AustinRecommendsMovies.com:

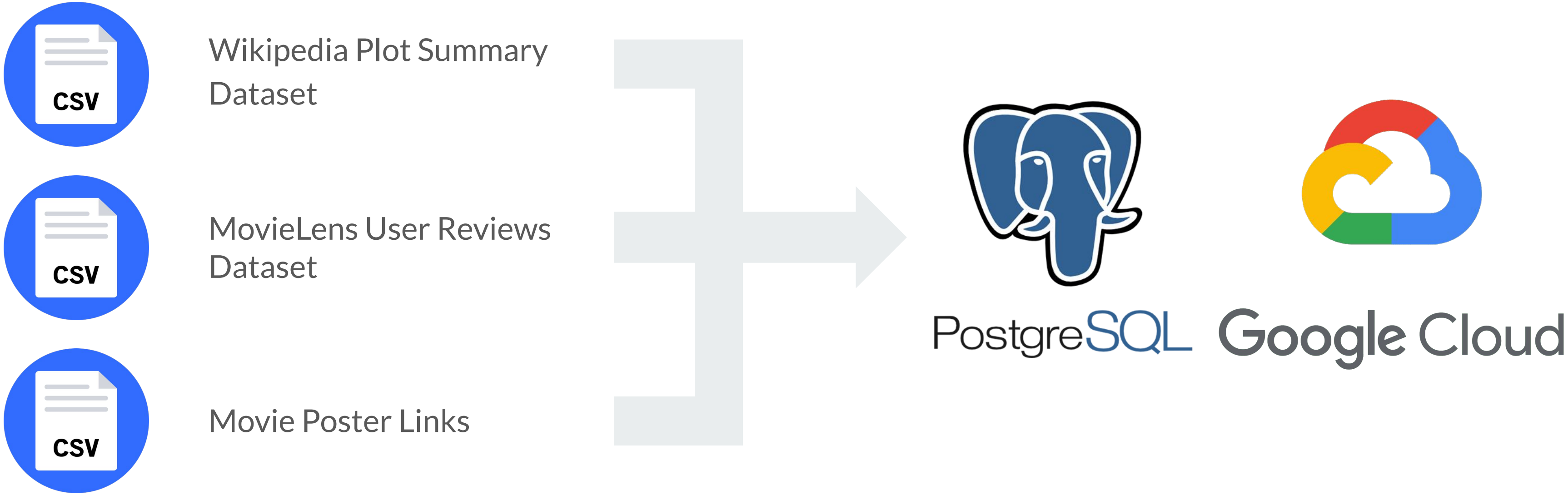
Recommending Movies with Data Science

Austin Poor

Goal: Building a movie recommendation website using film summaries and user ratings

