# IMDB Dataset of 50K Movie Reviews Sentiment Prediction

Maika Francis (UID: 605973753), Maneh Begi (UID: 605575829), Aaron Tae (UID: 705591367), Yuki Yu (UID: 605533518), Kuan-Ting (Andrew) Chen (UID: 505506493), Jun Ryu (UID: 605574052)

## I. Abstract and Dataset Introduction

The Internet Movie Database (IMDB) offers extensive information on movies, TV shows, video games, and streaming content. It enables users to rate and review movies. The 'IMDB Dataset of 50K Movie Reviews,' intended for binary sentiment classification, includes only positive and negative ratings. Our goal is to examine the relationship between ratings and reviews and to predict sentiments using Logistic Regression, Logistic Regression with Stochastic Gradient Descent, K-Nearest Neighbors, and Random Forest models.

The dataset includes:

1. Review: Audience-submitted textual reviews on movies.
2. Sentiment Rating: The overall sentiment of the movie, classified as positive or negative.

## II. Descriptive Statistics

In our analysis, we utilize Google Colaboratory as the primary platform for loading and collaboratively analyzing the data in real-time. To get a better understanding of the data, some descriptive statistics are constructed.

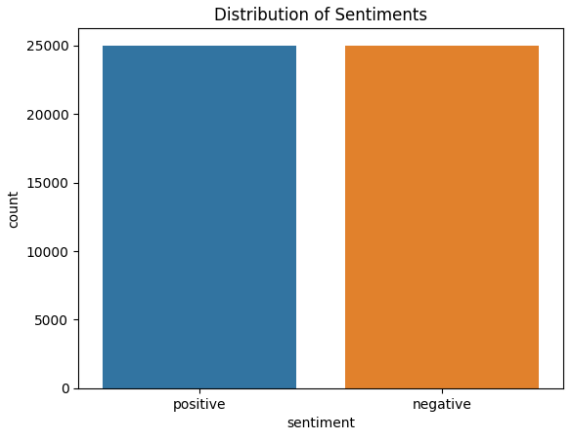


Fig. 1: Distribution of Sentiments

The dataset is balanced, featuring 50,000 reviews, with an equal split of 25,000 positive and 25,000 negative reviews.

## III. Preprocessing Step

Upon examining the initial reviews in the dataset, we note the presence of HTML tags, likely resulting from web-scraping, along with other elements requiring data cleaning.

| | review | sentiment |
|---|---|---|
| 0 | One of the other reviewers has mentioned that ... | positive |
| 1 | A wonderful little production. <br /><br />The... | positive |
| 2 | I thought this was a wonderful way to spend ti... | positive |
| 3 | Basically there's a family where a little boy ... | negative |
| 4 | Petter Mattei's "Love in the Time of Money" is... | positive |
| 5 | Probably my all-time favorite movie, a story o... | positive |
| 6 | I sure would like to see a resurrection of a u... | positive |
| 7 | This show was an amazing, fresh & innovative i... | negative |
| 8 | Encouraged by the positive comments about this... | negative |
| 9 | If you like original gut wrenching laughter yo... | positive |

Fig. 2: A subset of the dataset

In this dataset, we implement key data-cleaning steps for enhanced quality and consistency:

1. Duplicate Reviews: Removed, yielding 49,582 unique observations.
2. HTML Strips: Stripped from review text using BeautifulSoup.
3. Special Characters and Noisy Text: Cleared using Regular Expressions.
4. Stopwords and Tokenization: Stopwords removed and text tokenized.
5. Text Normalization: All review text is converted to lowercase for uniformity.

Then, our next objective is to visualize the frequency of words in relation to the sentiment expressed in reviews.

