000000
max      3428.000000
Name: rating, dtype: float64
```

```
ptile = 20
p = np.percentile(no_user_voted,ptile)
print(p)
print(len(no_user_voted)*(ptile/100))
```

```
23.0
741.2
```

```
f,ax = plt.subplots(1,1,figsize=(16,4))

plt.scatter(no_user_voted.index,no_user_voted,color="purple")
plt.axhline(y=23,color='r')
plt.xlabel('MovieId')
plt.ylabel('No. of users voted')
plt.show()
```

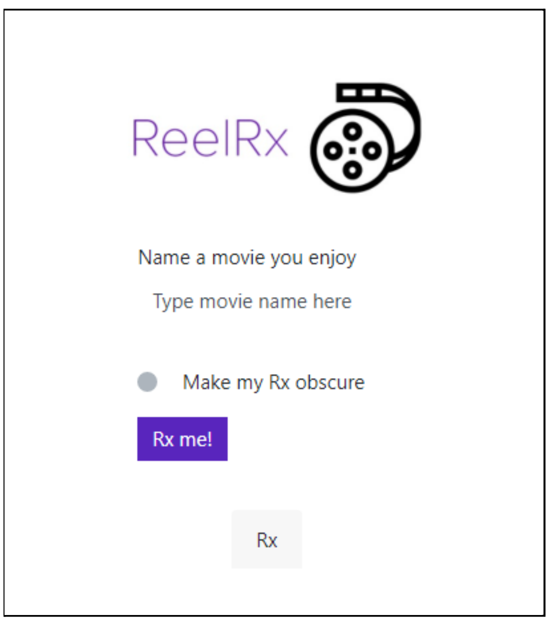


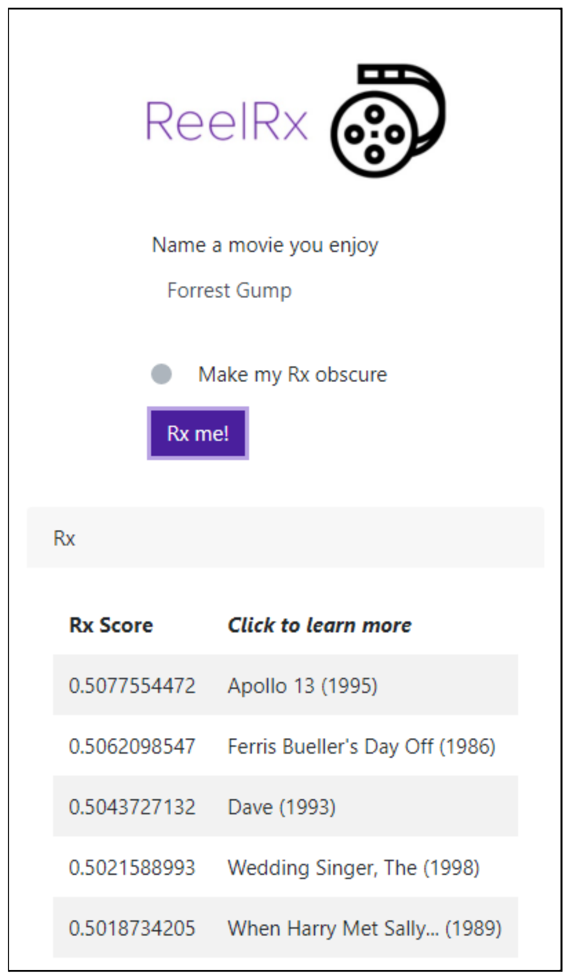
3.2.2 Machine Learning app in action:

User-friendly app interface is designed in the **ReelRx** tab of our website. Once a dataset is selected (obscure or not obscure) and a movie is entered, **ReelRx** applies a cosine similarity nearest neighbors model.

Cosine similarity is a measure of directional or angular similarity, as opposed to a measure of positional distance. This method gives qualitative preference to movies rated rather than quantitative preference to numbers of ratings applied.

