IMDb Movie Review Sentiment Prediction

Daphne Hidley (UID: 105502680), Kaylin Lee (UID: 906036630),
Megha Velakacharla (UID: 705587841), Noor Benny (UID: 605624764),
Rachel Ki (UID: 005590842)

Department of Statistics and Data Science UCLA

Abstract

IMDb, short for the Internet Movie Database, is an online database containing information about popular movies and television shows, including user ratings, reviews, plot summaries, cast lists, and more. Our objective is to explore the relationship between movie reviews and their associated sentiments, aiming to devise a methodology for predicting review sentiments. We will implement classification methods such as logistic regression, KNN, LDA, QDA, and random forests to build models capable of predicting sentiment from the words present in each review.

⋆ ⋆ ⋆ ⋆ ⋆

1 Introduction

Seeking cinematic guidance? IMDb is an online database that houses statistics for over 625,000 movies and consists of written reviews contributed by users [2]. Sifting through nearly eight million reviews within this extensive collection presents an opportunity to explore diverse viewpoints. For those curious about films they have not yet seen, IMDb reviews prove invaluable in gauging potential enjoyment based on fellow users' written opinions. These reviews typically vary in length and detail, reflecting users' perspectives on whether they liked or disliked the movie.

Our project involves analyzing a dataset of 50,000 IMDb reviews, each categorized by sentiment (negative or positive). Our objective is to explore which classification method performs the best under a combination of PCA and TF-IDF. By evaluating different classification models, we aim to identify the model that optimizes the accuracy of predicting a review's sentiment based on its content.

2 Preprocessing Step

To prepare the dataset of 50,000 IMDb reviews for classification modeling, the data was cleaned and processed in Python. We started with one file containing two columns: one column contained the reviews, while the other indicated the true sentiment of each review.

Using the tidyverse and tidytext packages in R, we first applied a tokenization process to the data which separated each review into the individual words. We proceeded by removing common stop words such as "a", "the", "is", and "are" from the reviews to leave only the meaningful words. Additionally, any numbers, special characters, pronouns, and line breaks (specifically the "
" tags left over from HTML) were removed while converting all text to lowercase for consistency. Finally, in order to refine the overall linguistic quality of the reviews, adverbs were transformed into adjectives using the udpipe package in R. For example, words like "happily" and "quickly" were changed to "happy" and "quick", respectively.

2.1 Descriptive Statistics

Once the data had been properly cleaned, we were able to begin our preliminary analysis and visualization of the reviews. This would help us gain a better understanding of the data through visualizations.

We began by tallying the count of reviews expressing positive and negative sentiments to assess the balance between each class (Figure 1). Our analysis revealed an equal number of positive and negative reviews within the dataset, establishing a well-balanced foundation for sentiment analysis. This equilibrium contributes to ensuring fairness in accurately understanding and predicting sentiments.

We also created a word cloud to quickly and intuitively visualize the most frequent or prominent words

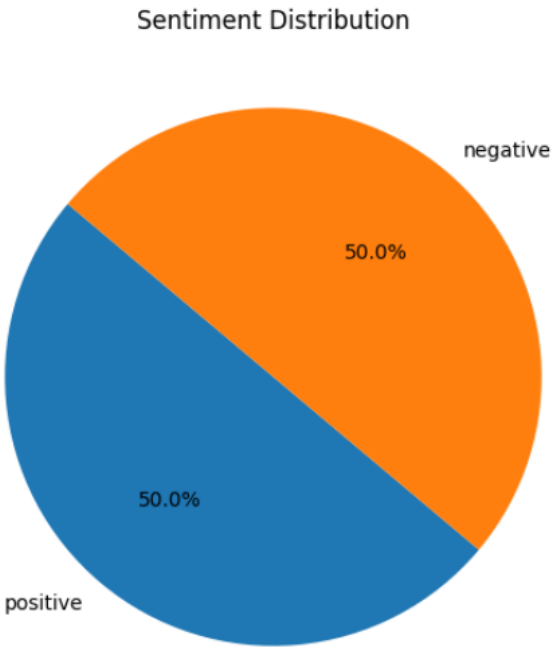

Figure 1: Sentiment Distribution

Figure 2: Word Cloud

in the set of reviews. Commonly used words like 'movie' and 'film' appear frequently but seem less critical in predicting sentiment compared to more emotionally loaded words like 'excellent' and 'wonderful'. This suggests that focusing on emotionally charged words might be more effective in understanding sentiments accurately.