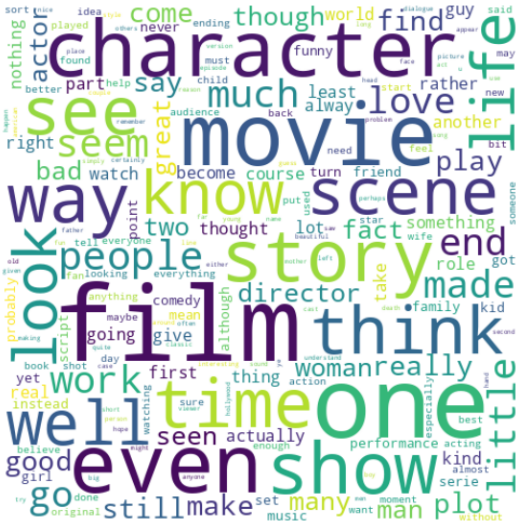


Fig. 3: Frequency of Words in Sentiment

As expected, the most frequently occurring words in the reviews are “movie,” “character,” and “film,” which generally do not provide significant insights for sentiment analysis. However, the dataset also includes a mix of some words potentially indicative of positive and negative sentiments, such as “well,” “good,” “great,” and “bad.”

## IV. Transforming the Corpus

After cleaning the data and removing noisy text, we move to reduce data dimensionality, essential in Natural Language Processing. Our goal is to condense words to their base forms, shortening sentences for easier application of sentiment analysis models. For instance, “smile” and “smiling” share a base but differ in encoding and tokenization.

To address the challenge, we apply either stemming or lemmatization to the text corpus to reduce words to their root forms. While lemmatization maintains word integrity, as seen with "episode," stemming occasionally alters words, such as transforming "episode" to "episod." Therefore, lemmatization appears more suitable for our needs. We use the Natural Language Toolkit (NLTK) for this process.

## V. Using different methods and classifiers to predict

Before fitting and testing the dataset with the four selected classifiers, we split the dataset into 80% for training (39,665 samples each for reviews and sentiments) and 20% for testing (9917 samples each for reviews and sentiments).

To determine which classifiers are most effective for predicting sentiments based on movie reviews, we fit the training data under three different methods: Bag-of-Words with and without considering the counts of words, and Term Frequency-Inverse

Document Frequency (TF-IDF). We apply these methods using the four classifiers as previously mentioned.

## A. Bag-of-Words (BOW) Method (considering the counts of words)

The Bag-of-Words (BOW) model is a widely utilized approach for extracting features from text for modeling purposes, including machine learning algorithms. This model represents text by noting the occurrence of words within a document and generally involves two components:

1. A defined vocabulary of known words.
2. A metric to quantify the presence of these known words.

The term "bag" in BOW signifies the disregard for the order or structure of words in the document. The model's primary focus is on the occurrence of known words, rather than their positions in the document.

To implement the BOW method, CountVectorizer (CV) is employed to vectorize the reviews training data. When configured with the setting `binary = True,` CV does not consider the frequency counts of words. In this configuration, the presence of a word is indicated by 1, and its absence by 0, regardless of the word's frequency of occurrence. Consequently, One-Hot Encoding is applied to normalize the given reviews training dataset, ensuring each word is appropriately represented for subsequent analysis.

**One-Hot Encoding**

One-Hot encoding is a method used to convert categorical data into binary values. In this process, each unique category of the variable is represented as a separate column in the dataset. For any given record, the column corresponding to the category that the record belongs to is marked with a 1, while all other category columns are marked with 0. This approach effectively represents categorical data in a binary, machine-readable format, facilitating its use in various statistical and machine-learning models.

### A.1 Logistic Regression

Logistic regression is a supervised machine learning algorithm commonly utilized for binary classification. It examines the relationship between one or multiple independent variables and a dependent binary variable. Although it is a simple and effective way to model binary data, challenges such as overfitting and limited generalizability to new data can arise. To address these issues, regularization techniques are employed to enhance the model's performance.

