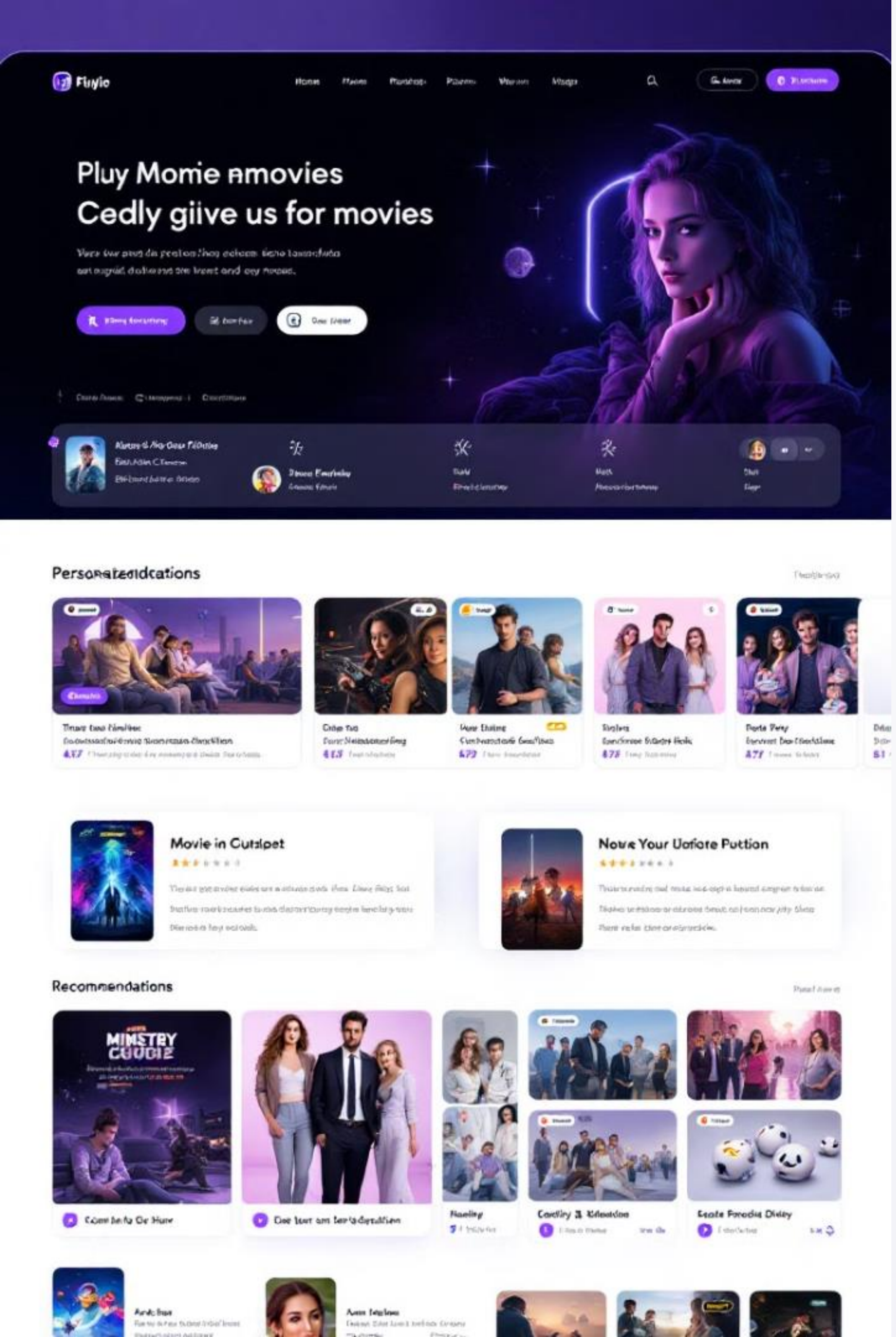


Movie Recommendation System

by

Colleta Kiilu

Business Understanding

This project creates a movie recommendation system. It addresses the challenge of providing personalized content. The system enhances user experience and engagement.

