# Decoding IMDb Ratings

A Data-Driven Approach to Understanding Audience Sentiment

# Understanding the Streaming Wars Through IMDb Ratings

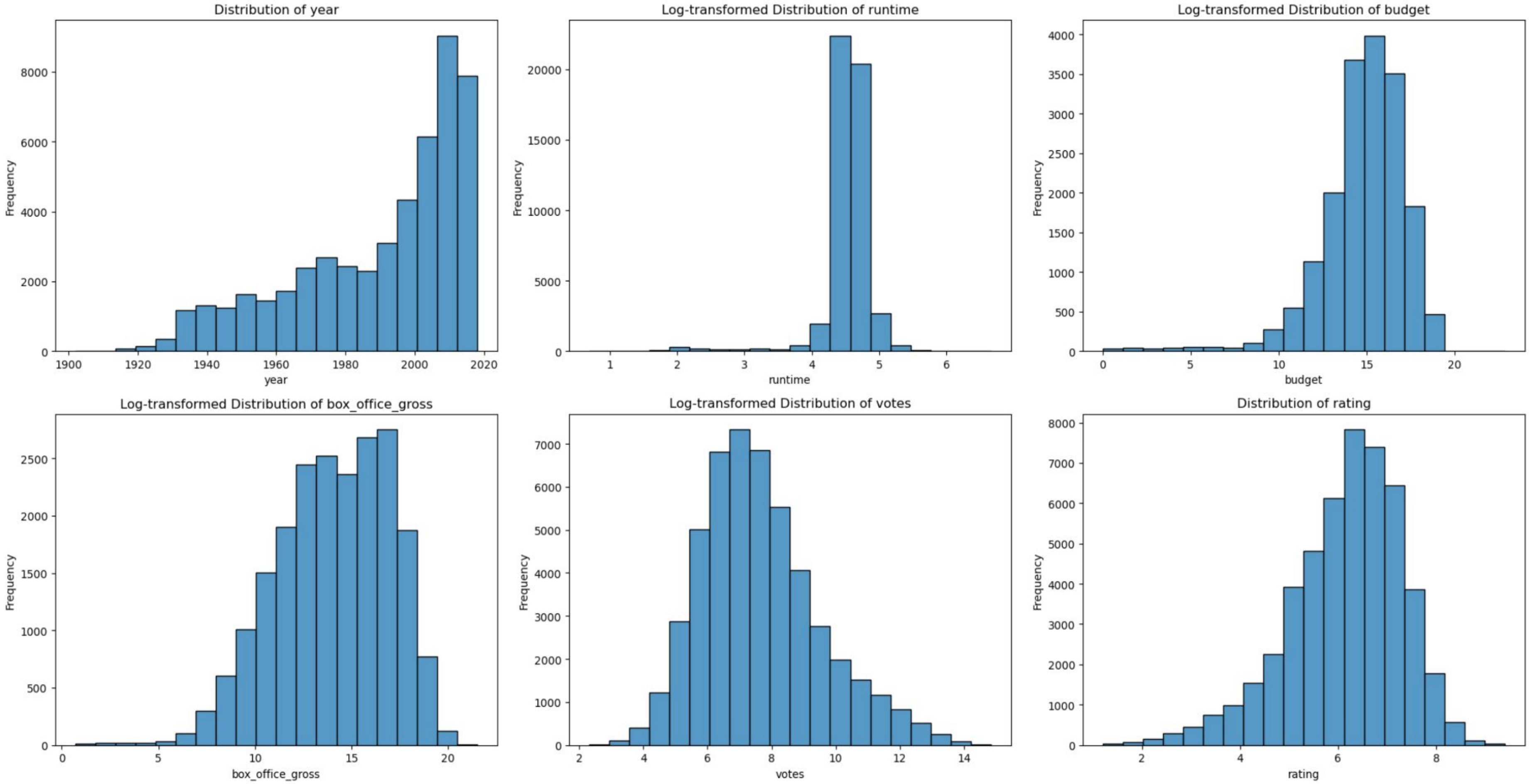
- **Predicting IMDb Ratings:** Gain real-time insights into audience sentiment crucial for industry success.
- **Streaming Wars:** Rapidly understand viewer preferences across genres for a competitive edge.
- **Data Science Edge:** Our model uses machine learning to forecast IMDb ratings, providing key insights for content success.
- **Beyond Traditional Methods:** Leveraging user-generated content, our model combines text from reviews with quantifiable data for a nuanced view of audience sentiment.