In our logistic regression model, the hyperparameter C is key for controlling regularization strength. Lower C values enhance regularization for simpler models, while higher values reduce it, allowing more complexity. We fine-tune C using Grid Search Cross Validation to find the optimal setting.

| C | Accuracy |
|---|---|
| 0.01 | 0.88111 |
| 0.02 | 0.88131 |
| 0.05 | 0.88424 |
| 0.25 | 0.88242 |
| 0.5 | 0.88010 |
| 0.75 | 0.87920 |
| 1 | 0.87859 |

We select C = 0.05 as this value yields the optimal estimator in our model.

Upon training the logistic regression model, we obtain the following results from the testing dataset:

Table 1: Logistic Regression Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| **False** | 0.89 | 0.88 | 0.88 | 4925 |
| **True** | 0.88 | 0.89 | 0.89 | 4992 |
| **Accuracy** | — | — | 0.88 | 9917 |
| **Macro Avg** | 0.88 | 0.88 | 0.88 | 9917 |
| **Weighted Avg** | 0.88 | 0.88 | 0.88 | 9917 |

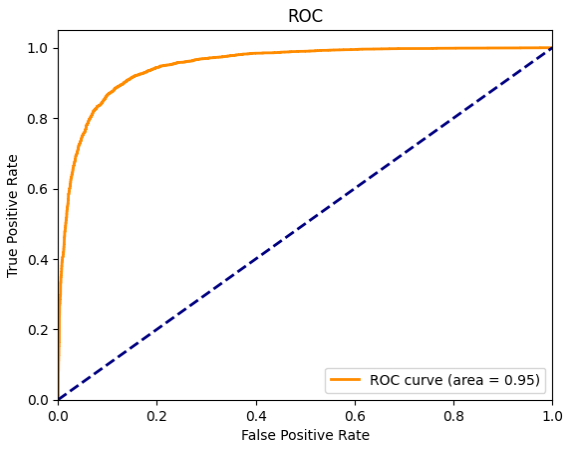


Fig. 4: ROC Curve of the Logistic Regression Model

The model shows strong performance in classifying movie reviews as Positive or Negative, with precision rates of 0.89 for Negative and 0.88 for Positive sentiments. This minor difference of 0.01% indicates balanced accuracy in identifying both sentiments, essential for impartial and reliable sentiment analysis. Additionally, the ROC curve's area of 0.95 highlights the model's effectiveness in differentiating between the two sentiment classes.

### A.2 Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is an optimization method that iteratively computes and updates the model parameters based on the gradient of the objective function, aiming to minimize this gradient. It is much more computationally efficient than the traditional gradient descent algorithm since it only computes the gradient for a single or a small batch of samples at each iteration. Upon training a new logistic regression model using SGD, we obtain the following results from the testing dataset:

Table 2: Logistic Regression with SGD Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| **False** | 0.88 | 0.88 | 0.88 | 4925 |
| **True** | 0.88 | 0.88 | 0.88 | 4992 |
| **Accuracy** | — | — | 0.88 | 9917 |
| **Macro Avg** | 0.88 | 0.88 | 0.88 | 9917 |
| **Weighted Avg** | 0.88 | 0.88 | 0.88 | 9917 |

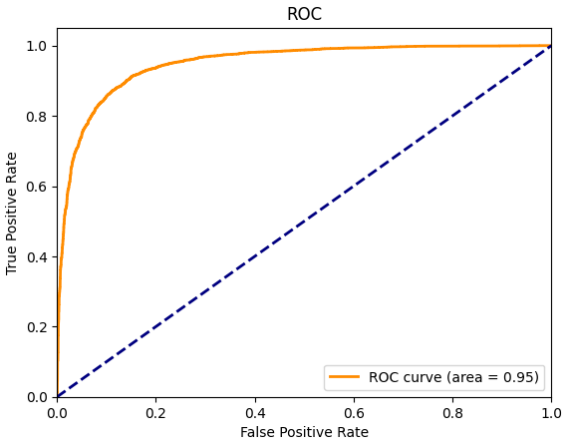


Fig. 5: ROC Curve of SGD Model

In our analysis, logistic regression with SGD as its optimization method demonstrates performance similar to traditional logistic regression in classifying movie review sentiments as Positive or Negative. This is reflected in a precision rate of 0.88 for both positive and negative sentiments, along with an ROC curve area of 0.95, indicative of a balanced classification between False and True outcomes. However, traditional logistic regression took approximately 0.162 minutes to run, while the implementation using SGD reduces the runtime to just 0.006 minutes, showcasing its efficiency.

