IMDB Reviews: Sentiment Analysis

Team 25

|  |  |  |
| --- | --- | --- |
| Nikola Dobricic | Milan Stankovic | Ioana Popescu |
| * LSTM Model * Embedding methods * Random Forests | * Bert Model * Model Performance * Keyword 3 | * Data Analysis * Bow Model |

# Highlights:

* **Multiple Methodologies**: Our project explored four different approaches to sentiment analysis, showcasing a comprehensive comparison:
  + Bag of Words
  + Random Forests with GloVe Embeddings
  + LSTMs
  + BERT

This allowed us to understand the strengths and limitations of traditional deep learning techniques in our specific task of predicting movie review ratings.

* **Word Embeddings**: The use of GloVe embeddings with Random Forests and a pre-trained WordPiece tokenization with the BERT model demonstrated how we can analyze text, by projecting it into a latent space of word vectors
* **Robust Data Preparation**: We emphasized thorough data preprocessing, including cleaning, tokenization, and handling of stop words. We conducted exploratory data analysis to understand the characteristics of the dataset before training.
* **Importance of context in NLP:** in sentiment analysis, models likeLSTMs, capable of remembering long-term dependencies in sequential data, outperform others in terms of accuracy and generalization, as they capture semantic relationships between words and contextual meaning

# Brief Report:

### 1. Project Scope and Approach

The **scope** of our project was to develop and train models to predict a **rating** (on a scale from 1 to 10) based on the text of movie reviews.

We aimed to answer the following key questions:

* **How accurately can we predict the rating of a movie based on its text reviews?**
* **Which modeling approaches are most effective for this task?**
* **What insights can we derive from the text that correlate with different rating levels?**

The dataset used for training and evaluation was sourced from **Kaggle** [1], which contains 50,000 IMDB movie reviews. This dataset is commonly used for **natural language processing (NLP)** and text analytics tasks, specifically for **binary sentiment classification.** However, our goal was to extend beyond binary classification to predict a continuous rating.

Our models can be used by **marketing** and **film companies** to gauge public opinion on movies from reviews on platforms like **Twitter**, **Instagram**, and **Facebook**, where explicit ratings are not provided. This can help in sentiment analysis and market research, providing deeper insights into **consumer opinions and trends.**

### 2. Methodology

Our project proceeded through the following steps:

* **Data Preprocessing**:
  + Text Cleaning: Removal of HTML tags, punctuation, and special characters.
  + Tokenization: Splitting text into words.
  + Stop Words Removal: Removing common but uninformative words.
* **Exploratory Data Analysis (EDA)**:
  + Visualization: Distribution of ratings, word cloud of frequent terms.
  + Insights: Understanding patterns and trends in the data.

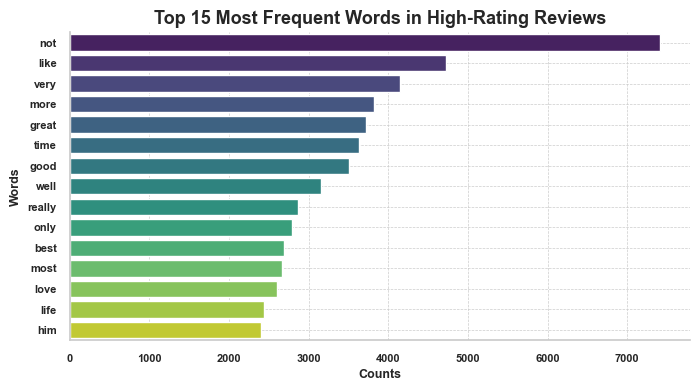
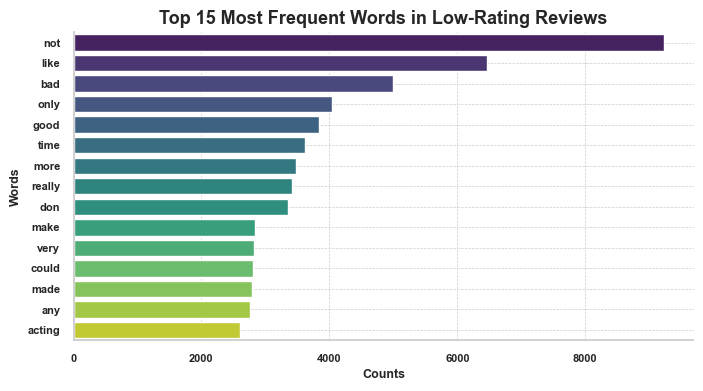