- **The Challenge**: Movie streaming services face user frustration due to lack of tailored content, leading to decreased interaction and potential revenue loss.
- **The Solution**: a recommendation system based on user ratings and preferences can improve the viewing experience.
- **Benefits**: the system uses data-driven insights to provide tailored movie recommendations, which increases user engagement and pleasure.
- **Business Impact**: Improves conversion rates, customer happiness, and user retention.

Project Objectives

1 **User Engagement**

Keep viewers engaged longer with tailored suggestions.

2 **Recommendation Diversity**

Balance familiarity with exploration of new content.

3 **Content Discoverability**

Showcase a broader selection of films to users.

Data Understanding

- ❖ The data was sourced from the popular [MovieLens]() dataset from the GroupLens research lab at the University of Minnesota.
- ❖ The different subsets (movies, ratings, links, tags) were merged.

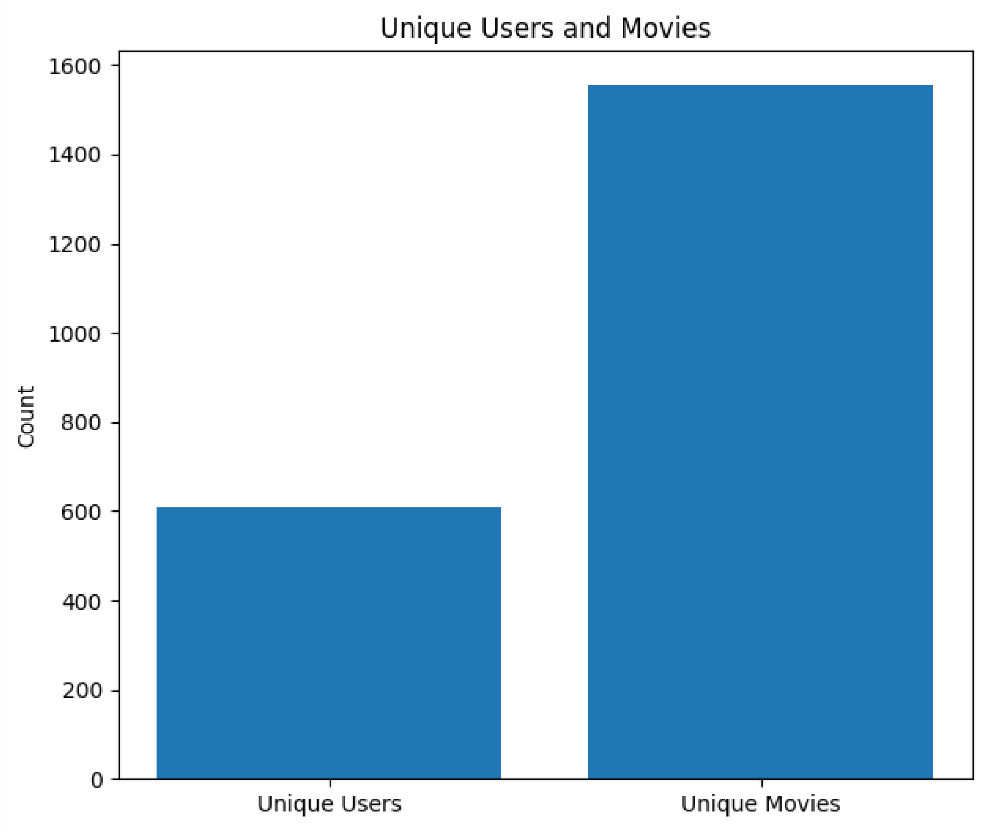
Dataset Size	Key Features	Missing Values
285,762 rows and 11 columns initially.	userIdrater, movieId, rating, title, genres.	userIdtag, tag, timestamptag columns had missing data.

