Streaming Service Movie Rater

Chris Shin, Dylan Clark-Boucher, Grace Bosma

Due: 12/22/2020

Objective

The purpose of this project was to both model and visualize IMDb ratings of movies and films currently available for streaming on various online platforms

Introduction

According to Statista, approximently 74% of homes have access to at least one streaming service in 2019 compared to about 52% in 2015. Additionally, as a result of the COVID-19 pandemic, many consumers have either purchased an additional streaming service subscription or increased time spent streaming. It is clear that movie and TV streaming services are increasing in popularity and are integrated into many of our day to day routines. Just last year, the movie and television streaming industry was valued $42.60 billion USD. Information on the popularity of movies available for streaming is critical to streaming platforms like Netflix, Disney+, Hulu to maintain customer interest in their platform rather than their competitors.

Methods

Data Cleaning (see cleaning_data.Rmd for details)

The data for this analysis was obtained from either Kaggle or International Movie Database (IMDb) websites. Data regarding streaming service platform and ratings from IMDb and Kaggle was obtained from Kaggle whereas information on actors involved in movies were obtained from IMDb.

The Kaggle Movie data set was downloaded on 12/4/2020. This dataset contained information on movies available on various streaming services as of May 22, 2020 from US locations. In the dataset there were 16,631 movie titles (entries) and 16 variables including imdb_id (IMDb-specific id), title, year (of release), age_rating (7+, 13+, 16+, 18+, or all), imdb_score (rating from IMDb, which is aggregated from IMDb registered users who cast a vote from 1 to 10 per title), rt_score (TOMATOMETER score from Rotten Tomatoes), netflix (whether it was available on Netflix), hulu (whether it was available on Hulu), amazon-prime (whether it was available on Amazon Prime Video), disneyplus (whether it was available on Disney+), type (whether it was a movie or not), directors, genres, country (of origin), languages (in which the movie is available, either voice dubbed or subtitles), and runtime (length of movie). During cleaning, movies that did not match to titles on IMDb were excluded. The final movie data set consisted of 11448 titles.

Data made publicly available on the website IMDb is updated daily and available for download. The dataset "name.basics.tsv.gz" was downloaded from IMDb on 12/4/2020 and used in this project. This data set contained information on actors and held information on 6,414,148 movie production team members, four "Known For" film names associated with each actor or actress, and additional roles the actor or actress held (i.e. writer, producer, ...). During cleaning, those that did not hold an actor/actress role or had no

"known for" titles were excluded from analyses. The final, cleaned IMBd actors dataset held information on 2,426,996 actors or actresses.

Using the actors dataset, we created three variables specific to each actor/actress to use to describe movie titles. Score1 takes into account the number of actors/actresses involved in the titles the specific actor/actress was known for, score2 considers the number of titles and additional roles the actor/actress took on, and score3 prioritizes producers, directors, and writers, respectively.

Data Merging (see merging_data.Rmd for details)

We merged the movies dataset from Kaggle and the actors dataset from IMDb using the unique imdb_id associated with each title. The final merged dataset contained 139,772 observations.

Prediction

Data for prediction using linear regression and random forest required complete cases, and the variables we chose to include were hulu, disneyplus, netflix, amazonprime, year, rt_score, imdb_score, actor_score1, actor_score2, and actor_score3.

The outcome to predict was imdb_score for both methods.

Random Forest

Random forest is a method of fitting multiple decision trees using a technique called bagging, which introduces randomness into each tree building process and averages the results across multiple trees. Random forest can be used for regression and classification tasks but requires complete data sets. We chose to implement random forest to observe the predictive ability of variables we chose to include in the linear regression models. Random forest is ideal for our situation because our final dataset is not too large.

