Making a Magnificent Movie

Marin Lolic, Ryan Viti, and Seth Galluzzi

Aim

"You gotta watch it! It's an instant classic! I don't care what Rotten Tomatoes says."

- *Coworker, every two months about the movie 'Infinite'*

"Did you see the new Top Gun movie? That Tom Cruise is so handsome."

- *Grandma, three times in the same conversation*

"What should we watch tonight?"

- *Girlfriend, every night - usually followed by "What should we eat."*

Selecting the perfect movie to watch has become somewhat of a tradition in American culture. Sifting through various streaming services, trying to find that one title that is right is an agonizing and tedious process that most would rather avoid. During this process, it is common to search reviews to determine what others recommend. One of the most popular online sources that ranks titles is the Rotten Tomatoes Tomatometer. The Tomatometer was created to separate the good movies from the bad movies in an effort to help viewers determine what to watch. A film's Tomatometer score is determined by numerous movie critic reviews. The categories within the score are *Rotten* for movies with poor reviews, *Fresh* for movies with good reviews, and *Certified Fresh* for movies with consistently excellent reviews. It appears the Tomatometer usually gets it right, with movies like *Top Gun* receiving a *Certified Fresh*, while movies like the dreadful *Fool's Gold* receiving a *Rotten*. However, there are some movies, such as epic cult classic *Space Jam,* that have clearly been unfairly categorized as *Rotten (*though it is important to note the correct categorization of *Space Jam 2 -- Rotten).* Furthermore*,* It is painful to think of the millions of people tricked into spending two and a half hours watching *Certified Fresh-- Star Wars: The Last Jedi* only to be majorly disappointed. Some of these head scratchers could make one wonder how the critics come up with this stuff, and what other factors could influence a movie's Tomatometer score. Thus, the aim of this exploration is to determine different movie characteristics associated with the Tomatometer, and to use Bayesian statistics to create models that can predict Tomatometer score. Hopefully, we can then use our insight to better determine what movies are good and what movies are bad, making the movie selection process a lot less painful.

Executive Summary

We tackle the problem of defining a probability distribution for our outcome variable: the critics' rating of a particular film. More specifically, we will categorize each movie using the critics_score variable, which uses three ratings, each summarized above: *Rotten, Fresh,* and *Certified Fresh.*

Our goal is to predict which movies received higher ratings by extracting their biggest sources of variance via data exploration. Using these as the categories of our response, we will explore relationships and trends within our remaining dataset to find the best combination of covariates to include in our final models. We take a Bayesian logistic regression modeling approach at first to see how one of the most tried and true methods of machine learning classification performs on our relatively small dataset. Further, we binarize our outcome variable and rerun a Bayesian logistic regression model with only our most important covariates to improve our results. Enhancing results from that of a logistic regression, we finally extend our analysis into Bayesian Additive Regression Trees as a bootstrapping method to combine many simple regression trees together to estimate a robust posterior probability distribution. We conclude that audience scores of movies have the largest magnitude of impact on critics rating. More specifically, a higher audience score on average calls for a higher critics rating on a given movie.

The Data

The dataset was obtained online from Duke University. The dataset consists of information about movies released before 2016. Some of the information includes the title, genre, release date, IMDB rating and Rotten Tomatoes critics score . We chose this dataset specifically due to its abundance of information regarding movies as well as the popularity of Rotten Tomatoes as a movie rating staple. The variable critics_rating will be the response variable during this exploration. Critics_rating is a categorical variable consistent with the Tomatometer. Again, the three categories of critics_rating are *Rotten, Fresh, and Certified Fresh.* Many of the columns in the dataset were also removed during the cleaning process including the five actor columns and the url column. Furthermore, the column good_or_bad was added to aid in the exploration. This new column is categorical and is coded 0 for movies with a *Rotten* critics_rating and 1 for movies with a *Fresh* or *Certified Fresh* critics_rating.

Data Exploration

To get a better understanding of the data, many of the variables were explored and some baseline data was collected. Figure 1 gives some interesting intel about the data. The average runtime of the movies in the data set is 106 minutes, while the average audience score is 62 and the average critic score is 58. Based on the Rotten Tomatoes website, critics score is the basis for classifying movies as *Certified Fresh, Fresh,* or *Rotten.* Thus, we will avoid using critics' scores for model development. However, it is interesting to note that audience score is similar to critics score in most of the data's quartiles, and also has a similar standard deviation.

