IMDB Movie Reviews Sentiment Analysis

By: Sofia Rueda
Data 4391

Project Overview

In this project, I used Natural Language Processing (NLP) to classify IMDb movie reviews as positive or negative.

Goal: Predict whether a movie review is **positive** or **negative**

Dataset: IMDB Movie Reviews (50,000 labeled reviews)

Three text-feature methods used:

- **TF-IDF Vectorizer**
- **Word2Vec Embeddings**
- **GloVe Embeddings**

Two ML models used:

- **Logistic Regression**
- **Linear SVC**

Also evaluated **balanced (50/50)** and **imbalanced (60/40)** class scenarios

Final step: Generated predictions for new, unseen reviews

///////

What is NLP?

Natural Language Processing (NLP) is a field of Artificial Intelligence that focuses on enabling computers to understand, interpret, and generate human language. It combines **linguistics**, **machine learning**, and **computer science** to process and analyze text the way humans do.

Why NLP Matters?

NLP allows computers to make sense of the huge amount of human language data produced every day.

Real-world applications include:

- **Sentiment Analysis** (reviews, customer feedback)
- **Chatbots & Virtual Assistants** (Siri, Alexa, customer support bots)
- **Spam Detection** (email filtering)
- **Translation** (Google Translate)
- **Summarization** (condensing long articles)

Data Preprocessing

To prepare raw movie reviews for NLP modeling, I applied standard text-cleaning steps:

- **Lowercasing** all text to keep words consistent
- **Removing punctuation & special characters**
- **Removing stopwords** (common words like *the, is, and*)
- **Tokenization** – splitting sentences into individual words
- **Lemmatization** – reducing words to their base form (*running* → *run*)

These preprocessing steps **reduce noise**, **standardize the text**, and help the models learn clearer patterns, leading to better overall accuracy.

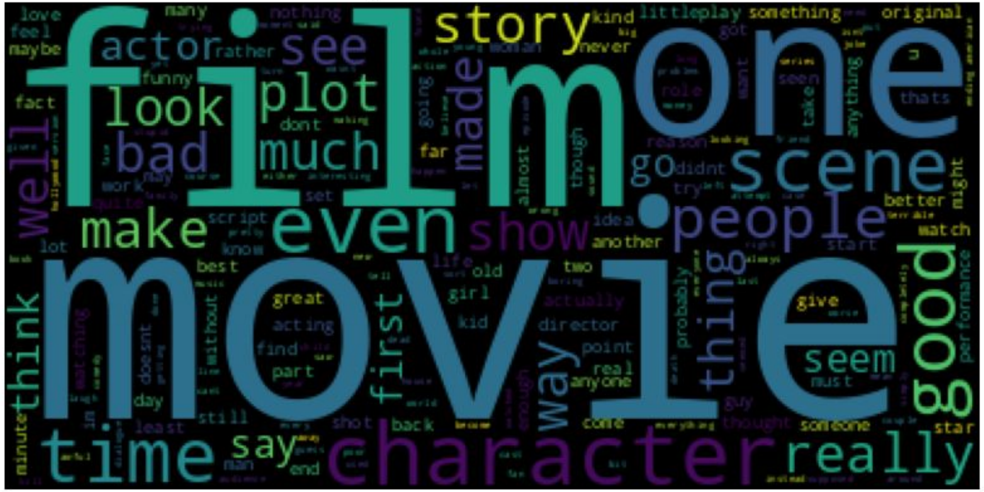
///////

Word Cloud

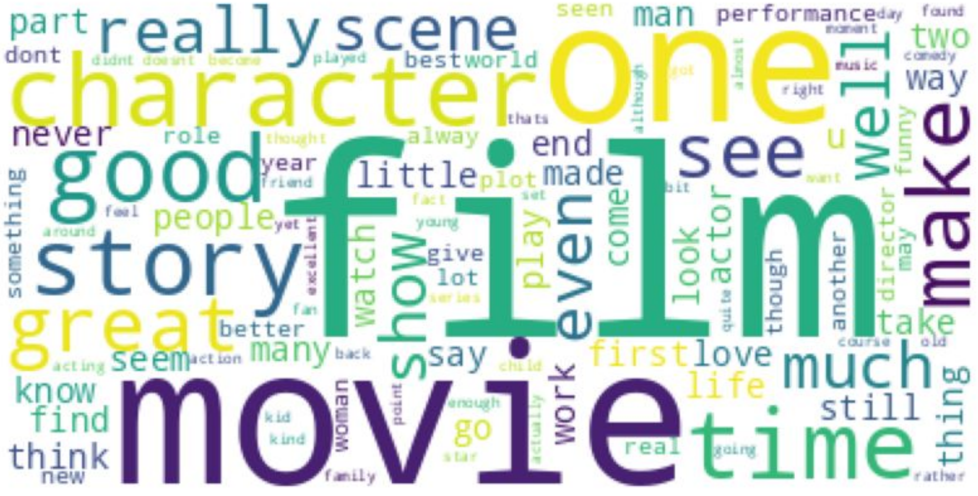
What this visual shows:

