Movie Box Office Revenue Prediction

Brian M

6/14/2020

Summary

In 2019, movie global box office revenue hit a record $42.5 billion (source: hollywoodreporter). We're hired by a movie production startup who wants to make it big producing movies. Using historical movie data, we're tasked with building a model that can predict international movie box office revenue using movie attributes. Stakeholders also want to be able to see what variables tend to be most predictive of a movie blockbuster (e.g. model interpretation will need to be considered).

We're given a Kaggle training dataset which has 23 initial features and 3000 observations. The Kaggle dataset is sourced from The Movie Database (TMDB). The dataset is curated by TMDB community members who input movie metadata. A glimpse of the starting features and data structure is shown below:

```
## Rows: 3,000
## Columns: 24
## $ id                    <dbl> 1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 1...
## $ belongs_to_collection <chr> "[{'id': 313576, 'name': 'Hot Tub Time Machin...
## $ budget                <dbl> 14000000, 3300000, 1200000, 0, 8000000, 14000...
## $ genres                <chr> "[{'id': 35, 'name': 'Comedy'}]", "[{'id': 18...
## $ homepage              <chr> NA, "http://sonyclassics.com/whiplash/", "htt...
## $ imdb_id               <chr> "tt2637294", "tt2582802", "tt1821480", "tt138...
## $ original_language     <chr> "en", "en", "hi", "ko", "en", "en", "en", "en...
## $ original_title        <chr> "Hot Tub Time Machine 2", "Whiplash", "Kahaan...
## $ overview              <chr> "When Lou, who has become the \"father of the...
## $ popularity            <dbl> 6.575393, 64.299990, 3.174936, 1.148070, 0.74...
## $ poster_path           <chr> "/tQtWuwvMf0hCc2QR2tkolwl7c3c.jpg", "/lIv1Qin...
## $ production_companies  <chr> "[{'name': 'Paramount Pictures', 'id': 4}, {'...
## $ production_countries  <chr> "[{'iso_3166_1': 'US', 'name': 'United States...
## $ release_date          <chr> "2/20/15", "10/10/14", "3/9/12", "2/5/09", "8...
## $ runtime               <dbl> 93, 105, 122, 118, 83, 92, 84, 100, 91, 119, ...
## $ spoken_languages      <chr> "[{'iso_639_1': 'en', 'name': 'English'}]", "...
## $ status                <chr> "Released", "Released", "Released", "Released...
## $ tagline               <chr> "The Laws of Space and Time are About to be V...
## $ title                 <chr> "Hot Tub Time Machine 2", "Whiplash", "Kahaan...
## $ Keywords              <chr> "[{'id': 4379, 'name': 'time travel'}, {'id':...
## $ cast                  <chr> "[{'cast_id': 4, 'character': 'Lou', 'credit_...
## $ crew                  <chr> "[{'credit_id': '59ac067c92514107af02c8c8', '...
## $ revenue               <dbl> 12314651, 13092000, 16000000, 3923970, 326163...
## $ label                 <chr> "train", "train", "train", "train", "train", ...
```

We'll use 80% of the data for training and 20% of the training data as a final test dataset. Given the dataset is only 3000 observations, we'll use cross validation on the training data to optimize model performance. By partioning only 20% of the available for testing we hope to leave ample data for training our models. The test dataset will only be used for final prediction assessment. Our project dataset has many features in JSON format and NA values are present. We'll treat the JSON features as characters strings and mine the text using regular experssions / text analysis techniques. Feature imputation techniques will be used to fill in the NA values. Additionally, the dataset includes a mix of numeric, time series, and text feeatures whcih we'll look to extract useful signal from.

Our target variable for prediction is revenue. In order to not over pentalize blockbluster movies we predict/measure performance on the log of revenue and use RMSE (root mean square error) for performance assessment.

Phases of this project:

1) Read in the project datasets
2) Data cleaning & feature engineering
3) Leverage Caret package to test various models and optimize tuning parameters
4) Select top performing model
5) Output final RMSE for the validation dataset

Analysis & Method

Feature cleaning and feature engineering steps

Below we highlight which features have NA values. When movie runtime is missing, we replace run time with the median runtime. Mode imputation is used to replace missing language and status variables. Additional text based features which are NA are replaced with "Missing" text. Movie year is corrupted for several movies and does not return NA. We create a custom function to clean movie year.

```
##                  feature NA_Count
## 1  belongs_to_collection     2396
## 2               homepage     2054
## 3                tagline      597
## 4               Keywords      276
## 5   production_companies      156
## 6   production_countries       55
## 7       spoken_languages       20
## 8               overview        8
## 9                 genres        7
## 10                  crew        3
## 11               runtime        2
## 12           poster_path        1
```

After the first phase of feature cleaning, we move to deriving new features off the base set of features with the goal of extracting useful signals for model prediction.