Figure 1: Numeric Data Summary

	runtime	thtr_rel_year	thtr_rel_month	thtr_rel_day	imdb_rating	imdb_num_votes	critics_score	audience_score
count	650.000000	651.000000	651.000000	651.000000	651.000000	651.000000	651.000000	651.000000
mean	105.821538	1997.941628	6.740399	14.416283	6.493088	57532.983103	57.688172	62.362519
std	19.445047	10.974501	3.554223	8.861167	1.084747	112124.386910	28.402971	20.222624
min	39.000000	1970.000000	1.000000	1.000000	1.900000	180.000000	1.000000	11.000000
25%	92.000000	1990.000000	4.000000	7.000000	5.900000	4545.500000	33.000000	46.000000
50%	103.000000	2000.000000	7.000000	15.000000	6.600000	15116.000000	61.000000	65.000000
75%	115.750000	2007.000000	10.000000	21.000000	7.300000	58300.500000	83.000000	80.000000
max	267.000000	2014.000000	12.000000	31.000000	9.000000	893008.000000	100.000000	97.000000

Another interesting piece of the data to explore are the value counts of some of the categorical variables. Our response variable, critics_score contains a lot of *Rotten* movies, while the genre of most movies seems to be *Drama.*

Figure 2: Counts of critics_score categories and genre types

```
Rotten              307
Fresh               209
Certified Fresh     135
Name: critics_rating, dtype: int64
```

```
Drama                          301
Comedy                          85
Action & Adventure              65
Mystery & Suspense              59
Horror                          23
Other                           15
Art House & International       14
Animation                        9
Science Fiction & Fantasy        9
Musical & Performing Arts        8
Documentary                      3
Name: genre, dtype: int64
```

Exploring the correlations between some of the variables in figure 3 shows there are strong, positive correlations between IMDB_rating, audience_score, and critics_score.

Figure 3: Correlation Matrix

	Unnamed: 0	runtime	thtr_rel_year	thtr_rel_month	thtr_rel_day	dvd_rel_year	dvd_rel_month	dvd_rel_day	imdb_rating	imdb_num_votes	critics_score	audience_score
Unnamed: 0	1.000000	-0.040722	-0.003676	-0.002700	0.038116	-0.006841	-0.013576	0.028463	0.017521	0.021794	-0.011316	0.011352
runtime	-0.040722	1.000000	-0.104377	0.220987	0.037396	-0.081902	-0.033309	0.024235	0.268240	0.347215	0.172499	0.180963
thtr_rel_year	-0.003676	-0.104377	1.000000	-0.000507	0.107400	0.660465	0.036678	-0.003763	-0.030003	0.155906	-0.080785	-0.054079
thtr_rel_month	-0.002700	0.220987	-0.000507	1.000000	0.116228	-0.009989	-0.168612	0.028240	0.072278	0.106326	0.032404	0.032690
thtr_rel_day	0.038116	0.037396	0.107400	0.116228	1.000000	0.041322	-0.025531	0.001797	0.021891	0.067976	0.013689	0.019221
dvd_rel_year	-0.006841	-0.081902	0.660465	-0.009989	0.041322	1.000000	-0.006506	-0.068134	-0.015263	0.093711	0.015212	-0.062970
dvd_rel_month	-0.013576	-0.033309	0.036678	-0.168612	-0.025531	-0.006506	1.000000	-0.030141	0.064806	0.030912	0.031070	0.057349
dvd_rel_day	0.028463	0.024235	-0.003763	0.028240	0.001797	-0.068134	-0.030141	1.000000	0.026912	-0.016419	-0.024228	0.021644
imdb_rating	0.017521	0.268240	-0.030003	0.072278	0.021891	-0.015263	0.064806	0.026912	1.000000	0.331152	0.765036	0.864865
imdb_num_votes	0.021794	0.347215	0.155906	0.106326	0.067976	0.093711	0.030912	-0.016419	0.331152	1.000000	0.209251	0.289813
critics_score	-0.011316	0.172499	-0.080785	0.032404	0.013689	0.015212	0.031070	-0.024228	0.765036	0.209251	1.000000	0.704276
audience_score	0.011352	0.180963	-0.054079	0.032690	0.019221	-0.062970	0.057349	0.021644	0.864865	0.289813	0.704276	1.000000

In figures 4 and 5, we gain a better understanding of the data as a whole, as well as a closer inspection of audience_score and critics_score in relation to critics_rating. The scores are normally distributed. It also appears that both scores seem somewhat bimodal. It is interesting to see that the *Certified Fresh* and *Fresh* categories resemble each other more closely in the audience score graph than in the critics score graph. The relationship between the two variables appears to be linear based on the scatter plots.