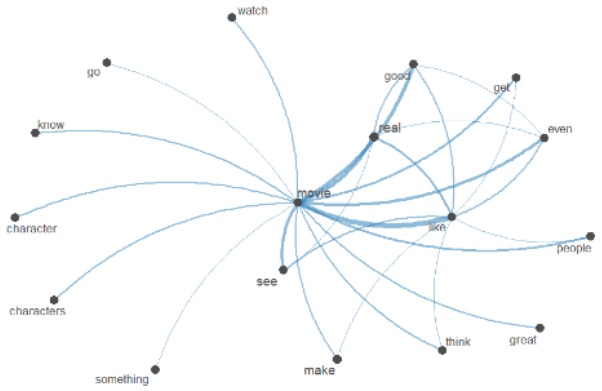

Figure 3: Word Association Network

Finally, we created a word association network to better understand the relationship between common words appearing in the set of reviews. The word association network displays connections between words, offering insights into how words relate to each other in reviews. Thicker connecting lines represent a stronger connection in how two words are used together. Additionally, words placed in the center of the network are most critical in connecting ideas.

Here, we see that the strongest connections occur between common words like "movie", "like", and "real" which seem common to all reviews and less powerful in predicting sentiment. Going forward, we would want to find a method to disregard common and useless words in favor of using emotional and qualitative words to conduct analysis and prediction.

2.2 Feature Engineering: Sentiment Lexicon and PCA

Our next step in preparing the data for classification was to begin constructing the dictionary of words to which the reviews would be compared. In our approach to building the word dictionary, we employed the TF-IDF (Term Frequency-Inverse Document Frequency) approach to transform textual data into numerical representations. This technique evaluates word importance within reviews and across the entire dataset. The selection of words was based on

their TF-IDF scores falling within specified thresholds (min_df = 0.1 to max_df = 0.7). By placing these thresholds, we hoped to omit highly appearing yet useless words like "movie" and any rarely appearing words. This process resulted in a dictionary of 108 words from the reviews. To include more words regarding sentiment in our dictionary, we leveraged external resources encapsulated in the positive-words.txt and negative-words.txt files [5]. These lists contained words classified as positive or negative based on predefined sentiments, adding two more features to the data - the count of positive words per review, and the count of negative words per review. After combining this with our dictionary from TF-IDF, we were left with 110 features.

Principal Component Analysis (PCA) was then employed to address high-dimensional feature spaces inherent in text data. We conducted PCA to mitigate the negative effects of dimensionality, aiming to reduce the computational complexity and potential noise within the dataset while preserving the most critical information. Initially, we transformed the textual features into a lower-dimensional space using PCA, a technique that identifies the principal components capturing the variance in the data. The choice of the reduction factor was determined by assessing the cumulative explained variance, ensuring a balance between dimensionality reduction and information retention, thereby enabling efficient sentiment analysis without compromising predictive power [7]. To achieve 95% of variance of the input explained by the generated components, we chose to utilize 95 principle components.