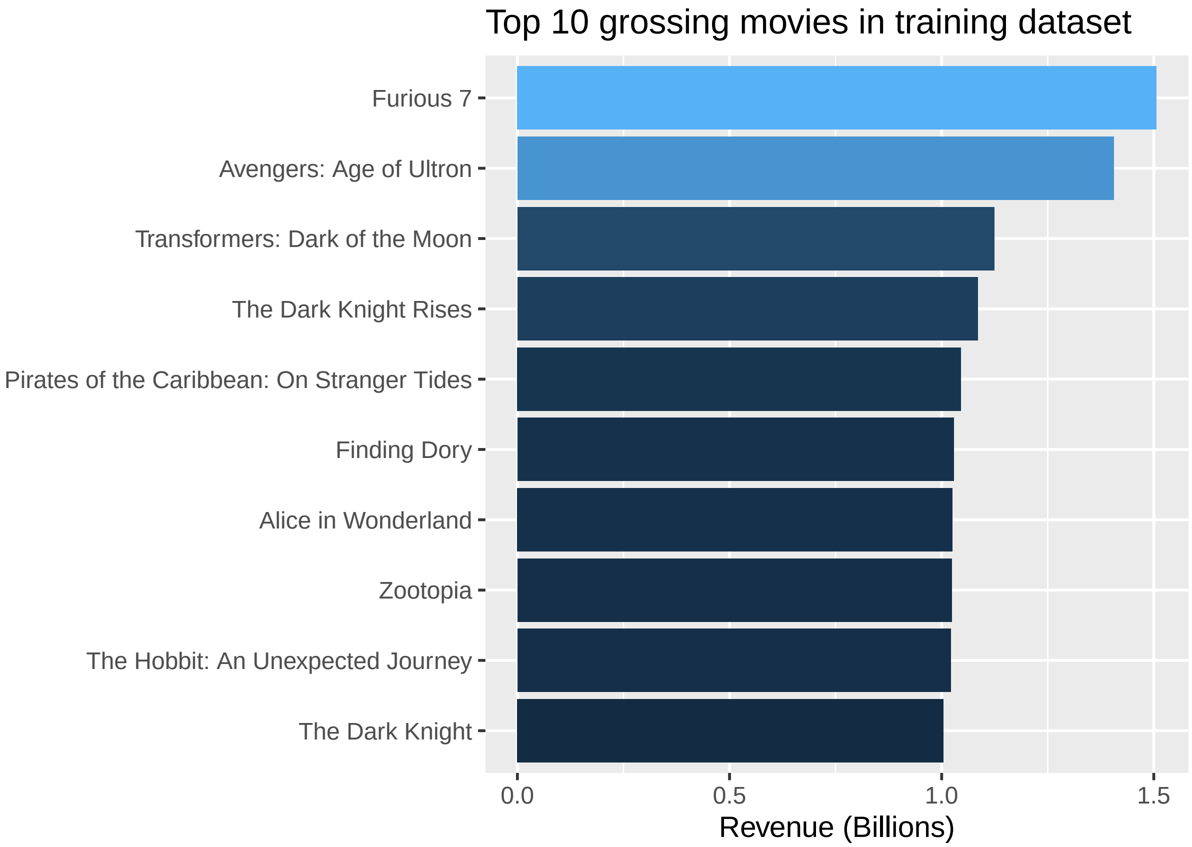
```
### derive features
all_data <- all_data %>%
  mutate(pre_process_budget = budget,
  pre_process_budget_available = ifelse(pre_process_budget>0,1,0),
  release_year = year(release_date_clean),
  before_2000_flag = ifelse(year(release_date_clean)<2000,1,0),
  before_1980_flag = ifelse(year(release_date_clean)<1980,1,0),
  release_year_bin = cut2(year(release_date_clean), g=10),
  release_month = month(release_date_clean),
  release_month_day = day(release_date_clean),
  release_week_number = week(release_date_clean),
  release_day_of_week = wday(release_date_clean, label = TRUE),
  release_year_quarter_str = paste0("Quarter","::",
  quarter(release_date_clean, with_year = FALSE, fiscal_start = 1)),
  title_length = str_length(title),
  belongs_to_collection_flag = ifelse(str_count(belongs_to_collection, "name")>0,1,0),
  tagline_available = ifelse(tagline=="Missing", 0, 1),
  homepage_available = ifelse(homepage=="Missing", 0, 1),
  homepage_disney_flag = ifelse(str_count(homepage, "disney")>0,1,0),
  homepage_sony_flag = ifelse(str_count(homepage, "sony")>0,1,0),
  homepage_warnerbros_flag = ifelse(str_count(homepage, "warnerbros")>0,1,0),
  homepage_focusfeatures_flag = ifelse(str_count(homepage, "focusfeatures")>0,1,0),
  homepage_fox_flag = ifelse(str_count(homepage, "foxmovies")>0 |
                                    str_count(homepage, "foxsearchlight")>0,1,0),
  homepage_magpictures_flag = ifelse(str_count(homepage, "magpictures")>0,1,0),
  homepage_mgm_flag = ifelse(str_count(homepage, ".mgm.")>0,1,0),
  homepage_miramax_flag = ifelse(str_count(homepage, ".miramax.")>0,1,0),
  homepage_facebook_flag = ifelse(str_count(homepage, ".facebook.")>0,1,0),
  genres_count = str_count(genres, "id"),
  production_company_count = str_count(production_companies, "name"),
  production_country_count = str_count(production_countries, "name"),
  spoken_languages_count = str_count(spoken_languages, "name"),
  cast = ifelse(cast=="[]","Missing", cast),
  cast = ifelse(cast=="#N/A","Missing", cast),
  cast_count = str_count(cast, "cast_id"),
  cast_gender_0_count = str_count(cast, "’gender’: 0,"),
  cast_gender_1_count = str_count(cast, "’gender’: 1,"),
  cast_gender_2_count = str_count(cast, "’gender’: 2,"),
  crew = ifelse(crew=="#N/A","Missing", crew),
  crew_count = str_count(crew, "credit_id"),
  director_count = str_count(crew, "job’: ’Director’, ’name’:"),
  producer_count = str_count(crew, "job’: ’Producer’, ’name’:"),
  exec_producer_count = str_count(crew, "’job’: ’Executive Producer’, ’name’:"),
  independent_film_flag = ifelse(str_count(Keywords, "independent film")>0,1,0)
)
```