# Final Dataset Overview

- Combined 3 separate datasets, leveraging the unique IMDb ID for each movie
- Final Dataset contains:
  - **49,378** unique movies
  - with **13** different metadata features
  - **3,146,437** rows of user review text data

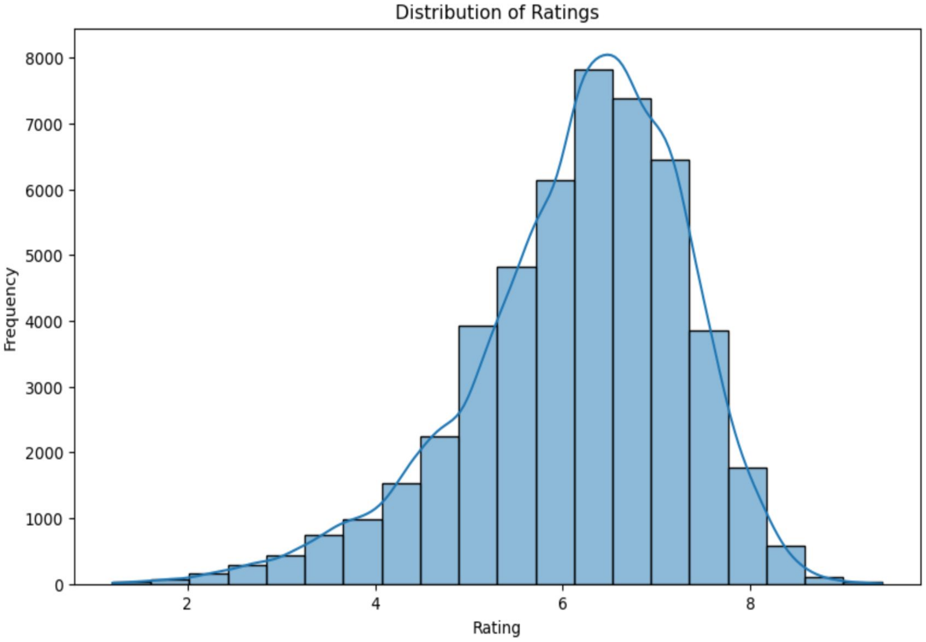
| Columns | Dtype |
|---|---|
| imdb_id | object |
| title | object |
| actors | object |
| directors | object |
| genres | int64 |
| language_en | int64 |
| year | int64 |
| runtime | int64 |
| budget | float64 |
| box_office_gross | float64 |
| production_companies | object |
| votes | int64 |
| rating | int64 |
| rating_category | object |
| decade | object |
| review_count | int64 |

# Distributions of Numeric Features



# Rating Binning

Target variable (star rating) was binned using quantiles to ensure balanced classes




| Rating Category | Count | Range |
| --- | --- | --- |
| Excellent | 11,059 | 7.1 – 9.4 |
| Good | 13,054 | 6.4 – 7.0 |
| Average | 12,640 | 5.6 – 6.3 |
| Poor | 12,626 | 1.2 – 5.5 |

# EDA Findings

1. **High Cardinality:** Too many unique values in actors, directors, and production companies for straightforward one-hot encoding.