Figure 4: Exploration of critics score and audience score

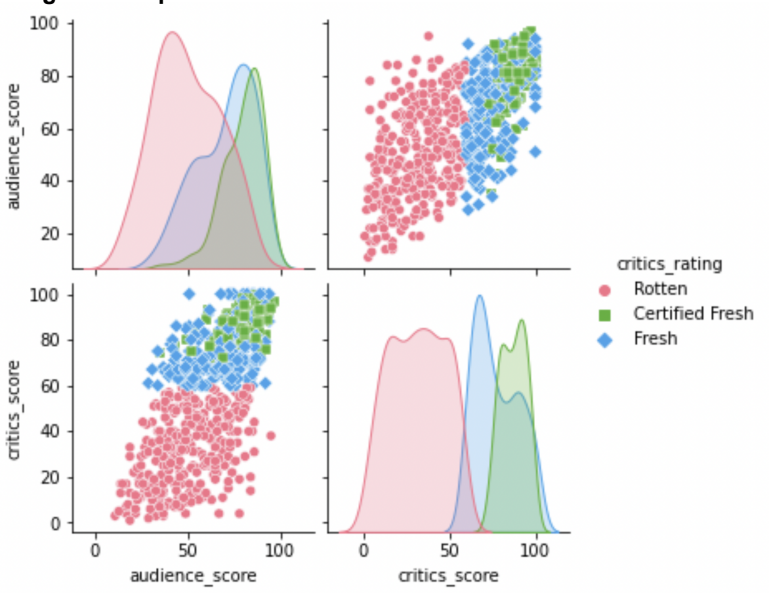
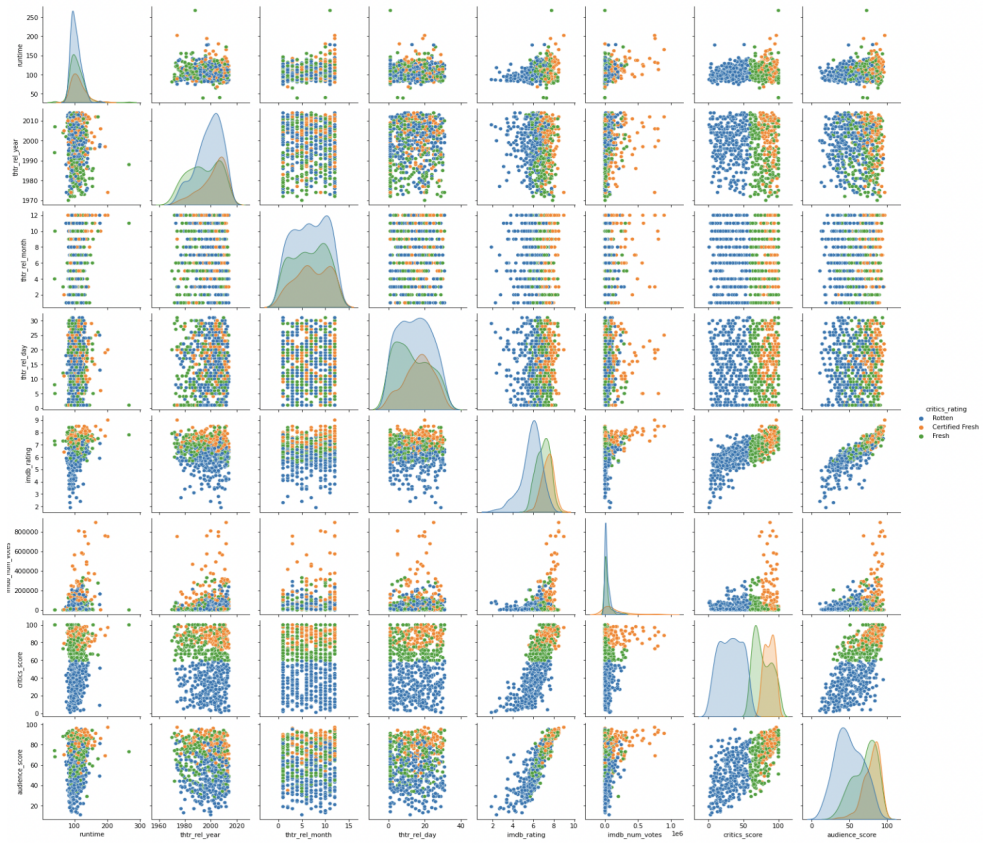


Figure 5: Overall Visualization of Data



In the movie industry, January, February, August, and September are known as dump months. During dump months, the movie industry releases movies that perform poorly during testing, have lower expectations, and are less likely to win any major awards. To gain further insight of dump months, proportions of the variable critics_rating are separated based on month in figure 6. There appears to be a small dip in *Certified Fresh* movies during dump months. However, if we combine the *Fresh and Certified Fresh* categories the proportions are similar across all the months. Still, this was an important piece of the puzzle to explore during the modeling phase.

