IMDB Movie Review Binary Sentiment Classification

Olivia Weisiger (605612247), Saaya Sitlani (605695030),
Meiyi Ye (505569664), and Alex Sapinoso (606037424)

Abstract

IMDB is a popular site for data related to movies and TV shows, including reviews written by viewers. These textual reviews are categorized as either positive or negative. Using the given dataset, our objective is to explore the words within each review, and determine if a numerical classification system developed through analyzing individual word sentiments can accurately categorize whether a textual review has an overall positive or negative sentiment. Through text mining, a *positive term frequency* and *negative term frequency* were created by taking the proportion of positive and negative words detected within each review. We utilized various classification models– including logistic regression, KNN, LDA, QDA, and random forest– to conduct the investigation, using *positive term frequency* and *negative term frequency* to predict the binary sentiment of each movie review, ultimately hoping to determine which model has the highest prediction accuracy. After training and testing these various models, the 10-fold LDA was found to have the highest predictive power, with a testing accuracy of 73.52%. Dimension reduction via PCA was conducted, revealing that both features explain a significant proportion of variance in the dataset, thus the developed numerical classification system was successful in extracting meaningful predictors that can accurately classify binary movie sentiment.

1 INTRODUCTION

IMDB is a reliable site utilized by many consumers of media as they make decisions on what shows and movies appeal to them. Textual reviews can be a beneficial way of understanding how other viewers feel about a particular movie or show. The dataset we are working with contains 50,000 textual movie reviews.

1.1 Research Goal

Our goal is to determine which model can best predict the overall positive or negative classification of a review given the proportion of negative and positive words within the review.

1.2 Variables of Interest

The main variable of interest in this study is the overall sentiment of each movie review as positive or negative. To predict this, the raw text from each movie review was transformed into numerical form. In this case, the proportion of positive and negative words in each review were calculated, and these became the predictor features for the response.

1.3 Methods

The 50,000 observations were randomly split into equal parts testing and training data (i.e., the testing and training data each contain 25,000 observations).

Using the training data, various models including logistic regression, KNN, LDA, QDA, and random forest were trained to analyze the relationship between the variables in this dataset. After, the accuracy of each model was tested by applying it to the testing data and comparing the predicted sentiments to the actual sentiments of the movie reviews.

2 FEATURE ENGINEERING

2.1 Preprocessing of Data

Text for each movie review was first preprocessed and standardized, in order to remove line breaks, punctuation, digits, and other special characters [1]. Stopwords including personal pronouns, determiners, coordinating conjunctions, and prepositions were omitted for their lack of meaning, ensuring the isolation of significant words in each movie review.

To further funnel words into their final state, *lemmatization*, the "grouping together the inflected forms of a word so they can be analyzed as a single item" was used to identify the word's lemmas and map them to their dictionary form [2].

Lastly, adverbs were transformed to adjectives using a custom built mapping function. Given an 'adverb' = 'adjective' list, the function split each review into individual words. If an adverb was detected, it was mapped to its corresponding adjective form.

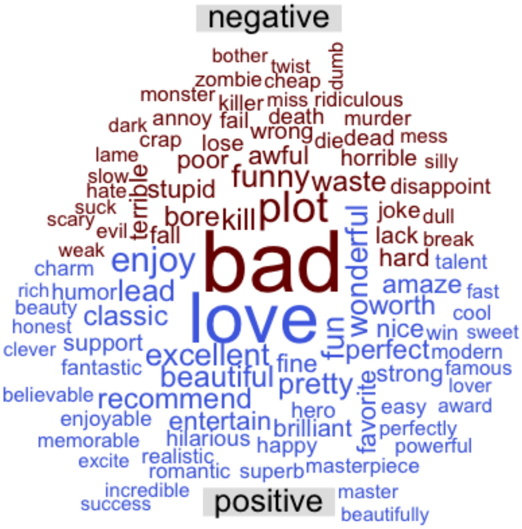
2.2 Text Mining & Feature Creation

The 'bing' lexicon [3] was used as the sentiment dictionary. Words within each review were mapped to 'positive' or 'negative' depending on if the 'bing' dictionary categorized their sentiments as such. The two features *positive term frequency* and *negative term frequency* were created by taking the proportion of positive and negative words detected within each review. Prior to model building, these features were standardized.

Other features such as word count (the number of words in each review) and sentence length (the length of sentences in each review) were considered, but omitted for their lack of significance.

