Stats 101C Final Project

Minhao Han, Laura Ngo, Nishaanth Krishnan, Damien Ha

December 2023

Abstract

This study explores the pivotal role of sentiment analysis in decoding audience responses within IMDB film reviews. We find that parametric models are better suited given their superior ability in high-dimensional feature spaces and we can apply dimension reduction techniques such as PCA to reduce performance metrics' differences.

⋆ ⋆ ⋆ ⋆ ⋆

1 Introduction

Ever wondered how film review sentiments shape the film's fate and influence our cinematic experiences?

With over 50K IMBD reviews, sentiment analysis in film reviews has played a critical role in gauging audience interpretation and reaction to determine the film's success level. Filmmakers often depend on sentiment analysis to refine projects and marketing tactics to adhere to and align with intended audience preferences and enhance viewer satisfaction.

Generally, an IMBD movie review includes the following components:

1. **1-10 rating:** An overall rating to reflect the overall film satisfaction with 1/10 being appalling and 10/10 being outstanding.
2. **Text review:** Includes the user's written commentary and opinions to the film's different aspects.

We will be analyzing the relationship between text reviews and the corresponding sentiment by employing both parametric and nonparametric machine learning models (Log Reg, KNN, Random Forest, LDA, QDA) to compare performances across different models.

2 Preprocessing Step

The IMDB dataset, integral to sentiment analysis, consists of about 50,000 movie reviews evenly divided between positive and negative sentiments. These reviews, varying in length and complexity, provide a diverse range of vocabulary and expressions reflecting user opinions on movies. Each review is labeled either 'positive' or 'negative'. At a high level, our data preprocessing includes text cleaning, normalization, and vectorization using TF-IDF. These steps are essential for preparing the data for subsequent analysis with machine learning models.

2.1 Text Cleaning and Normalization Process

The text cleaning and normalization process in the IMDB dataset involves several key steps. Initially, text cleaning is conducted to remove HTML tags and non-alphanumeric characters, along with standardizing text (e.g., converting to lowercase). This is followed by normalization, which includes handling contractions, removing stop words, and employing stemming or lemmatization to reduce words to their base or root forms. Next, TF-IDF vectorization is applied. This technique converts text data into numerical vectors by word frequency, where each word is assigned a value between [0, 1]. For our pipeline, we choose to keep all words (we set the minimum and maximum frequency to 0 and 1 respectively), resulting in 101895 feature columns, which we continue to engineer in the next section. The process of feature engineering, integral to preparing the data for machine learning models, focuses on extracting meaningful attributes from the preprocessed text to aid sentiment analysis.

2.2 Sentiment Lexicon Creation

The use of TF-IDF in the sentiment analysis of the IMDB dataset is tailored to construct numerical vectors based on word frequency, while incorporating a custom sentiment lexicon. This lexicon, consisting of words with clear positive or negative connotations, is created by filtering an existing sentiment lexicon to include only words marked as 'positive' or 'negative'. In feature selection, the lexicon is used to refine the TF-IDF vectorized matrix by retaining only those features (words) that are also in the sentiment lexicon. This approach narrows the focus to words with strong sentiment indicators, potentially enhancing the effectiveness of the machine learning models used. The subsequent analysis employs various algorithms such as Logistic Regression, SVM, Random Forest, and K-Nearest Neighbors. These models are selected for their suitability in handling high-dimensional data, and their performance is evaluated based on their ability to accurately classify the sentiments of movie reviews. This method highlights the significance of integrating TF-IDF analysis with domain-specific knowledge to improve feature selection relevance and accuracy in sentiment analysis.