Figure 6: Critics Rating Proportions by Month

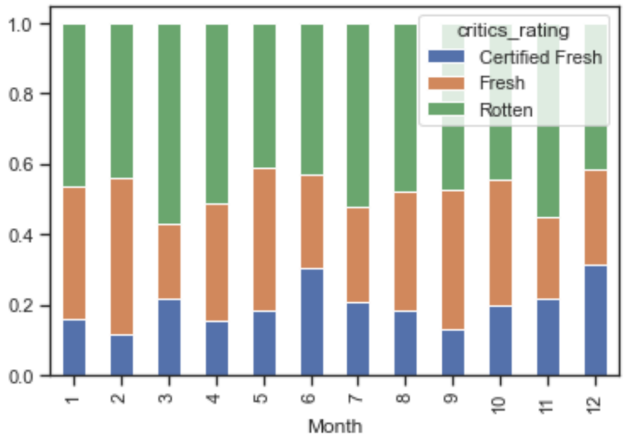
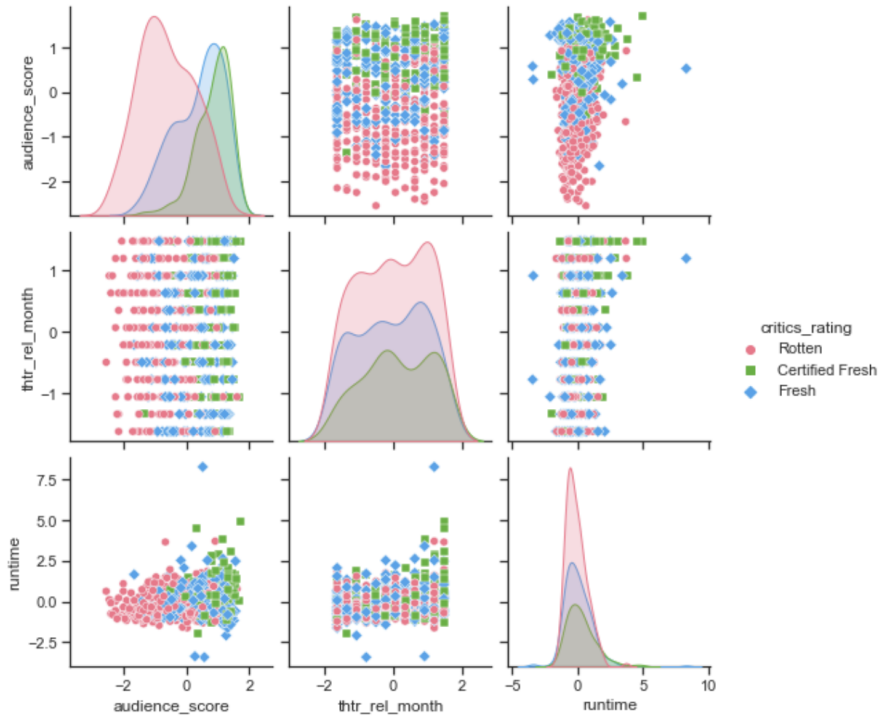


Figure 7: Pairplot of Audience Score, Month of Theater Release, and Runtime



Bayesian Logistic Classification

To gain further insight of the predictor variables for using logistic regression for classification on the movie data, we will perform MCMC using PyMC3. The variables explored will be audience_score because of the correlation to critics_score, thtr_rel_month because of the possibility of dump months, and runtime as a wildcard because long movies can be exhausting for a prospective viewer. The values were normalized and glm was used for the logistic model. To start, default priors were used. These are weak priors understood as $p(\theta) = N(0, 10^{12}I)$. The results depicted in figure 8 as well as the trace plot and forest plot in figures 9 and 10 respectively, indicate audience_score may be the only predictor that is useful when determining critics rating. The 94% credible interval of thtr_rel_month and runtime both contain zero. Thus, neither should be considered when exploring a model. Furthermore, figure 7 above shows there is a lot of overlap between the *Fresh* and *Certified Fresh* classifications. This could make a model with audience_score as a predictor inaccurate because of its inability to distinguish between the two classifications.