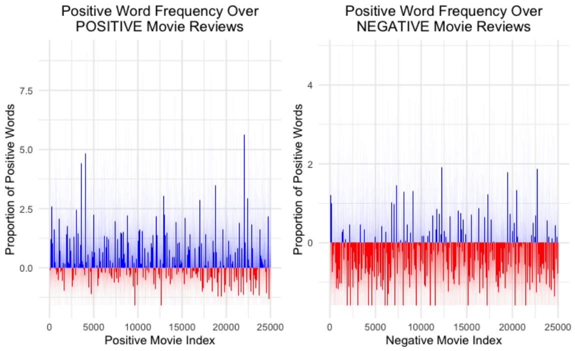
2.3 Exploratory Data Analysis

Before building any models, the sentiments of the words contained in the transformed movie reviews were visually investigated.



From the word cloud [3], we can see the most common positive and negative words detected, with "bad" being the most frequent negative term used, and "love" being the most common positive term used.

The positive and negative word proportions created for each movie were also examined, to better understand how these features may benefit the models.



The plots above display the proportion of positive words detected in each movie review, separated by whether the movie review is positive or negative. From the left plot, we can see that for positive movie reviews, the standardized positive word proportions tend to be greater than zero, while on the right plot, the standardized positive word proportions tend to be less than zero for negative movie reviews. This is a good indicator that positive word frequency per movie review will be a significant predictor of overall movie review sentiment.

3 MODEL BUILDING

Each model was constructed using the created features: *positive term frequency* and *negative term frequency*. The aim of each model is to accurately predict whether each review has a positive or negative sentiment based on these two features.

3.1 Logistic Regression

We performed logistic regression on the training data using *positive term frequency* and *negative term frequency* as predictors of the probability of the binary movie review sentiment outcome.

Summary of the Logistic Regression Model

	Estimate	Std. Error	P-Value
Intercept	0.02179	0.01513	0.15
Pos Freq	0.84924	0.01915	<2e-16***
Neg Freq	-1.04339	0.01925	<2e-16***

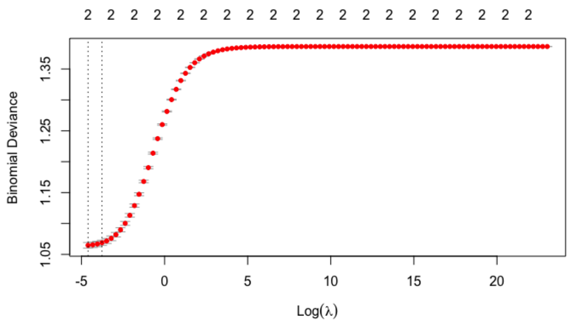
Additional Model Information

Null Deviance: 34657 on 24999 df
Residual Deviance: 26579 on 24997 df
AIC: 26585

After implementing the model on the testing data, we calculated its predictive performance using a confusion matrix, yielding an accuracy of 73.49%.

However, predictive models such as logistic regression are prone to overfitting. Therefore, regularization, a technique that reduces the complexity of a model by introducing a penalty term, was implemented to address this issue.

A logistic regression with the L2 regularization using cross-validation was conducted to find an optimal value for lambda, which controls the strength of the regularization. After combining the positive and negative term frequencies in the training data, we fit a logistic regression model and examined the plot of the cross-validated mean deviance against log(lambda) (pictured below) to observe the best lambda.



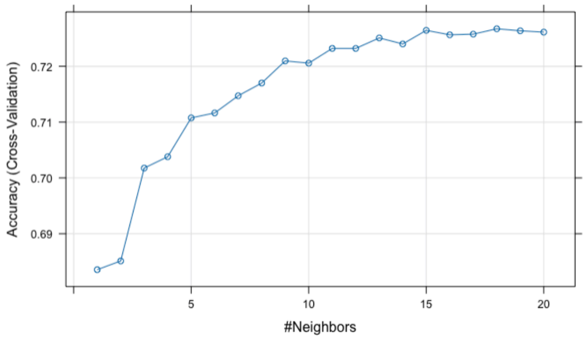
The best lambda was identified as 0.01. Using the optimal lambda, a new logistic regression model was fit. When evaluated on the testing data, the model had an accuracy of 73.50%, which is slightly better than the model without L2 regularization, although this improvement is negligible.

3.2 K-Nearest Neighbors (KNN)

K-nearest neighbors (KNN) is a supervised learning method used for classification and regression. It uses proximity to make classifications or predictions about the grouping of an individual data point [4].