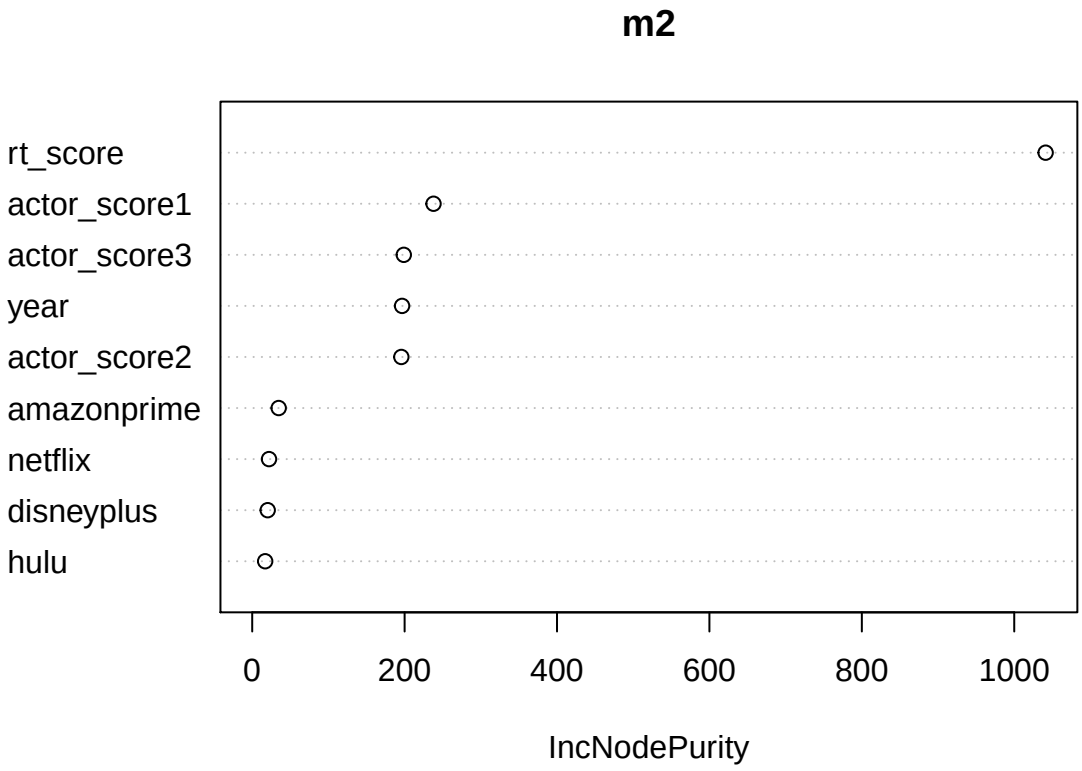
```
load("lm_data.rda")
library (randomForest)

set.seed(123)


model_form <- imdb_score ~ rt_score + year + disneyplus + hulu +
  amazonprime + netflix + actor_score1 + actor_score2 + actor_score3

m1 <- randomForest(
  formula = model_form,
  data = lm_data[complete.cases(lm_data),]
)
which.min(m1$mse)
```

```
## [1] 452
```

The default forest used 500 trees and chose 3 predictor variables at each split. When we plot the model, we see that we obtain the lowest error rate with 452 trees.

Because our goal for the random forest was to determine important variables, we did not continue with tuning the random forest and chose to look at the variable importance plot using a slightly modified random forest.

```r
m2 <- randomForest(
  formula = model_form,
  data    = lm_data[complete.cases(lm_data),],
  ntree = which.min(m1$mse)
)
varImpPlot(m2,type=2)
```