Data

Data Sources



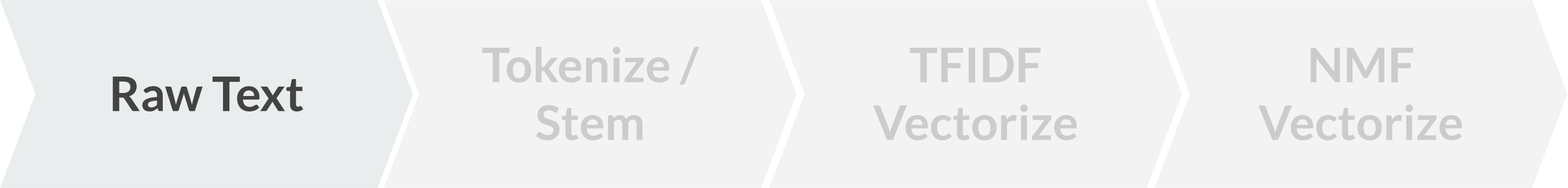
Wikipedia Plot Summary Dataset

MovieLens User Reviews Dataset

Movie Poster Links

Modeling

Model Building



Raw Text

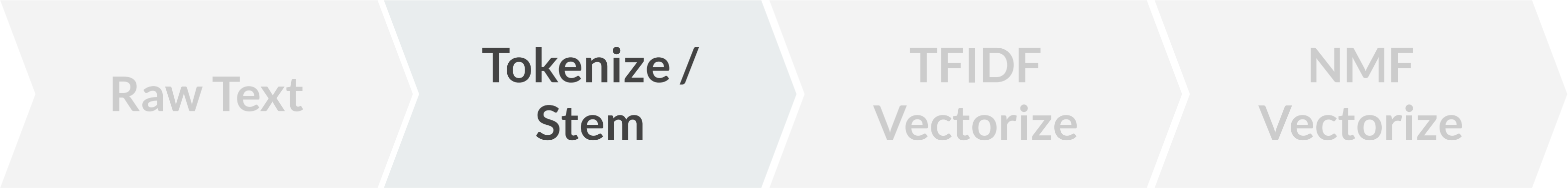
Tokenize / Stem

TFIDF Vectorize

NMF Vectorize

"Two years after his escape from France Jason Bourne and Marie Kreutz are living in Goa India Bourne continues to have flashbacks about his former life as a CIA assassin which he writes in a small diary Meanwhile in Berlin Germany CIA agents subordinate to Deputy Director Pamela Landy are paying..."

Model Building


Raw Text

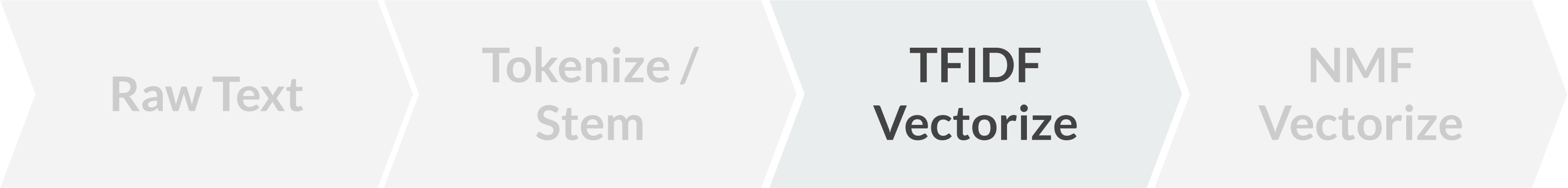
Tokenize / Stem

TFIDF Vectorize

NMF Vectorize

```
“['two', 'year', 'escap', 'franc', 'jason', 'bourn', 'mari',
'kreutz', 'live', 'goa', 'india', 'bourn', 'continu',
'flashback', 'former', 'cia', 'assassin', 'write', 'small',
'meanwhil', 'berlin', 'germani', 'cia', 'agent', 'subordin',
'deputi', 'director', 'pamela', 'landi', 'us', 'neski', 'file',
'document', 'theft', 'alloc', 'year', 'earlier', 'russian',
'feder', 'secur', 'servic',...]”
```

Model Building



Raw Text

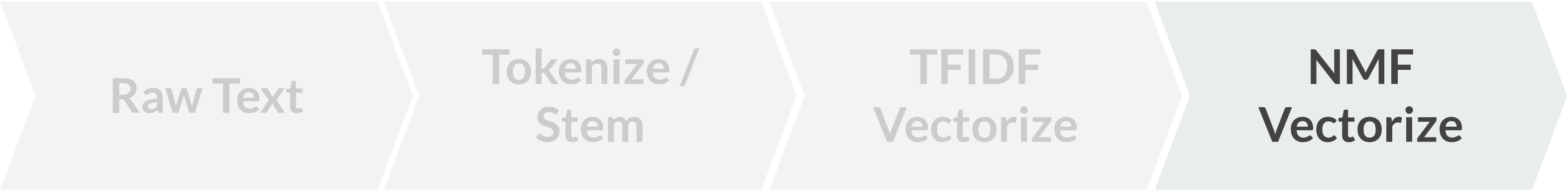
Tokenize / Stem

TFIDF Vectorize

NMF Vectorize

```
“[0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0.29752159 0.06511288 0. 0. 0. 0. 0. 0. 0.05109997 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.06792782 0. 0. 0. 0. 0. 0. 0.
0.05882215 0.05063095 0.03791007 0.04138681 0.13590716 0. 0.
0.05546180. 0.06569852 0. 0.03885992 0. 0. 0. 0. 0. 0. 0. 0. 0.
0.03796726 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. …]”
```