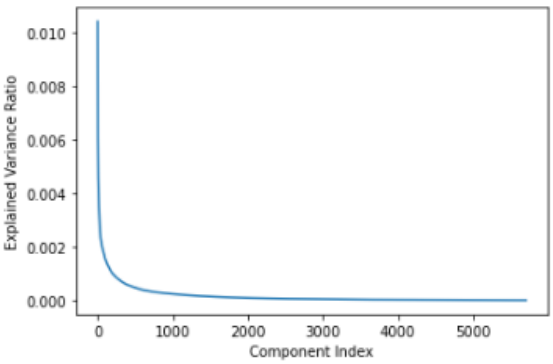

Figure 1: Explained Variance Ratio of all components

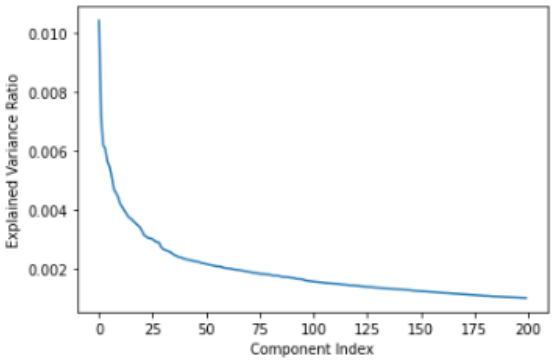

Figure 2: Explained Variance Ratio of top 200 components

2.3 Dimensionality Reduction and Principal Component Analysis

After applying TF-IDF vectorization and filtering by sentiment lexicon, our resulting feature matrix has 5706 columns, which is extremely high in dimensionality. We hypothesize that nonparametric models, especially KNN, will struggle in performance as a result, and that dimensionality reduction techniques will help reduce this gap in performance. We choose to use PCA to reduce the dimensionality of the feature matrix after plotting and visualizing the explained variance ratio of all components.

We start by constructing all components of our feature matrix and visualizing the explained variance ratio. From our visualization, our reasoning for selecting only the top 200 components for our new feature matrix is twofold. First, we observe a noticeable "elbow" in the explained variance ratio roughly around the 200th component, a popular PCA heuristic in machine learning literature. Second, we observe a plateau in explained variance ratio from the 200th component onward. Our subsequent analyses will cover both the raw feature matrix and also the dimensionality-reduced feature matrix.

3 Experiment

Given the models Logistic Regression, Random Forest, KNN, LDA, and QDA, we must find the most accurate and efficient model for prediction.

3.1 Logistic Regression

We tune the regularization strength hyperparameter (denoted by C) using five-fold cross-validation and determine C = 0.80 to be optimal.

Logistic regression is a statistical method of binary classification where the outcome variable is categorical and has two classes denoted 0 and 1.

In this case, those classes are negative and positive. It assumes a linear relationship between the independent variable and the log odds of the dependent variable and employs a sigmoid function to transform the linear combination of input features, mapping them to a value in the range [0, 1].

As previously mentioned, we have run logistic regression both before and after utilizing PCA, a mathematical technique used to simplify and reduce the complexity of data while retaining its essential information. The principle components are the directions in which the data varies the most, and the data is projected onto a lower dimensional space while retaining the most significant information.

	Precision	Recall	F1-Score	Support
Positive	0.84	0.88	0.86	5039
Negative	0.87	0.83	0.85	4961
Accuracy			0.85	10000
Macro Avg	0.85	0.85	0.85	10000
Weighted Avg	0.85	0.85	0.85	10000

Table 1: Logistic Regression Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.83	0.87	0.85	5039
Negative	0.86	0.81	0.84	4961
Accuracy			0.84	10000
Macro Avg	0.84	0.84	0.84	10000
Weighted Avg	0.84	0.84	0.84	10000

Table 2: Logistic Regression Classification Report (after applying PCA)

The tables above show the results of performing logistic regression on our data both before and after PCA. Precision is the ratio of true positive predictions to the total number of predicted positives. Recall, on the other hand, is the ratio of true positive predictions to actual positive instances. F1-Score is a harmonic mean between precision and recall and provides a balanced measure between both false positives and false negatives.

Before PCA, positive sentiment seems to have better recall and F1-score while negative sentiment performs better in terms of precision. This also seems to be the case after PCA as well, yet the overall performance is worse in all categories. This may be due to information loss in the dimensionality reduction, non-linearity between the independent variable and its predictors, amplification of noise due to PCA, or overfitting or underfitting. We can also see that our macro average, which assesses model performance with equal weight across all classes is fairly high, as is the weighted average which takes into account class imbalances.

Overall, logistic regression performed relatively well in this experiment, with overall accuracy of 85% and 84% before and after employing PCA respectively.

3.2 LDA

Linear Discriminant Analysis, or LDA, aims to find a linear combination of features that best characterizes or discriminates between two or more classes in the data. It assumes the normality of features within classes and equal covariance matrices for all classes. It also uses a decision rule based on the class means and the within-class and between-class scatter matrices to classify new instances. Generally speaking, this model classifies by maximizing the separation between classes.