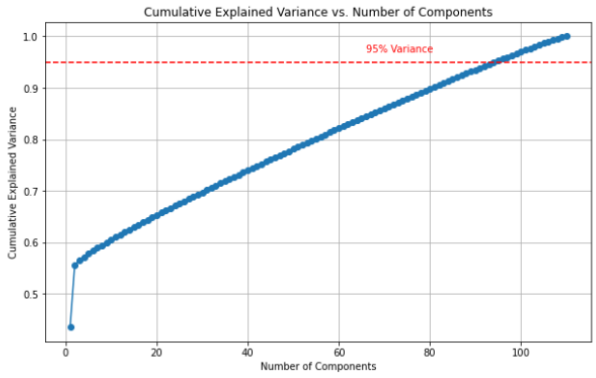

Figure 4: Principal Components by Cumulative Explained Variance

3 Experiment

In assessing the optimal prediction model, we employed four supervised learning algorithms: Logistic Regression, KNN, LDA/QDA, and Random Forests.

3.1 Logistic Regression

The first approach we employed was a logistic regression model. It served as our initial approach due to its relevance and efficacy in handling binary outcome predictions, aligning well with the sentiment analysis task. Logistic regression's simplicity, speed, and flexibility make it an ideal starting point for predictive analysis tasks [8]. Moreover, its computational efficiency and ability to process large volumes of data swiftly provided us with a pragmatic approach to handle the substantial amount of textual information present in movie reviews. Its simplicity also granted us enhanced visibility into the model's internal processes, facilitating easier error identification and correction. Upon training logistic regression on a subset of our dataset (50% for testing purposes), the model demonstrated an accuracy rate of 78.51%, as shown below.

Table 1: Logistic Regression Testing Results

	Precision	**Recall**	**F1-Score**	**Support**
False	0.7840	0.7823	0.7832	4961
True	0.7861	0.7879	0.7870	5039
Accuracy	-	-	0.7851	10000
Macro Avg	0.7851	0.7851	0.7851	10000
Weighted Avg	0.7851	0.7851	0.7851	10000

Table 2: Logistic Regression Confusion Matrix

	Predicted False	**Predicted True**
Actual False	3881	1080
Actual True	1069	3970

Furthermore, the analysis of the Receiver Operating Characteristic (ROC) curve unveiled an area under the curve (AUC) of 0.86, as displayed in Figure 5, affirming the model's robustness in discriminating between positive and negative sentiments.

While logistic regression served as an initial benchmark, our study further delved into employing a spectrum of classification methods, including KNN, LDA, QDA, and Random Forest. Nevertheless, the initial success of logistic regression at 78.5% accuracy and an AUC of 0.86 underscored its importance as an effective starting point in our pursuit of training effective sentiment prediction models for movie reviews.

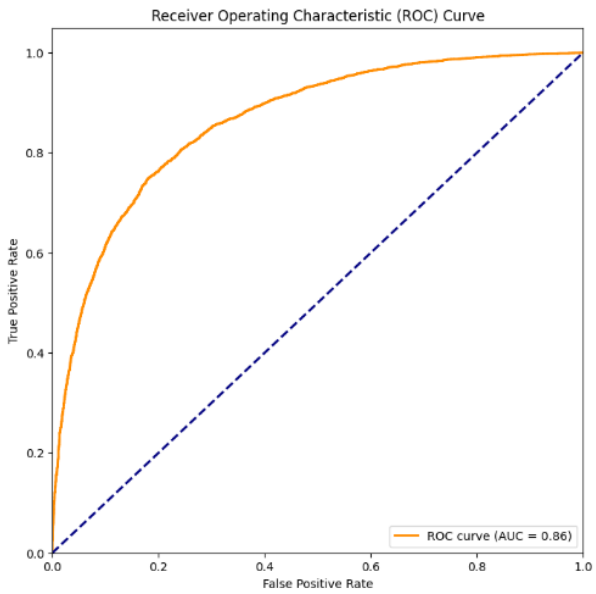

Figure 5: ROC Curve of the Logistic Regression Model

3.2 KNN

Next, we experimented with the performance of K-Nearest Neighbors classification on the dataset. KNN is a supervised learning classification algorithm that compares each testing data point to a training dataset to find a predetermined amount of data points (k) that are closest or most similar to it [3]. From this group of k data points, the algorithm takes the average of their classes to make a prediction for the testing data point. One important quality of KNN is that it is non-parametric which means that it does not make any assumptions about the data.