Model Building



Raw Text

Tokenize / Stem

TFIDF Vectorize

NMF Vectorize

```
[[0.02190351 0.         0.05002839 0.00091537 0.00067237 0.00050766
  0.         0.         0.         0.         0.         0.01514989
  0.03751799 0.00010761 0.01271533 0.         0.01349312 0.00531679
  0.         0.00809553]]
```

MVP

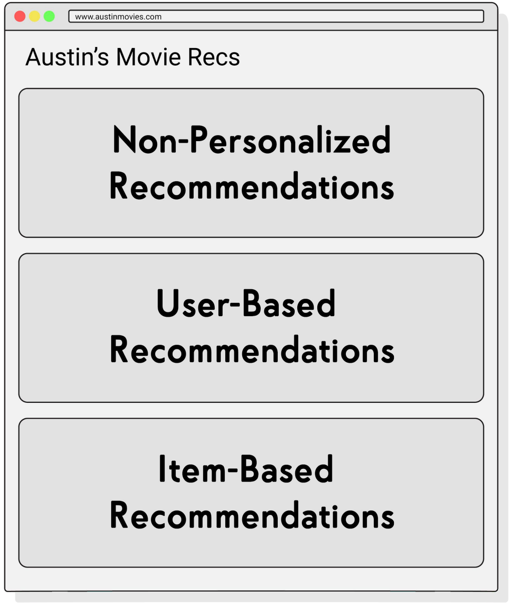
MVP App Features

Web app built with Flask, Bokeh, and Bootstrap

Hosted on GCP

Three recommendation sections:

- Non-personalized recs
- User-based recs
- Item-based recs



Austin's Movie Recs

Non-Personalized Recommendations

User-Based Recommendations

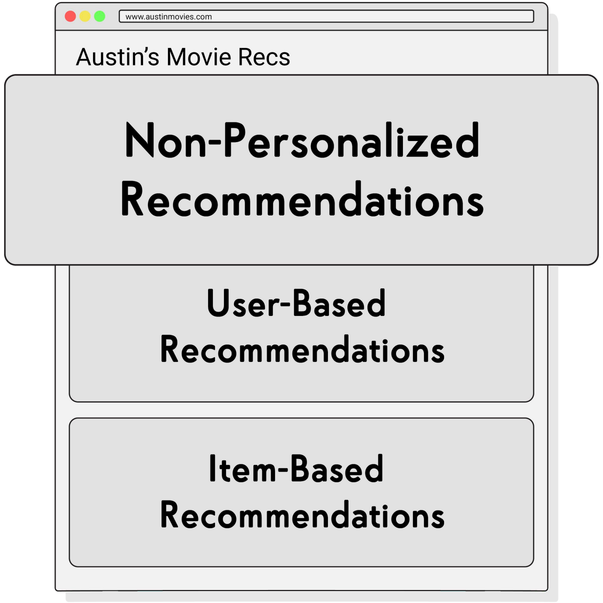
Item-Based Recommendations

Non-Personalized Recs.

What movies are currently popular?

