Methods

Overview

This study focuses on predicting numerical customer ratings from e-commerce reviews using advanced machine learning models. I employed three different models: Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and Transformer models, utilizing the ”Amazon Product Review Dataset.” My objective is to identify the most effective model for this task and to analyze the specific features that contribute to accurate sentiment prediction.

Data Collection and Preprocessing

The ”Amazon Product Review Dataset” from the University of California, San Diego, was used for this project. This dataset includes millions of reviews, offering a comprehensive view of customer feedback across a wide range of products. Key features of the dataset include:

- **Overall Rating:** Numeric customer rating from 1 to 5, providing a straightforward metric for sentiment analysis.
- **Review Content:** Full review text and summary, offering rich textual data for sentiment extraction.
- **Review Timing:** Date and UNIX timestamp, allowing analysis of temporal trends in customer feedback.

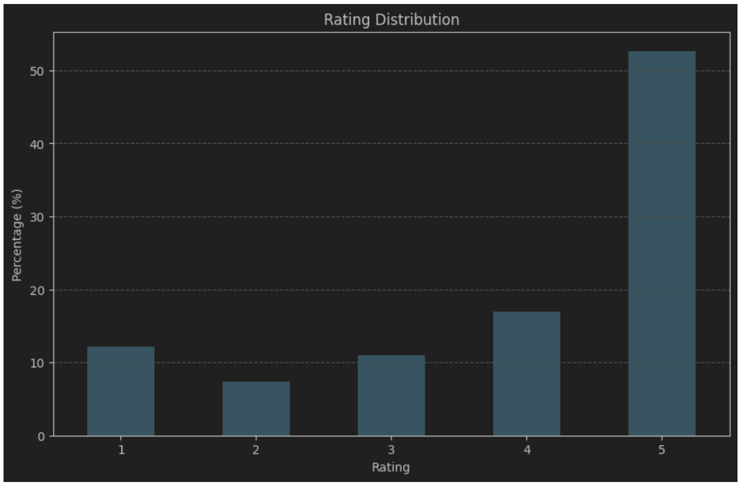


Figure 1: Overall rating distribution of the dataset.

Preprocessing Steps

Effective preprocessing is critical for the performance of machine learning models. Our preprocessing pipeline included the following steps:

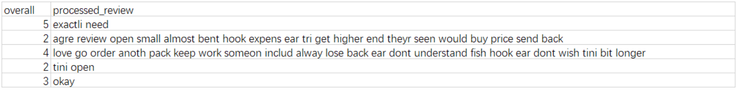
1. **Data Cleaning:** Removed non-textual elements (e.g., HTML tags, special characters) and handled missing values to ensure data quality.

2. **Text Normalization:** Converted text to lowercase, removed punctuation, and filtered out common stopwords to reduce noise and standardize the text.

3. **Tokenization:** Split text into individual tokens (words), providing the basic units for further processing.

4. **Embedding:** Used Word2Vec embeddings to transform text into vector representations, capturing semantic relationships between words.

After preprocessing, the dataset was ready for model training and evaluation. Here is the the first 5 lines of the dataset after processing.



overall	processed_review
5	exactli need
2	agre review open small almost bent hook expens ear tri get higher end theyr seen would buy price send back
4	love go order anoth pack keep work someon includ alway lose back ear dont understand fish hook ear dont wish tini bit longer
2	tini open
3	okay

Figure 2: Data example after preprocessing.

Model Architectures and Training

Convolutional Neural Network (CNN)

The CNN model was chosen for its ability to identify local patterns within text data, such as phrases and short sequences of words. The architecture consisted of:

- **Embedding Layer:** Transformed text into vectors using pre-trained Word2Vec embeddings, capturing word semantics.

- **Convolutional Layer:** Applied 128 filters of size 3x3 to extract local features from the text, enabling the detection of important n-grams.

- **Global Max Pooling:** Reduced dimensionality by selecting the most significant features from each filter, focusing on the most relevant information.

- **Dropout Layer:** Implemented dropout with a rate of 0.5 to prevent overfitting and enhance the model's generalization capability.

- **Dense Layer:** Final fully connected layer for sentiment prediction, outputting the numerical rating.