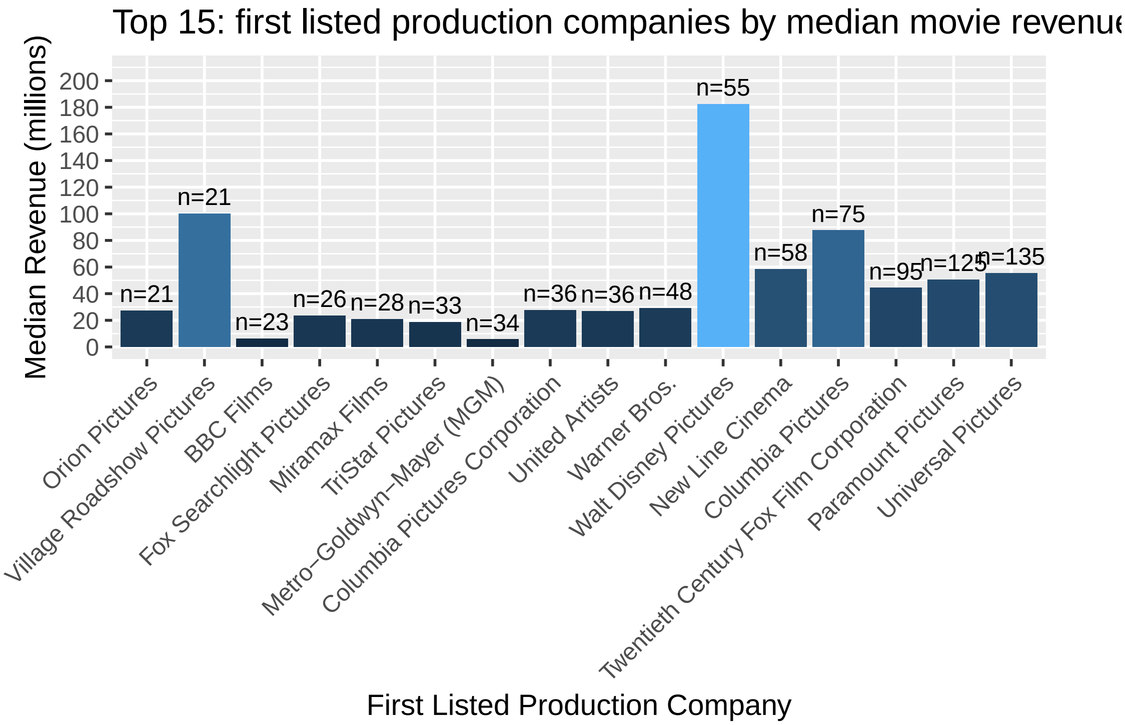
In the third phase of feature engineering, we look to clean up derived features and prepare for iterative exploratory analysis and modeling. When cast or crew count is zero, replace with median. A KNN model (trained on the training data only) is used to predict movie budget when budget is less than 1k. ~30% of observations in training set have budget less than 1k.

