Movie Recommendation

Data Science Project

Problem statement and importance

Design a **Movie Recommendation System** that given some tag or genre, suggests movies based on user-provided tags, historical rating patterns.

Its importance:

- Reduces the search people have to do to find movies that they would like
- Improves content discovery for niche interests and long-tail content
- Leverages user-generated tags for more contextual suggestions than traditional genre-based filtering

Dataset Summary

We used the public dataset 'MovieLens' provided by grouplens.org

Ratings

- Total Records: 33,832,162
- Unique Users: 330,975
- Unique Movies: 83,239
- Rating Range: 0.5 - 5.0
- Average Rating: 3.54
- Rating Standard Deviation: 1.06

Movies

- Total Movies: 86,537
- Unique Genres: 20

Input Features

- User-assigned tags (free text)
- Movie genres (categorical)
- Historical user ratings (numerical)
- Tag genome scores (numerical)
- Timestamp (temporal)

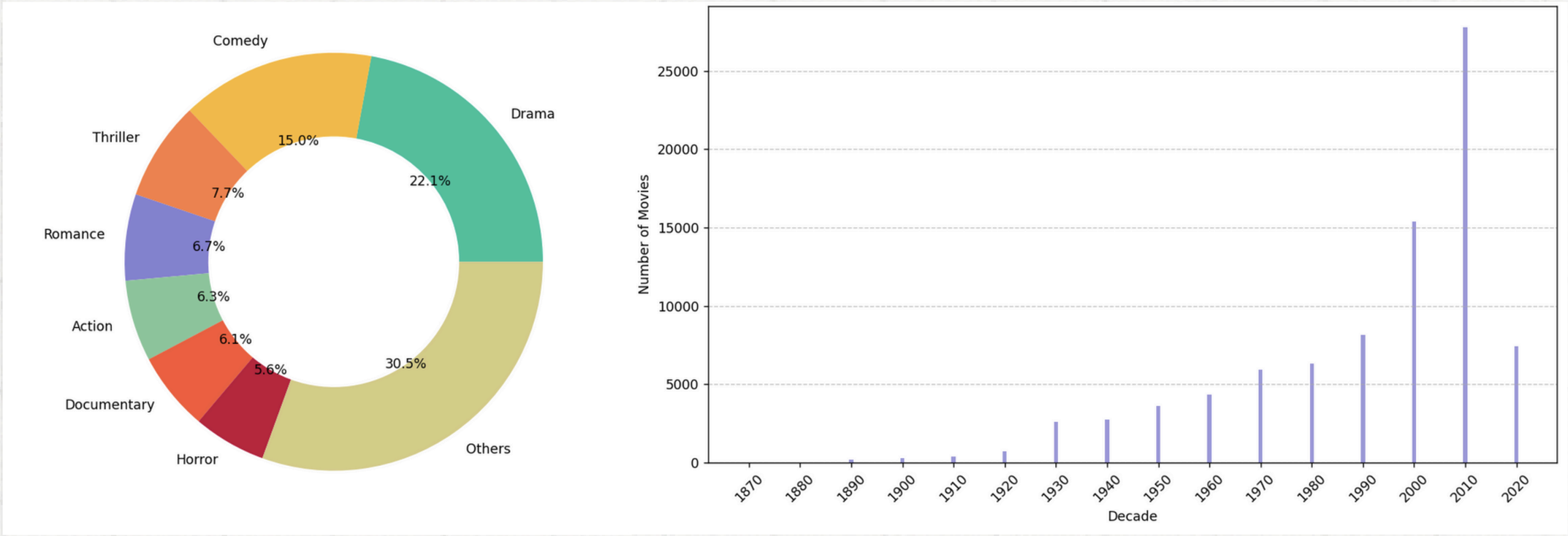
Tags

- Total User Tags: 2,328,315
- Unique Tags: 153,949
- Users Who Tagged: 25,280

Genome

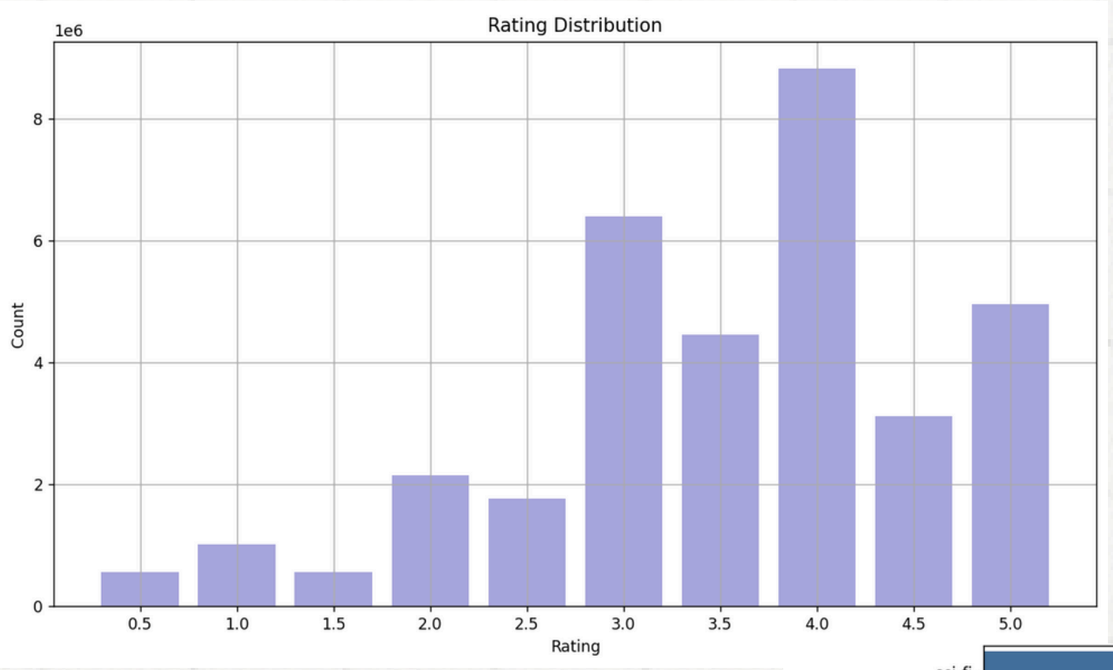
- Tag Genome Size: 1128
- Average Relevance Score: 0.111