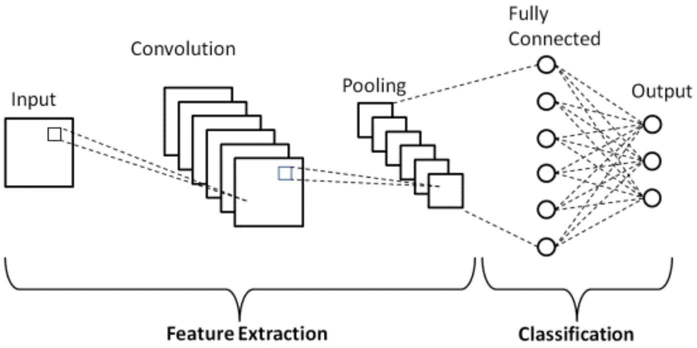


Figure 3: CNN Architecture

Long Short-Term Memory (LSTM)

LSTM networks were utilized for their ability to manage sequential data and capture long-term dependencies, which are essential for understanding context in text. The architecture included:

- **Embedding Layer:** Transformed text into vectors, similar to the CNN, using pre-trained Word2Vec embeddings.
- **LSTM Layers:** Processed text sequences with 128 hidden units, retaining important information over extended sequences.
- **Dropout Layer:** Added a dropout rate of 0.5 to mitigate overfitting.
- **Dense Layer:** Produced the final sentiment prediction by aggregating the information processed by the LSTM layers.

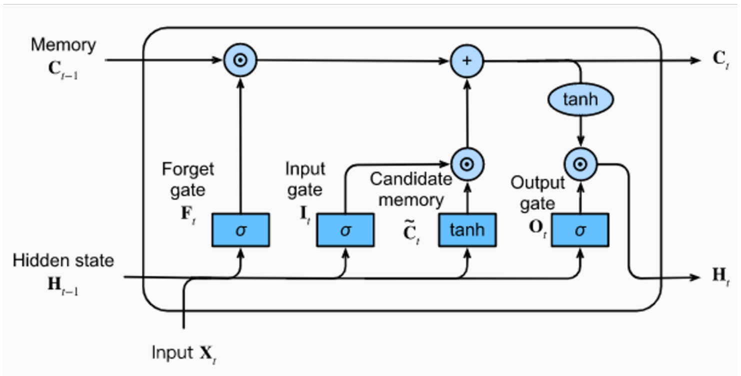


Figure 4: LSTM Architecture

Transformer

The Transformer model leverages self-attention mechanisms to process entire input sequences efficiently, allowing it to capture global context and relationships within the text. The architecture featured:

- **Embedding Layer:** Transformed text into vectors using pre-trained embeddings.
- **Positional Encoding:** Retained sequential order by adding positional information to the embeddings, which is crucial for understanding the structure of the text.
- **Multi-head Attention:** Applied multiple attention mechanisms to focus on different parts of the input sequence simultaneously, enhancing context capture.
- **Feedforward Layers:** Consisted of fully connected layers that processed the attention outputs to generate the final prediction.

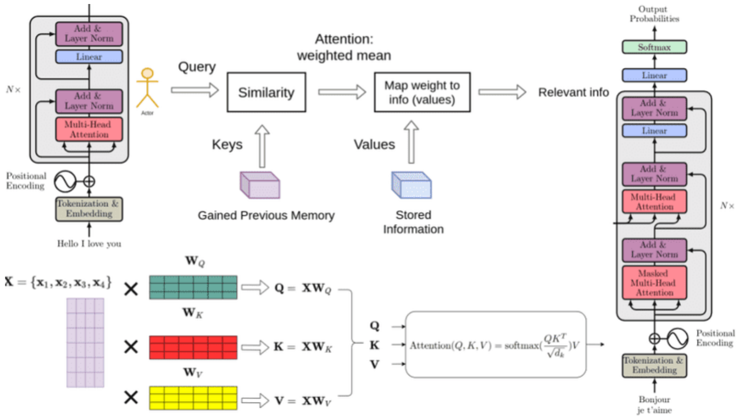


Figure 5: Transformer Architecture

Experimental Setup

Each model was trained and evaluated using the preprocessed dataset. The training process involved the following steps:

- **Training-Validation Split:** The dataset was split into training (80%) and validation (20%) sets to evaluate model performance on unseen data.
- **Optimization:** Models were optimized using the Adam optimizer, which is known for its efficiency and adaptability in training deep learning models.

- **Loss Function:** Mean Squared Error (MSE) was used as the loss function to measure the difference between predicted and actual ratings, with the goal of minimizing this error.

Parameter	**Value**
Training-Validation Split	80/20
Optimizer	Adam
Loss Function	Mean Squared Error (MSE)

Table 1: Experimental Parameters

Evaluation Metrics

The performance of the models was assessed using the following metrics:

- **Mean Absolute Error (MAE):** Measures the average magnitude of errors in predictions, providing a straightforward interpretation of prediction accuracy.
- **Root Mean Square Error (RMSE):** Measures the square root of the average squared differences between predicted and actual values, giving more weight to larger errors.

These metrics were chosen to comprehensively evaluate the models' ability to predict customer ratings accurately.

By implementing these methodologies, I aimed to accurately predict customer ratings from textual reviews, providing valuable insights for e-commerce platforms. This study also explores the specific words and features that significantly influence customer ratings, offering actionable insights for improving customer satisfaction.