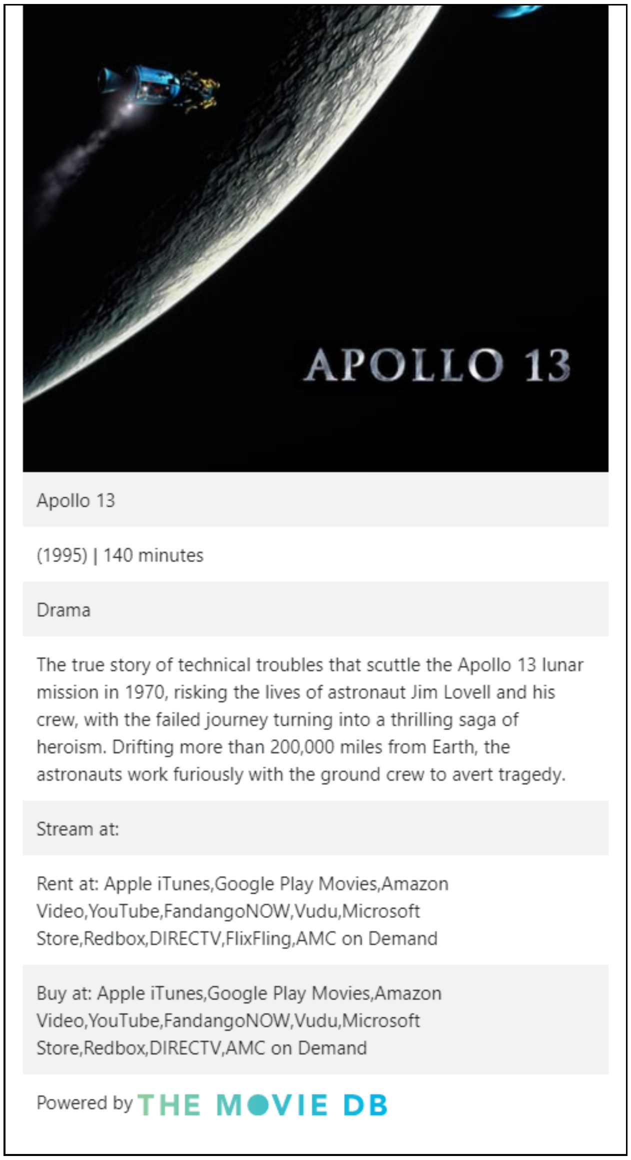
The app will return a top 5 recommended movies list, and will prescribe an "**Rx** Score" to each movie. We successfully added the suggested movie name as a dropdown so users don't need to type full movie titles. The detailed information and poster of the recommended movies will pop up when the user clicks on a recommended movie's title.





3.2.3 The Movie DB API:

Once a user has a recommendation list, they are able to click a movie title and have an API pull relevant information from The Movie DB.



ReelRx makes three separate API requests:

First, it makes a search request to find the tmbd id for the movie the user has clicked.

Next, it makes a detail request using the tmdb id to get the details about the selected movie.

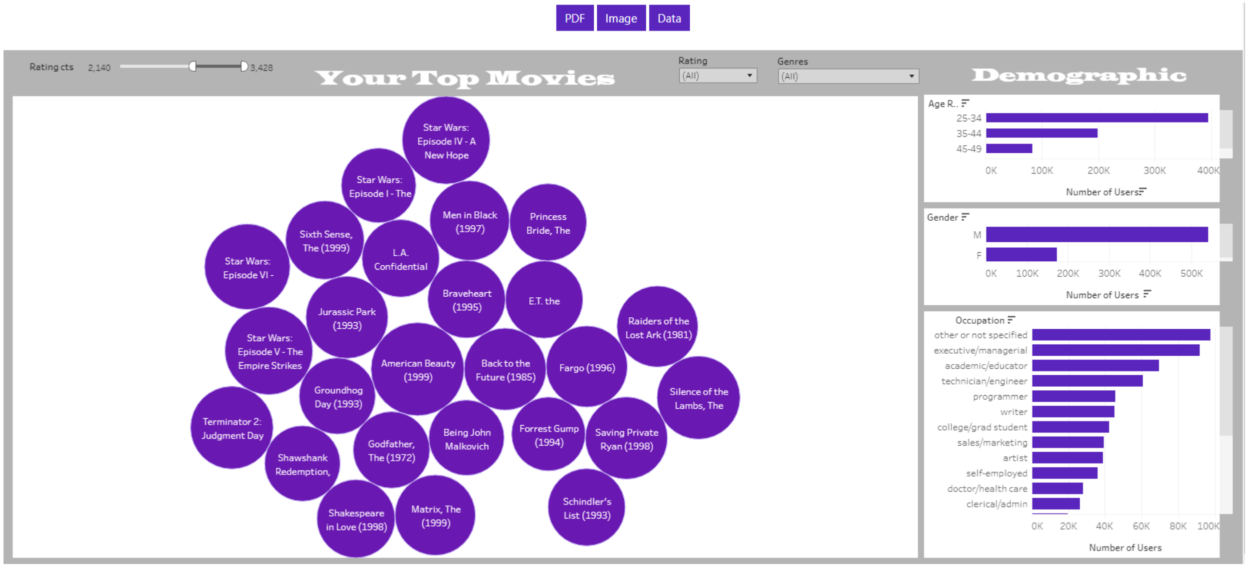
Finally, it once again uses the tmdb id to make a providers request to get information on where to stream, rent, or buy the selected movie.