Doesn’t take user preference into account

Score based on number of reviews, average rating, and age of the film

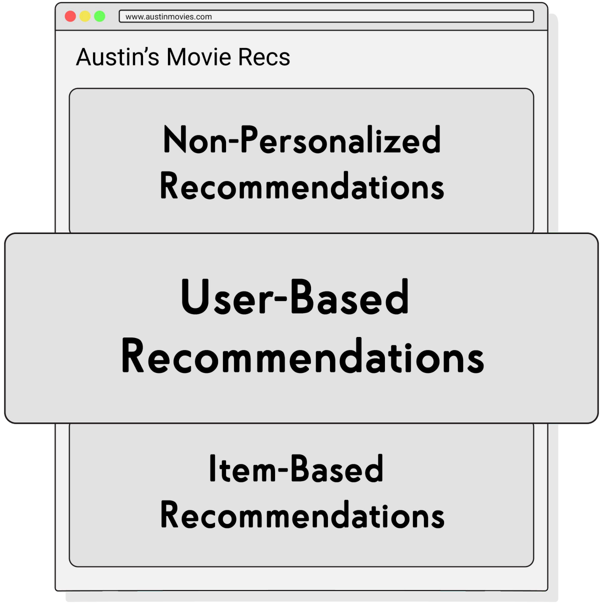

www.austinmovies.com
Austin’s Movie Recs
Non-Personalized Recommendations
User-Based Recommendations
Item-Based Recommendations

User-Based Recs.

Collaborative filtering

Steps:

1. Find similar users using Jaccard Similarity
2. Those users “vote” for candidate recommendations
3. Filter the top n suggestions

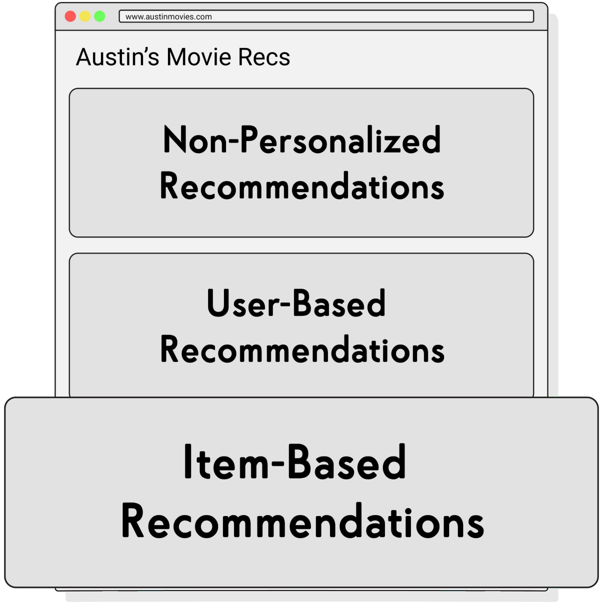

www.austinmovies.com
Austin’s Movie Recs
Non-Personalized Recommendations
User-Based Recommendations
Item-Based Recommendations

Item-Based Recs.

Content-based filtering

Steps:

1. Select some of user’s top rated movies
2. Rank un-seen films by their NMF vectors’ (Euclidian) distances
3. Filter the top n suggestions



