Machine Learning to Predict Netflix Top 10 Domestic Movies

Joe M, John A, Megan I, Morgan P

Value Proposition (Who might find this useful)

1. Streaming Platforms (e.g., Netflix, Hulu, Prime Video):
   a. Helps guide content acquisition and promotion strategies.
   b. Informs algorithmic recommendations and homepage placement to boost engagement.
2. Movie Studios & Distributors:
   a. Supports marketing decisions and release timing.
   b. Indicates which types of content (cast, genre, themes) are likely to succeed.
3. Content Creators & Producers:
   a. Offers data-driven guidance for casting, genre selection, and target demographics.
4. Advertisers & Brand Partners:
   a. Identifies which films are likely to draw the most eyeballs and therefore high-value ad placements.
5. Investors & Analysts:
   a. Useful for predicting platform performance, content ROI, and guiding decisions around partnerships or funding.
6. Academic & Market Researchers:
   a. Enables study of media consumption trends, cultural impact, and predictive modeling in entertainment analytics.

Data Acquisition

We found on Kaggle, two movie-related datasets:

- Netflix Top 10 Weekly Dataset (Global)
  - Contained Films and TV shows, included Weekly Rank and how many weeks a title was in the Top 10
- IMDB Movie Dataset Till Dec-2023
  - Contained Genre, Cast, Director, MPA Rating (PG, R, etc.), and IMDB's MetaScore

We planned on joining the two, so we could apply the IMDB attributes to the Movies that made it to Netflix's Top 10 list.

Data Transformation - Google Sheets

Some quick operations were done in Google Sheets.

<u>genre and cast transformation</u>

We thought it would be advantageous to separate the Genres into their own columns. ***split(genre0,”,”,true,true)*** did the trick.

<u>pr_rating transformation</u>

We converted the picture ratings to numeric ones: G - NC-17 where given the numbers 1-6. UNIQUE and XLOOKUP were used with a mapping column to add the *number_rating* column to the dataset.

Datasets placed in PostgreSQL

Tables were created for both datasets

```
CREATE TABLE IF NOT EXISTS public.imdb_movie_data_2023
(
    row_id integer NOT NULL,
    movie_name text,
    rating real,
    votes integer,
    meta_score real,
    genre0 text,
    genre1 text,
    genre2 text,
    genre3 text,
    pr_rating text,
    year integer,
    duration real,
    cast0 text,
    cast1 text,
    cast2 text,
    cast3 text,
    cast4 text,
    director text,
    number_rating integer,
    CONSTRAINT imdb_movie_data_2023_pkey PRIMARY KEY (row_id)
)
```

```
CREATE TABLE IF NOT EXISTS public.kaggle
(
    "UID" bigint NOT NULL DEFAULT nextval('"kaggle_UID_seq"'::regclass),
    week date,
    category text,
    weekly_rank integer,
    show_title text,
    weekly_hours_viewed bigint,
    runtime real,
    weekly_views bigint,
    cumulative_weeks_in_top_10 integer,
    CONSTRAINT kaggle_pkey PRIMARY KEY ("UID")
)
```

Creating View for ML Modeling and Visualization

The Machine Learning Model and Visualizations had different requirements

Machine Learning needs more numeric-based columns.

Visualization can use more text-based information digestible by people.

```sql
--updated with a new name.  copied the kaggle table as netflix
CREATE OR REPLACE VIEW view_full_data_set_no_nulls AS
SELECT
   imdb.*,
   netflix.weekly_rank,
   netflix.weekly_hours_viewed,
   netflix.weekly_views,
   netflix.cumulative_weeks_in_top_10
FROM
   imdb_movie_data_2023 AS imdb
LEFT JOIN
   netflix
ON
   imdb.movie_name = netflix.show_title

WHERE
   imdb.cast0 IS NOT NULL AND
   imdb.number_rating IS NOT NULL AND
   imdb.genre0 IS NOT NULL AND
   imdb.meta_score IS NOT NULL AND
   netflix.weekly_rank IS NOT NULL AND
   netflix.cumulative_weeks_in_top_10 IS NOT NULL
```

```sql
CREATE OR REPLACE VIEW view_full_data_set AS
SELECT
   imdb.*,
   netflix.weekly_rank,
   netflix.weekly_hours_viewed,
   netflix.weekly_views,
   netflix.cumulative_weeks_in_top_10
FROM
   imdb_movie_data_2023 AS imdb
LEFT JOIN
   netflix
ON
   imdb.movie_name = netflix.show_title;

CREATE TABLE tableau as
SELECT * FROM view_full_data_set
WHERE pr_rating not like '%TV%'
```