- Most common words found in the IMDB reviews
- Larger words = appear more frequently
- Helps identify dominant themes from user opinions
- Based on cleaned reviews after preprocessing

Negative Reviews WordCloud



Positive Reviews WordCloud



TF-IDF

01

- Converts text into high-dimensional sparse vectors
- Captures how important each word is across all reviews
- Works well for linear models
- First baseline model for the experiments

TF-IDF Results (Balanced Classes)

Logistic Regression

- **Macro F1:** 0.89
- Performs well on both classes:
  - Negative: Precision 0.90, Recall 0.87
  - Positive: Precision 0.88, Recall 0.90

Prediction Example
“The pacing of the movie was slow, and it didn't sustain my interest.”
→ **Negative 😞 (Confidence: 0.92)**

Linear SVC

- **Macro F1:** 0.88
- Very similar performance to LR:
  a. Negative: Precision 0.89, Recall 0.87
  b. Positive: Precision 0.88, Recall 0.89

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.80)**

/////// TF-IDF Results (Imbalanced Classes)

Logistic Regression

- **Macro F1:** 0.88
- Performs well on both classes:
  - Negative: Precision 0.89, Recall 0.92
  - Positive: Precision 0.87, Recall 0.83

Prediction Example
"The pacing of the movie was slow, and it didn't sustain my interest."
→ **Negative 😞 (Confidence: 0.93)**

Linear SVC

- **Macro F1:** 0.87
- Very similar performance to LR:
  a. Negative: Precision 0.89, Recall 0.91
  b. Positive: Precision 0.86, Recall 0.83

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.84)**

TF-IDF

01

Balanced Classes

The best TF-IDF model with balanced data was **Logistic Regression**, achieving **~0.89 macro F1** with strong, stable performance across both positive and negative reviews.

Imbalanced Classes

Under imbalanced TF-IDF training, **Logistic Regression** again performed best with a **macro F1 ~0.87**, but its recall dropped for the minority class.

Word2Vec

02

- Creates dense 100-dimensional vectors
- Words with similar meaning have similar vectors
- We averaged vectors to create one embedding per review
- Captures semantic relationships better than TF-IDF

/////// Word2Vec Results (Balanced Classes)

Logistic Regression

- **Macro F1:** 0.86
- Performs well on both classes:
  - Negative: Precision 0.87, Recall 0.86
  - Positive: Precision 0.86, Recall 0.87

Prediction Example
"The pacing of the movie was slow, and it didn't sustain my interest."
→ **Negative 😞 (Confidence: 1.00)**

Linear SVC

- **Macro F1:** 0.86
- Very similar performance to LR:
  a. Negative: Precision 0.87, Recall 0.86
  b. Positive: Precision 0.86, Recall 0.87

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.95)**

///////

Word2Vec Results (Imbalanced Classes)

Logistic Regression

- **Macro F1:** 0.85
- Performs well on both classes:
  - Negative: Precision 0.93, Recall 0.81
  - Positive: Precision 0.76, Recall 0.91

Prediction Example
"The pacing of the movie was slow, and it didn't sustain my interest."
→ **Negative 😞 (Confidence: 0.99)**

Linear SVC

- **Macro F1:** 0.84
- Very similar performance to LR:
  a. Negative: Precision 0.93, Recall 0.80
  b. Positive: Precision 0.76, Recall 0.91

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.82)**

Word2Vec

02

Balanced Classes

For Word2Vec embeddings with balanced classes, **Logistic Regression and Linear SVC tied**, both reaching **~0.86 macro F1**. Word2Vec captured semantic structure better than GloVe, leading to stronger embeddings and more stable predictions.

