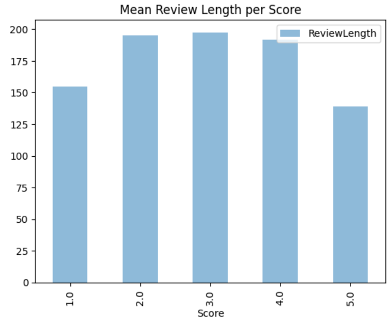
Midterm Report

Aidan Ruvins

November 8th, 2023

The key problem being solved is to create a model that predicts the star rating associated with user reviews from Amazon Movie Reviews. The features that are made available to train the model are the Review ID, ProductID, UserID, Helpfulness Numerator, Helpfulness Denominator, Time, Summary, Text, Helpfulness, and Review Length. Given these features, we need to predict the score. The score can be an integer number 1-5 i.e. the score must be 1.0, 2.0, 3.0, 4.0, or 5.0 and can not be a float like 4.5. Note that a user may write a score in the text field that is a float but the final score must be an integer score. Helpfulness is calculated by dividing the Helpfulness Numerator by the Helpfulness Denominator. A higher Helpfulness score is better The numerator and denominator are representative of how many people clicked the “Helpful” or “Not Helpful” buttons respectively. Review ID, ProductID, and UserID are features that identify the review, the product (movie), and the respective user. The time feature represents when the review was posted. The summary and text are text fields that were filled out by the user to explain their review. Lastly, the Review length is the number of words written in the text field.

Feature engineering was done to build an understanding of the data and build a more accurate model. Some features that were tested were utilizing the Review Length, Helpfulness, Summary, Keywords, Association of keywords with a score, and Sentiment analysis. The first question that was asked was if there was a correlation between a higher review length and an extreme score. The hypothesis was that a higher review length would mean that the score would be a more extreme score of either a 1 or a 5. This hypothesis was incorrect and it was found that a more neutral score tended to have a higher mean review length. Another possible feature that was tested was looking into helpfulness. The idea was to use helpfulness as a weight to determine the validity of a given review. For example, a review that has a low helpfulness score may be less valid since others tend to disagree with their review or a review with a high helpfulness score may be more valid since others agree with this review. In the implementation, this looked like a weight placed upon the review where a higher helpfulness score would help pull the estimated score in the direction of the reviewed score. Another feature that was tested was looking into sentiment analysis. This feature would aim to score the review based on keywords seen within the summary and text fields. This was broken down into 3 subfeatures. The first was to look at the summary field and grab the most common words used in that field. The second was to look at the most common words in the summary field per score. For example, the keyword “Great” was very common with 5-star reviews. The last subfeature was to assign a score value to the keyword. The code would then look at the written sections of the review to look for keywords and assign a score based on whether the keywords were positive or negative. A review with more positive keywords would then have a higher score.



The tested hypothesis was that this was a classification problem since the model should classify the reviews into the scores {1.0, 2.0, 3.0, 4.0, 5.0}. This was hypothesized since the score can't be a decimal number and must be one of the 5 options. Given this, the hypothesized best model would have been Naive Bayes. After implementation, the returned RMSE was 2.355. The next test was on a Regression model. The model selected was a Linear Regression model. This model gave an RMSE of 1.452.

Since the selected model was a linear regression, the model could be tuned using either a Lasso or Ridge regression. Lasso would be used if a coefficient could be deemed irrelevant and must be 0. A ridge regression would be used if all coefficients must carry a weight. It is important to note that a ridge regression can have small coefficients but they can never be 0. This difference is because Lasso minimizes coefficients as a function of an absolute value while ridge minimizes coefficients as a squared function. Thus the Lasso allows the coefficients to be 0. After experimenting, the Lasso regression was picked because it minimizes the RMSE the most. The final RMSE of the Linear Regression after tuning with Lasso was 1.33 with an alpha of 0.6.

To validate the model, RMSE was used. RMSE is a function that finds the error between the predicted values and the actual values. The goal of the model was to minimize the error.