Predicting Movie Popularity via Bayesian Analysis

Sameer Patel

Executive Summary

For this project, the team decided to explore the following research question:

Can Bayesian inference and associated statistical methods be used to determine the popularity of a movie according to select characteristics of that movie?

To answer this question, the team applied key components of Bayesian inferential statistics in R to build a predictive model; the model was then used to predict a movie rating based on specific values of the explanatory variables upon which the model was built.

The dataset used to build the model is a well-known dataset in data science - the IMDb/Rotten Tomatoes "movies" dataset. After appropriate data cleaning, Exploratory Data Analysis (EDA), and model tuning, the final model was used to determine a rating for the movie “Godzilla (2014)”, shown below:

Movie <fctr>	Estimated.IMDB.rating <dbl>	Real.IMDB.rating <dbl>
Godzilla	6.642862	6.5

The model yielded an estimated movie rating of 6.64, a difference of less than 3% from the actual rating, confirming the robustness of Bayesian inferential statistics with regards to the research question.

The following sections provide further details for each step of the analytical process that yielded this result; including the data cleaning/EDA processes, model fitting, prediction, and supporting visualizations.

Data Source and EDA

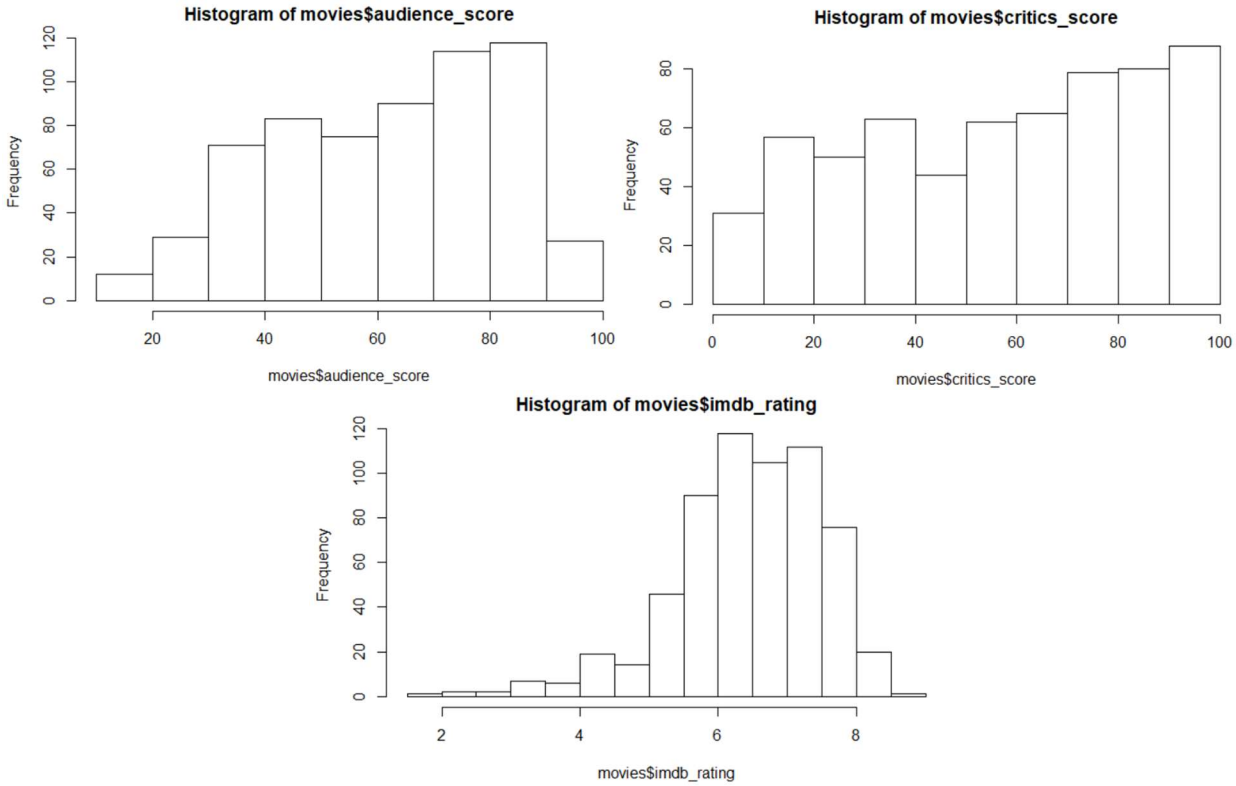
The “movies” dataset, sourced from Rotten Tomatoes and IMDb, contains 651 movies with information contained in 32 features. As with all forays into statistical analyses, appropriate data cleanup and exploration is required in order to extract meaningful information and properly fit a model.