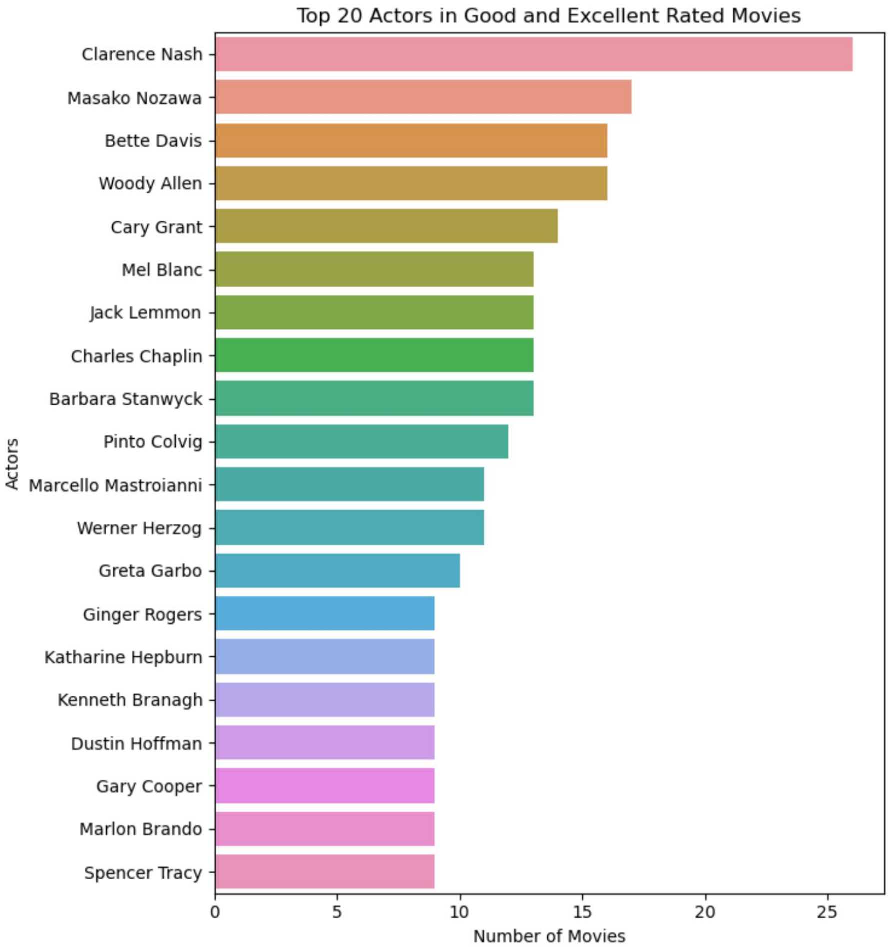
2. **Limited Actor-Director Influence:** No overwhelmingly frequent actor-director or director-production combinations that significantly influence movie ratings or earnings.

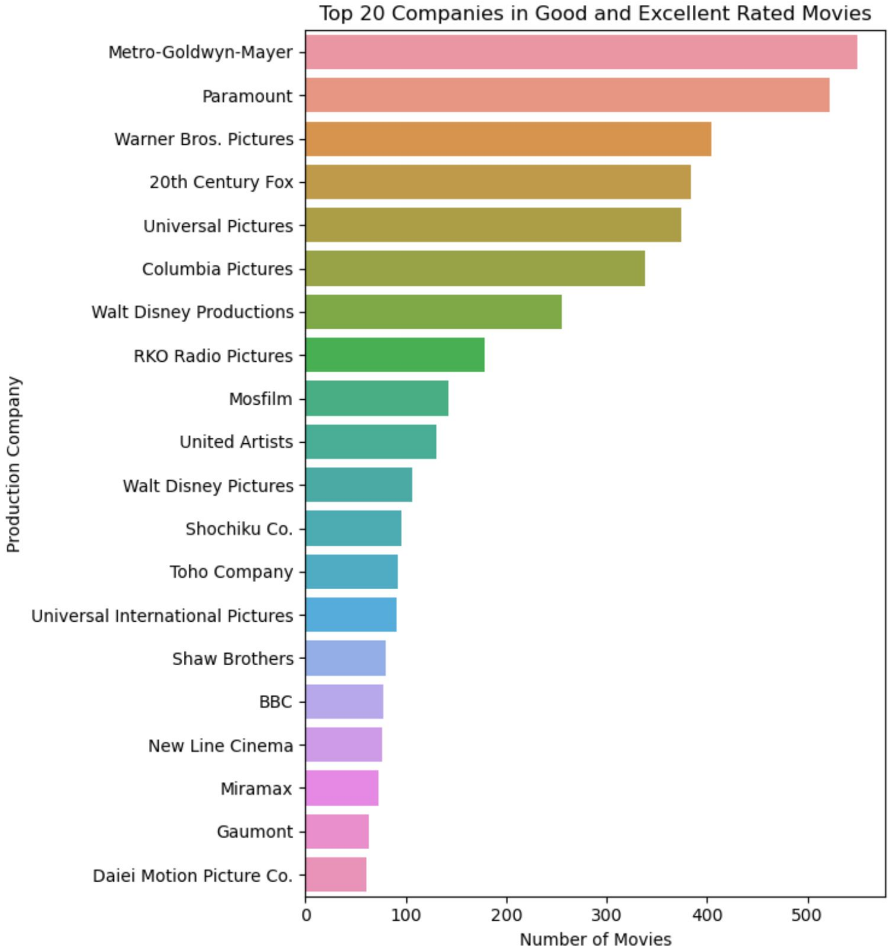
3. **Correlation**: Limited linear relationships or strong correlations, implying that film success is nuanced.