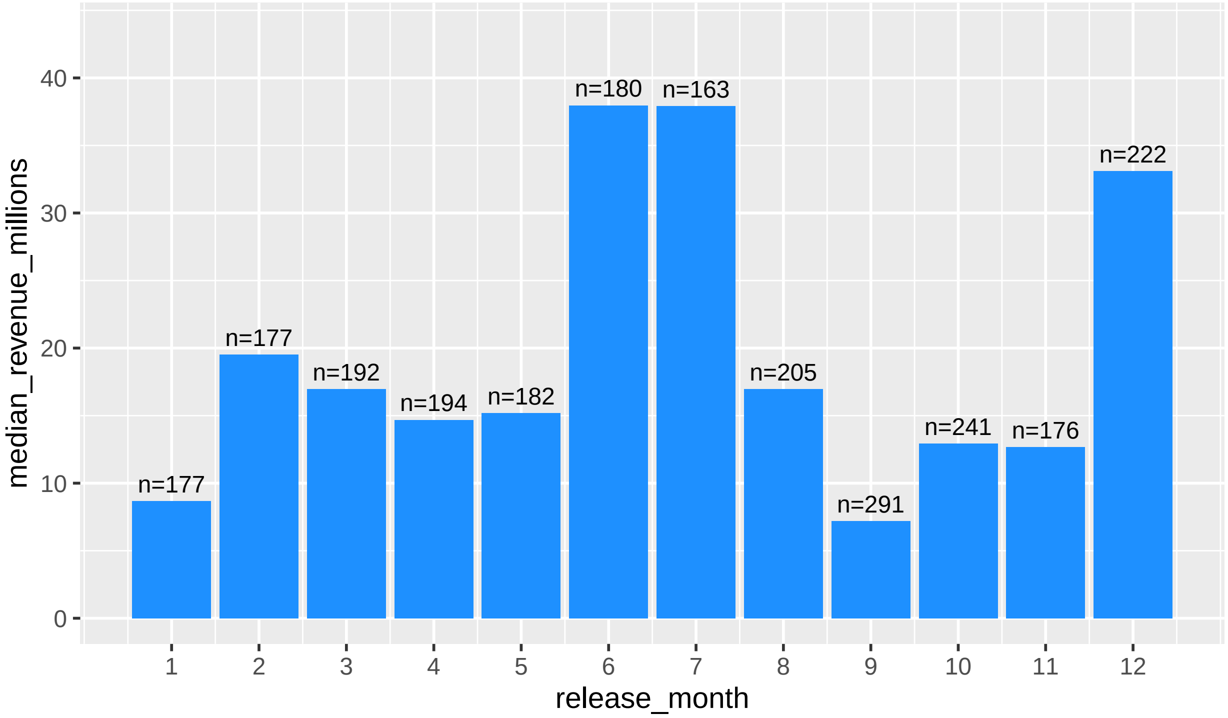
JSON string parsing is also used to extract various features about the movie production company and crew. For sparse values we set them to other and add flags if a movie was developed by one or more popular production companies.

