NYU

Movie Recommendation System

Popcorn Predictors - Team number 2

(Ethan Iwama, Meghna Sharma, Eren Kaval, Cherron Griffith, Santiago Bernheim, Patrick Wang)

04.02.25

About…

<u>Goal:</u> Recommend movies using user preferences and movie metadata

<u>Dataset:</u> TMDB 5000 Movie Dataset

<u>Techniques:</u> Demographic Filtering, Content-Based Filtering, Collaborative Filtering

<u>Tools:</u> Data cleaning (pandas & numpy) JSON parsing for structured metadata

Visualization (Seaborn & Matplotlib)

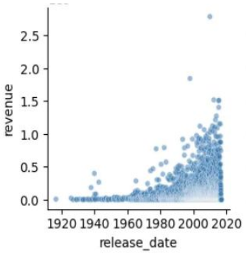
NYU

Data Cleaning

Exploratory Data Analysis (EDA) was conducted on both datasets to identify missing values, correct column data types, and uncover significant relationships relevant to our project goal.

```
movies.head()
movies.info()
movies.shape()
movies.dtypes
```

```
movie_id    0
title       0
cast        0
crew        0
dtype: int64
```



01.

Corrected column data types and extracted relevant values from columns containing lists of Python dictionaries

02.

Handled missing values by ensuring they were properly represented according to the column data type, without discarding too much data

03.

Refined the dataset to include only columns relevant to our project

04.

Created visualizations to uncover significant relationships within the data

NYU

Column Selection

Chose columns most relevant to our project objectives:

- **Performance Metrics:** popularity, vote_average, revenue, budget
- **Attributes:** genres, language, runtime, title
- **People Involved:** actors, directors

```
Credits Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   movie_id  4803 non-null   int64 
 1   title     4803 non-null   object
 2   cast      4803 non-null   object
 3   crew      4803 non-null   object
dtypes: int64(1), object(3)
```

```
Movies Dataset:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   budget                4803 non-null   int64  
 1   genres                4803 non-null   object 
 2   homepage              1712 non-null   object 
 3   id                    4803 non-null   int64  
 4   keywords              4803 non-null   object 
 5   original_language     4803 non-null   object 
 6   original_title        4803 non-null   object 
 7   overview              4800 non-null   object 
 8   popularity            4803 non-null   float64
 9   production_companies  4803 non-null   object 
 10  production_countries  4803 non-null   object 
 11  release_date          4802 non-null   object 
 12  revenue               4803 non-null   int64  
 13  runtime               4801 non-null   float64
 14  spoken_languages      4803 non-null   object 
 15  status                4803 non-null   object 
 16  tagline               3959 non-null   object 
 17  title                 4803 non-null   object 
 18  vote_average          4803 non-null   float64
 19  vote_count            4803 non-null   int64  
dtypes: float64(3), int64(4), object(13)
```

NYU

Handling Null Values

Obvious Null Values:

```
Missing values after joining and cleaning:
id                   0
title                0
original_language    0
genres               0
budget               0
revenue              0
runtime              0
popularity           0
vote_average         0
vote_count           0
release_date         1
release_year         1
director            30
main_actors          0
```

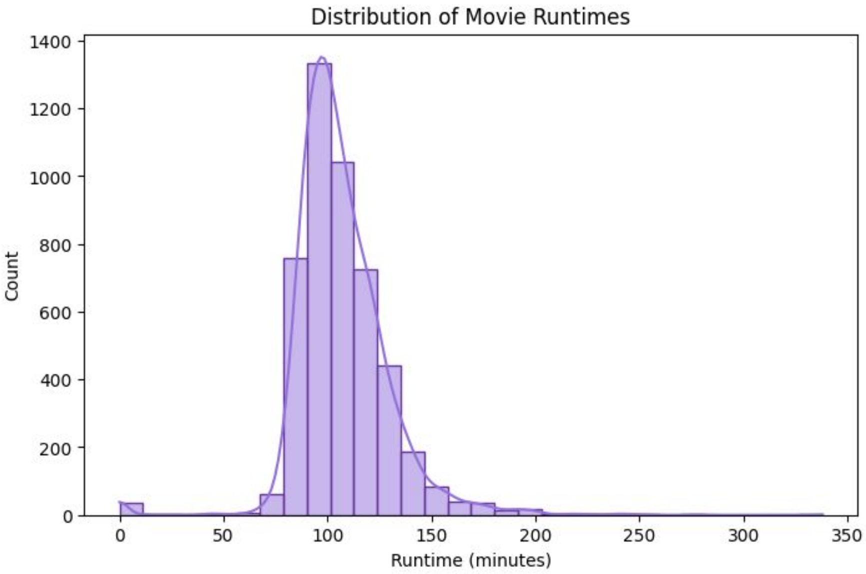
Hidden Null Example:

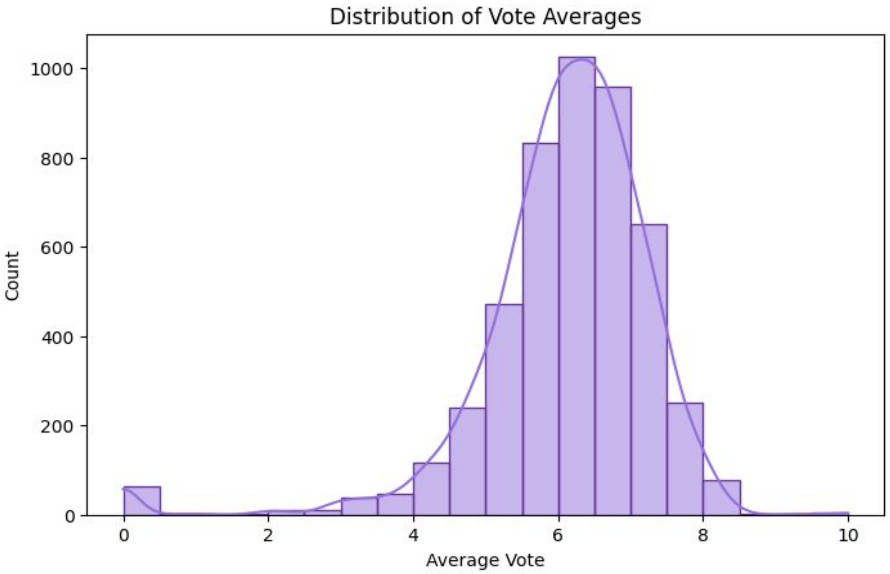
genre_names
[]
[]
[]
[]
[]
[]
[]

- The compiler did not catch all null values, so additional steps were necessary
- Adjusted null values to match the column data type
  - for columns containing lists, replaced empty lists with [Unknown]
  - set missing integer values to 0 and missing object (string) values to "Unknown"