4. **Frequent Production Companies:** Higher frequencies observed for production companies; possible feature for predictive modeling.

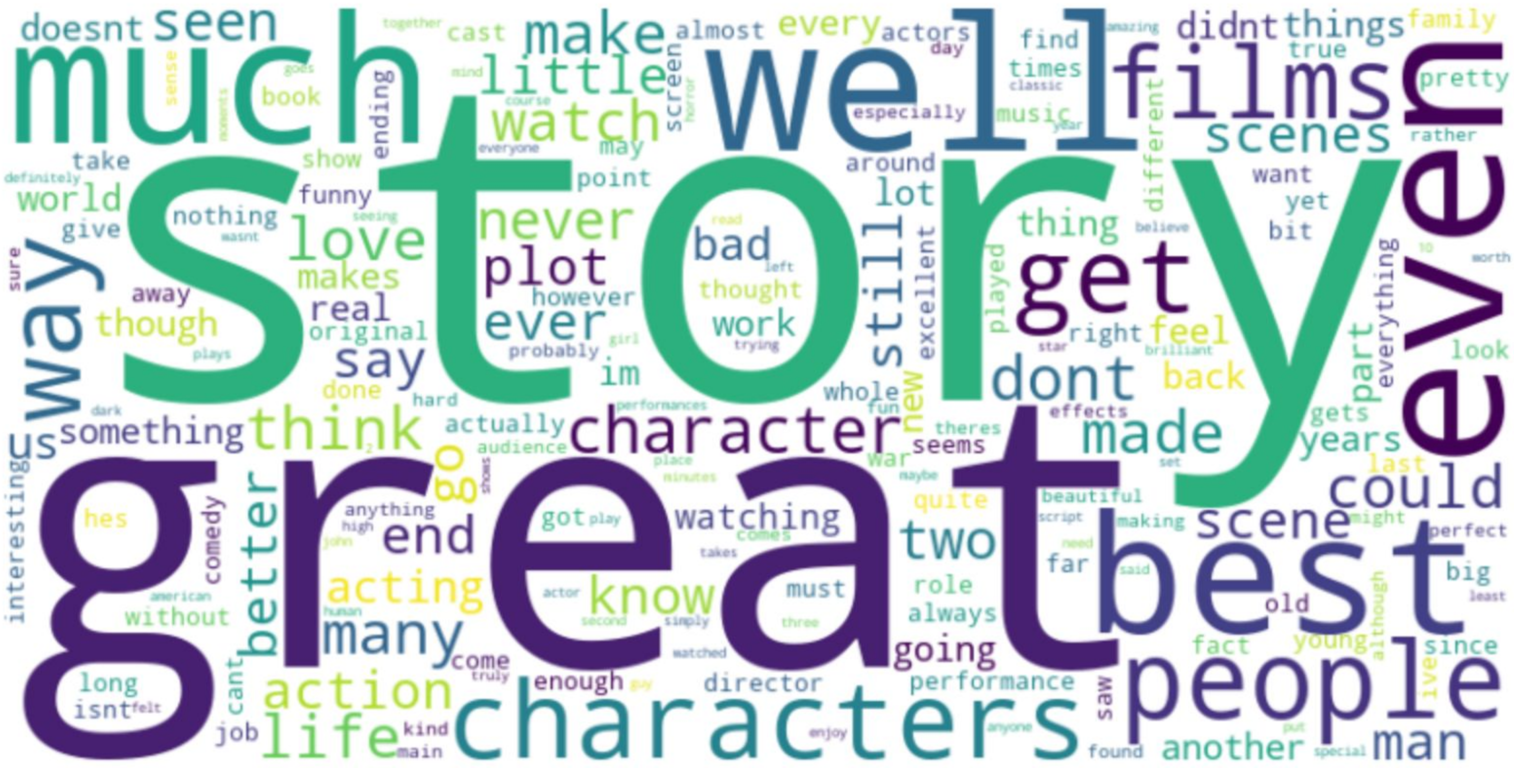
5. **Feature Engineering Strategy:** Use a "Top-N" approach for actors, directors, and production companies in top rated movies.

