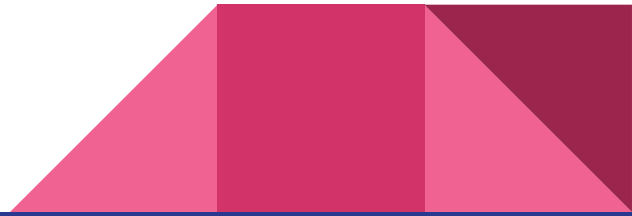
MovieMatch: Kanopy's Recommender System

Author: Chris Kucewicz

Introduction and Contents

1. Business Understanding
2. Data Understanding
3. Data Preparation
4. Modeling
5. Evaluation
6. Limitations
7. Recommendations
8. Next Steps



Business Understanding

Background | Goals | Success Criteria

- Kanopy App: streaming platform
  - Competing with Netflix, Hulu, HBO Max
- Looking to implement recommendation system

kanopy
THOUGHTFUL ENTERTAINMENT

Business Understanding

Background | **Goals** | Success Criteria

- Develop a recommendation system for Kanopy
  - Top 5 movies

kanopy
THOUGHTFUL ENTERTAINMENT

Business Understanding

Background | Goals | **Success Criteria**

- Root Mean Squared Error (RMSE)
  - Average distance between predictions & actual ratings

	Actual Rating	Prediction	Distance
User 1 Movie A	2.0	4.0	2

Business Understanding

Background | Goals | **Success Criteria**

- Root Mean Squared Error (RMSE)
  - Average distance between predictions & actual ratings

	Actual Rating	Prediction	Distance
User 1 Movie A	2.0	4.0	2
User 2 Movie A	5.0	1.0	4

Mean Distance = 3

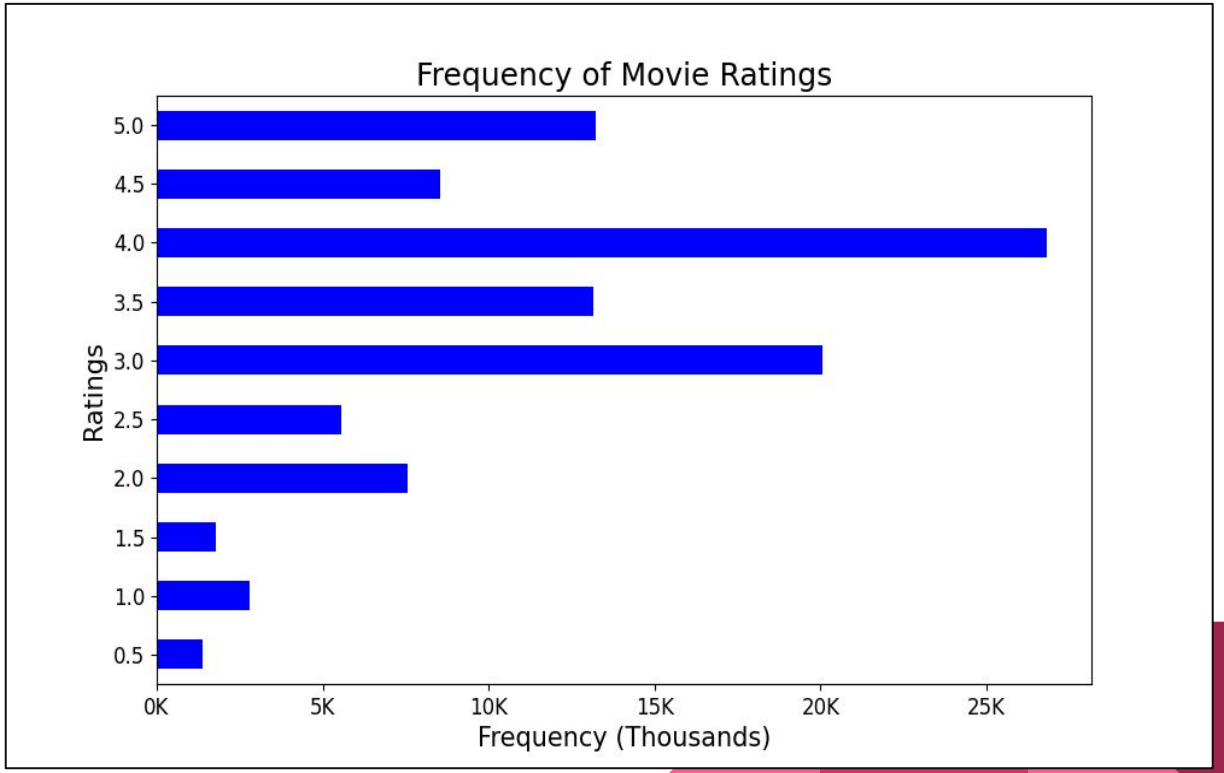
RMSE = 3.16

Data Understanding

MovieLens Database:

- `movies.csv`:
  - Over ***9k*** unique movies

- `ratings.csv`:
  - Over ***100k*** user ratings
  - Rating scale: 0.5-5.0

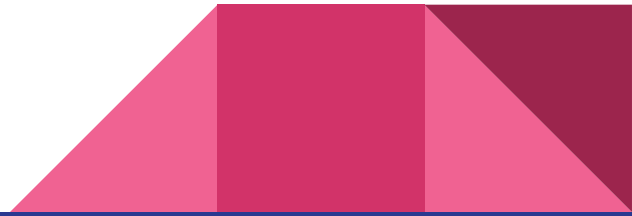


Data Preparation

- Minimal steps
- Dropped `timestamp` feature
- `ratings` dataset ready for modeling

Modeling

- Collaborative Filtering (CF)



Modeling

User A

Likes:

- Toy Story
- The Grinch

User B

Likes:

- Toy Story
- The Grinch
- Star Wars

Dislikes:

- Little Mermaid

Which movie should we recommend to User A?

