Movie Review Sentiment Prediction

Rebekah Limb (705576239), Elena Bateman (605520224) , Alyssa Lung (905536285), Ashley Jones (205479641), Misha Reiss (505580077), Ethan Warren (405608185)

Abstract

Internet Movie Database, or IMDb, is an online database related to films, television programs, home videos, video games, and streaming content. The platform allows registered users to submit reviews for content on the site, generally with a positive or negative sentiment. We will be analyzing the relationship between the review posted and the review sentiment, ultimately predicting sentiment from the review written. We will utilize TF-IDF scores, K-means clustering, logistic regression, K-nearest neighbors, determinant analysis, and random forests to perform our analysis.

1 Introduction

Want to watch a good movie but don't know where to start? IMDB is your go to platform for all information related to films, tv shows, and much more in the realm of entertainment media. But, with thousands of reviews, how do you quickly decide which film is worth your time? That is where our project steps in. We're looking into an IMDB dataset with 50,000 reviews, evenly split for training and testing. Our aim is to search for clues and key words in reviews that reveal whether or not the review is positive or negative. We're exploring the relationship between ratings and reviews and our goal is to have a system capable of sifting through thousands of words to detect the underlying sentiment of the review.

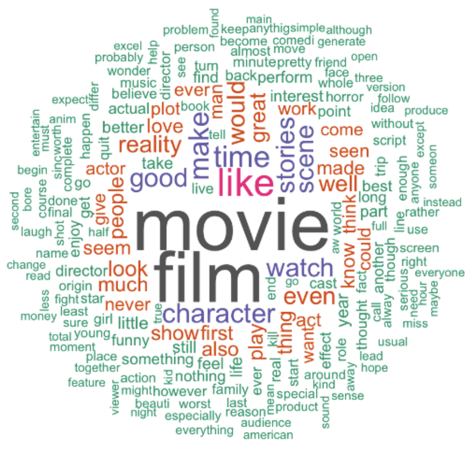
2 Descriptive Statistics

To best understand the review data, we constructed descriptive statistics. After randomly splitting the data set into 70% training and 30% testing, we were left with 700 training samples and 300 testing samples.

The training dataset consisted of 35,000 reviews, with 50% positive sentiment and 50% negative sentiment.

Our constructed testing dataset consisted of 15,000 observations, with 50% positive sentiment and 50% negative sentiment.

After cleaning the reviews, words such as "movie," "film," "like," "time," and "good" appear most frequently.



2.1 Feature Engineering

We are not able to extract meaningful information with the raw review data itself, so we performed feature engineering on our data to provide more useful predictors.

First, we cleaned the data to remove all stop words, unnecessary punctuation, and non-english words. We also used stemming in order to reduce the total number of words across the corpus and combine all variations of a root word into one group.

We first attempted to use the Bag-of-words numericalization method for our review test, by using a provided sentiment lexicon as the sentiment dictionary to vectorize the text. There were a total of 6783 words in the lexicon. Even after reducing the lexicon dictionary size by stemming to base words, we were left with 4034 columns. Moreover, we performed PCA on the features and we saw that there was not a viable dimension reduction that would be useful for modeling and still capture the majority of the variance. 75% of the variance was captured by the first 3000 components, which would be difficult to reduce.

We opted to use Term Frequency-Inverse Document Frequency scores to create our dictionary. It is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus). So we used TF-IDF scores to output words between a minimum of 0.05 and maximum 0.7 document frequency, which is how often a certain word appears in a document. We get a vector of only 331 words which simultaneously reduces our dimension of the model significantly. This list represents the most important and distinctive words relative to the entire corpus which we then use as our dictionary to vectorize the words in the review data.

3 Experiment

To properly analyze and model the dataset for prediction, we used 4 different methods: Logistic Regression, KNN, Discriminant Analysis, and Random Forest models.