An initial KNN model, using *positive term frequency* and *negative term frequency* to predict the binary sentiment of movie reviews, was fit to the training data using an arbitrary three nearest neighbors (k = 3). A testing accuracy of 68.84% was attained using this preliminary model.

To better fit a KNN model, cross-validation was leveraged as it provides a systematic way to tune hyperparameters, preventing overfitting, assessing generalization performance, and ensuring robustness across different subsets of the training data. This contributes to the development of a more reliable and effective KNN model for making predictions on new and unseen data.



From the figure above, which plots KNN model accuracy against increasing numbers of neighbors (k), we can see that the optimal k = 18. On the training data, k = 18 yields an accuracy of 72.67% and a moderately good kappa value of 0.4534. When applied to the testing data, the KNN model was 72.73% accurate in predicting the sentiment of movie reviews.

3.3 Linear Discriminant Analysis (LDA) & Quadratic Discriminant Analysis (QDA)

Unlike discriminative models such as logistic regression and KNN, which focus on learning the boundary between different classes for classification, generative models focus on understanding the data's underlying structure and generating new data points. LDA and QDA are two examples of generative models which were leveraged in our binary sentiment prediction.

After applying an LDA and QDA model, using *positive term frequency* and *negative term frequency* to predict the binary sentiment of movie reviews, on the testing data, an accuracy of 73.37% and 73.21% were achieved, respectively.

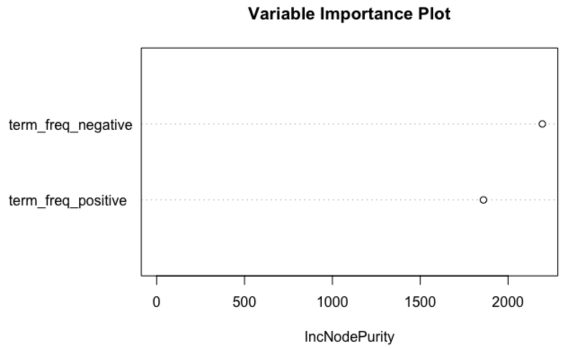
To improve the performance of the LDA and QDA, 10-fold cross-validation was used to help tune hyperparameters, prevent overfitting, and ensure robustness across different subsets of the training data, making the models more reliable. Using the cross-validation technique, testing accuracies of 73.52% for LDA and 73.33% for QDA were achieved.

In this case, LDA outperformed QDA in regards to accuracy– though the difference in model accuracy is negligible.

3.4 Random Forest

Random forest models apply decision trees to different subsets of the data and average their outcomes, leading to a higher accuracy and preventing overfitting to the dataset. Decision trees work like a flowchart, starting with a root node as the base of the tree and splitting it into multiple nodes until the final nodes, known as leaf nodes, represent one class label. Decision trees are prone to bias and overfitting, so random forest is used to help prevent this problem by creating an uncorrelated forest of decision trees [5].

A random forest model was built, using *positive term frequency* and *negative term frequency* to predict the binary sentiment of movie reviews. When applied to the testing data, an accuracy of 68.50% was attained, which is similar to the accuracy reached with the KNN model using k = 3.



When plotting the variable importance (see above) of the random forest model, we can see that both features, positive and negative word proportions, are important for predicting overall movie review sentiment, with the negative word proportion being slightly more important as measured by the increase in node purity.

Since random forest has many hyperparameters, we further enhanced the robustness of the model by introducing cross-validation. After using 10-fold cross-validation, the mean accuracy reached was 70.22%.

4 FINAL MODEL SELECTION

4.1 Comparing Each Model

Since the ultimate goal is determining which model can best predict the overall positive or negative classification of a review given the proportion of negative and positive words within the review, the testing accuracies of each model will be compared to determine which is the optimal model.

Model Type	***Testing Accuracy***	***False Positive Rate***	***False Negative Rate***
Logistic Regression	73.49 %	26.95 %	26.07 %
Logistic Regression (L2 Reg.)	73.50 %	26.94 %	26.05 %
KNN (k = 3)	68.84 %	30.27 %	32.00 %

KNN (k = 18)	72.13 %	26.66 %	29.08 %
LDA	73.37 %	27.1 %	26.17 %
LDA (10-fold)	73.52 %	-	-
QDA	73.21 %	26.14 %	27.44 %
QDA (10-fold)	73.33 %	-	-
Random Forest	68.50 %	31.12 %	31.88 %
Random Forest (10-fold)	70.22 %	-	-

After looking across the testing accuracies for each model type, the LDA model with 10-fold cross-validation is determined to be the most optimal model based on the level of testing accuracy reached compared to the other models.