### A.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a supervised learning algorithm that classifies based on proximity between test data and training points. Its non-parametric nature allows flexibility in modeling diverse data patterns. However, factors like dimensionality and computational efficiency influence its effectiveness with large datasets.

While working with the KNN algorithm, we must consider the bias-variance tradeoff. Increasing K (the number of 'voting' neighbor points) decreases model flexibility, leading to higher bias but potentially lower test error (variance). We use cross-validation to optimize K, finding that K = 100 provides the most accurate estimations for our model.

Table 3: KNN Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.81 | 0.57 | 0.67 | 4925 |
| True | 0.67 | 0.87 | 0.76 | 4992 |
| Accuracy | — | — | 0.72 | 9917 |
| Macro Avg | 0.74 | 0.72 | 0.71 | 9917 |
| Weighted Avg | 0.74 | 0.72 | 0.72 | 9917 |

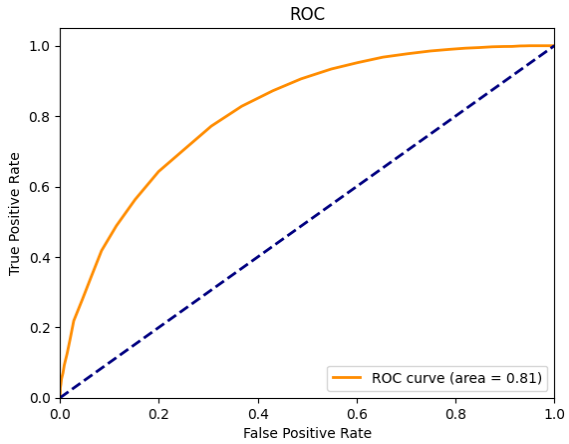


Fig. 6: ROC Curve of KNN Model

Compared to pure logistic regression and SGD logistic regression, KNN does not perform as effectively, as indicated by its lower precision rates, a larger difference between these rates, and a lower ROC curve area. This is believed to be due to KNN's less optimal performance with high-dimensional training datasets.

### A.4 Random Forest

Random Forest, widely used in machine learning for classification and regression, is an ensemble technique that aggregates the outcomes of multiple decision trees. Decision trees divide training data into smaller subsets until reaching a stopping criterion, like maximum depth, enabling predictions on new data. Random Forest enhances accuracy, mitigates overfitting, manages complex data, and minimizes outlier impact. After training our Random Forest model, we analyze results from the testing dataset.

Table 4: Random Forest Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.85 | 0.85 | 0.85 | 4925 |
| True | 0.85 | 0.85 | 0.85 | 4992 |
| Accuracy | — | — | 0.85 | 9917 |
| Macro Avg | 0.85 | 0.85 | 0.85 | 9917 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 9917 |

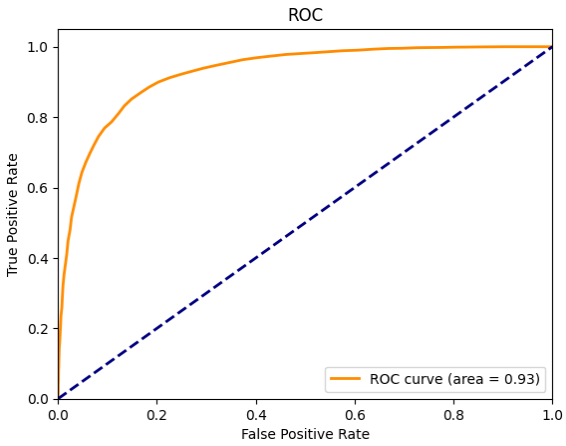


Fig. 7: ROC Curve of Random Forest Model

Random Forest effectively classifies movie review sentiments (Positive/Negative) with balanced outcomes of False and True. Its area under the ROC curve is comparable to that of both traditional and SGD logistic regression. However, it's noteworthy that Random Forest's precision rates and ROC area are slightly lower than the results achieved by both traditional and SGD logistic regression.