Figure 8: PyMC Summary

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	-0.10	0.10	-0.30	0.08	0.0	0.0	6254.0	2878.0	1.0
audience_score	-1.62	0.13	-1.87	-1.40	0.0	0.0	5944.0	3398.0	1.0
thtr_rel_month	0.03	0.10	-0.15	0.23	0.0	0.0	5987.0	3307.0	1.0
runtime	-0.04	0.10	-0.23	0.16	0.0	0.0	4762.0	3012.0	1.0

Figure 9: PyMC Trace Plot

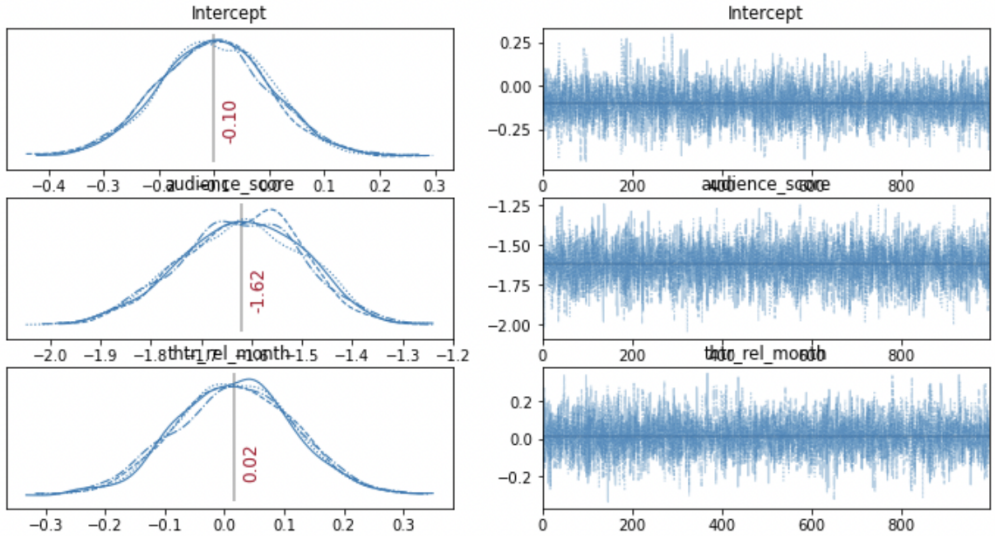
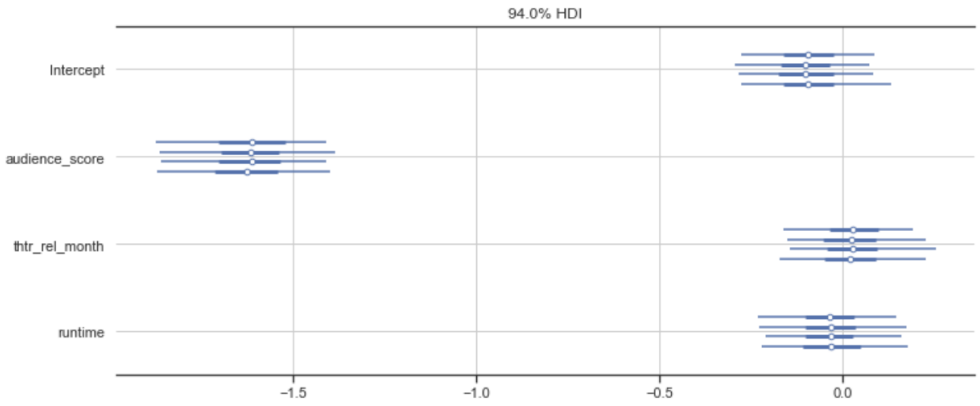