Tableau Visualizations

- With a clean csv file, all visuals were filtered with the wkly_top_10 value of 3 or 4 or more
- Types of visuals being showcased are going to be bar, treemap, and packed bubbles

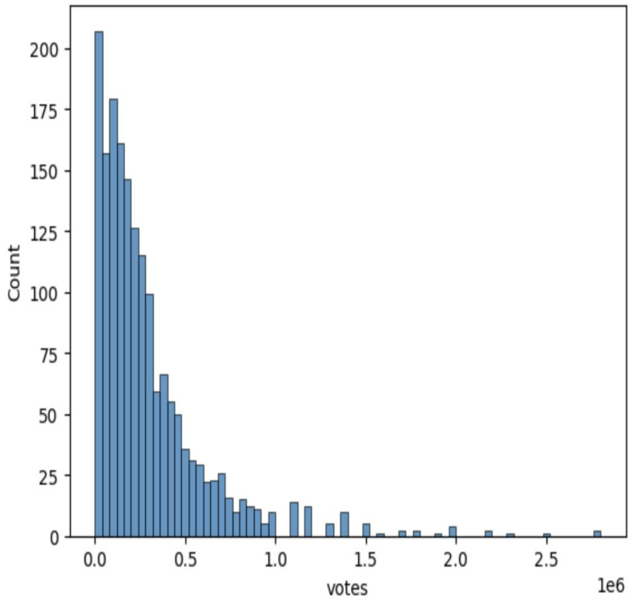
  Tableau workbook

- Be interesting to be able to have the option to filter any of these visuals with possible netflix original movies (we just couldn't find the dataset at the time to do this)

Exploratory Data Analysis (EDA)

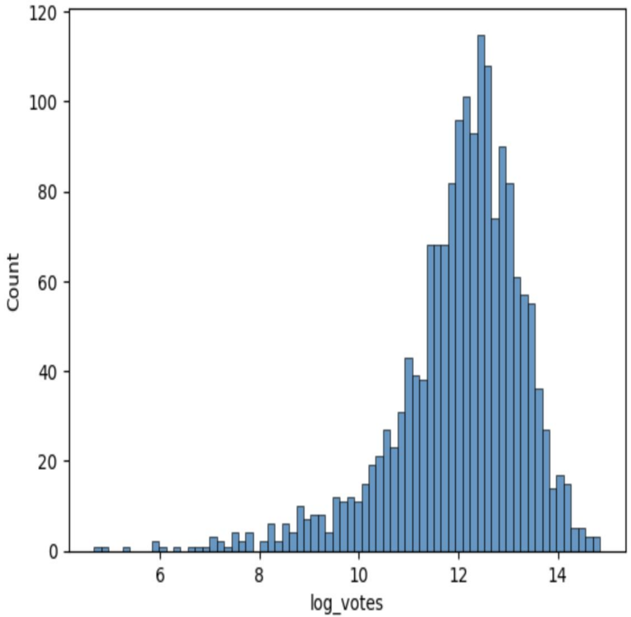
```python
# Create a histogram to better visualize the distribution of votes
sns.histplot(df['votes'], bins=70)
```

<Axes: xlabel='votes', ylabel='Count'>




```python
# Create histogram to see distribution of log transformed votes
sns.histplot(df['log_votes'], bins=70)
```

<Axes: xlabel='log_votes', ylabel='Count'>



Data Preparation

- Cleaned the data
- Handled missing values
- Created new features (feature engineering)
- One-hot encoded categorical variables
- Scaled all features
- Checked for class imbalance
- Applied SMOTE to balance the target classes

Investigate Top 10 movies Feature

```python
# Check out how many movies are in the Top 10 versus not (0=No, 1=Yes)
df['netflix_top_10'].value_counts()
```

netflix_top_10	count
0	1553
1	175

dtype: int64

Machine Learning Models

1. Logistic Regression
    - Accuracy 91%
    - 6 out of 36 right

2. **Random Forest Model**

3. Keras Model
    - Accuracy 90%
    - 9 out of 36 right

```
Accuracy: 0.9953703703703703

Confusion Matrix:
 [[396   0]
 [  2  34]]

Classification Report:
              precision    recall  f1-score   support

          No       0.99      1.00      1.00       396
         Yes       1.00      0.94      0.97        36

    accuracy                           1.00       432
   macro avg       1.00      0.97      0.98       432
weighted avg       1.00      1.00      1.00       432

['random_forest_model.pkl']
```