	Precision	Recall	F1-Score	Support
Positive	0.82	0.85	0.83	5039
Negative	0.84	0.81	0.83	4961
Accuracy			0.83	10000
Macro Avg	0.83	0.83	0.83	10000
Weighted Avg	0.83	0.83	0.83	10000

Table 3: LDA Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.82	0.87	0.84	5039
Negative	0.86	0.81	0.83	4961
Accuracy			0.84	10000
Macro Avg	0.84	0.84	0.84	10000
Weighted Avg	0.84	0.84	0.84	10000

Table 4: LDA Classification Report (after applying PCA)

The results of LDA before and after applying PCA are shown above. Overall Logistic Regression is slightly more accurate, but LDA seems to perform fairly accurately too. In general, positive sentiment has better recall and a higher F1 score while negative sentiment has better precision. LDA might be slightly better at detecting positive sentiment, though it seems to perform

decently well overall. The application of PCA seems to slightly improve the results of the model across the board, with accuracy increasing from 83% to 84%.

3.3 QDA

Quadratic discriminant analysis, or QDA, is a technique extending LDA to cases where the covariance matrices of different classes are not assumed to be equal and are thus, independent. It also assumes a quadratic decision boundary such that it models the probability density function (PDF) of each class separately. The decision rule in QDA involves assigning an observation to the class with the highest posterior probability which typically involves Bayes rule.

	Precision	Recall	F1-Score	Support
Positive	0.70	0.21	0.32	5039
Negative	0.53	0.91	0.67	4961
Accuracy			0.56	10000
Macro Avg	0.62	0.56	0.50	10000
Weighted Avg	0.62	0.56	0.50	10000

Table 5: QDA Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.76	0.87	0.81	5039
Negative	0.84	0.72	0.78	4961
Accuracy			0.79	10000
Macro Avg	0.80	0.79	0.79	10000
Weighted Avg	0.80	0.79	0.79	10000

Table 6: QDA Classification Report (after applying PCA)

The results of running QDA are shown above, and it is immediately obvious that this model has a poorer performance than logistic regression and LDA, at least on its own. QDA performs somewhat well with precision for positive sentiment, achieving 70% precision, so the model at least somewhat accurately matches positive predictions. However, other metrics perform rather poorly, with an overall accuracy of 56%.

When we employ PCA alongside QDA, the model's performance improves drastically with an overall accuracy of 70%. QDA appears to face challenges when handling high dimensionality, especially because covariance matrix estimation becomes difficult with many features. This increases the risk of overfitting. The curse of dimensionality can easily become a problem for QDA, especially if instances become spread thinly across the feature space.

Overall, PCA can help alleviate these problems through dimensionality and noise reduction. This way, QDA can become more generalizable, overfit less, and perform better on otherwise unseen instances.

3.4 Random Forest

Random Forest, an ensemble learning method, works by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees. In our Random Forest analysis, we dedicated considerable effort to hyperparameter tuning, focusing on two critical parameters: tree depth and the number of estimators. Through extensive five-fold cross-validation, we determined the optimal combination to be a tree depth of 80 and 120 estimators.

This model is particularly effective due to its ability to handle high-dimensional datasets and its robustness against overfitting, which is often a challenge with complex models. The diversity introduced by the ensemble reduces variance, making the model more stable and accurate. The following tables display the classification report of the Random Forest model before and after applying PCA. Metrics include Precision, Recall, F1-Score, and Support for both positive and negative sentiments, as well as overall accuracy and averages.

	Precision	Recall	F1-Score	Support
Positive	0.83	0.85	0.84	5039
Negative	0.85	0.83	0.84	4961
Accuracy			0.84	10000
Macro Avg	0.84	0.84	0.84	10000
Weighted Avg	0.84	0.84	0.84	10000

Table 7: Random Forest Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.81	0.82	0.81	5039
Negative	0.81	0.80	0.81	4961
Accuracy			0.81	10000
Macro Avg	0.81	0.81	0.81	10000
Weighted Avg	0.81	0.81	0.81	10000

Table 8: Random Forest Classification Report (after applying PCA)

The analysis of the model's performance in sentiment analysis, both before and after the application of PCA, reveals some insightful trends. Initially, the model demonstrates a slightly higher precision in classifying negative reviews (0.85) than positive ones (0.83), suggesting a marginally better accuracy in identifying negative sentiments. Conversely, its recall is higher for positive reviews (0.85) compared to negative ones (0.83), indicating a greater effectiveness in capturing actual positive cases. The F1-Scores, standing at a balanced 0.84 for both classes, reflect the model's overall robustness in maintaining a balance between precision and recall. This is further corroborated by an overall accuracy of 0.84, a strong indicator of the model's proficiency in sentiment classification.

