Sentiment-Based Rating Prediction for Health and Personal Care Products on Amazon

Harshavardhan Patekar
School of Computing
Dublin City University
Dublin, Ireland
harshavardhan.patekar2@mail.dcu.ie

Mervin Shibu George
School of Computing
Dublin City University
Dublin, Ireland
mervin.shibugeorge2@mail.dcu.ie

Saurav Nambiar
School of Computing
Dublin City University
Dublin, Ireland
saurav.nambiar2@mail.dcu.ie

Vignesh Jagdish Battulkar
School of Computing
Dublin City University
Dublin, Ireland
vigneshjagdish.battulkar2@mail.dcu.ie

***Abstract*—In the era of digital commerce, user-generated reviews serve as a critical resource for understanding customer sentiment and product performance. This study explores the feasibility of predicting product ratings based on textual reviews using Natural Language Processing (NLP) techniques. Focusing on Health and Personal Care products from the Amazon Reviews 2023 dataset, the project investigates the correlation between the content of user reviews and their corresponding star ratings. The primary objective is to evaluate the effectiveness of various text representation methods—such as TF-IDF, FastText, GloVe, and Word2Vec—in combination with machine learning and deep learning models, including Logistic Regression, Linear SVC, Random Forest, and BiLSTM. Through extensive experimentation, the study aims to determine which combinations offer the most accurate predictions. The findings have potential applications in automating product rating systems, enhancing recommendation engines, and identifying inconsistencies or biases in user reviews. By leveraging NLP for sentiment-based rating prediction, the research contributes to a more transparent and intelligent review ecosystem that can support better decision-making for consumers and sellers alike.**

***Index Terms*—Natural Language Processing (NLP), Rating Prediction, Amazon Reviews, Health and Personal Care, TF-IDF, Word Embeddings, FastText, GloVe, Word2Vec, BiLSTM, Machine Learning, Deep Learning, Text Classification, Consumer Reviews.**

I. Introduction

In today's digital economy, user-generated content such as online reviews plays a pivotal role in shaping consumer behavior and influencing purchase decisions. With the exponential growth of e-commerce platforms like Amazon, understanding customer sentiment has become increasingly vital not only for consumers but also for businesses striving to enhance their products and services [1], [2]. As reviews are often unstructured and voluminous, analyzing them manually is impractical. This has led to a surge in the use of Natural Language Processing (NLP) techniques for automating sentiment analysis and, more recently, for predicting product ratings from review content [3].

Sentiment-based rating prediction is a specific task within opinion mining that seeks to predict a product's star rating based solely on its textual reviews and summaries [4]. Unlike general sentiment classification which labels reviews as simply positive or negative, rating prediction requires mapping nuanced textual data to a more granular scale—typically 1 to 5 stars. This poses challenges due to linguistic variability, implicit sentiments, and diverse user expectations [5].

Several studies have tackled this problem using a combination of text vectorization techniques like Bag-of-Words (BoW), TF-IDF, Word2Vec, and GloVe, paired with supervised learning models such as Support Vector Machines (SVM), Logistic Regression (LR), and Random Forests (RF) [6], [3]. More recent work also explores deep learning approaches, including Long Short-Term Memory (LSTM) networks and Bidirectional LSTMs (BiLSTM), which are capable of capturing contextual dependencies in textual data [7].

This study focuses on the domain of Health and Personal Care products, which are particularly sensitive to user sentiment due to their direct impact on well-being. Using the 2023 Amazon Reviews dataset, we aim to answer the research question: Can NLP techniques accurately predict product ratings from user-generated reviews for Health & Personal Care products on Amazon? Specifically, we experiment with various combinations of feature extraction (TF-IDF, FastText, GloVe) and classification algorithms (Logistic Regression, Linear SVC, Random Forest, BiLSTM) to evaluate model performance. By comparing traditional machine learning methods with deep learning models, this research contributes to understanding the trade-offs in model complexity and interpretability for the rating prediction task. The outcomes of this work are expected to aid future recommender systems by automating rating predictions where explicit feedback is unavailable or incomplete [8], [9].

II. Related Work

Understanding sentiment in user-generated reviews has become central to various recommendation and rating systems, especially on platforms like Amazon. Several studies have examined how natural language processing (NLP) techniques can be applied to extract useful features from text and convert them into numerical representations for predictive modeling. For instance, Reddy et al. [5] proposed a star rating prediction system based on n-gram features and traditional classifiers like Naïve Bayes and Random Forest. Their findings show that bigram-trigram Naïve Bayes achieved comparable performance to Random Forest with reduced computational complexity, highlighting the trade-off between accuracy and efficiency. Similarly, Rathor et al. [1] demonstrated the effectiveness of unigrams and weighted unigrams when paired with classifiers like Support Vector Machines (SVM), Naïve Bayes, and Maximum Entropy, concluding that SVM consistently outperformed others in accuracy. However, these studies largely rely on syntactic features and do not incorporate semantic embeddings, limiting their ability to capture contextual meaning in reviews.