Imbalanced Classes

With imbalanced Word2Vec data, the best model was **Logistic Regression**, achieving **~0.85 macro F1**. While accuracy remained high, the model favored the majority class—showing again how imbalance affects semantic models.

GloVe

03

- Pretrained on billions of tokens
- Captures semantic meaning + global word co-occurrence
- Averaged embedding = 100-dimensional vector per review
- Often performs better than Word2Vec for sentiment tasks

//////

GloVe Results (Balanced Classes)

Logistic Regression

- **Macro F1:** 0.79
- Performance:
  - Negative: Precision 0.79, Recall 0.79
  - Positive: Precision 0.79, Recall 0.79

Prediction Example
"The pacing of the movie was slow, and it didn't sustain my interest."
→ **Negative 😞 (Confidence: 0.88)**

Linear SVC

- **Macro F1:** 0.80
- Very similar performance to LR:
  a. Negative: Precision 0.80, Recall 0.79
  b. Positive: Precision 0.79, Recall 0.80

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.66)**

GloVe Results (Imbalanced Classes)

Logistic Regression

- **Macro F1:** 0.77
- Performance:
  - Negative: Precision 0.89, Recall 0.71
  - Positive: Precision 0.66, Recall 0.86

Prediction Example
"The pacing of the movie was slow, and it didn't sustain my interest."
→ **Negative 😞 (Confidence: 0.81)**

Linear SVC

- **Macro F1:** 0.77
- Very similar performance to LR:
  a. Negative: Precision 0.88, Recall 0.71
  b. Positive: Precision 0.67, Recall 0.86

Prediction Example
Same review
→ **Negative 😞 (Confidence: 0.60)**

GloVe

03

Balanced Classes

The strongest GloVe model under balanced training was **Linear SVC**, reaching **~0.80 macro F1**.

Imbalanced Classes

For imbalanced GloVe data, both models performed similarly, but **Linear SVC was slightly better**, with **~0.78 macro F1**.

Feature Type	Class Balance	Model	Macro F1-score	Confidence
TF-IDF	Balanced	Logistic Regression	0.89	0.92
TF-IDF	Balanced	Linear SVC	0.88	0.80
TF-IDF	Imbalanced	Logistic Regression	0.88	0.93
TF-IDF	Imbalanced	Linear SVC	0.87	0.84
Word2Vec	Balanced	Logistic Regression	0.86	1.00
Word2Vec	Balanced	Linear SVC	0.86	0.95
Word2Vec	Imbalanced	Logistic Regression	0.85	0.99
Word2Vec	Imbalanced	Linear SVC	0.84	0.82
GloVe	Balanced	Logistic Regression	0.79	0.88
GloVe	Balanced	Linear SVC	0.80	0.66
GloVe	Imbalanced	Logistic Regression	0.77	0.81
GloVe	Imbalanced	Linear SVC	0.77	0.60

Final Conclusions

Best Overall Approach:

- **TF-IDF + Logistic Regression** was the strongest performer under balanced classes.
- It achieved **~89% accuracy** and **~0.89 macro F1-score**, outperforming both Word2Vec and GloVe embeddings.
- This makes it the most reliable traditional NLP setup when class distribution is even.

Best Semantic Embedding Method:

- **Word2Vec embeddings with balanced classes** slightly outperformed GloVe
- **Macro F1-score:** ~0.86
- **Example review confidence:** 1.00 (LR) / 0.95 (Linear SVC)
- Captures more nuanced sentiment patterns compared to GloVe (~0.77–0.79 macro F1 under imbalance).

Key Insights:

- Balancing helped improve fairness between classes, but the imbalanced results still reflect real-world distributions.
- Dense embeddings (Word2Vec, GloVe) work well, but TF-IDF + classical ML often gives better results for sentence-level sentiment.
- Logistic Regression vs Linear SVC: both strong; LR slightly better for confidence, SVC for speed on large datasets.

Practical Takeaways:

- **For real-world sentiment analysis:** TF-IDF + classical ML is simple and highly effective.
- **Use embeddings:** when deeper semantic understanding or neural models are involved.

Future Considerations

- **Explore Deep Learning Models:** Test LSTM or transformers (BERT) for richer contextual understanding. These typically outperform classical ML on semantic tasks.

Thank You!