Austin’s Movie Recs

Non-Personalized Recommendations

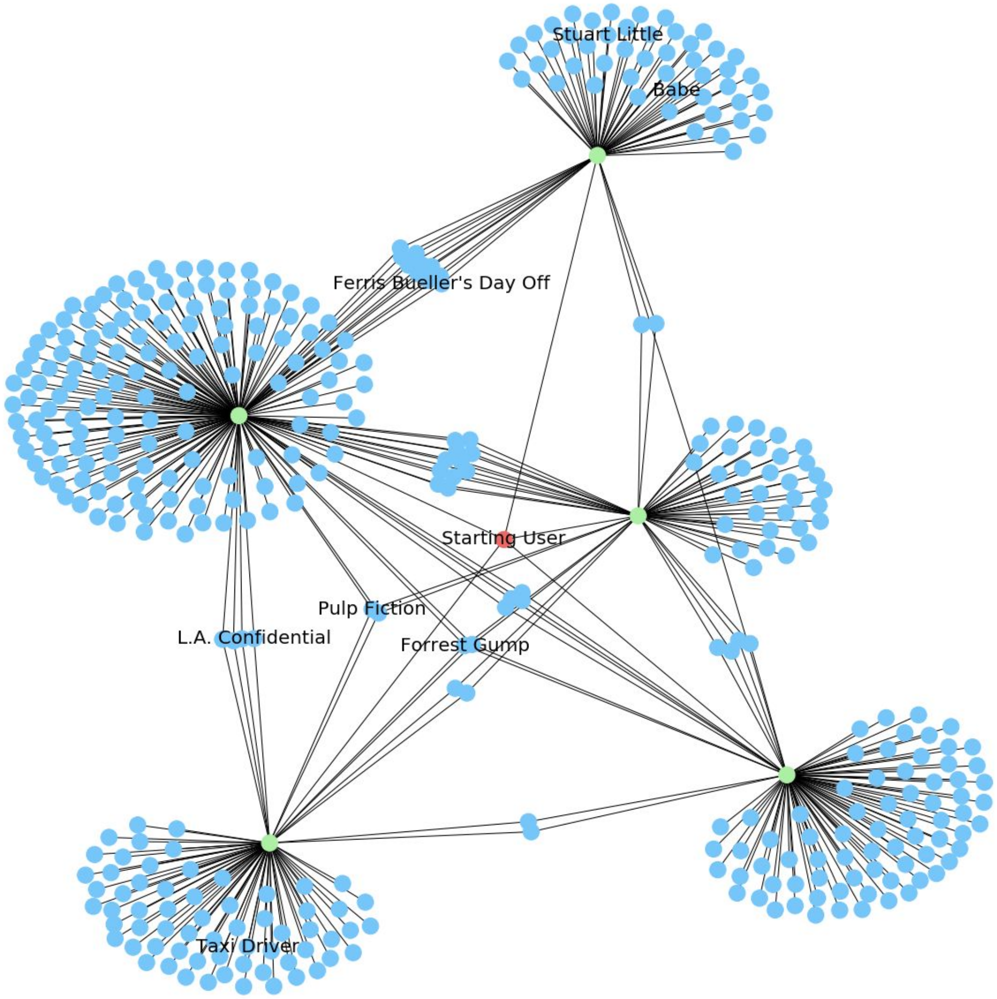
User-Based Recommendations

Item-Based Recommendations

Collaborative Filtering Example

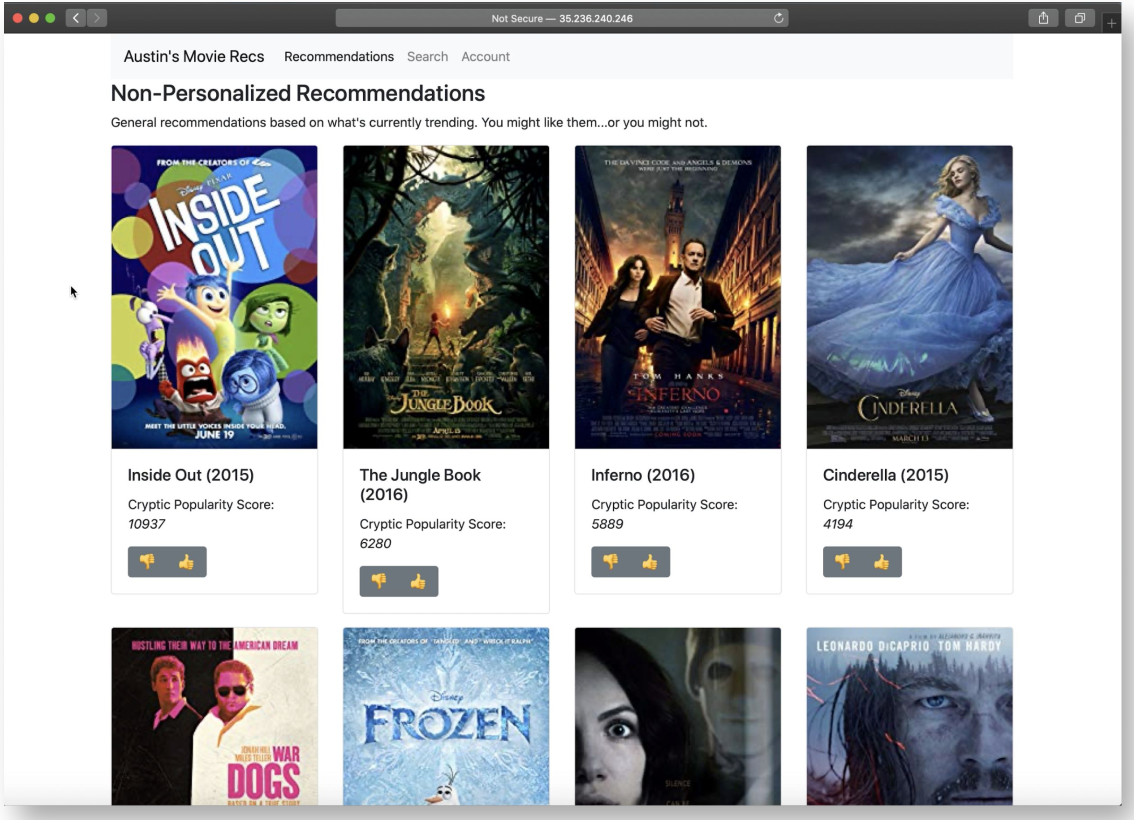
- 🔴 Target User
- 🟢 Similar Users
  (5 similar to users)
- 🔵 Movies (rated 5.0)