4.2 Dimension Reduction Using PCA

Although we have only defined two features to create a classification model to group the data (*positive term frequency* and *negative term frequency*), we decided to run Principal Component Analysis to determine whether dimension reduction was beneficial to our model. The results are summarized here.

Importance of Components		
	PC1	PC2
Std. Deviation	1.1478	0.8262
Prop. of Variance	0.6587	0.3413
Cumulative Prop.	0.6587	1.0000

Firstly, we determined that the standard deviation of the first principal component (*positive term frequency*) was 1.1478, while the standard deviation of the second principal component (*negative term frequency*) was 0.8262. The fact that positive term frequency had a higher standard deviation than negative term frequency indicates that it captures more of the variability in the data.

We also found the cumulative proportions of variance explained by each component, with positive term frequency explaining 65.87% of the variation in the data and negative term frequency explains the rest, bringing the total cumulative variance to 100%. This indicates that the use of these two features is sufficient to account for all the variance in this dataset.

5 CONCLUSION

5.1 Highest Accuracy Model

As we can see in the model selection table, the model that yielded the highest accuracy was the 10-fold LDA (73.52% accuracy). This is likely because LDA is a generative model, and unlike discriminative models, it aims to understand the data's structure and generate new points. Other models that performed almost as well as 10-fold LDA were logistic regression with L2 regularization (73.50% accuracy) and logistic regression (73.49% accuracy). This, especially paired with the fact that LDA was the model with the highest accuracy, may indicate that the classes within the dataset were separated by a linear boundary. The models that performed with the least accuracy in predicting the positive and negative reviews were KNN with k = 3 (68.75% accuracy) and random forest (68.50%

accuracy). Though the difference between the accuracy of these models and the most accurate models is not very significant, we believe that the higher flexibility of these models could have led to slight overfitting and therefore caused their lower accuracy.

5.2 False Positive Rate (FPR) & False Negative Rate (FNR)

Another remark regarding the accuracy of our data is that we paid more attention to the false positive rate than the false negative rate of the various classifiers; in the context of this dataset, having a low false positive rate is valuable because incorrectly stating that reviews are positive has a worse impact on movie viewers than doing the opposite. However, because the models all had similar false positive and false negative rates, this was not a major factor in determining which model performed the best on our dataset.

6 DISCUSSION

During the text mining step of cleaning the movie reviews, the process of tokenization to split movie reviews into individual words could be paired with part of speech tagging to more thoroughly transform adverbs in the text to their adjective form. This follows methods used in NLP, in which we can exploit the POS tags to better clean the data [6]. Using the ‘udpipe’ package in R, a pre-trained English model can be downloaded. We can then leverage the ‘adverb’ tags by isolating them and creating a more complete adverb to adjective mapping given the unique adverbs identified in the dataset.

Some limitations to the models built include the fact that they were only built on two features. In the future, more can be done to transform the text into significant numerical predictors for this classification problem.

Though the final model achieved a moderately high testing accuracy of 73.52%, there are other methods, such as Support Vector Machine (SVM), that could have been implemented to potentially get higher accuracies when conducting classification on the testing data.

7 REFERENCES

[1] Sudhaharan, Roshini. “Text Pre-Processing in R.” *Tilburg Science Hub*, tilburgsciencehub.com/building-blocks/prepare-your-data-for-analysis/data-preparation/text-preprocessing/. Accessed 8 Dec. 2023.

[2] “Lemmatizing.” *Textstem*, cran.r-project.org/web/packages/textstem/readme/README.html#lemmatizing. Accessed 8 Dec. 2023.

[3] Silge, Julia, and David Robinson. “Sentiment analysis with tidy data.” *Text Mining with R: A Tidy Approach*, www.tidytextmining.com/sentiment. Accessed 8 Dec. 2023.

[4] “What Is the K-Nearest Neighbors Algorithm?” *IBM*, www.ibm.com/topics/knn. Accessed 8 Dec. 2023.

[5] “What Is Random Forest?” *IBM*, https://www.ibm.com/topics/random-forest. Accessed 8 Dec. 2023.

[6] Bird, Steven, et al. “Categorizing and Tagging Words.” *Natural Language Processing with Python*, 2019, www.nltk.org/book/ch05.html.