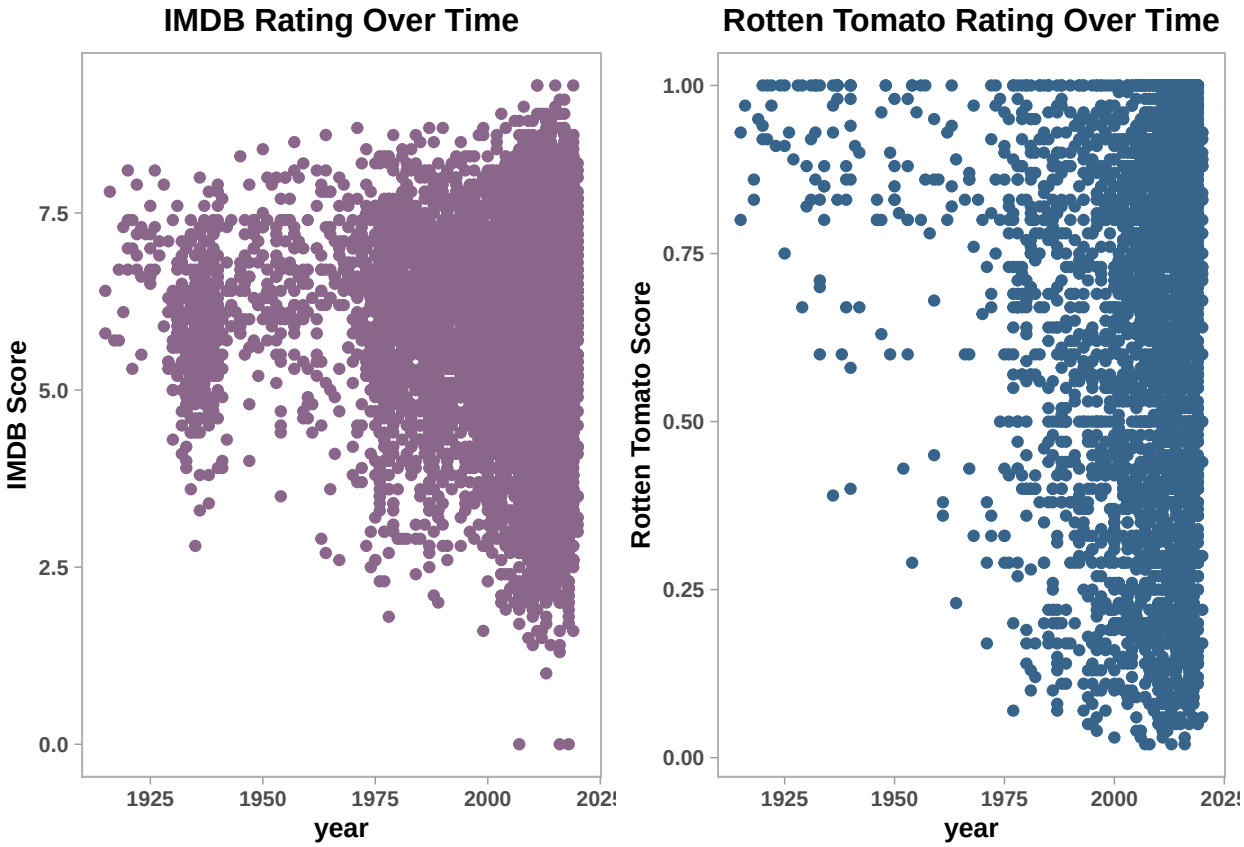


From this plot we see that rt_score (TOMATOMETER score from Rotten Tomatoes) is highly influential in determining imdb_score (rating from IMDb), followed by actor_score1, year (of release), actor_score2, and actor_score3. Availability on streaming service was not found to be highly influential in determining IMDb rating.

Linear Regression

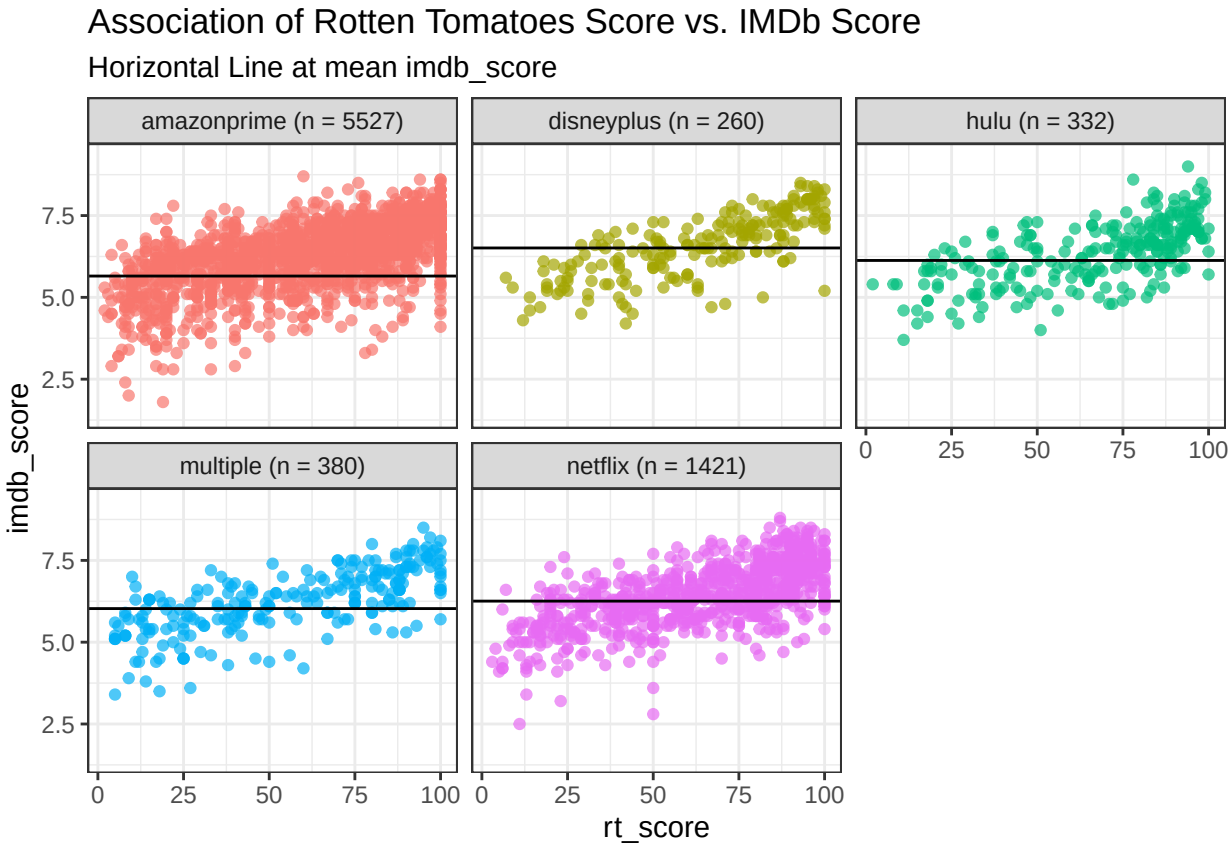
See section on RShiny Application for details.