Figure 1: Most frequent words in positive reviews and negative reviews differ significantly

* **Feature Extraction**:
  + Bag of Words: Converting text into vectors of word counts.
  + GloVe Embeddings: Using pre-trained word vectors to capture semantic meaning.
  + Sequential Data Representation: Preparing data for models that process text sequentially, like LSTMs.
* **Model Training**:
  + Bag of Words Model: A baseline approach using **Linear Regression**. We also removed frequent words which provide no meaningful information: “and”, “the”, “him”
  + **Random Forest** with **GloVe Embeddings**: An approach leveraging GloVe [2] for better feature representation into a rich latent space, and Random Forests for handling high-dimensional data, capturing non-linear relationships and insights into feature importance.

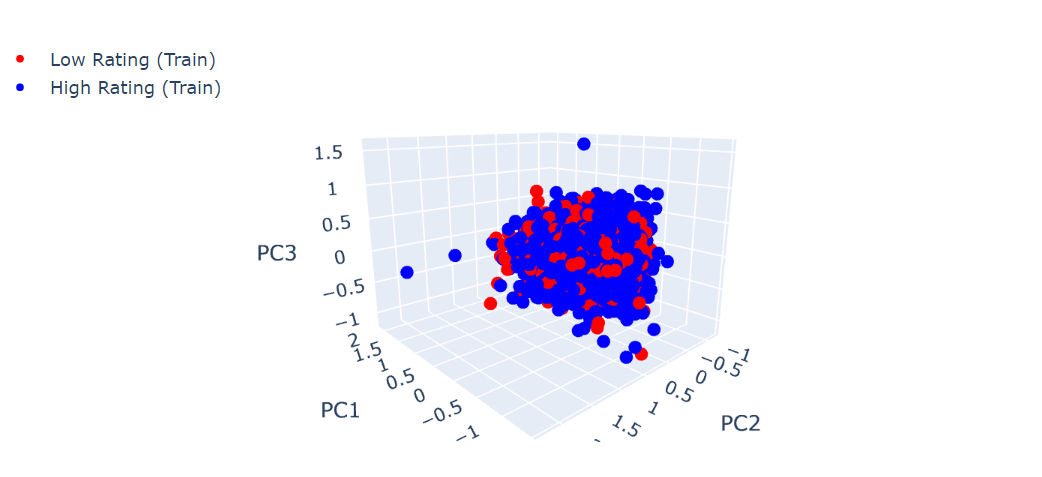


Figure 2: Projecting GloVe embeddings from 50 dimensions to 3D

using PCA does not show clear clustering

* + LSTM (Long Short-Term Memory): A type of **Recurrent Neural Network (RNN)** that captures long-term dependencies in text.
  + BERT (Bidirectional Encoder Representations from Transformers): An advanced **Autoencoder** model that uses Transformerarchitecture to understand context in both directions [3].