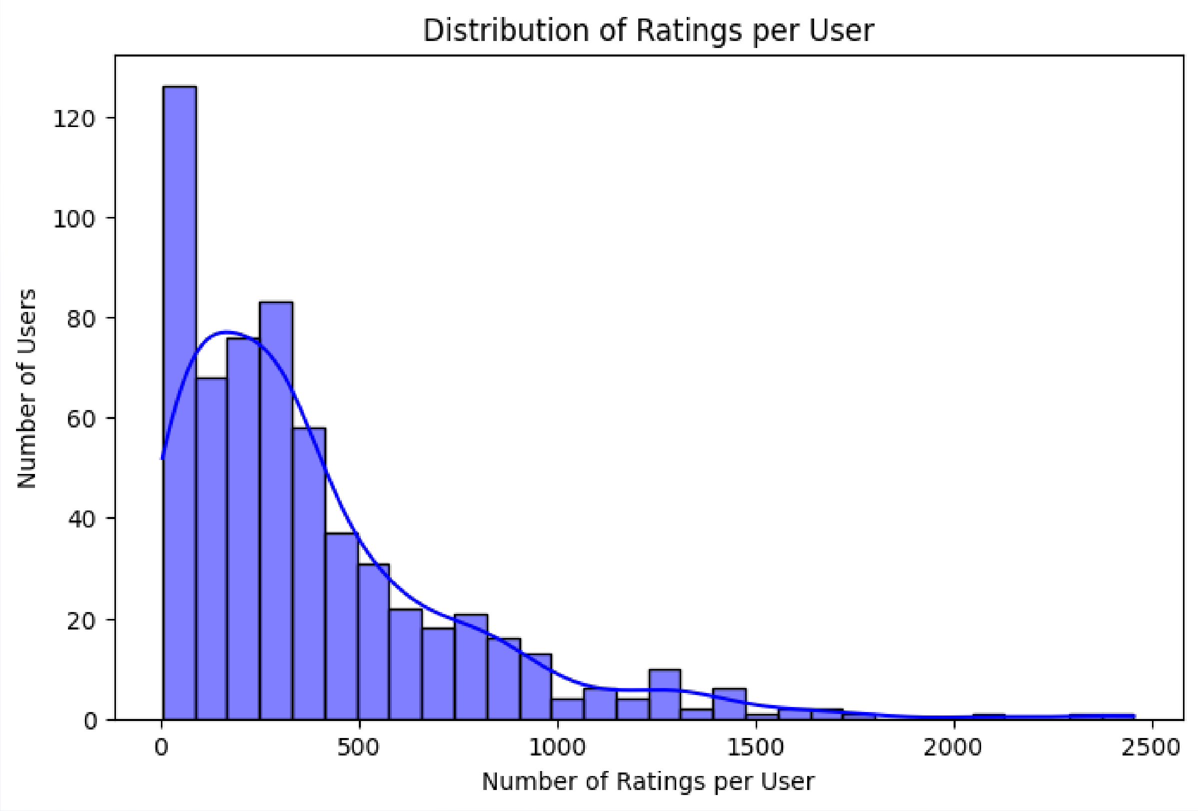
- ❖ After dropping null values, 233,213 records remained.
- ❖ The timestamps were also converted to datetime format.
- ❖ Other data types include string, integer and float.
- ❖ the data ranges from March 1996 to September 2018.

```
<class 'pandas.core.frame.DataFrame'>
Index: 233213 entries, 0 to 285760
Data columns (total 11 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   userIdrater     233213 non-null  int64
 1   movieId         233213 non-null  int64
 2   rating          233213 non-null  float64
 3   timestamprater  233213 non-null  datetime64[ns]
 4   title           233213 non-null  object
 5   genres          233213 non-null  object
 6   userIdtag       233213 non-null  float64
 7   tag             233213 non-null  object
 8   timestamptag    233213 non-null  datetime64[ns]
 9   imdbId          233213 non-null  int64
 10  tmdbId          233213 non-null  float64
dtypes: datetime64[ns](2), float64(3), int64(3), object(3)
memory usage: 21.4+ MB
```

Exploratory Data Analysis (EDA)

- There are 610 unique users and 1,554 unique movies.
- The average number of ratings per user is 382.32., while the median is 280, indicating most users rated fewer movies than average.
- The rating activity is highly skewed with a few users contributing the majority of ratings.
- The most active user rated 2,455 movies, while the 10th most active user rated 1,460 movies.
- The Top 10 Highest Rated Movies all had an averaging rating of 5.
- Bottom 10 Lowest Rated Movies had an average rating of between 0.5 to 1.8.