Figure 10: PyMC Forest Plot



After exploring the logistic regression, it is clear the model could be improved in a few ways. The first way is to binarize the response variable. Instead of using the tomatometer standard of *Rotten, Fresh, and Certified Fresh, we can combine* the categories *Fresh and Certified Fresh* together and make a new categorical variable: good_or_bad. The new variable is categorized as 0 when the movie is rated bad, and 1 when the movie is rated good. By doing this, we will no longer be able to identify the elite movies in the set, but we will give ourselves a better opportunity to create an accurate model. We will also focus our efforts on the predictor audience_score, since the other predictors provided little value in differentiating between good and bad movies. Finally, we will use a discrete beta prior with alpha 1 and beta 1 for the good_or_bad response variable, and a continuous normal distribution prior with mean 0 and standard deviation 1 for the audience_score predictor. We have done this because throughout our research, in obtaining a useful posterior probability distribution for a given movie, we have no reason to deviate from the average values for our chosen covariates. The results of the posterior distributions, displayed in figures 11 to 14, show the coefficient on audience_score has a credible interval centered at 1.59 with a standard deviation of 0.12. Using PyMC to perform Bayesian analysis has proved useful when determining that audience_score is a useful predictor in determining whether a movie is rated well by critics.

Figure 11: PyMC Summary 2

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
Intercept	0.10	0.10	-0.08	0.29	0.0	0.0	17673.0	14413.0	1.0
audience_score	1.59	0.12	1.35	1.82	0.0	0.0	18145.0	13663.0	1.0

Figure 12: PyMC Trace Plot 2

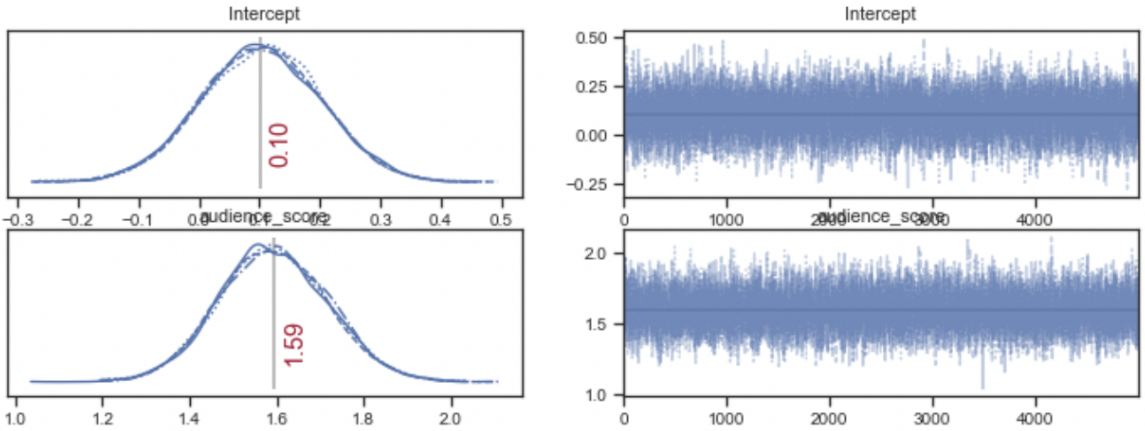


Figure 13: PyMC Forest Plot 2

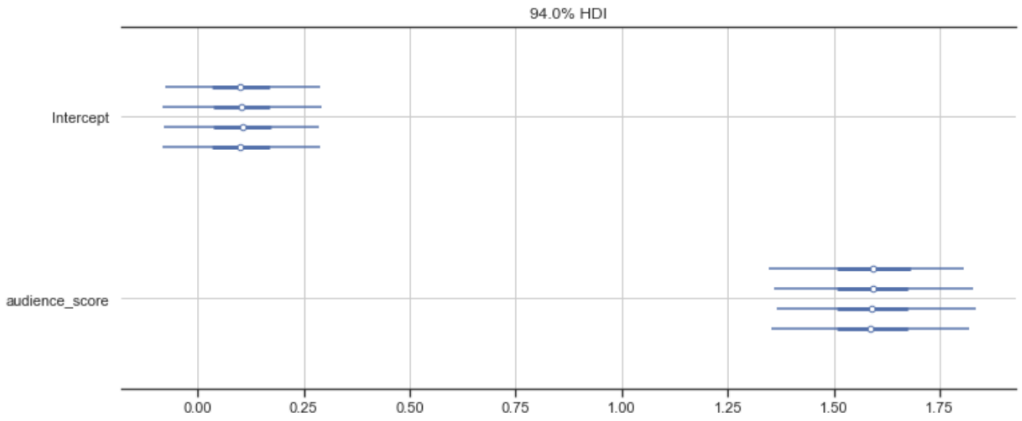
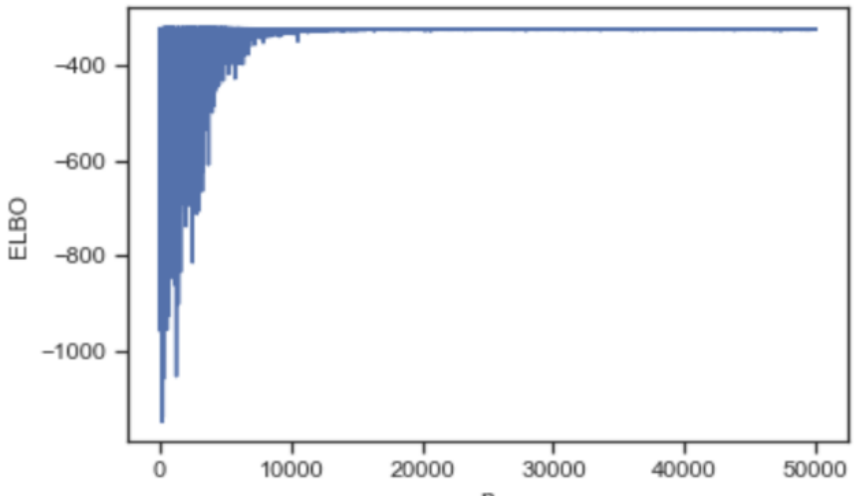