Data Understanding by Visualization and EDA



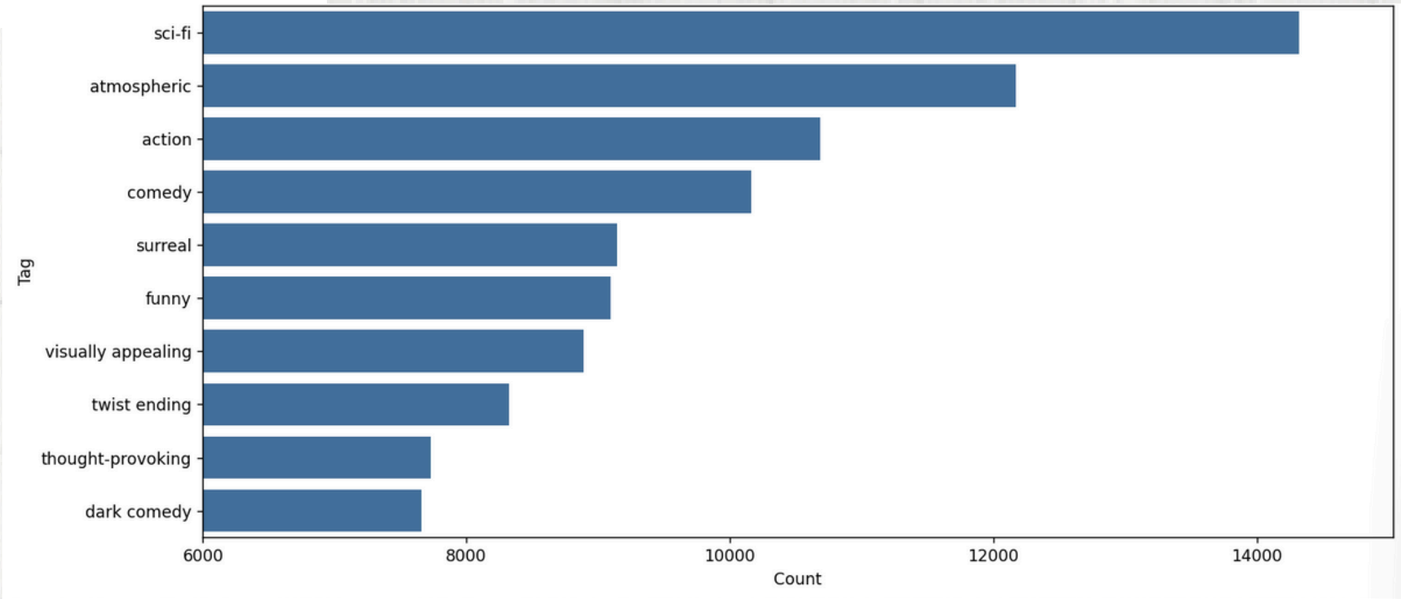
Genre Distribution

Movies temporal distribution



Ratings Distribution

Count of top-10 most used tags



Preprocessing

Checking For Duplicates

- Ensure that each user-movie pair has a unique rating, which **prevents skewed recommendations** from duplicate ratings.
- **Check for duplicates** in the ratings dataset and retain the most recent for that movie.

Handling Missing Values

- **Fill missing values** to avoid issues during recommendation generation.
- Check each dataset for **missing values in key columns.**

Removing Movies With No Genres

- Movies without genres **do not provide useful information** for genre-based recommendations, as they lack context for content discovery.
- **Excludes movies with (no genres listed)**, focusing on content with descriptive genres.

One Hot Encoding

- Enable **genre-based filtering** in the recommendation system.
- Transform genres in movies dataset into a **binary format** where each genre has a separate column indicating its presence.

Preprocessing Results

```
Before Cleaning:
Number of indices of movies: 86537
Number of indices of tags: 2328315
Number of indices of ratings: 33832162

After Cleaning:
Number of indices of movies: 79477
Number of indices of tags: 2307418
Number of indices of ratings: 33776103
```

User-Tag Consistency Test

Investigating whether users show consistency in their tag usage patterns.

Hypothesis:

- H0: Users are random in their tag assignments.
- H1: Users show consistency in their tag usage patterns.

Method:

- Group tags by user.
- Calculate the tag repetition rate for each user.
- Perform a one-sample t-test against a random repetition rate of 0.0.

Results:

- Statistic: 86.034
- p-value: 0.0
- Mean Repetition Rate: 0.106

Conclusion:

- The low p-value (< 0.05) indicates that users show significant consistency in their tag usage patterns, rejecting the null hypothesis.

Rating Patterns Test

Assessing whether movie ratings are uniformly distributed or exhibit significant patterns.

Hypothesis:

- H0: Ratings are uniformly distributed.
- H1: Ratings show significant patterns or biases.

Method:

- Perform a normality test on the rating distribution.
- Analyze hourly rating patterns.
- Conduct ANOVA on ratings grouped by hour of the day.

Results:

- Normality Test p-value: 0.0
- Time Pattern ANOVA F-statistic: 161.476
- Time Pattern ANOVA p-value: 0.0
- Mean Rating: 3.543
- Rating Standard Deviation: 1.064
- Number of Ratings: 33,776,103
- Rating Range: [0.5, 5.0]

Conclusion:

- The significant results from the normality test and ANOVA indicate that movie ratings are not uniformly distributed and show significant patterns, rejecting the null hypothesis.

Experiments conducted to validate the hypothesis tests.

User-Tag Consistency Test

Method:

1. Data Collection:
   - Extract tags assigned by users from the dataset.
   - Group tags by individual users.
2. Tag Repetition Rate Calculation:
   - For each user, calculate the repetition rate of tags.
   - Formula: **Repetition Rate = 1 - Number of Unique Tags / Total Number of Tags**
3. Statistical Testing:
   - Conduct a one-sample t-test to compare the observed repetition rates against a random repetition rate of 0.0.

Rating Patterns Test

Method:

1. Normality Test:
   - Perform a normality test (D'Agostino and Pearson's test) on the rating distribution.
2. Time-Based Analysis:
   - Convert timestamps to datetime and extract the hour of the day.
   - Calculate average ratings for each hour.
3. ANOVA Test:
   - Conduct an ANOVA test to compare rating distributions across different hours of the day.

Thank you very much!