## B. Bag-of-Words (BOW) Method (with binary = False)

Setting the binary parameter equal to false, we will calculate the frequency of occurrence of the given reviews training dataset (i.e. its word counts).

### B.1 Logistic Regression

Similar to part A, we also tune the hyperparameter using Grid Search Cross Validation and obtain results for possible choices of C.

| **C** | **Accuracy** |
|---|---|
| 0.01 | 0.88242 |
| 0.02 | 0.88414 |
| 0.05 | 0.88626 |
| 0.25 | 0.88434 |
| 0.5 | 0.88232 |
| 0.75 | 0.88071 |
| 1 | 0.88041 |

We pick C = 0.05, as it produces the best estimator.

Table 5: Logistic Regression Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.89 | 0.88 | 0.88 | 4925 |
| True | 0.88 | 0.89 | 0.89 | 4992 |
| Accuracy | — | — | 0.89 | 9917 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 9917 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 9917 |

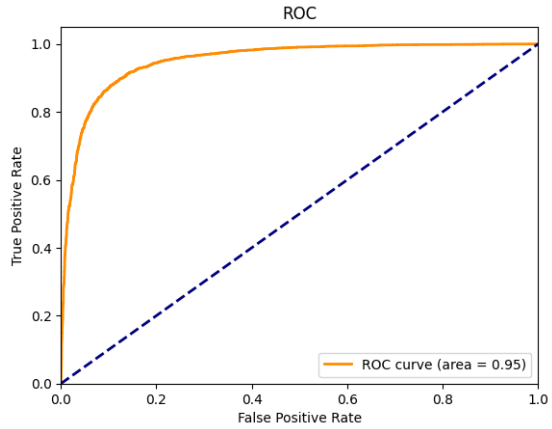


Fig. 8: ROC Curve of the Logistic Regression Model

With the BOW method with binary = FALSE, logistic regression performs similarly to BOW using counts. In both cases, there is only a 0.01% difference in the precision of TRUE's and FALSE's, and the precision rates and ROC area are the same.

## B.2 Logistic Regression with SGD

Table 6: SGD Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.87 | 0.88 | 0.87 | 4925 |
| True | 0.88 | 0.87 | 0.88 | 4992 |
| Accuracy | — | — | 0.87 | 9917 |
| Macro Avg | 0.87 | 0.87 | 0.87 | 9917 |
| Weighted Avg | 0.88 | 0.87 | 0.87 | 9917 |

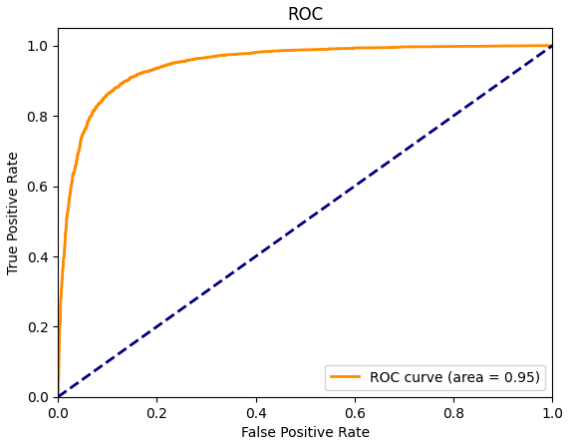


Fig. 9: ROC Curve of the Log Reg SGD Model

Logistic regression with an SGD implementation exhibits similar performance to the other BOW method. From the tables, we can see that precision, recall, F1 score, and support values are the same for both methods. The ROC area is also the same.

## B.3 K-Nearest Neighbors (KNN)

Similar to part A, we use cross-validation to optimize K, finding that K = 250 provides the most accurate estimations for our model.