Within each step of the API requests, error handling and cleaning occurs.

3.3. Data Visualization:

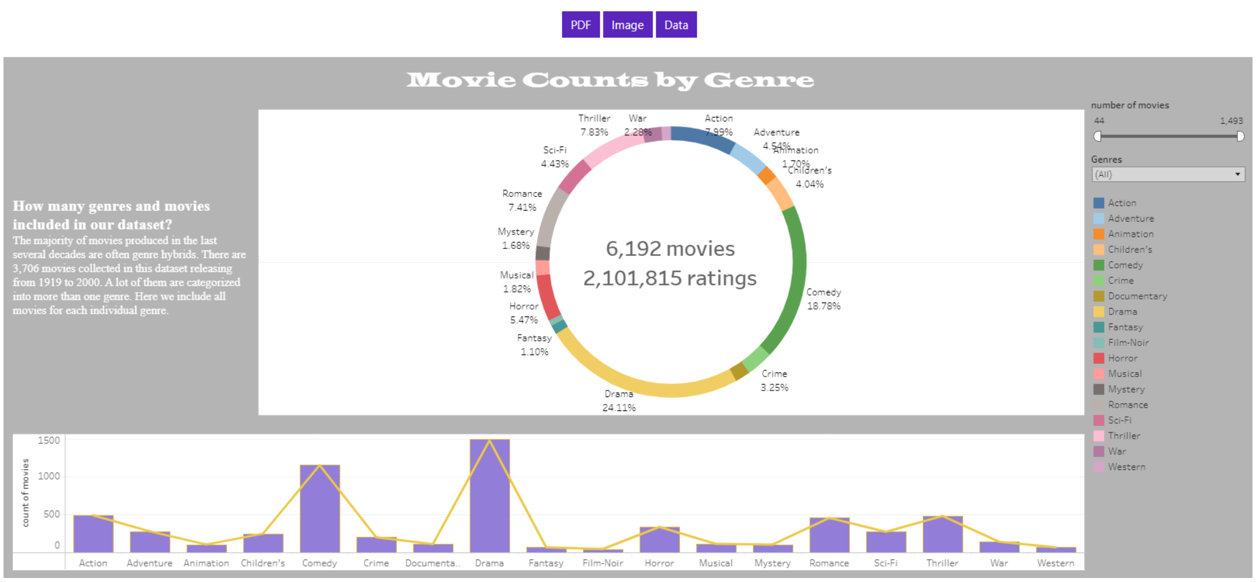
3.3.1 User Visualization

Two Tableau dashboards were built to aid in user exploration of data. The first dashboard is visualization of top films by demographic. Because the dataset is skewed by the user base, this would allow users to filter by gender, age range, and occupation.



