Sentiment Analysis using Logistic Regression and TF-IDF

Submitted for

Natural Language Processing CBCA275

Submitted by:
(E23BCAU0196) RITIKA SIROHI
(E23BCAU0059) TANISHA CHAUHAN

Submitted to
DR. SHAKSHI SHARMA

Jan-May 2025

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

INDEX

Abstract

This project aims to perform sentiment analysis using machine learning techniques on the "Sentiment Labelled Sentences Data Set" from Kaggle. The dataset consists of sentences labelled as positive or negative from sources like IMDb, Amazon, and Yelp. Our goal is to train a classifier that can predict the sentiment of new, unseen sentences accurately. Techniques like TF-IDF vectorization and Logistic Regression are employed for text processing and classification.

Introduction

Sentiment analysis, also known as opinion mining, refers to the use of natural language processing to determine whether a piece of text is positive, negative, or neutral. It is a crucial component in applications such as product reviews, social media monitoring, and customer feedback analysis. This project focuses on binary sentiment classification using a labeled dataset of sentences from real-world platforms.

Related Work

Various machine learning models have been applied to sentiment analysis tasks, ranging from traditional approaches like Naive Bayes, Logistic Regression, and SVMs to deep learning techniques like RNNs and transformers. Our approach is inspired by the simplicity and effectiveness of using TF-IDF in combination with a Logistic Regression classifier, as demonstrated in earlier studies on sentiment classification tasks.

Methodology

4.1 Dataset Loading

- We used three datasets: amazon_cells_labelled.txt, imdb_labelled.txt, yelp_labelled.txt.
- Each contains 1000 sentences labeled as either positive (1) or negative (0).

4.2 Preprocessing

- Lowercasing, removal of special characters/numbers, tokenization, stopword removal, lemmatization.

4.3 Feature Extraction

- TF-IDF Vectorizer with unigrams and bigrams.

4.4 Model Building

- Logistic Regression trained using 80-20 split.

4.5 Prediction Function

- predict_sentiment() to classify new input.

4.6 Feature Visualization

- Top positive and negative words visualized using bar charts.

Hardware/Software Required

Hardware:

- Minimum 4GB RAM
- Any processor capable of running Python

Software:

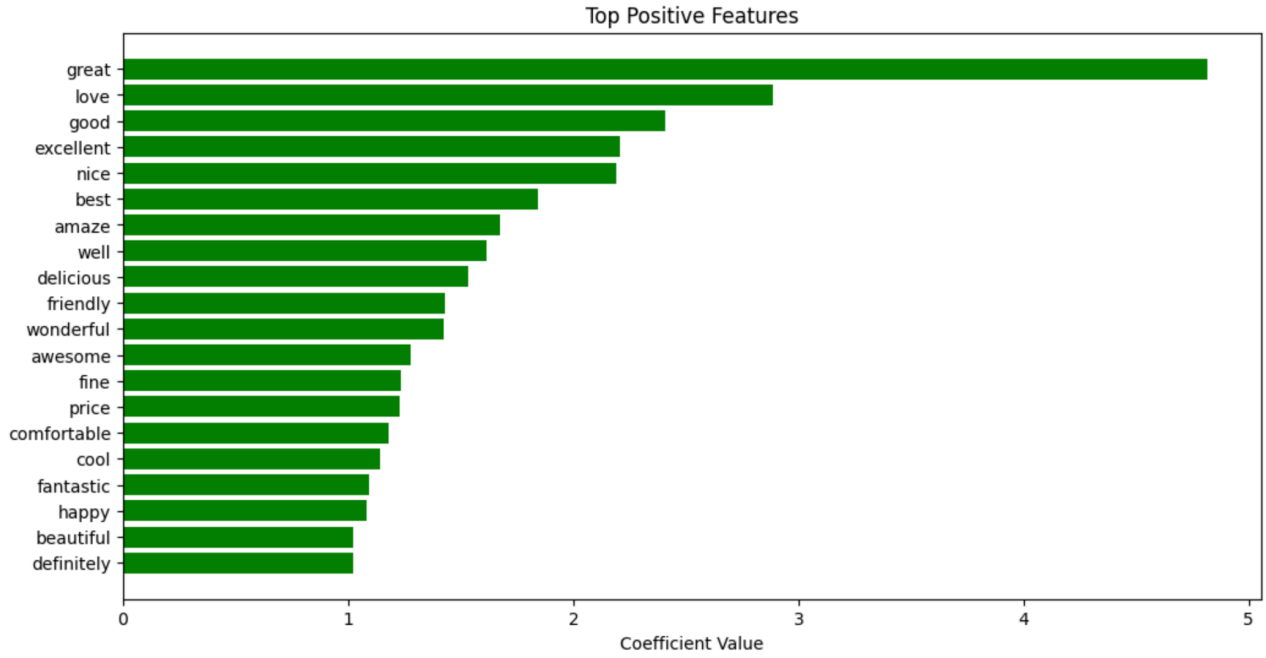
- Python 3.x
- Jupyter Notebook / Google Colab
- Libraries: pandas, scikit-learn, nltk, matplotlib