Figure 14: ELBO using ADVI



Model Evaluation

To compare our two logistic regression models, we used the Arviz package to calculate WAIC values for each model, indicating a relative performance measurement amongst several models. For our first logistic regression with three covariates and a multilevel response, we found a value of -322.77, whereas our second logistic regression which employed solely audience score as a predictor and used our engineered, binary response, the WAIC emerged as -321.06. As a result, we can conclude that of the two models compared by the WAIC metric, our first model with all predictors included performed better, as it has the smallest value. This fact will be supported by the Bayesian version of a traditional machine learning technique we explore next.

Bayesian Additive Regression Trees

Bayesian Additive Regression Trees (BART) are an application of Bayesian machine learning principles to an ensemble of decision trees. First introduced in 2008 by Chipman et al, BART takes concepts from older approaches such as boosting and imposes structural priors on the model. Most of these priors have the goal of minimizing overfitting by keeping any one tree from becoming too influential within the ensemble. The first prior controls the depth of the trees, and typical BART formulations lead to trees with 2 or 3 levels of depth. The second prior governs the contribution of a given tree to the total model by shrinking each tree's predictions toward zero. The third prior is the standard deviation of the response variable, which can utilize either a naive prior (the sample standard deviation of y) or more informed priors if those are available. Finally, BART requires selecting a total number of trees and the number of burn-in samples, though results are robust to a wide variety of choices.

The fitting process requires what the authors describe as "tailored version of Bayesian back-fitting MCMC that iteratively constructs and fits successive residuals" and is similar to Gibbs sampling. Like other tree-based methods of machine learning, BART tracks which predictors appear in trees most often and uses this information to assess the relative importance of the predictors. While the name BART suggests a method for estimating continuous dependent variables, it has been extended to classification problems by the original authors. Numerous packages exist in Python, R and other languages for rapid implementation of BART.

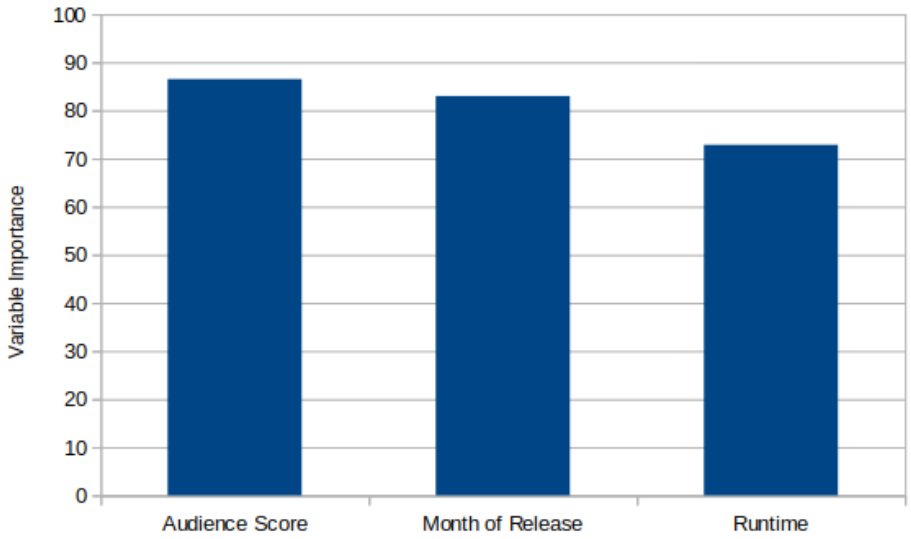
We implemented BART through the “BART” library in R. As this library requires a binary response variable, we aggregated “Certified Fresh” and “Fresh” into a dummy variable with value “1”, while ‘’Rotten” took value “0”. We used the library’s default settings on priors, which are set to match the recommendations in the original Chipman paper. We used a total of 200 trees, and the MCMC algorithm ran for a total of 1100 iterations, of which the first 100 were discarded as burn-in. Finally, we utilized the same three predictors as in our first Logistic Regression, namely Audience Score, Month of Release, and Runtime.