Data Exploration & Trends

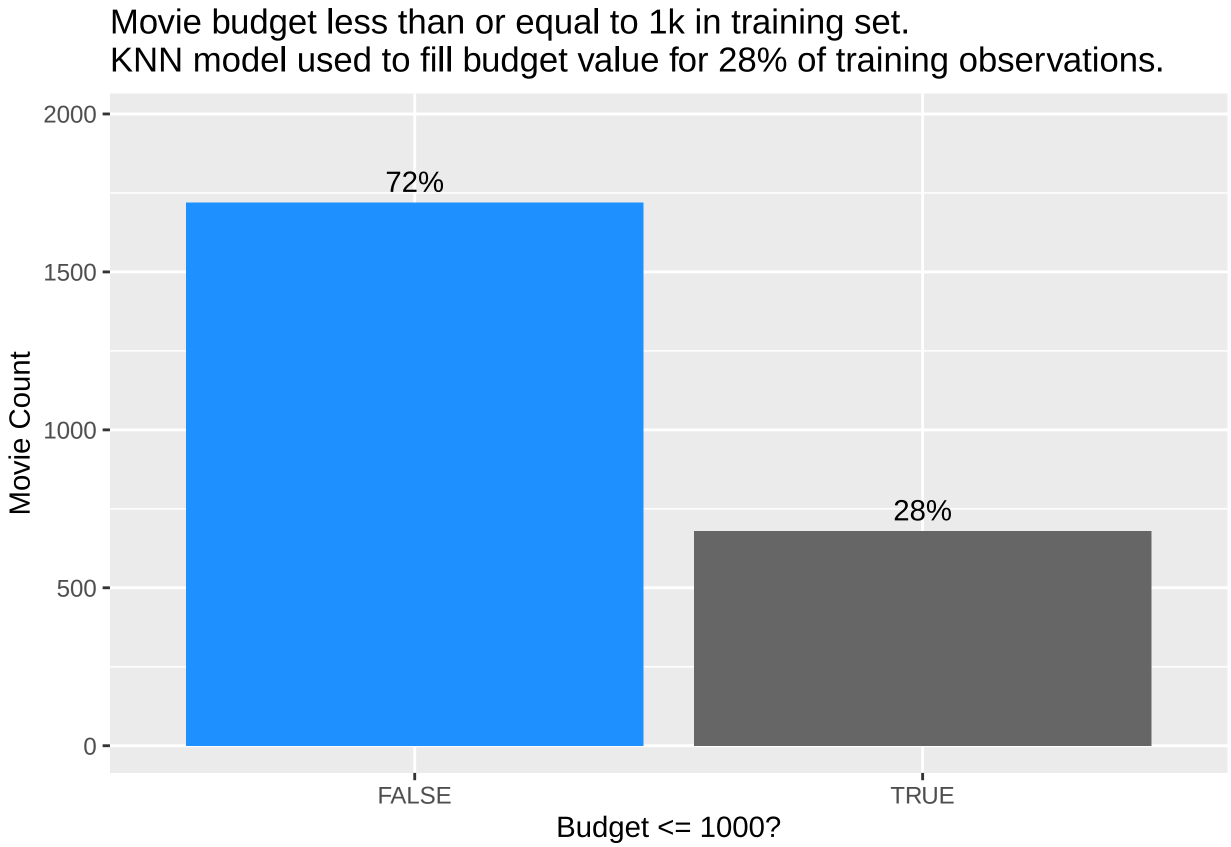
Next, we use data visualization to surface context and insights about the problem space.