Vizualization



- not married to these plots just figured we should show something

```
streaming_services = c("hulu (n = 332)", "disneyplus (n = 260)", "netflix (n = 1421)", "amazonprime (n =

lm_data$availability = streaming_services[apply(lm_data[,c("hulu", "disneyplus", "netflix", "amazonprime
lm_data = lm_data %>%
  mutate (availability = ifelse (sum (hulu + disneyplus + netflix + amazonprime) > 1,
                                 "multiple (n = 380)",
                                 availability))


lm_data %>%
  group_by (availability) %>%
  mutate (mean_imdb = mean (imdb_score, na.rm = T)) %>%
  ungroup() %>%
  ggplot ( aes (x = rt_score, y = imdb_score, color = availability, group = availability)) +
  geom_point ( alpha = 0.7) +
  geom_hline(aes(yintercept=mean_imdb)) +
  facet_wrap(~availability, nrow = 2) +
  theme_bw() +
  labs(title = "Association of Rotten Tomatoes Score vs. IMDb Score",
       subtitle = "Horizontal Line at mean imdb_score") +
  theme(legend.position = "none")
```



When we plot rt_score vs. imdb_score, we see that there is a positive association between the two across all platforms.

RShiny Application

An RShiny Application was created using our data set to display our results and can be accessed at: `https://dylanclark-boucher.shinyapps.io/movie_rater_shiny/`

There are four tabs in the upper left that allows you to navigate through the application; Ratings Filter Tool, Ratings by Category, Score Prediction, and About. The Ratings filter tool presents a histogram and box plot of movie ratings and allows the user to filter based on recommended age of target audience, streaming service, Rotten Tomato Score, and release year and select a preferred choice of graph color and bin width. The Ratings By Category page presents three tabs on the left side bar. The first, Streaming Service, presents the distributions of movie ratings for each streaming service in a box plot format. The second, Age Rating, presents the distributions of movie ratings for by age group in a box plot format. The third, presents a scatter plot of IMBd movie ratings plotted against Tomatometer Score. Below the scatter plot the user is again able to filter but age group and streaming service. THe third tab, Score Prediction, [FILL IN LATER XXX]. The final tab, About, briefly summarized the background, app, model and contributors.

- potentially integrate an example here?

Discussion

- conclusion from lm model

  - what variables are important
  - did we predict well
- future direction

Contributions

Our group collaborated well and distributed work evenly among all members. While all members participated in searching for potential data sets, organizing, drafting and proofreading, Dylan created the RShiny app and organized a shared folder on the biostat computing cluster, Chris cleaned the Kaggle and movie rating data sets, and Grace cleaned the actors data set and created a "cast score" to be used in model.

- be sure to update and include model