When employing KNN and testing its performance, it is important to choose a good value for k. Using an inadequate value can upset the balance between underfitting and overfitting of the model [3]. The exact tradeoff point for this balance, otherwise known as the optimal value of k, depends on the sample size of the data. For this reason, we decided to test multiple values of k to determine which value would maximize the accuracy.

Figure 6 compares the testing error and value of k. Usually, we would expect the plot to initially decrease and slowly start increasing as the value of k increases. The minimum point of the plot would be our optimal k. However, the plot seems to consistently decrease. It is possible that the sheer size of the dataset, or overfitting from training, is affecting

Figure 6: Testing Error by Value of K

the optimal value of k. Therefore, we will need to employ a different technique to select our value of k. One common yet less precise method of determining an appropriate choice of k is to let k equal the square root of the sample size; here, the square root of the sample is about 223. To reduce the risk of overfitting, we would prefer to select a medium value of k. Using the scikit-learn library in Python, we fit a KNN model with our medium k = 101 to our data and we found the following results:

Table 3: KNN Testing Results

	Precision	**Recall**	**F1-Score**	**Support**
False	0.7044	0.6648	0.6840	4961
True	0.6873	0.7253	0.7058	5039
Accuracy	-	-	0.6953	10000
Macro Avg	0.6958	0.6951	0.6949	10000
Weighted Avg	0.6958	0.6953	0.6950	10000

Table 4: KNN Confusion Matrix

	Predicted False	**Predicted True**
Actual False	3298	1663
Actual True	1384	3655

Using the KNN algorithm, we achieved an accuracy of 69%. According to the ROC curve, the area under the curve for our KNN model is 0.76. Both figures 6 and 7 reflect a performance that is not as accurate as we would have hoped, especially since KNN is supposed to be flexible and relies on the data instead of parameters for prediction. Additionally, since the KNN algorithm compares each data point one by one, this classification method proved to be computationally expensive. These limitations, coupled with the low accuracy score, led us to explore alternative clas-

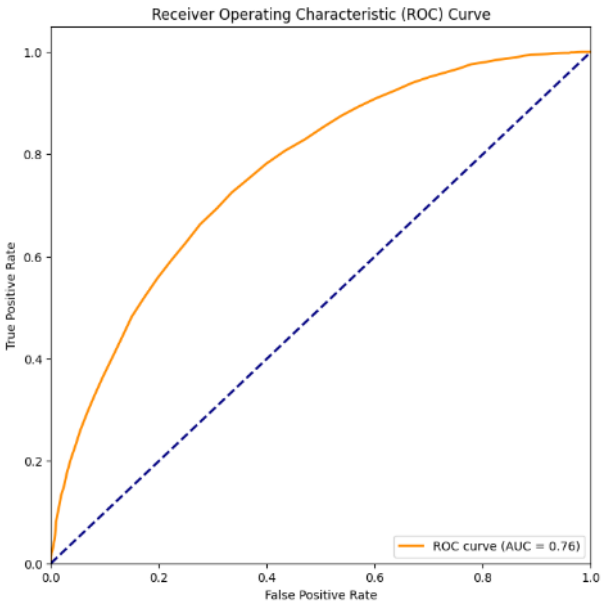

Figure 7: ROC Curve of the KNN Model

sification methods.

3.3 LDA/QDA

We also chose to explore the performance of LDA and QDA on our dataset. Unlike the other models that predict the estimate of the posterior class probabilities, LDA and QDA are generative models, which estimate class-conditional densities [6].

LDA assumes that the features of each class have the same covariance matrix and differ only in their means, making it a linear classifier. In contrast, QDA relaxes this assumption and allows for different covariance matrices for each class, making it a more flexible quadratic classifier.