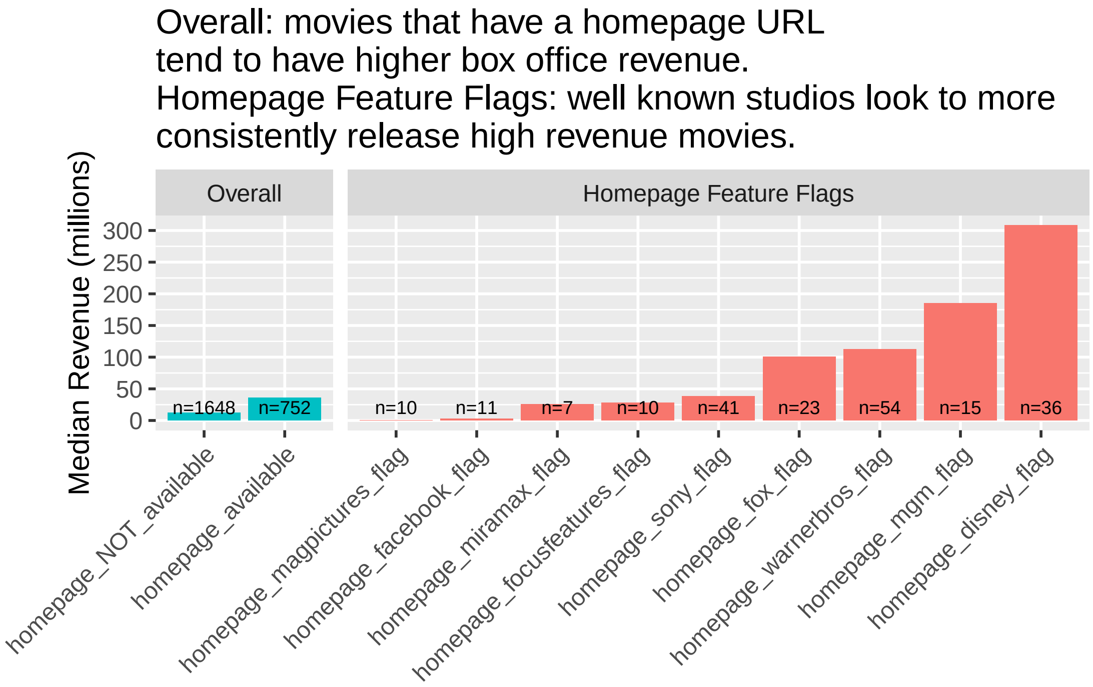

Top 10 grossing movies in training dataset
Furious 7
Avengers: Age of Ultron
Transformers: Dark of the Moon
The Dark Knight Rises
Pirates of the Caribbean: On Stranger Tides
Finding Dory
Alice in Wonderland
Zootopia
The Hobbit: An Unexpected Journey
The Dark Knight
0.0
0.5
1.0
1.5
Revenue (Billions)


Top 15: first listed production companies by median movie revenue
Median Revenue (millions)
0
20
40
60
80
100
120
140
160
180
200
n=21
n=21
n=23
n=26
n=28
n=33
n=34
n=36
n=36
n=48
n=55
n=58
n=75
n=95
n=125
n=135
Orion Pictures
Village Roadshow Pictures
BBC Films
Fox Searchlight Pictures
Miramax Films
TriStar Pictures
Metro–Goldwyn–Mayer (MGM)
Columbia Pictures Corporation
United Artists
Warner Bros.
Walt Disney Pictures
New Line Cinema
Columbia Pictures
Twentieth Century Fox Film Corporation
Paramount Pictures
Universal Pictures
First Listed Production Company



Movies released in June, July, and December