3.1 Logistic Regression

Logistic regression is a statistical method used for modeling the probability of a binary outcome. It's a type of regression analysis commonly used in the field of machine learning and statistics for predicting the probability of an event occurring. The logistic regression model uses the logistic function (also called the sigmoid function) to model the relationship between the independent variables and the probability of the event occurring. The logistic function has an S-shaped curve that maps any real-valued number to a value between 0 and 1. The formula for the logistic function is:

$$P(Y=1) = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_k X_k}}$$

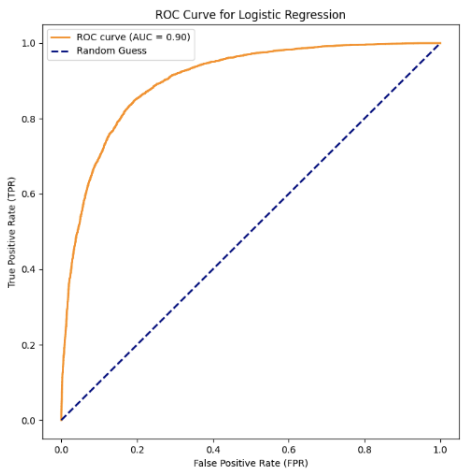
Where $P(Y=1)$ is the probability of the next event occurring, $\beta_0, \beta_1, ..., \beta_k$ are the coefficients and $X_1, ..., X_k$ are independent variables.

Analyzing the results from the model, we get an overall accuracy of 83%. The model is better at classifying positive sentiment reviews compared to negative reviews as displayed by the recall rates in the classification report. The classification report displays better precision for negative reviews (negative reviews were coded as 0 and positive were 1) with a value of 0.84 compared to 0.81. The F-1 score, which evaluates both the precision and true positive or negative rate was nearly equal for both positive and negative reviews, at 0.83 and 0.82 respectively.

Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.81	0.85	0.83	7526
Negative	0.84	0.8	0.82	7474
Accuracy	-	-	0.83	15000
Macro Avg	0.83	0.83	0.83	15000
Weighted Avg	0.83	0.83	0.83	15000

The ROC curve for logistic regression compares the False Positive Rate and True positive rates. The area under the curve (AUC) is 0.90, indicating a strong performance.



3.2 K-Nearest Neighbors

The K nearest neighbors (KNN) algorithm is widely used in machine learning. It is particularly helpful for classification and regression tasks. Its main principle involves predicting the label of a data point based on the labels of its neighbors in the training dataset. This algorithm is non-parametric and thus is highly flexible with no assumptions needed.

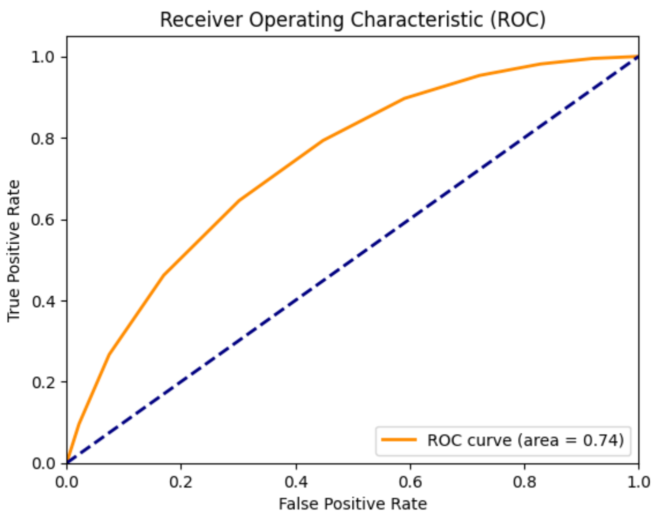
After training the model on the data set using five-fold cross validation, we have the ability to then select the optimal "K" value

K-Values

	K	Accuracy	Kappa
0	5	0.667400	0.334510
1	7	0.671267	0.342221
2	9	0.672733	0.345128

When you observe the accuracy of each various K-value, it can be revealed that K=9 had the highest accuracy of points that were classified correctly (Accuracy = 67.27%). Kappa is similar but is adjusted for the possibility of chance. A value of 1 implies perfect agreement and a value of 0 implies that the agreement is no better than chance. Each kappa is between 0.33 and 0.35, indicating that the level of agreement between the model's predictions and the actual data can be considered as "fair," but definitely not strong. The model is performing better than random chance, but its predictive accuracy is not highly reliable. A possible reason for this is that KNN may be too simple for the complexity of the data.