Experimental Results

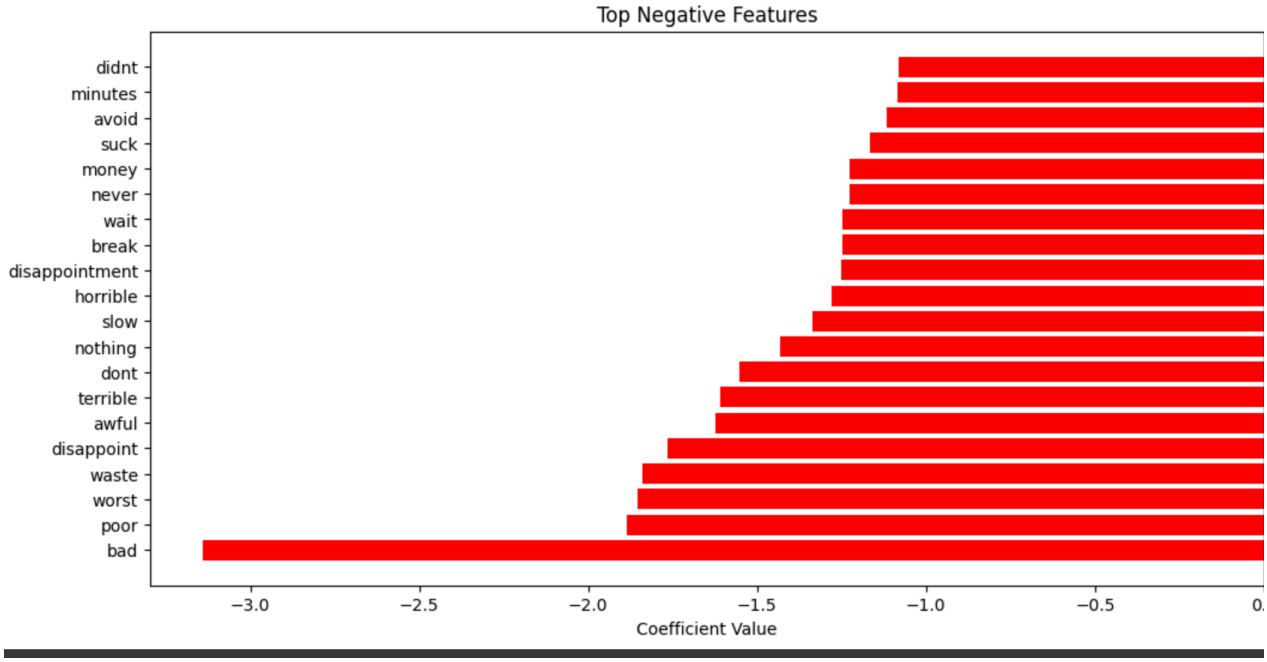
- Accuracy: 81.6%
- Precision, Recall, F1-score: Derived using classification report

Class	Precision	Recall	F1-Score	Support
Negative (0)	0.84	0.80	0.82	291
Positive (1)	0.79	0.83	0.81	259
Accuracy			0.82	550
Macro Avg	0.82	0.82	0.82	550
Weighted Avg	0.82	0.82	0.82	550

- Top Features:
  - Positive: great, love, good



- Negative: bad, poor, worst



Sample Predictions:
- "It was amazing and I loved it!" → Positive
- "This was the worst experience ever." → Negative

Conclusions

The Logistic Regression classifier achieved satisfactory performance for binary sentiment classification using simple text preprocessing and TF-IDF vectorization. The results demonstrate that even basic models can effectively analyze sentiment from text when paired with appropriate preprocessing techniques.

Future Scope

- Integrate deep learning models (e.g., LSTM, BERT)
- Expand the dataset
- Deploy via web/mobile app
- Multi-class sentiment classification

GitHub Link of Complete Project

https://github.com/ritikasirohi/sentiment-analysis-app