Movie Review Sentiment Prediction

Rita Chau[†] (UID: 606040558), Lingjun Chen[†] (UID: 906041231), Zesen Ma[†] (UID: 706066010)
[†]Department of Statistics UCLA

Abstract

IMDB, which stands for Internet Movie Database[1], is an online database of information related to films, television programs, home videos, video games, and streaming content online. We will be analyzing a dataset that contains 50k movie reviews. The dataset is for binary sentiment classification,i.e. positive review and negative review. Our goal is to build and validate predictive models to identify whether the sentiment of a review is positive or negative. Various machine learning methods are employed, including integrated methods such as logistic regression, K-nearest neighbor (KNN), linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), decision tree and random forest, in order to select the best performing and most appropriate model for IMDB movie review sentiment prediction.

1 Introduction

The success of a movie is often measured in box office numbers, but the true resonance of a movie is often captured in the audience's response. A collective voice of approval or disapproval can be found in the audience's reaction.

In our project, we are tasked with analyzing an IMDB dataset containing a large number of 50,000 movie reviews, which provided a set of 25,000 highly polarized movie reviews for training and 25,000 for testing. Our goal is to use this rich dataset to explore the relationship between sentiment classification and reviews, and ultimately identify the statistical model with the highest accuracy.

2 Preprocessing Step

Preprocessing raw data into a structured format that can be interpreted by algorithms is the basis of any text analytics or natural language processing work. In order to convert the unstructured movie reviews from the IMDB dataset into a format that can be used for sentiment classification, we have carefully selected the dataset for this project.

2.1 Text Cleaning

Firstly, in order to reduce the size of our dataset, we eliminated the unnecessary components that usually contribute noise instead of value to the forecasting models. This meant removing determiners and personal pronouns, which are structurally important for human language but have little predictive power in sentiment analysis. Prepositions and coordinating conjunctions were also removed from the text.

2.2 Adverb to Adjective Transformation

We converted adverbs into their adjective forms wherever it was possible since our raw dictionary is made of the adjective. This was accomplished by eliminating adverbial suffixes like ”ly” and ”fully,” streamlining the language, and focusing on the material that was emotionally charged. By converting adverbs into adjective, we are now able to choose our words based on the raw dictionary.

2.3 Descriptive Statistics

To get a better understanding of the data we have, we then constructed some descriptive statistics.

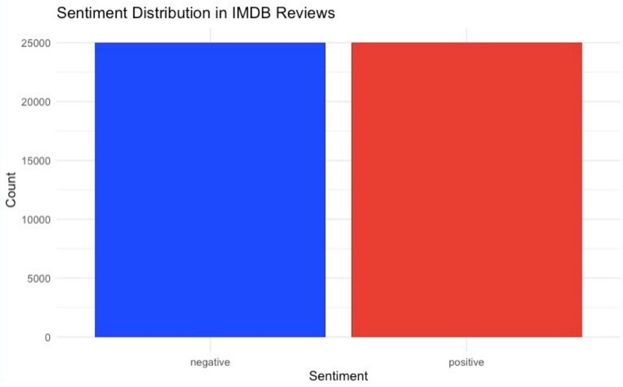

Figure 1: Distribution of the Sentiments

The sentiment distribution figure shows that there are approximately as many positive reviews in the dataset as negative ones, with a notable proportion of favorable ratings. The two bars' almost equal heights imply that there may be a reasonable balance between positive and negative reviews in the dataset.

2.4 Numeration and Term Weighting

After that, we used a sentiment lexicon[2] to number the textual material, which was quite helpful in figuring out each review's underlying emotions. To determine a term's relevance within the corpus, the frequency of terms was computed and then compared to the inverse document frequency. The most notable terms were ultimately chosen for our sentiment dictionary by using TF-IDF (Term Frequency-Inverse Document Frequency) weighting.