The team began by using the na.omit() function in R to remove incomplete cases. Accessory columns, including non-factor strings (URLs, titles, actor/actress names, etc.), DVD release information, and release month/day were then dropped from the set of explanatory variables, as they would not contribute meaningfully to a Bayesian model. The resulting dataframe contained 619 cases with 17 explanatory variables.

Going forward with the EDA aspect of the process, the team identified 5 exploratory variables as potential response variables: imdb_rating, critics_rating, critics_score, audience_rating, and audience_score. Using domain knowledge, critic/audience score was dropped – as both are simply binned representations of the corresponding critic/audience rating. A correlation table was created for the remaining 3 response variables:

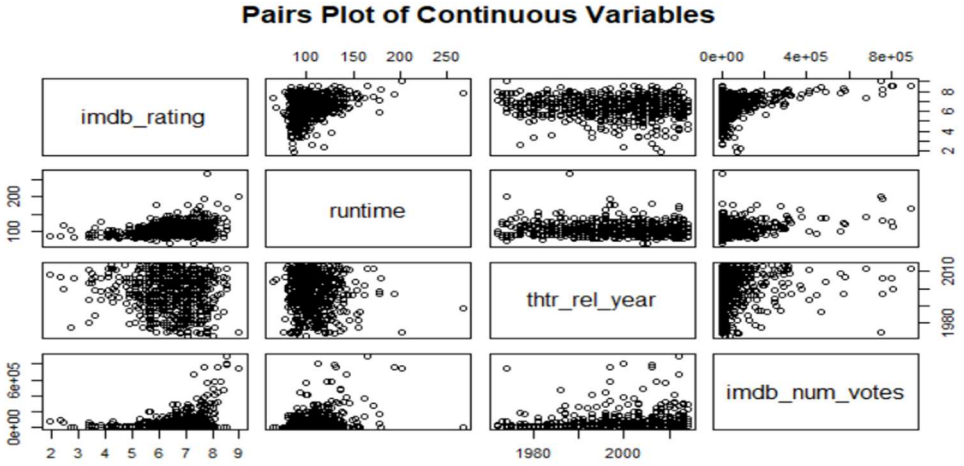
	imdb_rating	critics_score	audience_score
imdb_rating	1.0000000	0.7619990	0.8605425
critics_score	0.7619990	1.0000000	0.7015256
audience_score	0.8605425	0.7015256	1.0000000