While LDA is simpler and more computationally efficient, it might not capture complex relationships between variables as effectively as QDA, which can better handle nonlinear boundaries between classes. However, QDA requires more data to estimate covariance matrices accurately and can be more prone to overfitting, especially with smaller datasets.

LDA and QDA offer parametric simplicity and explicit probabilistic interpretation compared to nonparametric models like KNN and random forests, especially in scenarios with smaller datasets. Their flexibility in modeling linear and quadratic decision boundaries equips them to better capture complex data relationships, offering valuable insights and computational efficiency in classification tasks.

Table 5: LDA Testing Results

	Precision	**Recall**	**F1-Score**	**Support**
False	0.7818	0.7754	0.7768	4961
True	0.7807	0.7869	0.7838	5039
Accuracy	-	-	0.7812	10000
Macro Avg	0.7812	0.7812	0.7812	10000
Weighted Avg	0.7812	0.7812	0.7812	10000

Table 6: LDA Confusion Matrix

	Predicted False	**Predicted True**
Actual False	3847	1114
Actual True	1074	3965

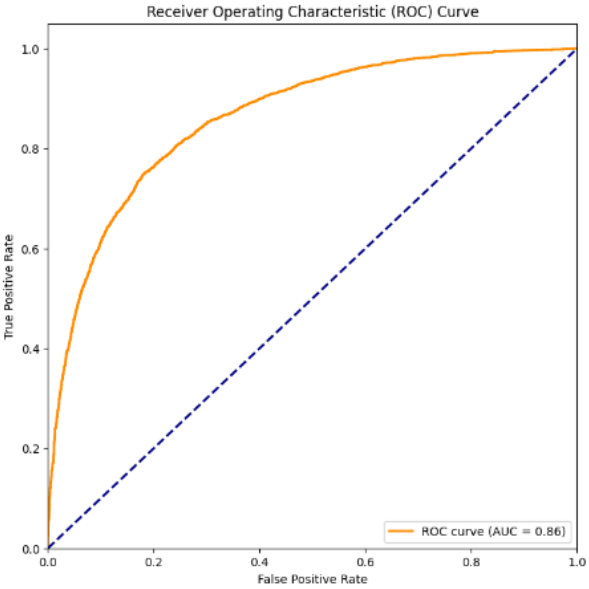

Figure 8: ROC Curve of the LDA Model

Table 7: QDA Testing Results

	Precision	**Recall**	**F1-Score**	**Support**
False	0.7211	0.7349	0.7280	4961
True	0.7340	0.7202	0.7270	5039
Accuracy	-	-	0.7275	10000
Macro Avg	0.7276	0.7276	0.7275	10000
Weighted Avg	0.7276	0.7275	0.7275	10000

Table 8: QDA Confusion Matrix

	Predicted False	**Predicted True**
Actual False	3646	1315
Actual True	1410	3629

Looking at the results, LDA displays higher precision and recall values classes compared to QDA. This implies that LDA has a better overall performance in

correctly identifying instances of both classes. However, QDA, despite a slightly lower accuracy, maintains a comparable balance between precision and recall for the given classes.

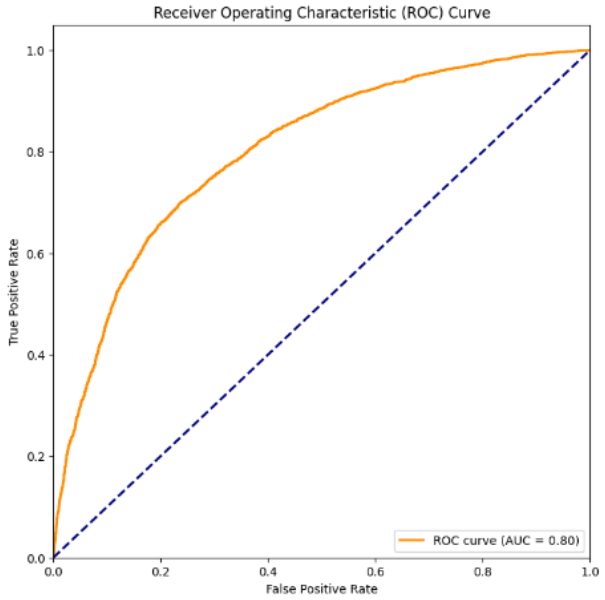

Figure 9: ROC Curve of the QDA Model

Overall, LDA seems to perform marginally better in precision and recall, while QDA shows a balanced performance with competitive metrics, even with a marginally lower accuracy.