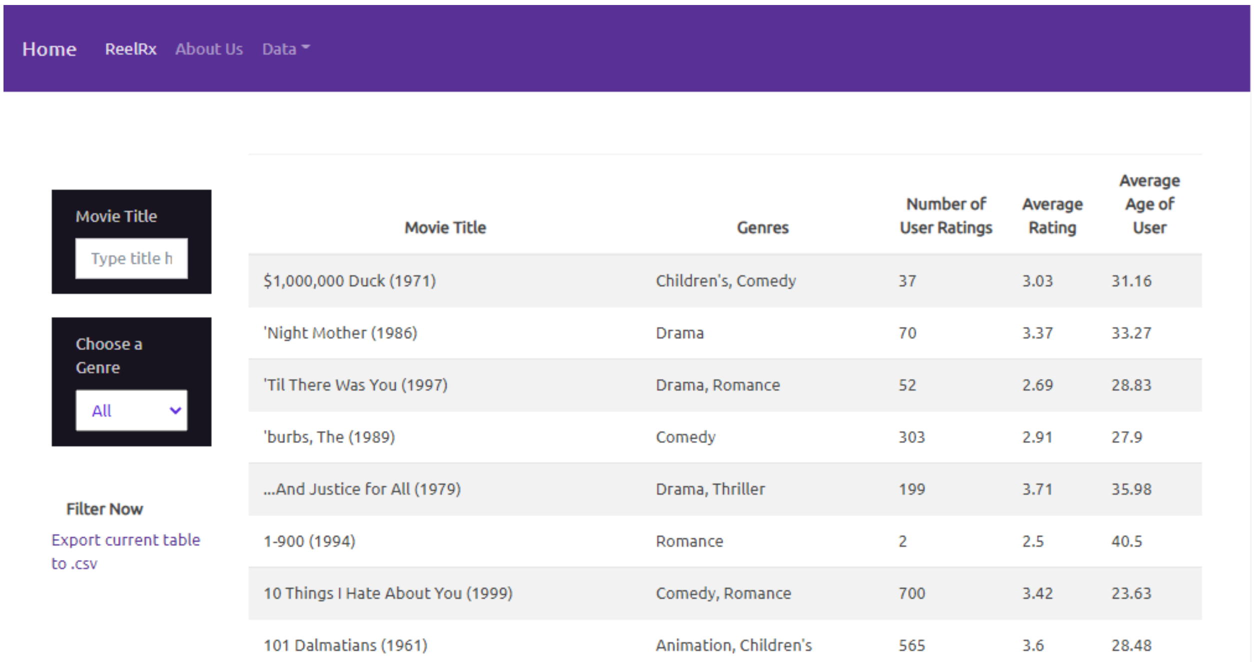
3.3.2 Genre Visualization

Second dashboard is inspired by one famous online banking's reward activity visualization. It aims to optimize ease of use while offering functional features and fun information regarding movie genres that interest any age group of user.



3.4. Data Tables:

We used D3.js and JQuery to load data tables to our website. A filtered table was created to allow users the opportunity to explore the raw dataset. Users can type movie title or a part of movie title to look up the movie's information and its rating such as genres of the movie, number of users rated for that movie, average rating, and average age of users. The genre drop down allows users to filter movies based on its genres.



Movie Title	Genres	Number of User Ratings	Average Rating	Average Age of User
$1,000,000 Duck (1971)	Children's, Comedy	37	3.03	31.16
'Night Mother (1986)	Drama	70	3.37	33.27
'Til There Was You (1997)	Drama, Romance	52	2.69	28.83
'burbs, The (1989)	Comedy	303	2.91	27.9
...And Justice for All (1979)	Drama, Thriller	199	3.71	35.98
1-900 (1994)	Romance	2	2.5	40.5
10 Things I Hate About You (1999)	Comedy, Romance	700	3.42	23.63
101 Dalmatians (1961)	Animation, Children's	565	3.6	28.48

The data table can be downloaded from this page in the form of a csv file by clicking on the export button.

4. How ReelRx Changes the World:

ReelRx provides users a clean, quick new way to get watching movies faster. The ability to simply enter a movie and receive unique recommendations overcomes the daily problem of "option overload." Stack on top of that the ability to choose obscure recommendations, and every user will find the right viewing prescription on **ReelRx.**

4.1. Opportunities for Future Development and Growth

ReelRx has the capability to be an all in one movie recommendation app. There are several opportunities that could help improve the functionality and breadth of work. The data set is currently limited in terms of demographics, future work would require gathering a sample from a more widespread demographic. Additionally, movies within the app are currently limited to release dates of 2000 or earlier, to further the recommendation capability, it would need to connect to an API that would allow monthly releases of new movies. Finally, a polished version of the app would connect to an existing/create a streaming service that would allow users to create profiles and watch the movies that are recommended.

5. References:

https://grouplens.org/datasets/movielens/latest/

https://www.kaggle.com/johnwill225/movie-recommendations

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

http://themoviedb.org