# Sentiment Analysis: A Machine Learning Approach to Text Classification

**Alexandros Daskalakis**
Department of Mathematics
Applied Mathematics
University of Crete
math1p0011@math.uoc.gr

**Abstract**

Sentiment analysis, a key task in natural language processing (NLP), involves classifying textual data according to the sentiment it conveys, such as positive, negative, or neutral. This work investigates a machine learning approach to text classification for sentiment detection, focusing on the application of various supervised learning algorithms. We explore models including Logistic Regression (LR), Naive Bayes Classifier (NBC), leveraging labeled datasets to train and evaluate the models' performance. Comprehensive evaluation is conducted using standard metrics such as accuracy and F1-score, to determine the best-performing model. The results underscore the potential of machine learning techniques for automating sentiment classification, with implications for fields such as social media analysis, consumer feedback, and market research.

## 1 Introduction to Sentiment Analysis

### 1.1 What is Sentiment Analysis?

Sentiment analysis is a subfield of natural language processing (NLP) that focuses on identifying and classifying the sentiment expressed in text. The goal is to determine whether the sentiment conveyed in a piece of text is positive, negative, or neutral.

### 1.2 Applications of Sentiment Analysis

Sentiment analysis has various real-world applications across different domains:

- **Stock Market**: Analyzing news articles and social media posts to gauge market sentiment and make predictions about stock price movements.
- **Customer Feedback**: Processing product reviews and survey responses to understand customer satisfaction and identify areas for improvement.
- **Social Media Monitoring**: Tracking public opinion on platforms like Twitter and Facebook to monitor trends and brand perception.
- **Political Sentiment**: Analyzing public opinion regarding political candidates, events, or policies through social media and news sources.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

# 2 Dataset Description

## 2.1 Source and Characteristics of the Dataset

The dataset used in this study consists of product reviews collected from the e-commerce platform Amazon.com. It contains over one million samples spanning multiple product categories. Each sample includes a numerical star rating (ranging from 1 to 5) and a corresponding textual review provided by the buyer. For sentiment classification purposes, the star ratings are mapped to three sentiment classes: Positive, Negative, and Neutral. Specifically, reviews with ratings greater than 3 stars are classified as Positive, those with ratings less than 3 are classified as Negative, and ratings of 3 are categorized as Neutral. This transformation is based on the assumption that higher star ratings reflect a positive sentiment toward the product, while lower ratings indicate dissatisfaction.

## 2.2 Preprocessing

The preprocessing steps involve transforming the raw text into a format suitable for use by the classification models. This process includes several stages, such as **Tokenization**, where the text is split into individual words or tokens. Next, **stop-word removal** is performed to eliminate common words (e.g., "and", "the", "is") that do not contribute meaningful information for sentiment classification. Finally, the text is represented numerically using **TF-IDF (Term Frequency-Inverse Document Frequency)**, a technique that captures the importance of each word within the document relative to the entire corpus, converting the text into a numerical feature vector that can be fed into machine learning models.

# 3 Text Representation: Converting Text to Numbers

## 3.1 TF-IDF (Term Frequency-Inverse Document Frequency)

To convert individual words into numerical representations, we employ the TF-IDF technique, which involves calculating two key frequencies: Term Frequency (TF) and Inverse Document Frequency (IDF).

- **Term Frequency (TF)**: This measures how often a word appears in a specific document. It is given by the formula:

$$\text{TF}(t, d) = \frac{\text{Frequency of term } t \text{ in document } d}{\text{Total number of terms in document } d}$$

- **Inverse Document Frequency (IDF)**: This measures how important a word is across the entire data set. Words that appear frequently in many documents are less informative. The IDF of a term is calculated as:

$$\text{IDF}(t) = \log\left(\frac{N}{1 + \text{DF}(t)}\right)$$

Where:

- **N** is the total number of reviews in the data set.
- **DF(t)** is the number of documents containing the term $t$.

Thus **TF-IDF** giving us the final score for a term $t$ in a document $d$ is :

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t).$$

This scoring method assigns higher weights to terms that are frequent in a specific document but rare across the entire corpus, ensuring that the most informative terms are given more significance in the model.

In the following figures, we can see a full example of the transformation. We start with our original data set :

| | text |
|---|---|
| 0 | We love this magazine! |
| 1 | I love this product! |
| 2 | I hate this product! |
| 3 | This is very bad! |

Figure 1: Original Sample.

| Text_ID | Processed Text |
|---|---|
| 0 | We love magazine |
| 1 | I love product |
| 2 | I hate product |
| 3 | This bad |

Figure 2: Processed Documents.

| Text_ID | bad | hate | love | magazine | product | this | we |
|---|---|---|---|---|---|---|---|
| 0 | 0.000000 | 0.000000 | 0.486934 | 0.617614 | 0.000000 | 0.000000 | 0.617614 |
| 1 | 0.000000 | 0.000000 | 0.707107 | 0.000000 | 0.707107 | 0.000000 | 0.000000 |
| 2 | 0.000000 | 0.785288 | 0.000000 | 0.000000 | 0.619130 | 0.000000 | 0.000000 |
| 3 | 0.707107 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.707107 | 0.000000 |

Figure 3: Final TF-IDF Values.