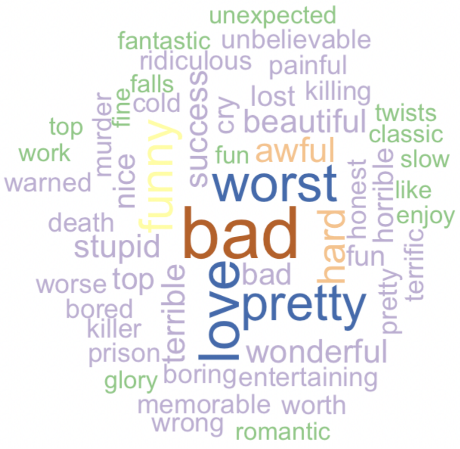

Figure 2: Frequency of Words in Reviews

2.5 Feature Engineering: Sentiment Lexicon and PCA

We decided to use Principal component analysis. Since the number of texts is abundant for movie reviews. We needed PCA to reduce the high dimensionality of this large dataset. The number of words selected for our models is significant and important to our models' accuracy and efficiency. Due to the nature of language, we would need a large number of words to train our models. We have discovered that the variances are highly spread out and follow slightly decreasing trend shown in Figure 3 (Top 25 words). After examining models under different number of words used, we discovered that using 300 words would result in satisfying accuracy and time efficiency for training models.

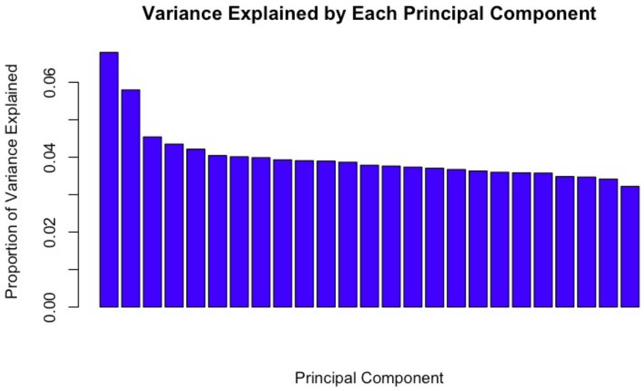

Figure 3: Variance Explained by Each Principal Component

3 Experiment

To consider the best model for prediction, we used six supervised learning algorithms.

3.1 Logistic Regression

Logistic Regression is a robust statistical method used to predict the probability of a binary outcome based on one or more predictor variables. We have used logistic regression in our research to estimate the probability of a positive or negative occurrence. This approach models the binary dependent variable using a logistic function, extending the idea of linear regression to classification issues.

Table 1: Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	9953	2032
Positive	2530	10427
Accuracy	0.8171	
95% CI	(0.8122, 0.8219)	
No Information Rate	0.5005	
P-Value [Acc > NIR]	$< 2.2e - 16$	
Kappa	0.6342	
Mcnemar's Test P-Value	1.862e-13	
Sensitivity	0.7973	
Specificity	0.8369	
Pos Pred Value	0.8305	
Neg Pred Value	0.8047	
Prevalence	0.5005	
Detection Rate	0.3990	
Detection Prevalence	0.4805	
Balanced Accuracy	0.8171	

With an accuracy of 81.71%, the model predicts correctly more than 81 out of 100 times. The 95% confidence interval (81.22% to 82.19%) indicates high reliability in the model's performance. A very small p-value reinforces that the model's accuracy is significantly better than random chance, with the No Information Rate at 50.05%. The model's balanced accuracy is 81.71%, demonstrating its effectiveness across both positive and negative classes.

The Kappa statistic of 0.6342 suggests a substantial agreement beyond chance. Sensitivity, or the true positive rate, is 79.73%, and specificity, the true negative rate, is 83.69%. These figures indicate the model's robustness in correctly identifying both classes. With a positive predictive value of 83.05% and a negative predictive value of 80.47%, the model is reliable for practical applications where precision in predicting both outcomes is critical.

3.2 Linear and Quadratic Determinant Analysis

Now, let's look at generative models, which represent the combined probability of the predictors and the response variable. Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) are two examples of such generative models.

We closely examine the confusion matrices and the resulting statistics in order to evaluate the effectiveness of our classification models. These measurements offer a thorough understanding of how well each model differentiates between the positive and negative classifications. The results of the Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA) models are summarized in Tables 2 and 3, respectively. We may compare the models quantitatively using these tables and see whether there are any notable variations in the models' capacity for prediction. We fit both LDA and QDA models to the training set and received the following results on the test set:

Table 2: LDA Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	9747	1862
Positive	2736	10597
Accuracy	0.8157	
95% CI	(0.8108, 0.8204)	
No Information Rate	0.5005	
P-Value [Acc ¿ NIR]	¡ 2.2e-16	
Kappa	0.6313	
Mcnemar's Test P-Value	¡ 2.2e-16	
Sensitivity	0.7808	
Specificity	0.8505	
Pos Pred Value	0.8396	
Neg Pred Value	0.7948	
Prevalence	0.5005	
Detection Rate	0.3908	
Detection Prevalence	0.4654	
Balanced Accuracy	0.8157	