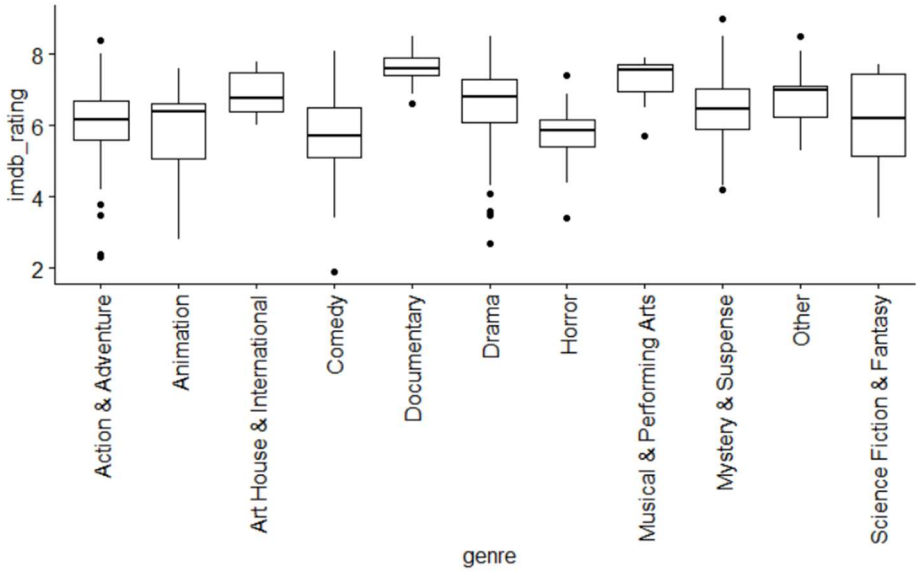
Collinear variables are generally problematic with respect to model accuracy. In order to determine which response variable to use, the team visually assessed the normality of the three variables by plotting their histograms.



As evidenced by the plots above, imdb_rating has the most normal distribution of the three, albeit with a slight left-skew (as would be expected given a 1 to 10 rating scale). The other two variables have more uniform distributions, and as such were dropped from the dataset.

In order to evaluate the distributions of the explanatory variables and analyze their interactions with the chosen response variable, boxplots were created for categorical variables and scatter plots were created for continuous variables. Given space constraints, a sample of the plots is displayed below (full visualizations are viewable in the R code).





Before continuing, it is important to note that Bayesian methods are much more flexible with respect to variable assumption violations (homoscedasticity and normality) than their sister frequentist methods. While these violations are evidenced in some of the boxplots displayed above, all categorical variables were selected to remain in the model. Model accuracy often improves with more explanatory variables, even with minor violations of normality/homoscedasticity, at the cost of model runtime. From the scatter plots, we can see that num_votes has a right-skewed relationship with the response variable. To correct for this, num_votes was replaced with log(num_votes) in the dataframe.

Bayesian Inference, Requirements, and Application

In order to address our research question, let's first dive a little more into the details of Bayesian methods. Bayesian statistics are built upon Bayes' Theorem, which is a mathematical way of quantifying conditional probability – or the probability that an event will happen given that another event has happened. Bayesian inference is simply the diachronic interpretation of Bayes' Theorem, which just means that the probability of an event happening is updated over time as new data is provided. Bayes' Theorem and some key definitions are presented below:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

- $p(A)$ is the probability of the hypothesis before we see the data, called the prior probability, or just **prior**.
- $p(\mathrm{A}|\mathrm{B})$ is our goal, this is the probability of the hypothesis after we see the data, called the **posterior**.
- $p(B|\mathrm{A})$ is the probability of the data under the hypothesis, called the **likelihood**.
- $p(B)$ is the probability of the data under any hypothesis, called the **normalizing constant**.

The important takeaways from the information presented above are the **prior** and the **posterior**. At its heart, a Bayesian model is simply an iterative series of models whereby the posterior – in our case the predicted movie rating – is updated based on a set prior and the **likelihood**, or new data presented in steps formulated from the explanatory variables upon which the model is built.

There is another term that is important to Bayesian inference – the Bayes Factor – defined as the ratio of the likelihood probability of two competing hypotheses. The Bayes Factor helps R to quantify supporting one model over another in the iterative model-building process.

For our analysis, the team used the Bayesian Adaptive Sampling (BAS) package in R to conduct Bayesian inference via Bayesian Model Averaging, or BMA. BMA yields a single model obtained by averaging results of coefficients for all models using their posterior probabilities. The BAS package contains functions to implement BMA for variable selection, regression, and visualization of results. The requirements to use Bayesian analysis are minimal: simply formulate a hypothesis and feed data iteratively in order to update said hypothesis. Assumptions for Bayesian statistics are the same as those for frequentist (normality/homoscedasticity of variables) but, as aforementioned, Bayesian methods are much more flexible with respect to assumption violations.