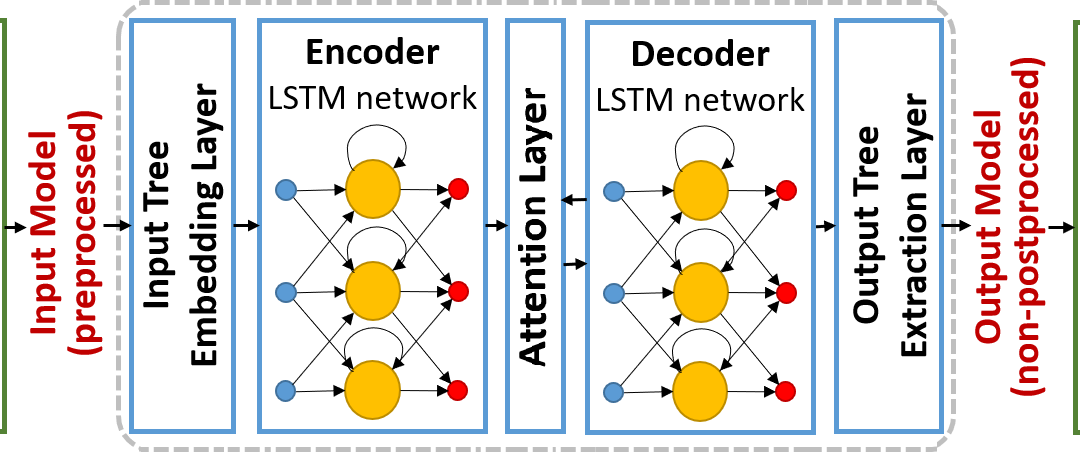


Figure 3: LSTM Architecture [4]

* **Model Evaluation**:
  + Metrics: Evaluated models using Mean Squared Error (MSE) and Mean Absolute Error (MAE), **Precision and Recall**, and also a **Confusion Matrix**.
  + Loss function: We monitored the loss function on both the training and validation sets in real-time throughout the training process
  + Comparison: Performance comparison of different models to identify the best approach.

|  | Bow | Bow with words removed | Random Forest | LSTM | LSTM  Plus | Bert |
| --- | --- | --- | --- | --- | --- | --- |
| Loss | - |  | - | MSE |  | MSE |
| Optimizer | - |  | - | Adam |  | Adam |
| MSE |  |  |  |  |  |  |
| MAE |  |  |  |  |  |  |
| Precision |  |  |  |  |  |  |
| Recall |  |  |  |  |  |  |

Figure 4: Performance metric table

### 3. Key Findings:

* The **Bag of Words** approach, while simple, was ineffective due to its inability to capture word order and context. Even when **removing filler words**, performance does not improve. We would roughly get the same score just by always predicting a rating of 5. We used it as our baseline.
* The **Random Forest** with GloVe embeddings provided a good balance of performance and computational efficiency.
* Pre-trained Embeddings such as GloVe or Word2Vec are trained on extensive datasets and capture general semantic relationships between words. They save computational resources and enable transfer learning. However, in our case, **training our own embeddings** worked better, as movie reviews often contain specific vocabulary and idiomatic expressions related to films, genres, and subjective opinions that may not be well-represented in general pre-trained embeddings. Hence, we can capture context and sentiment-specific nuances more accurately, leading to better performance (as can be seen in Figure 4).
* **LSTMs** outperformed other models due to their ability to **capture sequential dependencies and context** in the text. Moreover, LSTMs can handle input sequences of varying lengths, making them suitable for movie reviews that can range from a few sentences to several paragraphs.
* **BERT** showed strong performance, considering its deep contextual understanding and its handling of complex language constructs. As BERT sets new benchmarks in various NLP tasks and given that we only **fine-tuned** the last layers for our specific task, we felt like we were 'cheating' by leveraging its advanced pre-trained architecture rather than constructing our own model from scratch.

### Future steps: Unsupervised learning on text extracted from social media platforms - Twitter, Facebook, Instagram

### 4. Limitations & Omitted Material

# While our project aims to provide a robust framework for predicting movie ratings from text reviews, there are certain limitations we acknowledge:

# Dataset Specificity: Our models are trained on a specific dataset from IMDB, which may limit generalizability to other types of reviews or platforms without further adaptation.

# Scope of Analysis: We focused on predicting ratings but did not explore other aspects such as the impact of review length or reviewer credibility on prediction accuracy.

# Due to time constraints, we did not include:

# Detailed hyperparameter tuning processes for each model.

# Extensive analysis of model interpretability and feature importance.

### 5. References

**[1] IMDB Dataset**: Kaggle, https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

**[2] GloVe**: Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global Vectors for Word Representation.

**[3] BERT**: Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

**[4] LSTM Architecture:** https://modeling-languages.com/lstm-neural-network-model-transformations/