3.4 Random Forests

Our last model, Random Forest, is a method that combines the capabilities of a multi-decision tree to improve predictive accuracy and control overfitting. When employing Random Forest, the balance between model complexity and overfitting is important in achieving optimal performance. The hyperparameters in Random Forest, such as the number of trees and their depth, play a crucial role in shaping the model.

Since our aim is not only to optimize accuracy but to also understand how different configurations impact the model's interpretability and generalization across the data review, we focused into the depth of individual trees within the Random Forest.

When we trained the model, we initially chose a practical approach and used the default hyper-parameter values provided by the scikit-learn library. More specifically, our model was instantiated with 100 trees (n_estimators = 100) and various maximum tree depths (max_depths) [4].

We explored the accuracy of different tree depths (5, 10, 15, and 20) to understand how the changing tree depth affects the performance of the Random Forest classifier. Based on the exploration, we identified the optimal tree depth that maximized accuracy. The chosen depth was then used to train a Random Forest classifier on the entire training dataset [1].

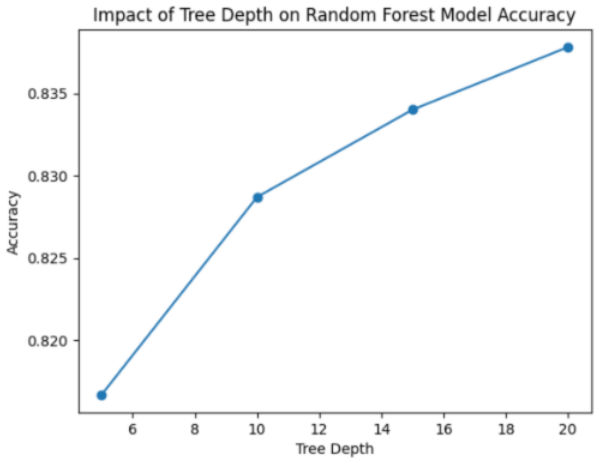

Figure 10: Impact of Tree Depth on Random Forest Model Accuracy

Figure 10 shows that the accuracy of our Random Forest model increased gradually with the tree depth. Starting at an accuracy of 0.82 for a depth of 5, the model showed improvements with accuracies of 0.83 at depth 10 and same at depth 15. The highest accuracy, reaching 0.84, was achieved with a maximum tree depth of 20.

Table 9: Random Forest Testing Results

	Precision	**Recall**	**F1-Score**	**Support**
False	0.8583	0.8061	0.8314	4961
True	0.8199	0.8690	0.8437	5039
Accuracy	-	-	0.8378	10000
Macro Avg	0.8391	0.8376	0.8376	10000
Weighted Avg	0.8390	0.8378	0.8376	10000

Table 10: Random Forest Confusion Matrix

	Predicted False	**Predicted True**
Actual False	3999	962
Actual True	660	4379

Overall, the model seemed to be performing well, with a balanced trade-off between precision and recall. The values in the confusion matrix also suggest that the model made reasonably accurate pre-