# 4 Model Selection and Evaluation

## 4.1 Model Selection

In sentiment analysis with TF-IDF, multi-class classification is used to categorize text into multiple sentiment categories, such as positive, neutral, or negative. TF-IDF helps capture the importance of words by weighing terms based on their frequency across documents, making it an ideal representation for text data. Naive Bayes Classifier (NBC) is a strong choice for this task due to its simplicity and effectiveness in high-dimensional sparse data, which is common in text-based features like TF-IDF. Logistic regression, being a powerful linear model, also works well in multi-class settings, efficiently modeling relationships between features and sentiment labels. Both models are computationally efficient, easy to implement, and perform well in sentiment analysis tasks, especially when distinguishing between multiple classes like positive, neutral, and negative sentiments.

## 4.2 Training the Model

We performed 5-fold cross-validation to evaluate and tune the hyper-parameters of our models, aiming to identify the best-performing classifier based on accuracy. For the NBC, we tested different values of the alpha parameter(Laplace smoothing) (1, 2, and 3), while LR we explored various values of the regularization parameter C (1, 2, and 3). This process allowed us to optimize the models and select the one that achieved the highest accuracy in predicting sentiment labels, ensuring the best performance for our sentiment analysis task.

## 4.3 Evaluation Metrics

In this subsection, we evaluate the performance of our models using F1-Score as the key metric. Our dataset is imbalanced, meaning that the distribution of instances across the sentiment classes (positive, neutral, and negative) is not uniform. In such cases, accuracy can be misleading, as a model could predict the majority class well while performing poorly on the minority class. F1-Score is preferred because it balances precision and recall, providing a more comprehensive evaluation of the model's performance, especially for underrepresented classes. This is particularly important in multi-class sentiment analysis tasks, where the goal is to classify text into one of three categories: positive, neutral, or negative. By using F1-Score, we ensure that the model performs effectively across all classes, mitigating the impact of the class imbalance and avoiding bias toward the majority class.

In total, we trained **6 different models**(3 configurations of Naive Bayes and 3 configurations of Logistic Regression), and each model was evaluated using 5-fold cross-validation, meaning that each unique model was trained and tested 5 times, once for each fold of the cross-validation process. This ensures that we get a more robust estimate of each model's performance. The F1-Score values reported in the table reflect the F1-Score across these 5 folds for each model and hyper-parameter configuration.

After performing 5-fold cross-validation and tuning the hyper-parameters, we found that Logistic Regression with a regularization parameter **C=2** achieved the best performance. Below is a table summarizing the average F1-Score for each model and hyper-parameter configuration:

Table 1: Model accuracy with Hyper-parameter Tuning

| Part | | |
|---|---|---|
| Model | Hyper-parameter | **F1-Score** |
| NBC | $a = 0.05$ | 0.74 |
| NBC | $a = 0.5$ | 0.67 |
| NBC | $a = 0.025$ | 0.73 |
| LR | $C = 1$ | 0.77 |
| LR | $C = 1.5$ | 0.78 |
| LR | $C = 2$ | 0.79 |

# 5 Conclusion

In conclusion, this project explored the application of machine learning techniques to sentiment analysis, specifically for classifying text into positive, neutral, or negative categories. Various models, including Logistic Regression, Naive Bayes Classifier, were evaluated using 5-fold cross-validation and performance metrics such as accuracy and F1-score. After experimenting with multiple models and hyper parameter tuning, it was found that Logistic Regression with a ridge penalty (C=2, where C=1/$\lambda$) provided the best performance. This model yielded the highest F1-Score and demonstrated strong classification potential. Training and testing one last the best model on the full data set , we manage to achieve **accuracy 81%**.

Additionally, the total corpus contained around 12,000 unique words. However, limiting the feature set to only the 5,000 most frequently used words did not significantly decrease performance. This suggests that, in practice, reducing the feature size can help save computation time without a notable sacrifice in model accuracy. By focusing on the most important features, we can streamline the model and make it more efficient, especially when dealing with large-scale datasets.

However, several limitations were observed during the implementation. One significant drawback of using the TF-IDF method for text representation is that it does not capture the order or sequence of words in a sentence. As a result, TF-IDF fails to understand context-specific elements like sarcasm or irony, which are often conveyed through word order or juxtaposition.

Furthermore, while the models performed well on the dataset, they may struggle when applied to different or more complex datasets, particularly if those datasets contain more diverse or ambiguous language. The absence of word order consideration and the reliance on statistical associations may result in reduced effectiveness when analyzing subtle sentiment cues or complex phrases.

Despite these limitations, the best-performing Logistic Regression model provides a strong baseline for sentiment classification tasks. Future work could explore more advanced techniques, such as deep learning models that capture word order and context, to further improve performance and address the challenges associated with sentiment analysis in varied linguistic contexts.

# References


Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In Advances in Neural Information Processing Systems (pp. 5998–6008).

Das, B., Chakraborty, S. (2018). An improved text sentiment classification model using TF-IDF and next word negation.

Jeni, L. A., Cohn, J. F., De La Torre, F. (2013). Facing imbalanced data: Recommendations for the use of performance metrics. Proceedings of the 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (ACII), 245–251.

Amazon Reviews 2023 Dataset. Retrieved from https://amazon-reviews-2023.github.io/ (Accessed: Jan 31, 2025).