# Frequent Actors & Production Companies



# Wordcloud: Good & Excellent



# Wordcloud: Poor & Average

# Data Pre-Processing

1. **New Features:** Created one-hot encoded columns for the top 10 most frequent actors, directors, and production companies. Created review count column.

2. **Genres:** Exploded and one hot encoded all genres

3. **Decades:** Binned movies into decades to capture general patterns across the years

4. **Scaling:** Scaled data using StandardScaler

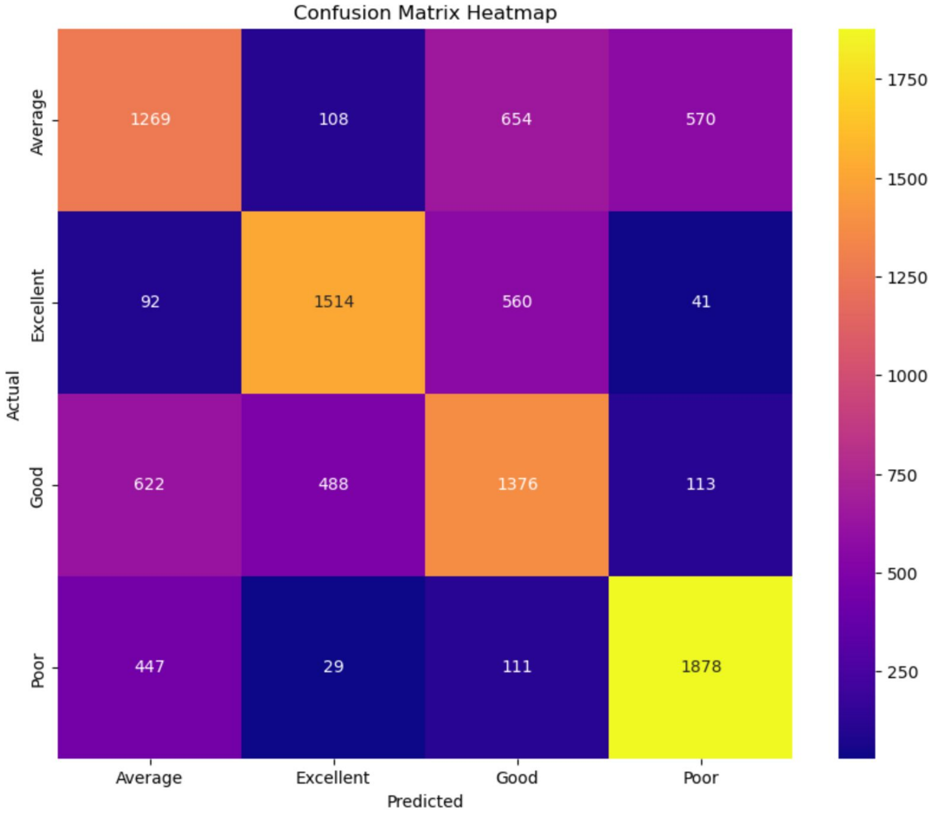
5. **Imputation:** Imputed missing budget and box office gross information using KNN Imputer

6. **Text Pre-processing:** Used lemmatization and TF-IDF to clean and vectorize text data, limiting it to the top 5000 features

# Baseline Model: Logistic Regression

- C = 0.5
- **Training Accuracy:** 0.67900109
- **Test Accuracy:** 0.61152755

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| **Average** | 0.522 | 0.488 | 0.504 | 2601.000 |
| **Excellent** | 0.708 | 0.686 | 0.697 | 2207.000 |
| **Good** | 0.509 | 0.529 | 0.519 | 2599.000 |
| **Poor** | 0.722 | 0.762 | 0.741 | 2465.000 |
| **accuracy** | 0.612 | 0.612 | 0.612 | 0.612 |
| **macro avg** | 0.615 | 0.616 | 0.615 | 9872.000 |
| **weighted avg** | 0.610 | 0.612 | 0.610 | 9872.000 |



# Next Steps

- Refine feature selection and try to understand feature importance based on coefficients from logistic regression model
- Further refine text vectorization, potentially with Word2Vec
- Try more advanced algorithms to improve accuracy, potentially Random Forest and/or XGBoost
- Ensure model's technical performance translates to business value