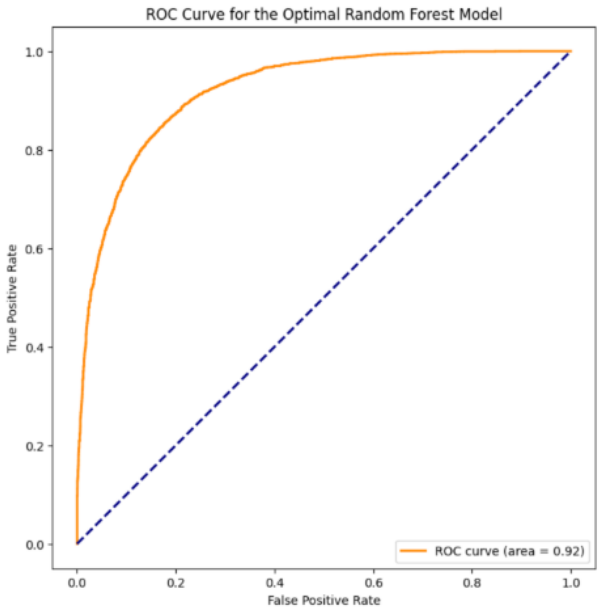

Figure 11: ROC Curve of the Random Forest Model

dictions for both classes. Also, Figure 11 illustrates that the ROC curve is positioned near the top-left corner of the graph, which implies that the model exhibits strong predictive capabilities and achieves a good balance between true positive predictions and false positive predictions.

4 Conclusion and Summary

4.1 Results and Analysis

Table 11: Model Accuracy Results

	Accuracy
KNN	0.6953
QDA	0.7275
LDA	0.7812
LR	0.7851
RF	0.8378

By comparing the model accuracies, we can see that our Random Forest model demonstrated the highest accuracy of 0.84 in our evaluation. This seems to be reasonable because Random Forest naturally captures nonlinear relationships in complex patterns in tasks involving high-dimensional data, such as text analysis on large datasets. We sought to enhance the model's complexity and representation for the Random Forest, allowing it to capture more complex patterns and nonlinear relationships within the data.

Following closely, logistic regression analysis proved its efficiency and simplicity with an accuracy of 0.79. It looks like a good starting point for processing data with simplicity and speed, especially since it is suitable for dealing with binomial outcome predictions in this sentiment analysis.

Logistic regression demonstrated its efficiency and simplicity, establishing itself as an ideal starting point for data processing. Its quick performance and suitability for binomial outcome predictions made it particularly adept for sentiment analysis tasks, offering a straightforward yet effective approach to handle data.

KNN, with an accuracy of 0.69, presented unique challenges, especially concerning the choice of the optimal k value. One possible explanation for the underperformance of KNN is the fact that the algorithm's accuracy suffers when the dimension of the dataset is too large. Although we applied PCA early in the preprocessing stage, it is possible that this was not sufficient to reduce the dimensionality.

LDA/QDA presented their own specific challenges in our analysis. The flexibility of these models led to instances of overfitting, particularly evident in scenarios with numerous classes. This suggests a tendency for these models to excel in training data but may result in suboptimal performance when confronted with a broader, more diverse test set.

This means is that this particular model is excellent at accurately identifying instances of positive classes, meaning that it has a high true positivity rate or high precision for positive classes. In the context of sentiment analysis, this means that the model is particularly effective at accurately identifying and classifying positive sentiments in a given dataset. The higher the "True" classification of the model, the more accurate and reliable it is in predicting positive sentiments or instances of target categories.

Table 12: True Classification Results

	Precision	**Recall**	**F1-Score**
KNN	0.6873	0.7253	0.7058
QDA	0.7340	0.7202	0.7270
LDA	0.7807	0.7869	0.7838
LR	0.7861	0.7879	0.7870
RF	0.8199	0.8690	0.8437

Looking at Table 12, the Random Forest (RF) model achieves the highest precision, recall, and F1 score, indicating its highest accuracy in identifying positive sentiments. Contrarily, the KNN model records the lowest precision among the presented models, suggesting a comparatively lower accuracy in correctly classifying positive instances.

Model "False" Classification signifies the model's accuracy in identifying instances where predictions are incorrect, encompassing both false positives and false negatives. False positives occur when the model incorrectly predicts instances as positive, while false negatives occur when it predicts instances as negative. In sentiment analysis, this means the model may misclassify negative sentiments as positive or overlook positive sentiments.