However, upon applying PCA, a technique intended for dimensionality reduction, there is a noticeable though slight decrease in both precision and recall across both classes. The F1-scores dip to 0.81 for both positive and negative sentiments, implying a minor reduction in the balance between precision and recall. This drop is also mirrored in the overall accuracy, which falls to 0.81. This decline, although not substantial, suggests that in this specific context, the Random Forest model might be more effective without the application of PCA. The decrease in performance post-PCA could be attributed to the loss of certain informative features during the dimensionality reduction process, underscoring the model's inherent strength in handling high-dimensional data without significant information loss.

3.5 KNN

For the K-Nearest Neighbors (KNN) model, we conducted a thorough hyperparameter tuning process, particularly focusing on the number of neighbors. Through five-fold cross-validation, we determined that K = 10 neighbors yields the optimal results. KNN operates on a simple principle of detecting the closest data points in the feature space and predicting the class based on the majority class of these neighboring points.

KNN's simplicity can be both an advantage and a limitation. Its performance heavily relies on the feature space's dimensionality and the distance metric used. In high-dimensional spaces, distances can become less meaningful, leading to poorer performance. This phenomenon is often referred to as the "curse of dimensionality". To mitigate this, PCA can be employed to reduce the number of dimensions while retaining the most critical information.

The tables below show the classification reports for KNN before and after the application of PCA. These reports include key metrics such as Precision, Recall, F1-Score, and Support for the positive and negative classes, as well as overall accuracy and averages.

	Precision	Recall	F1-Score	Support
Positive	0.54	0.98	0.70	5039
Negative	0.89	0.14	0.24	4961
Accuracy			0.57	10000
Macro Avg	0.71	0.56	0.47	10000
Weighted Avg	0.71	0.57	0.47	10000

Table 9: KNN (K = 10) Classification Report

	Precision	Recall	F1-Score	Support
Positive	0.64	0.91	0.75	5039
Negative	0.84	0.49	0.62	4961
Accuracy			0.70	10000
Macro Avg	0.74	0.70	0.69	10000
Weighted Avg	0.74	0.70	0.69	10000

Table 10: KNN (K = 10) Classification Report (after applying PCA)

The K-Nearest Neighbors (KNN) model, evaluated without the use of Principal Component Analysis (PCA), demonstrates a distinct performance characteristic in sentiment classification, as shown in the provided classification report.

Notably, there is a significant disparity in precision between positive and negative sentiments. For positive sentiments, the model achieves a lower precision of 0.54, indicating a less reliable prediction for positive reviews. In contrast, it exhibits a much higher precision of 0.89 for negative sentiments, suggesting a greater accuracy in identifying negative reviews. This disparity in precision is further mirrored in the recall scores, where the model shows an exceptionally high recall of 0.98 for positive reviews, implying it is highly effective at capturing most of the actual positive cases. However, for negative reviews, the recall drastically drops to 0.14, indicating a significant number of negative sentiments are being missed.

The overall accuracy of the model stands at 0.57, reflecting a moderate level of correctness in

classifying sentiments. The macro average and weighted average metrics, both hovering around the 0.70 mark, suggest that while the model may be somewhat effective in certain aspects (like detecting positive sentiments), it is considerably imbalanced, struggling notably with the accurate classification of negative reviews. This imbalance in performance highlights the model's sensitivity to the distribution and nature of the data, a common characteristic in distance-based algorithms like KNN.

4 Conclusion and Summary

We will wrap up our findings and analysis, along with some concluding thoughts.

4.1 Results and Analysis

In this study, we observed the pivotal role of sentiment analysis in decoding audience responses within IMDB film reviews. Our exploration of our 50K IMBD film review dataset involved the application of several parametric models including Log Regression, LDA, QDA, and non-parametric models including Random Forest and K-Nearest Neighbors. These models were then evaluated based on their accuracy before and after dimensionality reduction through Principle Component Analysis.

We begin by reviewing and comparing our accuracy scores for each model. We have created a table below from the greatest to least accurate model (after PCA).

Model	Accuracy (%) Before/After PCA
Log Regression	85 %, 84 %
LDA	83 %, 84 %
Random Forest	84 %, 81 %
QDA	56 %, 79 %
KNN	57 %, 70 %

Table 11: Model Accuracy Comparison

The accuracy comparison table provides a quick overview of each model's performance and allows us to identify which has the highest accuracy.