The steps by which Bayesian inference takes place are as follows:

1. Set a prior
2. Calculate posterior probabilities based on a prior and likelihood
3. Update prior probabilities through an iterative process of data collection.

The code involves invoking the bas.lm() function from the BAS package as follows:

```
```{r - Build Bayesian Model}
movie_bas<-bas.lm(imdb_rating ~ .,
                  data = movies,
                  method = "MCMC",
                  prior = "ZS-null",
                  modelprior = uniform())
```
```

Bayesian methods are an impossibly complex field of statistical sciences, and options for fine-tuning the model are seemingly endless. As this analysis is an exploratory foray into using Bayesian methods, the options selected for our model were the default calls for the bas.lm() function, and are listed as follows:

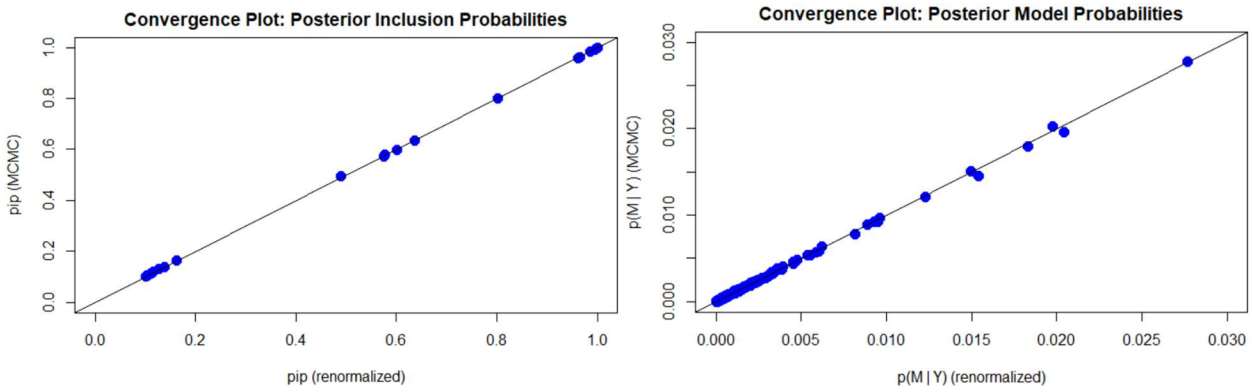
- Prior – Zellner-Siow Cauchy
    - Coefficient distributions based on Zellner's g-prior
    - Bayes' factors compared to null model
- Method – MCMC (Markov Chain Monte Carlo)
    - Sampling algorithm to sample without replacement from the space of models
- Modelprior – uniform()
    - Equal probabilities for all models

The function then runs several thousand models (the exact number given by 2^P, where P is the number of explanatory variables included in the dataframe) and returns a table of the explanatory variables, their associated marginal inclusion probabilities (0 to 1 scale), the top 5 models, and model summary statistics. The summary statistics include the Bayes factor (BF) for each model in comparison to the