Table 3: QDA Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	9720	2455
Positive	2763	10004
Accuracy	0.7908	
95% CI	(0.7857, 0.7958)	
No Information Rate	0.5005	
P-Value [Acc ¿ NIR]	¡ 2.2e-16	
Kappa	0.5816	
Mcnemar's Test P-Value	2.138e-05	
Sensitivity	0.7787	
Specificity	0.8030	
Pos Pred Value	0.7984	
Neg Pred Value	0.7836	
Prevalence	0.5005	
Detection Rate	0.3897	
Detection Prevalence	0.4881	
Balanced Accuracy	0.7908	

When comparing Linear Discriminant Analysis (LDA) and Quadratic Discriminant Analysis (QDA), LDA shows better sensitivity and a marginally higher accuracy (81.57% vs. 79.08%), indicating more precise positive detections. On the other hand, QDA predicts real positive instances more accurately and with higher specificity. While LDA has a somewhat tighter confidence interval, indicating more accurate results, both models are significant with low P-values in McNemar's test. Whether it is more important to decrease false positives or increase overall prediction accuracy will choose which of LDA and

QDA to use.

3.3 K-Nearest Neighbors

We also implemented KNN method to look at the performance. The KNN model, at k = 5, achieved, an accuracy of 50.29, which is only slightly better than random guessing. Both sensitivity (true positive rate) and specificity (true negative rate) are approximately 50, meaning that the model overall is not efficient. Further adjustment are required for KNN models: Other dimensionality reduction techniques and feature selections methods might be necessary.

Table 4: KNN Model Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	6325	6241
Positive	6158	6218
Accuracy	0.5029	
95% CI	(0.4967, 0.5091)	
No Information Rate	0.5005	
P-Value [Acc ¿ NIR]	0.2256	
Kappa	0.0058	
Mcnemar's Test P-Value	0.4615	
Sensitivity	0.5067	
Specificity	0.4991	
Pos Pred Value	0.5033	
Neg Pred Value	0.5024	
Prevalence	0.5005	
Detection Rate	0.2536	
Detection Prevalence	0.5038	
Balanced Accuracy	0.5029	

3.4 Random Forest

Random forest is also implemented. Random Forest is known for its high accuracy in many applications. It can handle binary, categorical, and numerical data and is effective for both classification and regression tasks. The model is trained with 15 trees. After running the model: The model achieved perfect classification with no misclassifications. 10013 negative and 9540 positive instances were correctly classified, achieving 78.39 percent accuracy. This is a reasonable accuracy. However, further examinations such as cross validations may be necessary to consider for potential over fitting.

Table 5: Random Forest Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	10013	2919
Positive	2470	9540
Accuracy	0.7839	
95% CI	(0.7788, 0.789)	
No Information Rate	0.5005	
P-Value [Acc ¿ NIR]	$< 2.2 \times 10^{-16}$	
Kappa	0.5679	
Mcnemar's Test P-Value	1.043×10^{-09}	
Sensitivity	0.8021	
Specificity	0.7657	
Pos Pred Value	0.7743	
Neg Pred Value	0.7943	
Prevalence	0.5005	
Detection Rate	0.4015	
Detection Prevalence	0.5185	
Balanced Accuracy	0.7839	

3.5 Decision Tree

An even starting split in sentiment is indicated by the root node of our decision tree, which predicts negative sentiment with 50% certainty. The first decision splits in half based on the term ”bad,” which indicates that it is a strong negative indication because its existence accounts for 24% of the cases and carries a 25% chance of negative sentiment. On the other hand, the lack of ’bad’ leads to nodes containing phrases such as ’worst, waste, lousy, dreadful, and dull,’ all of which gradually lead to a positive outcome with probabilities of 61%, 63%, 65%, 66%, and 68% respectively.

Even though these nodes use terms that have negative connotations, they are linked to positive results. This could be due to the dataset's particular context or complex language usage, such as sarcasm. In its branch, the presence of ’worst’ yields a positive prediction 71% of the time, whilst ’boring’ yields a favorable result in 60% of situations.