tend to have higher revenues. Early summer and holidays are historically
when studios look to attract movie goers with films that have tested well.


Movie budget less than or equal to 1k in training set.
KNN model used to fill budget value for 28% of training observations.
Movie Count
2000
1500
1000
500
0
72%
28%
FALSE
TRUE
Budget <= 1000?


Overall: movies that have a homepage URL
tend to have higher box office revenue.
Homepage Feature Flags: well known studios look to more
consistently release high revenue movies.

Modeling approach

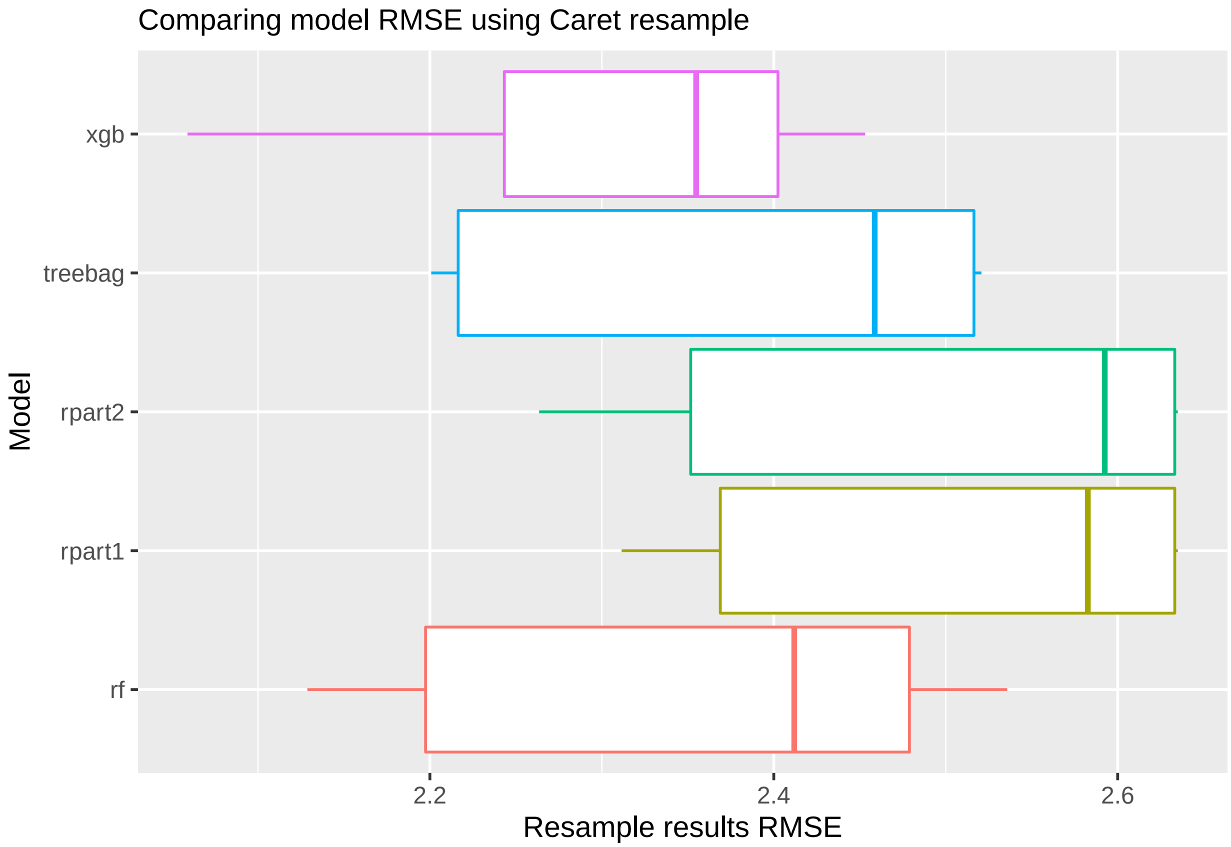
Next, we subset down the feature set to variables we want to prioritize as model inputs. Including all initial and derived character variables will drastically increase the feature space and increase the chance for overfitting. We'll use the Caret package to power our modeling phase. After converting character variables tofactors, Caret by default creates dummy variables for each of the factor variables.