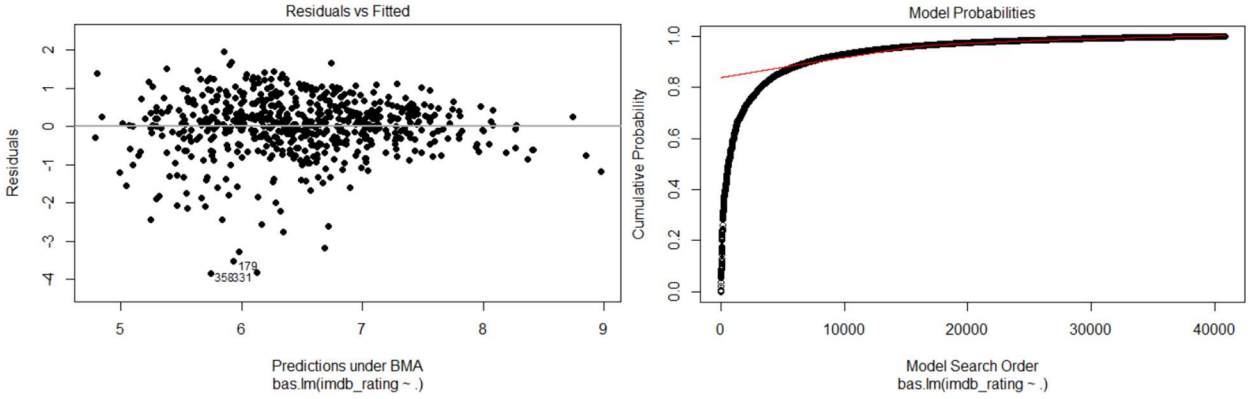
highest probability model, the posterior probabilities (PostProbs), the R-squared (R2), the dimension (dim), and the log marginal likelihood (logmarg) of each model under the selected prior distribution. We will not delve into full details of each of these outputs; the combination of all 5 of them are used by R to quantify support of one model over another, with Bayes' Factor being weighted the heaviest.

	P(B != 0 | Y)	model 1	model 2	model 3	model 4	model 5
Intercept	1.0000000	1.0000	1.0000000	1.0000000	1.0000000	1.000000
title_typeFeature Film	0.9961729	1.0000	1.0000000	1.0000000	1.0000000	1.000000
title_typeTV Movie	0.5994789	1.0000	0.0000000	1.0000000	1.0000000	1.000000
genreAnimation	0.1195684	0.0000	0.0000000	0.0000000	0.0000000	0.000000
genreArt House & International	0.9999765	1.0000	1.0000000	1.0000000	1.0000000	1.000000
genreComedy	0.1395699	0.0000	0.0000000	0.0000000	0.0000000	0.000000
...						
BF	NA	1.0000	0.7132033	0.7374176	0.6615126	0.540659
PostProbs	NA	0.0278	0.0202000	0.0195000	0.0180000	0.015000
R2	NA	0.4756	0.4712000	0.4712000	0.4671000	0.470700
dim	NA	16.0000	15.0000000	15.0000000	14.0000000	15.000000
logmarg	NA	159.0930	158.7549701	158.7883579	158.6797327	158.477992

The BAS package and the diagnostics() function allows R to easily display diagnostic plots for our model.

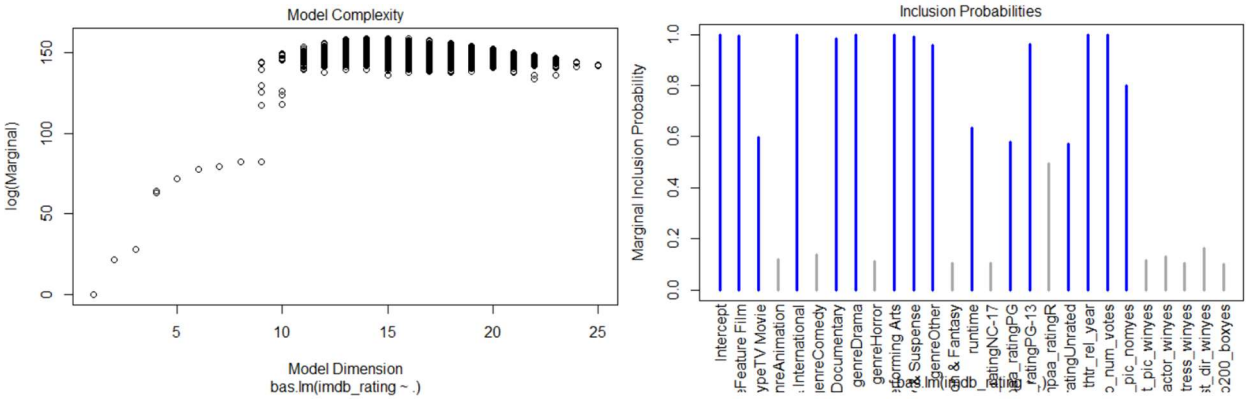


The Posterior Inclusion Probabilities (PIP) and Posterior Model Probabilities (PMP) Convergence plots show that all points are on or near the 45 degree diagonal. We can then say that the PIP of each variable from the MCMC have converged well enough to the theoretical PIP, and similar for the PMPs. In theory, we could increase the number of MCMC iterations to improve the model. However, model improvement would be negligible in comparison to the greatly increased runtime of bas.lm() method.