With that being said, the highest kappa is at K=9 where Kappa = 0.345, thus showing that K=9 is the best option for our KNN model with an expected accuracy of 67.27%. From there we run the KNN with K=9:



Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.64	0.79	0.71	7516
Negative	0.73	0.55	0.63	7484
Accuracy	-	-	0.67	15000
Macro Avg	0.68	0.67	0.67	15000
Weighted Avg	0.68	0.67	0.67	15000

Confusion Matrix on Test Set	
4129	3355
1554	5962

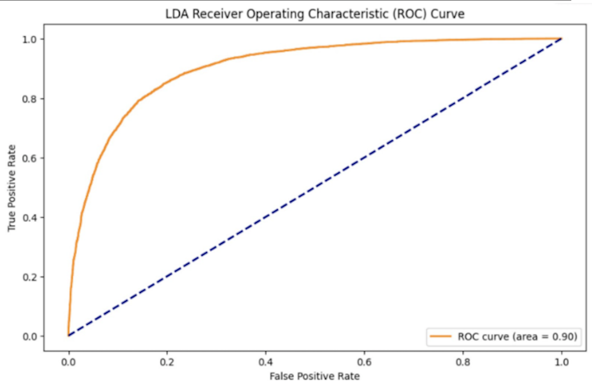
With an Area Under the Curve (AUC) of 0.74, the model demonstrates a pretty fair ability to discriminate between positive and negative reviews. This indicates that, on the test set, the KNN classifier is considerably better than a random guess at distinguishing the sentiment of the reviews. The curve does not approach the upper left corner, indicating a higher true positive rate (TPR) and a lower false positive rate (FPR). Although, while the model is reasonably good at correctly identifying positive reviews (true positives), there is still a noticeable rate of false positives. For this reason, it is important that we take a look at the other models.

3.3 Discriminant Analysis

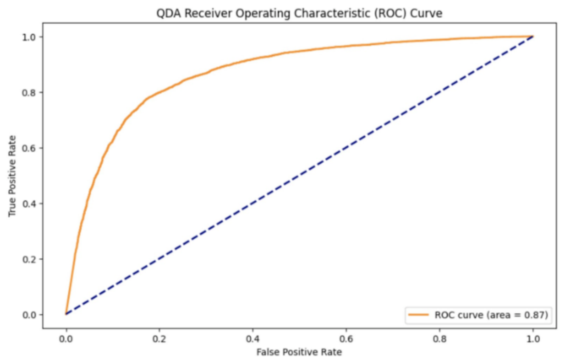
Discriminant analysis (DA) is generally used in classification problems where the response variable is categorical. It is a generative model, which studies the joint probability distribution of the predictors and response variable. The previously mentioned methods K-nearest neighbors and logistic regression are examples of discriminative models which, alternatively, study the conditional probability of the response variable given the data. Discriminant analysis has a few key advantages. It is a simple and computationally efficient algorithm and it can work well even when the number of features is much larger than the number of training samples. Additionally, there are no hyperparameters to tune in DA. The two types of discriminant analysis used in this study are Linear Discriminant Analysis (LDA) and Quadratic

Discriminant Analysis (QDA). In both LDA and QDA, it is assumed that the conditional distribution of the predictors given the response variable follows a multivariate normal distribution for each possible response value ("positive" or "negative" in this study). In addition, the assumption is made that each group has a different mean vector. In LDA, the covariance matrices are assumed to be the same. However, in QDA, they are assumed to be different.

LDA				
	Precision	Recall	F1-Score	Support
Positive	0.8080	0.8540	0.8310	7501
Negative	0.8450	0.7980	0.8260	7499
Accuracy	-	-	0.8259	15000
Macro Avg	0.8265	0.8260	0.8285	15000
Weighted Avg	0.8248	0.8265	0.8278	15000



QDA				
	Precision	Recall	F-1 Score	Support
Positive	0.7740	0.8320	0.8020	7501
Negative	0.8180	0.7570	0.7940	7499
Accuracy	-	-	0.7941	15000
Macro Avg	0.7960	0.7945	0.7980	15000
Weighted Avg	0.7942	0.7955	0.7974	15000



A Receiver Operating Characteristic (ROC) Curve shows the performance of a classification model at all classification thresholds. The Area Under the Curve (AUC) is an accuracy measure that quantifies the ability of the classifier to correctly distinguish between classes. The AUC in the ROC for LDA is 0.903 and the AUC in the ROC for QDA is 0.869. After examining both tables and graphs, LDA outperforms QDA in almost every metric including AUC and the other accuracy metrics displayed in the tables. There are a few reasons why this was the case. LDA performs well in binary classification when there is linear separability, meaning that the boundary that separates the two classes is linear. Also, as previously mentioned, QDA estimates a covariance matrix for each class and this estimate can be very sensitive to noise in the data. This noise can lead to suboptimal performance which may be the reason for the accuracy dropoff.