Movies rated highly by similar users would make good recommendations



App Demo

App Demo



Austin's Movie Recs Recommendations Search Account

Non-Personalized Recommendations

General recommendations based on what's currently trending. You might like them...or you might not.

Inside Out (2015)

Cryptic Popularity Score: *10937*

👎 👍

The Jungle Book (2016)

Cryptic Popularity Score: *6280*

👎 👍

Inferno (2016)

Cryptic Popularity Score: *5889*

👎 👍

Cinderella (2015)

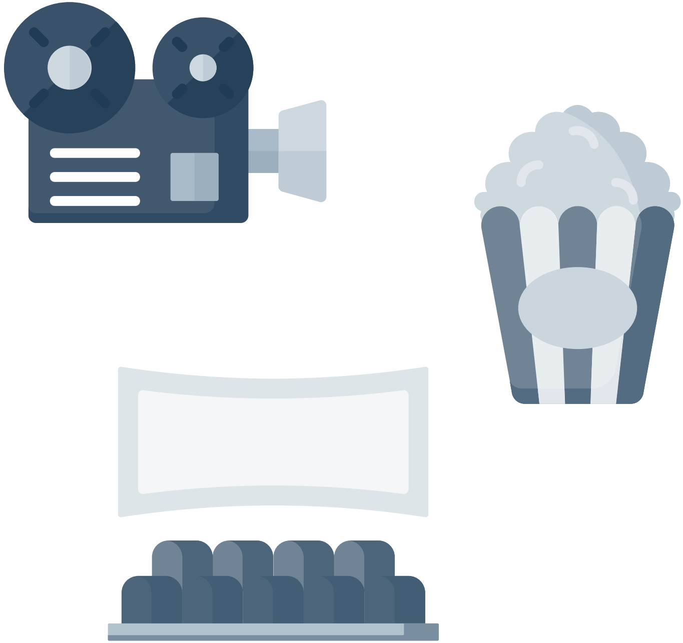
Cryptic Popularity Score: *4194*

👎 👍

[Alt Link]

Future Work

- Update the datasets (add more recent movies)
- Finish implementing the search feature
- Add a feature for users to sign-up and sign-in
- Optimize search queries



Thank you

Appendix

Score Formula

$$score = \frac{n\_votes \times ln(avg\_rating+1))}{(2017-movie\_year)^5}$$