NYU

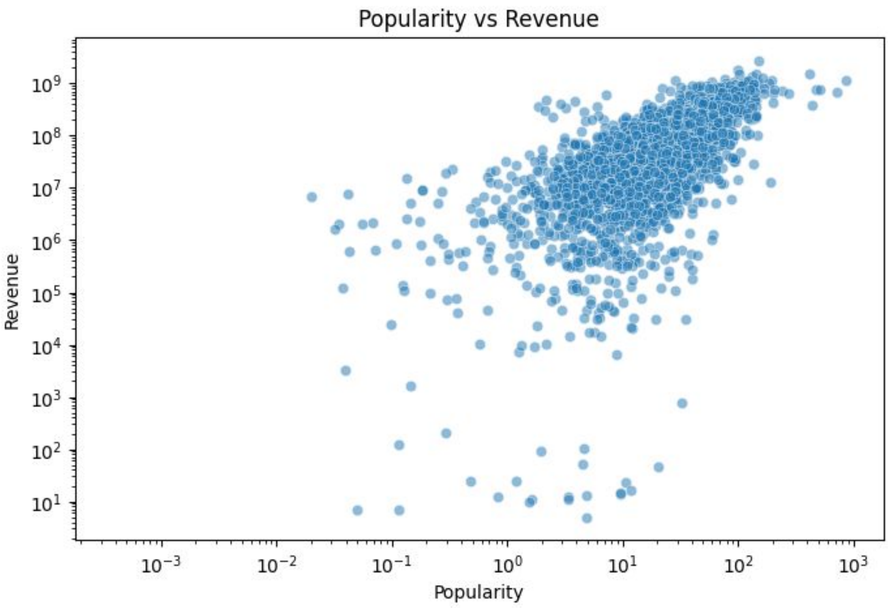
Key Observations: Value Distributions

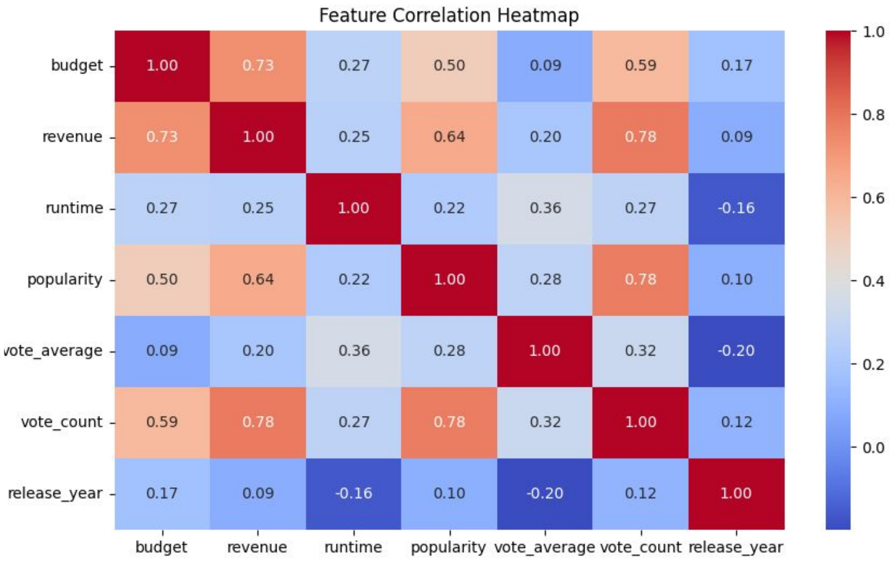




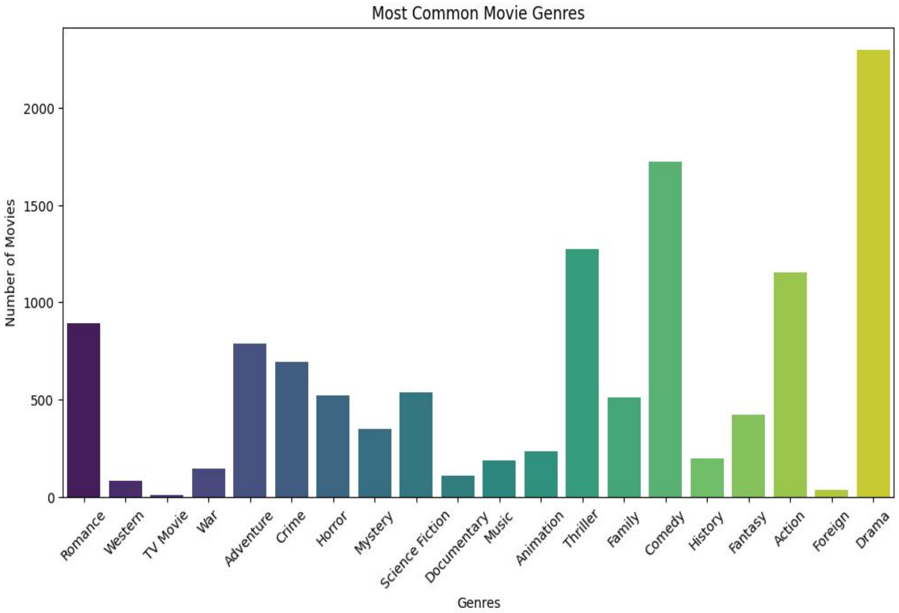
NYU

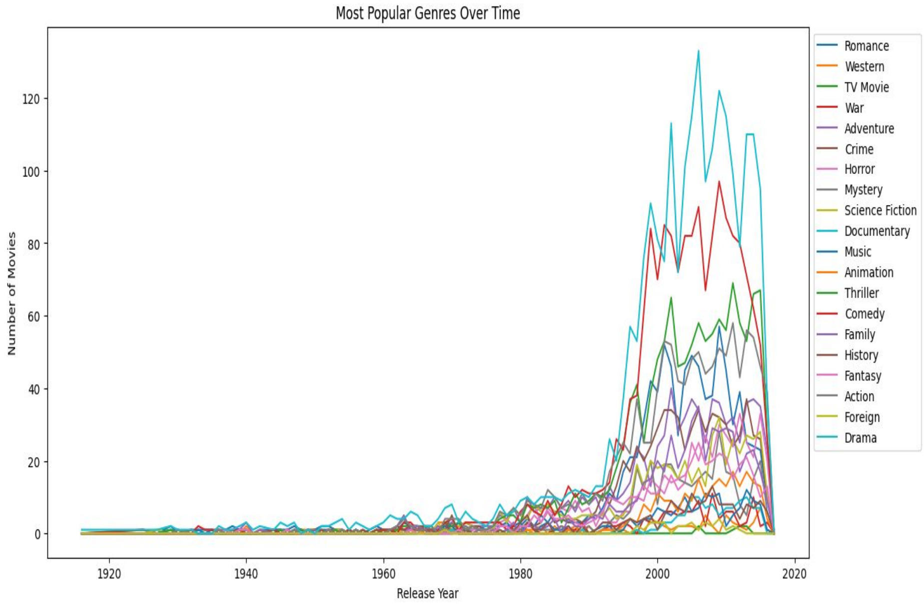
Key Observations: Column Correlations





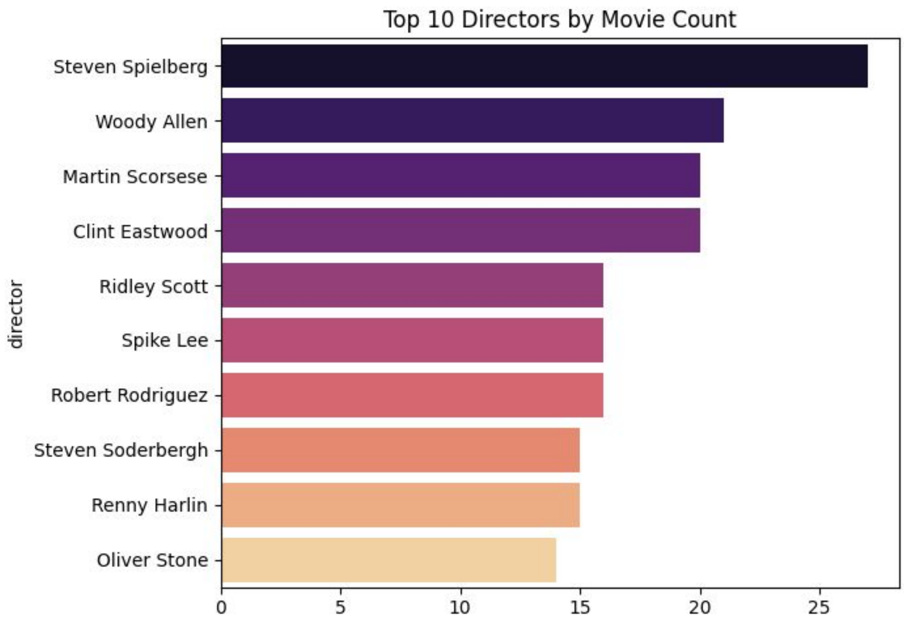
Key Observations: Genres

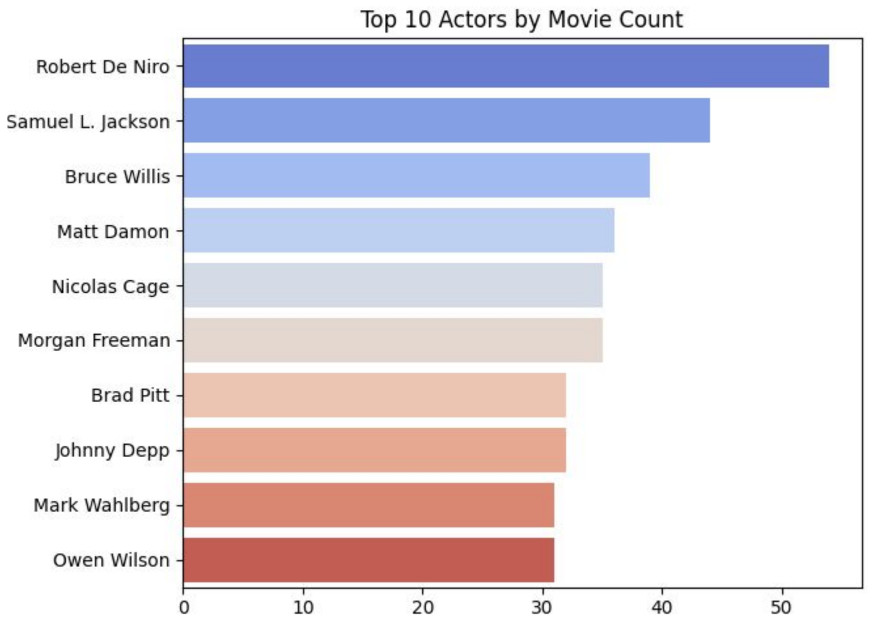




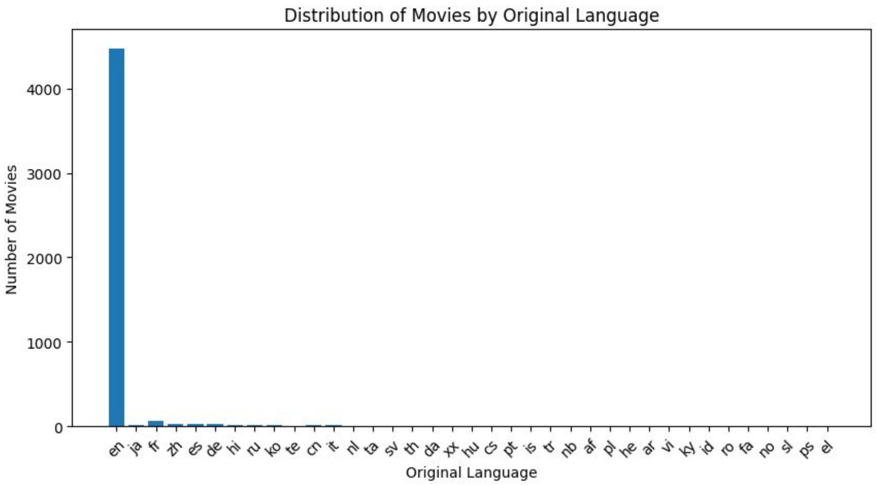
NYU

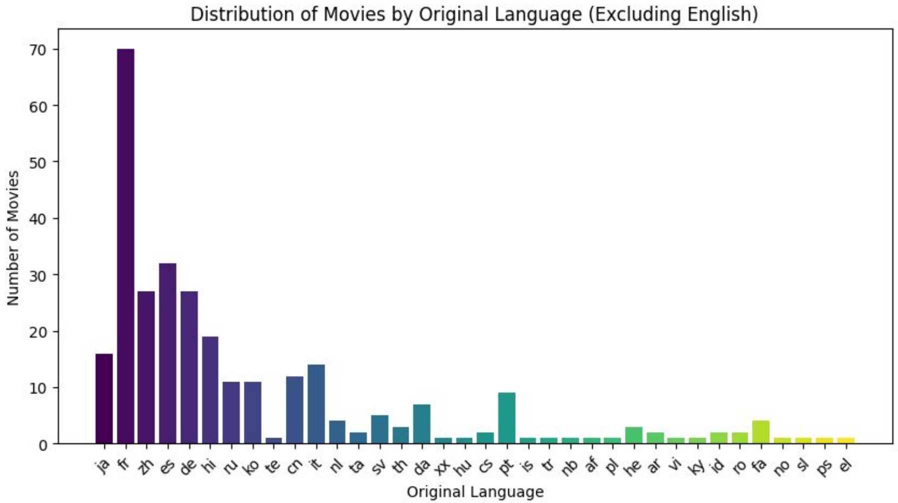
Key Observations: Actors & Directors





Key Observations: Languages





Next Steps

- **Baseline Model Development:** Create weighted function metric (IMDB model could be used as reference) that would use vote averages and counts to recommend top rated movies
- **Content-Based Filtering:** After determining top movies, use cleaned dataset used for analysis and train machine learning models using features to recommend movies that are similar
  - Based on initial impression a KNN method makes sense
  - Other considerations: Tree based methods
- **Model Evaluation and Visualization:** Evaluate the models using various methods such as precision & recall
  - Precision: "Among movies recommended, how many are actually relevant to the user?"
  - Recall: "Of all the relevant movies, how many did we successfully recommend to the user?"

NYU

Thankyou…

Popcorn Predictors - Team number 2

(Ethan Iwama, Meghna Sharma, Eren Kaval, Cherron Griffith, Santiago Bernheim, Patrick Wang)

04.02.25