Table 7: KNN Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.83 | 0.56 | 0.67 | 4925 |
| True | 0.67 | 0.88 | 0.76 | 4992 |
| Accuracy | — | — | 0.72 | 9917 |
| Macro Avg | 0.75 | 0.72 | 0.72 | 9917 |
| Weighted Avg | 0.75 | 0.72 | 0.72 | 9917 |

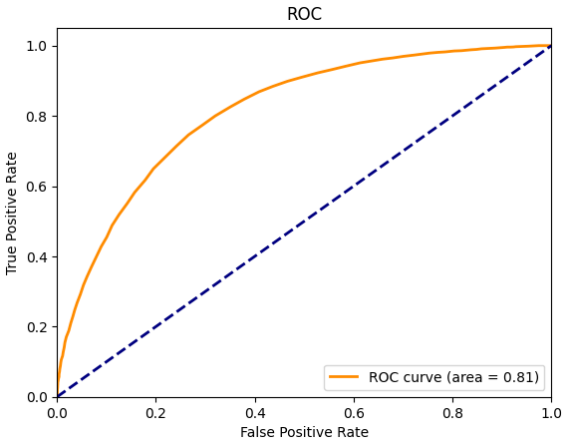


Fig. 10: ROC Curve of the KNN Model

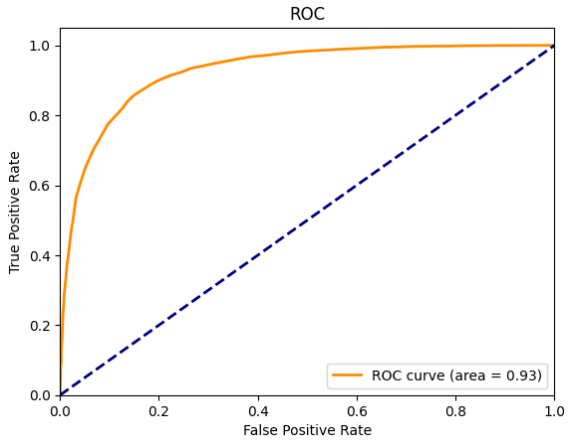


Fig. 11: ROC Curve of the Random Forest Model

Compared to the other BOW method, this method also gives similar results. The accuracy for true values is the same, whereas the accuracy for false values is slightly higher at 0.83, compared to 0.81. The ROC area remains the same.

### B.4 Random Forest

Table 8: Random Forest Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.85 | 0.85 | 0.85 | 4925 |
| True | 0.85 | 0.85 | 0.85 | 4992 |
| Accuracy | — | — | 0.85 | 9917 |
| Macro Avg | 0.85 | 0.85 | 0.85 | 9917 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 9917 |

Similar to the previous models we have discussed so far, Random Forest follows the trend of results being similar for both binary = TRUE and binary = FALSE. In this case, the results are the same.

Across both methods using BOW, our results using all 4 algorithms were very similar. In the case of Random Forest, logistic regression with SGD, and pure logistic regression, the results were exactly the same across both methods.

## C. Term Frequency and Inverse Document Frequency (TF-IDF)

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical metric used in NLP to assess word importance in documents. It considers two factors:

1. TF (Term Frequency): This measures word frequency within a document.
2. IDF (Inverse Document Frequency): This gauges word significance across a document corpus.

The TF-IDF score for a word is the product of its TF and IDF values. Higher scores

indicate greater word importance in a document relative to the corpus.

TF-IDF = TF * IDF
TF-IDF = TF * log(N/DF)

Where:

- TF is the term frequency of a word in a document
- N is the total number of documents in the corpus
- DF is the document frequency of a word in the corpus (i.e., the number of documents that contain the word)

The best threshold values for each method is computed individually using RandomSearchCV. For the first logistic regression model, we also use RandomSearchCV to find the optimal C value.