3.4 Random Forest

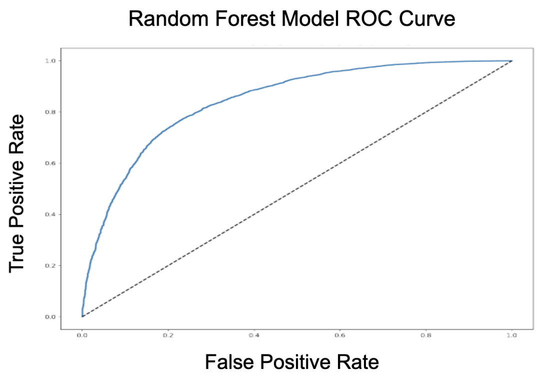
Decision Trees are a powerful tool in statistical modelling, because they are non-parametric models, meaning that they can approximate any function. Random Forests are built on the idea of decision trees, but enhance the idea by incorporating many non-correlated decision trees into the model, each trained on a unique, randomly selected subset of predictors and data. The strength of a Random Forest model lies in its collective approach: it bases its final prediction on the most frequent outcome from its individual Decision Trees. This aggregation results in a model that often surpasses other models in predictive accuracy.

We trained a random forest model to see how well it could predict the sentiment of a movie review. This model, like our previous models, uses the engineered columns instead of the whole review to make predictions, and whether the review was positive or negative as the response variable.

Random Forest Models are a bit unique in how they are trained because they have a lot of hyperparameters that can dramatically affect the model's performance. Because of this, in our training we used random search cross validation to search through many different combinations of hyperparameters to find the best one. We found that the best hyperparameters to use were bootstrap set to True with the entropy criterion, a max depth of 13 and the min samples per leaf to be 8, min samples split to be 6, and the number of estimators to be 714.

However, a drawback of the random forest model is that it is very computationally expensive to train. Because of this, we were not feasibly able to test every combination of hyperparameters for the model. The results of the final model are displayed below.

Random Forest				
	Precision	Recall	F1-Score	Support
Negative	0.8113	0.7885	0.7997	7474
Positive	0.7956	0.8178	0.8066	7526
Accuracy	-	-	0.8032	15000
Macro Avg	0.8034	0.8031	0.8031	15000
Weighted Avg	0.8034	0.8032	0.8031	15000



We can see that the Accuracy is pretty consistent, and the F1 score is high for the model across both positive and negative reviews. The ROC curve looks good but is not as close to the top-left as we would hope, which would indicate good classification for true positives and low false positives. Since the data is imbalanced, we should be extra considerate of this.

4 Conclusion

Here we conclude with our results and analysis.

4.1 Results and Analysis

To evaluate our models, we initially assess their performance based on accuracy scores using our test dataset. Here, we present the accuracy scores of each model, arranged in ascending order from the least accurate to the most accurate.

Table 9: Model Accuracy

	Accuracy
KNN	0.6727
QDA	0.7941
RF	0.8032
LDA	0.8259
LR	0.83

We can see that our Logistic Regression model performed the best with an accuracy of 0.83. This makes sense as Logistic Regression Models models are expected to do very well when predicting because of its simple, linear decision boundary which reduces overfitting.

Following our logistic regression model, we see that Linear Discriminant Analysis model was the next best with an accuracy of 0.8259. We think this is because it maximizes the ratio of the between-class variance to the within-class variance and reduces overfitting.

Interestingly, K-Nearest Neighbors performed the lowest in terms of accuracy scores, with a score of 0.6727. This guides us to believe that this is because KNN models are too flexible and they may overfit the training data, leading to a lack of performance.

To deepen our comprehension of our models' effectiveness and to substantiate our conclusions, we proceed to contrast their capabilities in predicting positive vs negative, designated as "True" responses, which constitute the predominant category in our test set. We then outline the precision, recall, and F1-scores of each model for these "True" responses, organized by ascending F1-scores.