We'll test tree models (rpart, rpart2, treebag, ranger, xgb) to see which performs best using cross validation RMSE. Tree models are selected due to interpretability strengths. We'llbe able to leverage feature importance charts to see which variables are most useful to the final prediction model. Five fold cross validation is used to assess model RMSE and do parameter tuning.Compared to partioning off an additional validation dataset, 5 k CV allows us to use more data for training and helps generate a conversative error estimate.

Features used for the final model:

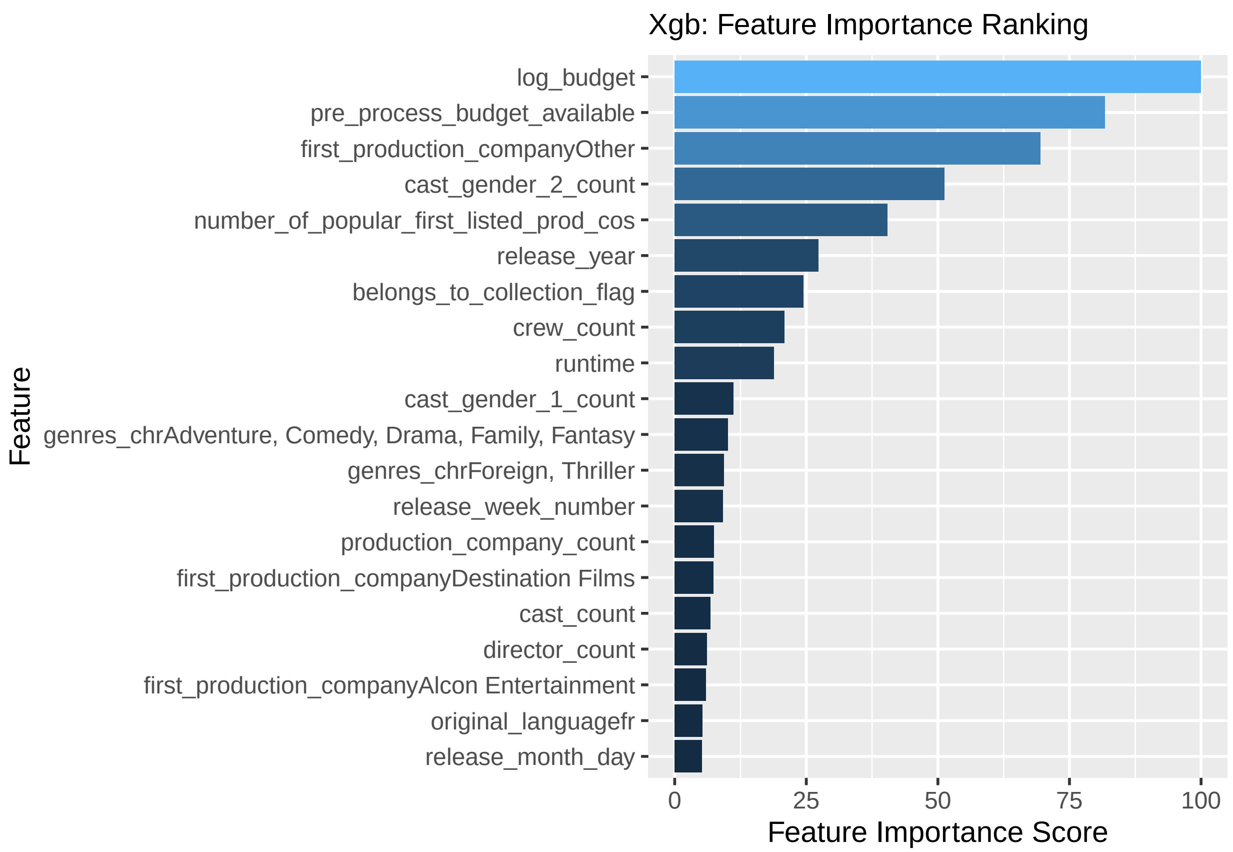
```
##  [1] "original_language"
##  [2] "runtime"
##  [3] "status"
##  [4] "pre_process_budget_available"
##  [5] "release_year"
##  [6] "before_2000_flag"
##  [7] "before_1980_flag"
##  [8] "release_year_bin"
##  [9] "release_month"
## [10] "release_month_day"
## [11] "release_week_number"
## [12] "release_day_of_week"
## [13] "release_year_quarter_str"
## [14] "title_length"
## [15] "belongs_to_collection_flag"
## [16] "tagline_available"
## [17] "homepage_available"
## [18] "homepage_disney_flag"
## [19] "homepage_sony_flag"
## [20] "homepage_warnerbros_flag"
## [21] "homepage_focusfeatures_flag"
## [22] "homepage_fox_flag"
## [23] "homepage_magpictures_flag"
## [24] "homepage_mgm_flag"
## [25] "homepage_miramax_flag"
## [26] "homepage_facebook_flag"
## [27] "genres_count"
## [28] "production_company_count"
## [29] "production_country_count"
## [30] "spoken_languages_count"
## [31] "cast_count"
## [32] "cast_gender_0_count"
## [33] "cast_gender_1_count"
## [34] "cast_gender_2_count"
## [35] "crew_count"
## [36] "director_count"
## [37] "producer_count"
## [38] "exec_producer_count"
## [39] "independent_film_flag"
## [40] "log_budget"
## [41] "first_genre_listed"
## [42] "first_production_company"
## [43] "first_production_country"
## [44] "genres_chr"
## [45] "number_of_popular_first_listed_prod_cos"
## [46] "log_revenue"
```

Comparing model performance is done by setting a seed before each model run, we can compare model performance downstream using the same folds (in other words, a fair comparison between models trained on the same data folds). We'll use Caret resamples() to compare model performance using the cross validation resampling results. We select xgb as the final model based on median resample RMSE.



Feature Importance

Using the final xgb model, we can highlight the most important features.



Results

Prediction Results

Prediction performance on the holdout test dataset:

Test set RMSE: 2.2646511

Conclusion

Takeaways

On average, predicted movie revenue is 2.26 times larger than the actual revenue or 1/2.26 times less the actual movie revenue. Compared to a baseline model which predicts the test set average, our final model RMSE is 56% less than a naive baseline.

We can see movie budget is the most important variable for predicting movie box office revenue. It'd be wise for startup movie execs to consider budget size as key factor to competing with established studios.

Movie startup execs might also consider partnering with well established studios and recruiting multiple directors / producers / exec products as these factors look to be predictive.

Size of cast and crew are also predictive variables to keep top of mind for movie execs. The more people contributing to the movie might help increase the overall quality.

Limitations

Only 3000 movies were considered for this project. Revenue was not adjusted for inflation. The community sourced dataset has impefections and missing data. International movie box office revenue can vary by online source. Human curated datasets can be used for initial model creation and analysis. However, a machine generated dataset might be more reliable for future work on this topic.

Future work areas

- Using additional data sources to further clean input data (i.e. revenue data for some movies doesn't match other online sources).
- Use TMDB API to generate larger sample set of movies.
- Explore more robust / complex models.
- Incorporate more data about the movie trailer and marketing content.
- Include features about the historicla awards the movie team has won.
- Consider adjusting revenue for inflation.
- Add more compute resources and utlize 10 fold cross validation with three repeats.
- Revisit leverage JSON R packages that could help with JSON parsing.