### C.1 Logistic Regression

Table 9: Logistic Regression Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.90 | 0.88 | 0.89 | 4925 |
| True | 0.88 | 0.90 | 0.89 | 4992 |
| Accuracy | — | — | 0.89 | 9917 |
| Macro Avg | 0.89 | 0.89 | 0.89 | 9917 |
| Weighted Avg | 0.89 | 0.89 | 0.89 | 9917 |

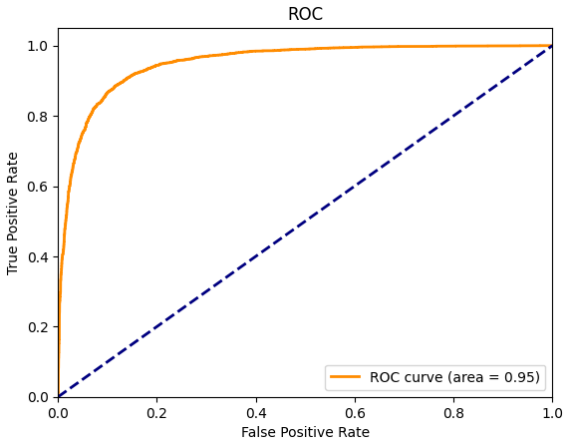


Fig. 12: ROC Curve of the Logistic Regression Model

We use a threshold value of around 0.29 and C value of around 0.83.

Compared to the logistic regression results from the previous BOW methods, this TF-IDF variant of logistic regression displays the highest performance by far because its precision rate for negative sentiments is 0.90 even though the rest of the results, including the ROC curve are the same as the other logistic regression models. performance and accuracy.

### C.2 Logistic Regression with SGD

Table 10: Log Reg SGD Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.89 | 0.86 | 0.88 | 4925 |
| True | 0.87 | 0.90 | 0.88 | 4992 |
| Accuracy | — | — | 0.88 | 9917 |
| Macro Avg | 0.88 | 0.88 | 0.88 | 9917 |
| Weighted Avg | 0.88 | 0.88 | 0.88 | 9917 |

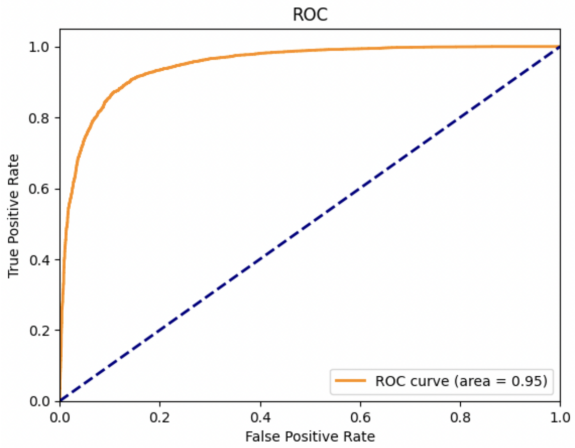


Fig. 13: ROC Curve of the Log Reg SGD Model

We use a threshold value of around 0.44.

Logistic regression with SGD and the TF-IDF method also performs well but is not as good as pure logistic regression in terms of precision rates.

### C.3 K-Nearest Neighbors (KNN)

Despite fitting our testing data under the TF-IDF method into the KNN algorithm for over an hour, it failed to produce any result. We therefore conclude that the KNN algorithm is not an effective method for this scenario due to the high-dimensional spaces. As dimensionality increases, it becomes more computationally intensive to calculate the distance and find the nearest neighbors in the data space, which is a phenomenon known as "the curse of dimensionality". Therefore, we do not recommend using the KNN algorithm in this case.