Table 13: False Classification Results

	Precision	**Recall**	**F1-Score**
KNN	0.7044	0.6648	0.6840
QDA	0.7211	0.7349	0.7280
LDA	0.7818	0.7754	0.7786
LR	0.7840	0.7823	0.7832
RF	0.8583	0.8061	0.8314

Table 13 illustrates that the Random Forest model attains the highest precision, recall, and F1 score, indicating its exceptional accuracy in identifying instances where predictions are incorrect. In contrast, the KNN model records the lowest precision among the presented models, suggesting a comparatively lower accuracy in correctly classifying instances with false predictions.

4.2 Final Remarks

Limitations

Despite the comprehensive exploration and analysis conducted on IMDb movie reviews, there were several limitations throughout our study. The dataset, although sizable, might not encompass the entirety of sentiments expressed by moviegoers. The focus on textual data and the associated sentiment might overlook other influential factors such as user demographics or cultural nuances that impact sentiment perception. Also, while efforts were made to optimize model performance, the challenge of dimensionality in text-based datasets, especially in algorithms like KNN, could have affected the accuracy of predictions despite PCA utilization. Futhermore, while predicting the sentiments, the model may not have been able to capture the complex nuances of sentiments in highly subjective review.

Achievement of Objectives

Our primary objective was to evaluate the performance of various classification methods within the context of combined PCA and TF-IDF techniques. Through meticulous experimentation, we successfully compared the efficacy of Logistic Regression, KNN, LDA/QDA, and Random Forest models in conjunction with PCA and TF-IDF. This allowed us to ascertain the most effective method for sentiment prediction in IMDb movie reviews.

Recommendations for the Future

As a recommendation for future studies, incorporating user-specific features or using context-aware sentiment analysis could improve the accuracy and relevance of sentiment predictions in the dynamic landscape of online reviews.

Furthermore, in order to refine the efficacy of classification methods when integrated with PCA and TF-IDF for sentiment prediction, several avenues for future research stand out. Firstly, exploring more advanced dimensionality reduction techniques beyond PCA, such as t-SNE (t-Distributed Stochastic Neighbor Embedding) or UMAP (Uniform Manifold Approximation and Projection), could offer enhanced feature representations for classification models. Additionally, refining the TF-IDF thresholds based on in-depth analysis or experimenting with different tokenization strategies might further optimize the feature selection process. Evaluating techniques that combine the strengths of multiple classifiers within the PCA-TF-IDF framework could also be promising for sentiment analysis.

References

[1] Bailey, Alexander. "Sentiment Analysis in Less than 50 Lines of Python." Medium, Towards Data Science, 2020, towardsdatascience.com/sentiment-analysis-in-less-than-50-lines-of-python-fc6451114c6.

[2] "IMDb Statistics." IMDb, www.imdb.com/pressroom/stats/. Accessed 1 Dec. 2023.

[3] "K-Nearest Neighbors Algorithm." IBM, www.ibm.com/topics/knn#: :text=Next

[4] Koehrsen, Will. "Random Forest in Python." Medium, Towards Data Science, 2018, towardsdatascience.com/random-forest-in-python-24d0893d51c0.

[5] Liu, Bing. "Opinion Lexicon." Kaggle, 2017, www.kaggle.com/datasets/nltkdata/opinion-lexicon/.

[**6**] Maugis. "Variable Selection in Model-Based Discriminant Analysis." Journal of Multivariate Analysis, Academic Press, 2011, www.sciencedirect.com/science/article/pii/S0047259X11000753#: :text=Generative

[**7**] "PCA-How to Choose the Number of Components?" Bartosz Mikulski - AI Consultant, mikulskibartosz.name/pca-how-to-choose-the-number-of-components. Accessed 4 Dec. 2023.

[**8**] "What Is Logistic Regression?" Amazon, The University, 1978, aws.amazon.com/what-is/logistic-regression/.