- Star Wars or Little Mermaid?

Modeling

User A

Likes:

- Toy Story
- The Grinch

User B

Likes:

- Toy Story
- The Grinch
- Star Wars

Dislikes:

- Little Mermaid

Which movie should we recommend to User A?

- Star Wars or Little Mermaid?

Modeling

- Collaborative Filtering (CF)
  - Key Idea: users with *similar* preferences tend to like *similar* items

User A

Likes:

- Toy Story
- The Grinch
- Star Wars

User B

Likes:

- Toy Story
- The Grinch
- Star Wars

Modeling

- Collaborative Filtering (CF)
- *Surprise* library
  - Common for Recommendation Systems
- Baseline models for comparison
- Multiple iterations of model tuning

	Model	RMSE
0	NormalPredictor	1.4239
1	BaselineOnly	0.8785
2	SVD_basic	0.8808
3	KNNBasic (Cosine)	0.9823
4	KNNBasic (Pearson)	0.9829
5	SVD_GS1	0.8632
6	SVD_GS2	0.8571
7	SVD_best	0.8559

Model Evaluation

- Best model: Singular Value Decomposition (SVD)
  - RMSE: ***0.86***
- Strengths:
  - Identifying patterns in sparse datasets
  - Scalability
- Weaknesses:
  - Struggles with Cold Start Problem

	Model	RMSE
0	NormalPredictor	1.4239
1	BaselineOnly	0.8785
2	SVD_basic	0.8808
3	KNNBasic (Cosine)	0.9823
4	KNNBasic (Pearson)	0.9829
5	SVD_GS1	0.8632
6	SVD_GS2	0.8571
7	SVD_best	0.8559

Model Evaluation: Addressing Cold Start Problem

- What movie should we recommend to User A?

User C

Likes:

- ??
- ??

User D

Likes:

- ??
- ??

Model Evaluation: Addressing Cold Start Problem

- Cold Start Problem:
  - Limited or no data

User C

Likes:

- ??
- ??

User D

Likes:

- ??
- ??

Model Evaluation: Addressing Cold Start Problem

1. Ratings Function:
    a. User rates 5 movies
2. Recommendation system function:
    a. User rates 5 movies
    b. User selects genre
    c. System provides 5 movie recommendations

Model Evaluation: Addressing Cold Start Problem

In [57]:

```
# testing out function
new_user_genre_recommend(model=svd_basic, df=cleaned_ratings, movies_df=movies, n=5)
```

```
Please enter your user ID: 100000000
It looks like you're a new user!
To provide better recommendations, we'll ask you to rate 5 movies.
Enter a genre you like (e.g., 'Action', 'Comedy', 'Drama'): Drama
You've selected the genre: Drama
Please rate 5 movies from the Drama genre (1-5), or type 'N/A' if you haven't seen the movi
e.
Rate the movie 'Waiting to Exhale (1995)' (1-5) or 'N/A' if you haven't seen it: 1
Rate the movie 'American President, The (1995)' (1-5) or 'N/A' if you haven't seen it: -1.5
Invalid rating. Please enter a number between 1 and 5 or 'N/A'.
Rate the movie 'American President, The (1995)' (1-5) or 'N/A' if you haven't seen it: 1.5
Rate the movie 'Nixon (1995)' (1-5) or 'N/A' if you haven't seen it: 5
Rate the movie 'Casino (1995)' (1-5) or 'N/A' if you haven't seen it: n/a
Rate the movie 'Sense and Sensibility (1995)' (1-5) or 'N/A' if you haven't seen it: 4

Here are your top movie recommendations based on your ratings:
Recommendation #1: Shawshank Redemption, The (1994) (Predicted Rating: 4.37)

Recommendation #2: Rear Window (1954) (Predicted Rating: 4.35)

Recommendation #3: Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb (19
64) (Predicted Rating: 4.34)

Recommendation #4: Lawrence of Arabia (1962) (Predicted Rating: 4.31)

Recommendation #5: Cinema Paradiso (Nuovo cinema Paradiso) (1989) (Predicted Rating: 4.28)
```

User supplying ratings

Recommendations

Limitations

- **Treating as Regression (Metric: RMSE)**
  - Ratings are ordinal data
- **Cold Start Problem**
  - Collaborative filtering weakness
- **Time Constraints**
  - Takes time to build out user ratings database

Recommendations

Recommendation 1:
Begin ratings collection

Recommendation 2:

Recommendation 3:

Use `user_rating_collection()` function to gather user ratings

Recommendations

Recommendation 1:
Begin ratings collection

Use `user_rating_collection()` function to gather user ratings

Recommendation 2:
Model Deployment

Implement recommender system: `new_user_genre_recommendation()`

Recommendation 3:

Recommendations

Recommendation 1:
Begin ratings collection

Use `user_rating_collection()` function to gather user ratings

Recommendation 2:
Model Deployment

Implement recommender system: `new_user_genre_recommendation()`

Recommendation 3:
Cold Start Solutions

Explore additional solutions:

- Implicit Feedback
- Hybrid model

Next Steps

1. **Explore Additional Models within Surprise**
    a. SVD ++
2. **Address Cold Start Problem with Clustering**
    a. K-Means
    b. HAC

Thank you!

Github Repository:
https://github.com/ckucewicz/movie_recommendation_system

Contact Chris Kucewicz at
cfkucewicz@gmail.com with additional questions