Table 10: Model "True" Classification Results

	T Precision	T Recall	T F1-Score
KNN	0.64	0.79	0.71
QDA	0.7740	0.8320	0.8020
RF	0.8113	0.7885	0.7997
LDA	0.8080	0.8540	0.8310
LR	0.81	0.85	0.83

We can see that Random Forest and Logistic Regression had the highest precision (0.8113 and 0.81, respectively), meaning most of its predictions of "True" were actually true. Thus their linear boundary from training was able to capture a sizable portion of the true values. KNN had the lowest precision at 0.64 and the second lowest recall of 0.79. LDA and Logistic Regression had the highest recall at 0.854 and 0.85, respectively. Even though LDA had the highest recall value, it had the third highest precision value of 0.8080. LDA is identifying a large portion of the relevant instances (high recall), but a significant number of the instances it predicts as positive are actually not relevant (low precision). This scenario is often referred to as a "high false positive rate" or "low positive predictive value." With Logistic Regression having the highest precision and recall value, we see that it can most accurately a good job of generalizing the information it gained from the training data to classify "True"s. Additionally, this is why we can see it has the highest F1-Score, which is a balance of precision and recall.

Table 11: Model "False" Classification Results

	F Precision	F Recall	F F1-Score
KNN	0.73	0.55	0.63
QDA	0.8180	0.7570	0.7940
RF	0.8113	0.7885	0.7997
LDA	0.8450	0.7980	0.8260
LR	0.84	0.8	0.82

As seen in the table, LDA and Logistic Regression outperform the other models in terms of "false" classification precision (the ratio of correct "false" classifications to total "false" classifications) with values of 0.845 and 0.84, respectively. LDA and Random Forest had the highest false recall. Models with better recall capture a higher proportion of the actual negative instances. Overall, LDA appears to be the most balanced model, achieving the highest score in precision and the second highest in recall, resulting in a high F1-score. LDA involves finding linear combinations of features that maximize the separation between classes. This causes LDA to accurately extract discriminative information. Alternatively, KNN produced the lowest precision, recall, and F1-score. This is a result of KNN's sensitivity to noise which can skew the decision boundaries. QDA and Random Forest did not excel in any one specific metric but were able to provide generally accurate classifications.

4.2 Final Remarks

Although all of our models performed well above the 50% classification rate, the threshold indicating random chance, we recognize there are more methods to improve our results.

A main concern is the potential violation of a key assumption while modeling – independence in the data. Since the data was collected on an online public forum, different reviewers could be influenced by other reviews, impacting the true perception and ultimately the independent review.

Another area of consideration was during our feature engineering task where we implemented the use of TF-IDF scores to create our numericalization dictionary. This approach employed a basic text analysis counting word frequencies in isolation, which results in not being able to capture the contextual nuances of positive or negative words.

The dictionary of words produced by the TF-IDF scoring system included words such as "movie", "film", "people". These words are not necessarily related to a sentiment as these types of words can appear in both positive and negative reviews. Thus these words might not be as meaningful as predictors.

Lastly, another area for improvement would be in our dimension reduction. We filtered the TF-IDF scores to only include words with a maximum document frequency of 0.7, and a minimum document frequency of 0.1. This parameter could be tuned to include more or less words.

References

[1] Bessa, A. (2022, March 17). Lexicon-Based Sentiment Analysis: A Tutorial. KNIME. https://www.knime.com/blog/lexicon-based-sentiment-analysis

[2] Education Ecosystem. (2018, September 12) Understanding K-means Clustering in Machine Learning | by Education Ecosystem (LEDU) | Towards Data Science. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1

[3] Geeks for Geeks. Understanding TF-IDF (Term Frequency-Inverse Document Frequency). https://www.geeksforgeeks.org/understanding-tf-idf-term-frequency-inverse-document-frequency/

[4] Heavy AI. What is Feature Engineering? Definition and FAQs | HEAVY.AI. (n.d.). Retrieved December 6, 2023, from https://www.heavy.ai/technical-glossary/feature-engineering

[5] Ziqi, Y. (2019). IMDB dataset (Sentiment analysis) in CSV format | Kaggle. https://www.kaggle.com/datasets/columbine/imdb-dataset-sentiment-analysis-in-csv-format