Modeling

1 Data Preparation

- ❖ userIdrater, movieId, rating columns were used for the modeling data.
- ❖ The surprise library was used to instantiate reader and data.
- ❖ train_test_split() was used to generate the train and test datasets

2 Models Trained

Baseline - Singular Value Decomposition with basic parameters

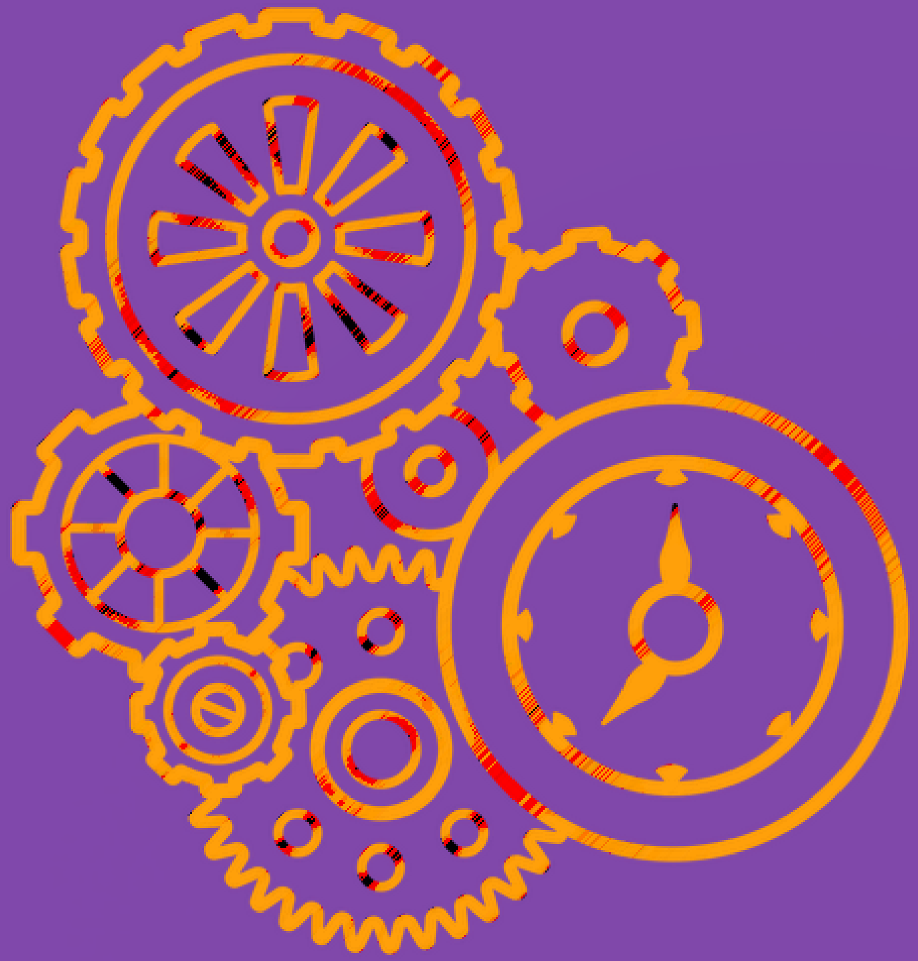
Model 2: KNNBasic

Model 3: KNNWithMeans

3 Evaluation

	Model	RMSE	MAE
0	SVD Baseline	0.3834	0.2346
1	KNNBasic	0.5274	0.2951
2	KNNWithMeans	0.5251	0.3176

- ❖ A low Root Mean Squared Error (RMSE) means that a model is able to make more accurate predictions.
- ❖ A low Mean Absolute Error (MAE) indicates less difference between the predicted and actual ratings



Tuned SVD Model

- The best initial model was the basic SVD. It had an RMSE of 0.3834 and an MAE: 0.2346.
- It made the best candidate for hyperparameter tuning with GridSearchCV.

Best Performance

RMSE: 0.3642

MAE: 0.2188

Hyperparameter Tuning

Improved prediction accuracy.

Conclusion

The project developed a movie recommendation system. A tuned SVD model was most effective. It improves user engagement and content discoverability.

Recommendations

1. Deployment
2. Monitoring
3. Enhancement

Integrate into streaming platform. Continuously monitor performance. Add features like genre preferences.