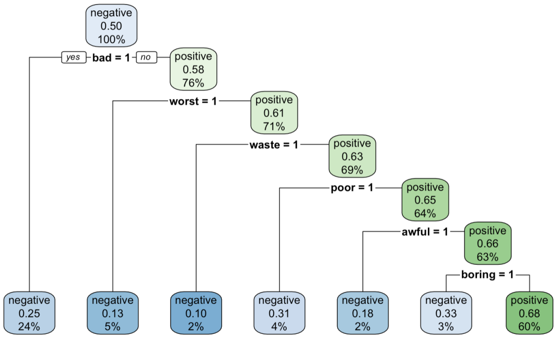

Figure 4: Decision Tree

Table 6: Decision Tree Model Confusion Matrix and Statistics

	Reference	
Prediction	Negative	Positive
Negative	7637	4846
Positive	2366	10093
Accuracy	0.7108	
95% CI	(0.7052, 0.7165)	
No Information Rate	0.5989	
P-Value [Acc > NIR]	$< 2.2e - 16$	
Kappa	0.4218	
Mcnemar's Test P-Value	$< 2.2e - 16$	
Sensitivity	0.7635	
Specificity	0.6756	
Pos Pred Value	0.6118	
Neg Pred Value	0.8101	
Prevalence	0.4011	
Detection Rate	0.3062	
Detection Prevalence	0.5005	
Balanced Accuracy	0.7195	

With an overall accuracy of 71.08%, the decision tree model outperforms a random guess, as seen by its higher accuracy than the No Information Rate of 58.99%. The percentage of positive identifications that were actually correct is 61.18%, while the percentage of negative identifications that were actually correct is greater at 81.01% and is known as the positive predictive value. The true positive rate, or sensitivity, is 76.35%, and the true negative rate, or specificity, is 67.56%. According to these numbers, the model can identify both positive and negative situations with a moderate degree of accuracy, with a substantial advantage in predicting real negatives. Beyond pure chance, the 0.4218 Kappa score indicates a reasonable degree of agreement between the actual and projected values. The balanced accuracy, which is the average of sensitivity and specificity, is 71.95%, reflecting a balanced performance across both classes.

4 Conclusion: Results and Analysis

4.1 Results

To analyze our models, we proceeded to compare the accuracy of the models used in our experiment. Below is our table summary of accuracy based on different models:

	Accuracy
KNN	50.29%
DS	71.08%
RF	78.39%
QDA	79.08%
LDA	81.57%
LR	81.71%

Table 7: Model Accuracy

We can see that, when setting top 300 most frequent used sentiment words, our logistic regression model performed the best in terms of accuracy at 81.57 percent. Following the LR model, LDA model obtains the second highest accuracy, slightly below LR, indicating that the linear assumption in LDA is more suitable for this dataset.

RF shows good performance, but more trees and feature engineering may be necessary in order to obtain a higher accuracy. Random forest usually performs well under high-dimensional dataset.

More complex models like Random Forest and QDA did not significantly outperform simpler models like Logistic Regression and LDA.

KNN shows the lowest performance among all models. This might be due to its sensitivity to high-dimensional data and noise.

Overall, the best option, in a practical application perspective, logistic regression is the best model to be considered, due to its highest accuracy and simplicity

4.2 Final Remarks

Based on our model performances on the dataset, we can conclude that using sentiment analysis model for IMDB movie review will be a more accurate and more effient way than just random guessing. However, there are still potentially areas that we can address to improve our models and analysis.

There may be some inherent issues with our data preprocessing, discovering and removing more un-sentiment related words can potentially result in a better training dataset. There appears to be a warning regarding the data frame of certain issues with word names can be addresses in the future. The reduction of dimensionality process may require a closer examination, as we chose to select the testing range from 25 - 300 words. A larger portion of numbers could potentially provide a more clear range of variances that may help us reduce the dimensionality. When performing KNN, the capability of our computers could be the limitations for us testing more K values in order to obtain a most satisfied model. Cross validation may be necessary and potentially useful for our models as well. Our random forest model is not as effient as expected, which leaves potentials to be explored and adjusted more in the future. In summary, further improvements on models might be achieved with different and more advanced feature engineering and, exploring more text processing techniques.

Acknowledgments

We would like to thank Professor Shirong Xu and Professor Yijia Zhao of UCLA for their professional guidance and valuable resources throughout the research process.

We thank team members Rita Chau, Lingjun Chen and Zesen Ma for their cooperation and dedication in this project come to fruition. All group members equally contributed to this project.

References

[1] IMDB Dataset of 50K Movie Reviews. Kaggle. `https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews`.

[2] Opinion Lexicon Dataset. Kaggle. `https://www.kaggle.com/datasets/nltkdata/opinion-lexicon`.