The Residual vs. Fitted plot shows an even spread with no apparent heteroscedasticity and minimal outliers. The Model Probabilities plot displays the cumulative probability of the models in the order that

they are sampled. From the graph we can see the cumulative probability starts to level off at around 6000 models, with each additional model adding only a small increment to the cumulative probability.



The Model Complexity plot shows the number of regression coefficients, including the intercept, versus the log of the marginal likelihood of the model. The plot for our model indicates the highest log marginal being reached at 14 dimensions, with no significant increase in log marginal likelihood beyond 10 dimensions. Finally, the Marginal Inclusion Probability plot displays the relative importance of the different predictors included in the bas.lm() function call. The blue lines correspond to the variables where the marginal PIP is greater than 0.5, suggesting that these variables are important to the accuracy of the model's predictions.

Prediction, Conclusion, and Future Considerations

The team selected the movie Godzilla (2014) for which to predict a movie rating based on the BMA model returned by the bas.lm() function call. The predict() function in R, along with Bayesian-specific option calls, can be used on the model object returned as follows:

```r
prediction <- predict(movie_bas, godzilla, estimator="BMA", interval = "predict", se.fit=TRUE)
```

By setting the "estimator" option in the predict function to "BMA", a single model is obtained by averaging the coefficients for all the models using their posterior probabilities. This approach leads to all explanatory variables being factored into the prediction. The model prediction for Godzilla (2014), along with the real rating, is shown below:

Movie <fctr>	Estimated.IMDB.rating <dbl>	Real.IMDB.rating <dbl>
Godzilla	6.642862	6.5

Our model estimates an IMDb rating of 6.64 for based on the BMA, which is very close to the real IMDb rating of 6.5 (within 3%). This seems to indicate an acceptably robust model performance. A consideration for further analysis would be to return predictions for several/all of the movies and to then run statistical analyses to compare predicted values vs. actual. However, the predict() function for a BMA model takes an oppressively long time to run given the number of models that are averaged with each function call, and consequently such an analysis would not be feasible at this juncture given time constraints. The team could also reduce the number of explanatory variables fed to the model, with a tradeoff of model accuracy.

Works Cited

Clyde, Merlise A. *Using the Bayesian Adaptive Sampling (BAS) Package for Bayesian Model Averaging and Variable Selection*, 24 Jan. 2020, cran.r-project.org/web/packages/BAS/vignettes/BAS-vignette.html.

Clyde, Merlise, et al. "An Introduction to Bayesian Thinking." *Chapter 8 Stochastic Explorations Using MCMC*, 2013, statswithr.github.io/book/stochastic-explorations-using-mcmc.html.

Cook, John D. "Four Reasons to Use Bayesian Inference." *John D. Cook*, 28 Apr. 2009, www.johndcook.com/blog/2009/04/28/reasons-to-use-bayesian-inference/.

Gray, Kevin. "What Is Bayesian Statistics?" *LinkedIn*, 2018, www.linkedin.com/pulse/what-bayesian-statistics-kevin-gray/.

Inzaugarat, Euge. "Linear and Bayesian Modeling in R: Predicting Movie Popularity." *Medium*, Towards Data Science, 3 May 2019, towardsdatascience.com/linear-and-bayesian-modelling-in-r-predicting-movie-popularity-6c8ef0a44184.

Ni, Chen. "Movie Dataset." *Kaggle*, 2 Apr. 2019, www.kaggle.com/nichen301/movie-data.