Here we can observe that Logistic Regression demonstrated the highest level of accuracy among the models, reaching 85 % and 84 % before and after PCA, respectively. This aligns with our previous expectations that parametric models are efficient in providing accurate predictions in high-dimensional spaces.

Random Forest displayed the second-highest with an accuracy of 84 % and 81 % for before and after PCA. This method is known for its robustness in handling complex and diverse feature sets. Although its accuracy is slightly less than Logistic Regression, it also effectively captures patterns within the IMDB film reviews.

LDA followed after Random Forest as predicted given that it is flexible and adaptable to changes in dimensionality.

QDA had the lowest accuracy before PCA but demonstrated a substantial improvement after PCA and dimensionality reduction, surpassing KNN.

K-Nearest Neighbors was deemed to have the lowest accuracy after PCA due to facing challenges in high-dimensional spaces such as text data where Log Regression and Random Forest excelled.

In practical applications, the choice of such models depends on a plethora of factors including but not limited to interpretability, computational expense, and implementation.

In other words, Log Regression and Random Forest display that they are the most solid and effective choices regarding sentiment analysis given their robustness. LDA is also shown to be a consistent model such that it demonstrates stability in accuracy. Moreover, QDA and KNN benefit significantly after our PCA application on the dataset which indicates that these models may face challenges in high-dimensional spaces and thus, become more efficient as their performance enhances after dimensionality reduction.

All in all, our study illuminates the impact of dimensionality reduction through PCA and the strengths of different models applied to sentiment analysis. More so, it provides us with practical insights for those in the film industry who seek effective tools for understanding the audience and their expressed opinions in reviews.

4.2 Final Remarks

Based on our results, all models have surpassed the 50 % accuracy threshold. However, there are several considerations and avenues for future re-

search for which we can enhance our methodology and the application of sentiment analysis in the film industry.

For example, we could have explored additional feature engineering techniques to consider sentiment-related features or more word-embedding. Additionally, we can implement other techniques to address issues such as class imbalance to improve our models' ability to handle minority class sentiments. Furthermore, we can consider a dynamically updating lexicon rather than only a static sentiment lexicon that does not regularly update to adapt to changes in user expressions. Implementing this can enhance the model's adaptability to changing sentiments in user reviews. Lastly, we can consider fine-tuning hyper-parameters for each model to potentially improve their performance or reveal more insights.

Regarding future research, one could involve deep learning models to refine the accuracy of sentiment analysis and consider the dynamics of the sentiment lexicon. There are a plethora of instances of complex and contradictory statements that can alter positive and negative sentiments. Furthermore, one could also analyze trends or patterns in audience sentiments based on demographic information or how sentiments could evolve.

5 References

1. N. Lakshmipathi. IMDB Dataset for 50K Movie Reviews. *Kaggle, 2018*
   https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews/data
   Accessed 6-December-2023.

2. University of Pittsburg. Sentiment Lexicon. *MPQA, 2005*
   https://mpqa.cs.pitt.edu/
   Accessed 6-December-2023

3. UCLA: Statistical Consulting Group. Log Regression — Stata Data Analysis Examples. *OARC, 2006*
   https://stats.oarc.ucla.edu/stata/dae/logistic-regression/
   Accessed 6-December-2023

4. J. Brownlee. Linear Discriminant Analysis for Machine Learning. *Machine Learning Mastery, 2020*
   https://machinelearningmastery.com/linear-discriminant-analysis-for-machine-learning/
   Accessed 6-December-2023

5. T Srivastava. A Complete Guide to K-Nearest Neighbors. *Analytics Vidhya, 2023*
   https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
   Accessed 6-December-2023

6. E.R. Sruthi. Understand Random Forest Algorithms with Examples. *Analytics Vidhya, 2023*
   https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
   Accessed 6-December-2023

7. **Pandas:** McKinney, W. (2010). *Data Structures for Statistical Computing in Python.* Proceedings of the 9th Python in Science Conference, 51-56. https://pandas.pydata.org/

8. **Scikit-learn:** Pedregosa, F., Varoquaux, ... (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. https://scikit-learn.org/

9. **NumPy:** Harris, C. R., Millman, K. J., ... (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362. https://numpy.org/

10. **Matplotlib:** Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95. https://matplotlib.org/