### C.4 Random Forest

Table 11: Random Forest Test Results

| | Precision | Recall | F1 Score | Support |
|---|---|---|---|---|
| False | 0.84 | 0.85 | 0.85 | 4925 |
| True | 0.85 | 0.84 | 0.85 | 4992 |
| Accuracy | — | — | 0.85 | 9917 |
| Macro Avg | 0.85 | 0.85 | 0.85 | 9917 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 9917 |

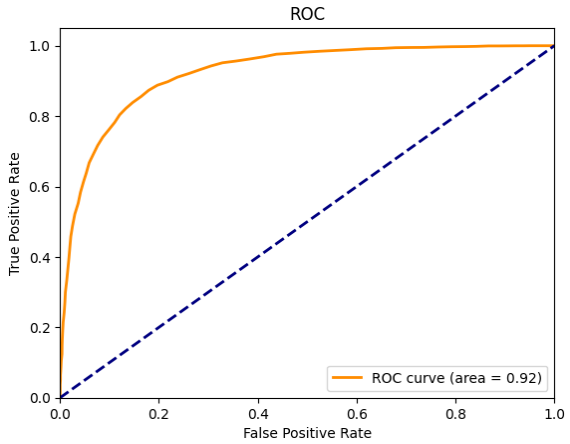


Fig. 14: ROC Curve of the Random Forest Model

We use a threshold value of around 0.44.

The random forest model performs well, but it is not as good as the logistic regression models. It is also slightly worse performing than the random forest models from both of the BOW methods based on the lower precision rates and smaller ROC area.

## VI. Conclusion and Limitation

### 6.1 Conclusion

In summary, employing pure logistic regression with the implementation of the TF-IDF method exhibits superior performance, evidenced by both the highest accuracy percentage and the largest area under the ROC Curve. Generally, TF-IDF proves more effective than the Bag-of-Words method, as the latter sacrifices semantic meaning. For instance, the phrase "not bad" conveys a positive sentiment, but when analyzed individually, the words "not" and "bad" may suggest negativity. TF-IDF addresses this by assigning a weighted score to each word, normalizing its count in a specific document by its frequency across all documents. This approach highlights words unique to a particular document, accentuating rare words while diminishing the significance of common words prevalent in many documents.

As of now, none of the classifiers has surpassed a 90% accuracy threshold. To potentially enhance our performance, consideration could be given to incorporating the Opinion Lexicon approach.

### 6.2 Limitation

In selecting classifiers for fitting and testing the training dataset, the preference was given to random forest over decision trees, despite its higher computational cost. This decision was driven by random forest's ability to yield more accurate predictions. Although decision trees are advantageous for large datasets due to its simplicity, it tends to produce less accurate results. On the other hand, Random forest stands out for its effectiveness in preventing overfitting and its enhanced accuracy, particularly in high-dimensional spaces.

Among the four classifiers, KNN exhibits the weakest performance across all three methods, primarily attributed to its limitations in handling high-dimensional datasets. Traditional logistic regression and logistic regression with SGD demonstrate comparable performance and accuracy. Meanwhile, Random Forest displays a commendable performance, albeit slightly lower than the other models.

## References


Bhoi, Niraj. “Stemming vs Lemmatization in NLP.” *Medium*, Medium, 14 Dec. 2022, nirajbhoi.medium.com/stemming-vs-lemmatization-in-nlp-efc280d4e845.

Brownlee, Jason. “A Gentle Introduction to the Bag-of-Words Model.” *MachineLearningMastery.Com*, 7 Aug. 2019, machinelearningmastery.com/gentle-introduction-bag-words-model/.

Jain, Pratyaksh. “Basics of Countvectorizer.” *Medium*, Towards Data Science, 1 June 2021, towardsdatascience.com/basics-of-countvectorizer-e26677900f9c.

Pradeep. “Understanding TF-IDF in NLP: A Comprehensive Guide.” *Medium*, Medium, 21 Mar. 2023, medium.com/@er.iit.pradeep09/understanding-tf-idf-in-nlp-a-comprehensive-guide-26707db0cec5.

Rithp. “Logistic Regression and Regularization: Avoiding Overfitting and Improving Generalization.” *Medium*, Medium, 5 Jan. 2023, medium.com/@rithpansanga/logistic-regression-and-regularization-avoiding-overfitting-and-improving-generalization-e9afdcddd09d.