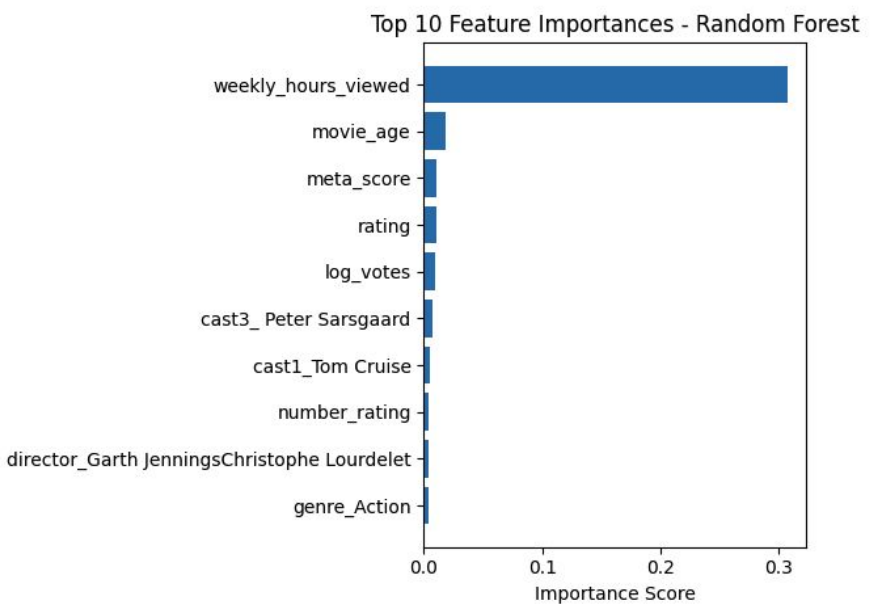
Random Forest Model

Training Classification Report:

	precision	recall	f1-score	support
No	1.00	1.00	1.00	1157
Yes	1.00	1.00	1.00	139
accuracy			1.00	1296
macro avg	1.00	1.00	1.00	1296
weighted avg	1.00	1.00	1.00	1296

Test Classification Report:

	precision	recall	f1-score	support
No	0.99	1.00	1.00	396
Yes	1.00	0.94	0.97	36
accuracy			1.00	432
macro avg	1.00	0.97	0.98	432
weighted avg	1.00	1.00	1.00	432



What Data Would Strengthen the Model?

- Views or hours watched in the first 91 days
- Weekly hours viewed for all titles, not just top 10
- Completion rate (how many people finish watching)
- User interaction signals (likes, watchlist adds, re-watches)

API/Flask process

We selected features that reflect both audience appeal and engagement potential.

- **Cast & Director**: Recognizable names boost visibility and drive traffic.
- **User Ratings & Reviews**: IMDb scores, meta scores, and vote counts show audience reception.
- **Weekly Hours Viewed**: A strong indicator of current popularity.
- **Movie Age**: Newer releases often get more promotion and interest.
- **Genre**: One-hot encoded to highlight trends (e.g., horror, comedy).

These features were tested, scaled, and used to train our Random Forest model—then integrated into a user-friendly **Flask API** that delivers Top 10 predictions.

API/Flask process

To deploy our model, we built a Flask API that uses the fully preprocessed and tested data from our Random Forest training. We saved the entire preprocessing pipeline (scaling, encoding, and feature transformations) so that new inputs would be handled exactly as they were during training. The API connects to a custom-built `index.html` page, allowing users to input movie details directly into a simple web form. Once submitted, the inputs are passed through the preprocessing steps, run through the model, and return a real-time prediction on whether the movie is likely to appear in Netflix's Top 10.

API/Flask process - Possible Improvements

To improve the model, we could have added engineered features like director success, cast popularity, or holiday release timing. External data like marketing spend or social media buzz could provide deeper insights. We also could have used advanced encoding or tried models like XGBoost, and checked for target leakage in features like weekly hours viewed. Still, our current model reached over 99% accuracy

This Model is Truly Useful Only to Netflix

- **Predictive Power for New Releases:**
  With access to **early viewership trends**, Netflix could forecast Top 10 potential quickly.
- **Content Strategy Insights:**
  Spot patterns in the attributes of successful titles to guide future content development and licensing.
- **Platform Optimization:**
  Suggest which titles to feature or promote more heavily for engagement.
- **Library Forecasting:**
  Estimate which existing titles may **trend or resurge**, especially around holidays or new seasons.