Figures 15 and 16 below summarize our results. Overall, we achieved an accuracy of 76.7%, approximately balanced between Type I and Type II errors; this accuracy was much higher than the no-information rate of 52.8%. Our variable importance measures, based on mean decrease in Gini Index, suggest that all three variables contributed to the model, in contrast to what we saw in the logistic regression. A possible reason for the discrepancy is the highly non-linear nature of decision trees.

Figure 15: BART Summary Output

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 234  79
         1  73 265

               Accuracy : 0.7665
                 95% CI : (0.7321, 0.7985)
    No Information Rate : 0.5284
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.532

 Mcnemar's Test P-Value : 0.6851

            Sensitivity : 0.7622
            Specificity : 0.7703
         Pos Pred Value : 0.7476
         Neg Pred Value : 0.7840
             Prevalence : 0.4716
         Detection Rate : 0.3594
   Detection Prevalence : 0.4808
      Balanced Accuracy : 0.7663
```

Figure 16: BART Variable Importance



Summary

Through this process, we gained a better understanding of how movie variables such as audience score, release month, and runtime could be used to determine a movie's Tomatometer score. We also explored many different variables related to the Tomatometer, created clear visualizations to inform others about our findings, and used Bayesian statistics (via PyMC) to determine important predictor variables of different logistic regressions. We then highlighted the usefulness of BART as an application of Bayesian machine learning, and used BART to gain a deeper understanding of our predictor variables. As is usually the case, our new found understanding led to more questions, and shed some light on what further exploration could be done. For example, examining how we could predict critics score (a variable closely related to critics rating) by using audience score, release month, and runtime using a linear regression could be an insightful and useful next step in research and development. Finally, the next time your significant other asks you what to watch, know that by exploring this research you can make a decision effectively and efficiently without taking too much time sifting through subscriptions.

This is our **Exploration Folder.** It contains our research, code, and other assorted documents.

References

About. Rotten Tomatoes. (n.d.). Retrieved August 4, 2022, from https://www.rottentomatoes.com/about#:~:text=The%20Tomatometer%20score%20represents%20the,Fresh%20red%20tomato

Billbasener. (2021, April 2). *Pymc3 bayesian logistic regression classification*. Kaggle. Retrieved August 4, 2022, from https://www.kaggle.com/code/billbasener/pymc3-bayesian-logistic-regression-classification

Billbasener. (2021, October 5). *Bayesian linear regression in PYMC3*. Kaggle. Retrieved August 4, 2022, from https://www.kaggle.com/code/billbasener/bayesian-linear-regression-in-pymc3

Generalized Linear Models (formula)¶. statsmodels. (n.d.). Retrieved August 4, 2022, from https://www.statsmodels.org/dev/examples/notebooks/generated/glm_formula.html

Index of /~CR173/sta523_fa16/data/movies. (n.d.). Retrieved August 4, 2022, from http://www2.stat.duke.edu/~cr173/Sta523_Fa16/data/movies/

Katievickers. (2021, May 23). *Rotten tomatoes: Critic ratings study*. Kaggle. Retrieved August 4, 2022, from https://www.kaggle.com/code/katievickers/rotten-tomatoes-critic-ratings-study/notebook

Movies: TV shows: Movie trailers: Reviews. Rotten Tomatoes. (n.d.). Retrieved August 4, 2022, from https://www.rottentomatoes.com/

Are new movies longer than they were 10, 20, 50 years ago? . Towards Data Science. Retrieved August 7th, 2022, from https://towardsdatascience.com/are-new-movies-longer-than-they-were-10hh20-50-year-ago-a35356b2ca5b