To address the limitations of bag-of-words-based approaches, researchers have explored advanced vector representations like TF-IDF and semantic embeddings. Asghar [4] evaluated multiple machine learning models—including Logistic Regression and Linear SVC—combined with TF-IDF, unigrams, bigrams, and trigrams. While the study highlighted that TF-IDF and SVC yielded competitive performance, it also emphasized the inherent challenges in capturing nuanced sentiments that go beyond surface-level word frequency. Haque et al. [6], working on a large-scale Amazon review dataset, also employed preprocessing and sentiment analysis using traditional classifiers. However, they observed that imbalanced rating distributions, especially the dominance of 4- and 5-star ratings, could bias classifier performance. These observations informed our decision to incorporate stratified sampling and evaluate with metrics beyond accuracy.

In addition to frequency-based approaches, there has been significant interest in sentiment analysis using deep learning models, particularly for linguistically diverse or noisy data. Roy [7] proposed an ensemble framework combining BiLSTM, CNN, and transformer-based models like BERT for low-resource, code-mixed languages. While the dataset differed, the architectural choice of BiLSTM with contextual embeddings demonstrated substantial improvements in F1-scores—validating our inclusion of BiLSTM with FastText and GloVe in this study. Similarly, Shukla and Dwivedi [3] evaluated various deep learning and classical models for Amazon review classification, confirming that BiLSTM performed particularly well when combined with robust preprocessing methods. This reinforces the potential of deep learning techniques, especially when dealing with informal, subjective review text common in Health and Personal Care products.

Some studies have also focused on review structure and social dynamics. Lei et al. [8] proposed a sentiment-based rating prediction model that incorporated user sentiment similarity, interpersonal influence, and item reputation. Although designed for a social network context, the sentiment modeling techniques are directly relevant to our objective of extracting meaning from review text. In a similar vein, AlZu'bi et al. [2] analyzed Amazon review metadata such as helpfulness votes, underlining how community interactions influence perceived sentiment. While our study does not utilize such metadata, it further supports the argument that the quality and informativeness of review text are critical for predicting star ratings accurately.

Bringing all of these perspectives together, it becomes clear that traditional machine learning models perform well on structured and shallow features like TF-IDF and n-grams, as demonstrated by Asghar [4], Reddy et al. [5], and Rathor et al. [1], but they fall short in capturing deeper semantics, which are crucial in sentiment-rich domains like Health and Personal Care. Deep learning methods, particularly BiLSTM, as discussed by Roy [7] and Shukla and Dwivedi [3], offer promising accuracy gains by leveraging semantic context, though at higher computational costs. Our approach integrates the best of both domains—testing combinations of TF-IDF, FastText, and GloVe with both traditional classifiers and BiLSTM—to evaluate their effectiveness in sentiment-based star rating prediction on Amazon Health & Personal Care product reviews.

III. Data Mining Methodology

To answer the research question—Can NLP techniques accurately predict product ratings from user-generated reviews for Health & Personal Care products on Amazon?—we adopted the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. This iterative and structured methodology enabled us to systematically approach each stage of the project, from understanding the business problem to evaluating results.

A. Business Understanding

Online reviews are a vital form of user-generated content, particularly in the Health and Personal Care sector, where they offer insights into product efficacy, safety, and overall well-being. However, the volume and inconsistency of such reviews pose challenges in effective interpretation.

Predicting product ratings from textual reviews has several strategic advantages:

- **Enhanced recommendations:** Improves personalization by integrating predicted ratings into collaborative filtering models.
- **Anomaly detection:** Flags reviews where sentiment and star ratings are misaligned.
- **Quality monitoring:** Helps brands track sentiment trends and identify potential product issues.
- **Support for unrated products:** Allows sentiment-based quality estimation when numerical ratings are insufficient.

This research addresses the question: *Can NLP techniques accurately predict product ratings from user reviews in the Health & Personal Care category on Amazon?*

To explore this, we combine text representation methods—TF-IDF, FastText, GloVe, and Word2Vec—with traditional classifiers such as Logistic Regression, Linear SVC, and Random Forest, as well as deep learning models like BiLSTM.

Given the nuanced and emotionally rich nature of health-related feedback, traditional rating aggregation methods may fail to capture the true sentiment. Our goal is to build a scalable, interpretable NLP-based model that supports both businesses and consumers with more reliable review analysis.

B. Data Understanding

The initial phase of the CRISP-DM methodology involved developing a thorough understanding of the dataset used for predicting Amazon product review ratings. This process aimed to familiarize ourselves with the data's characteristics, identify potential quality issues, and uncover preliminary insights relevant to the classification task. The dataset chosen was the "Health and Personal Care" category subset from the well-known Amazon Product Data collection, typically sourced from repositories like UCSD or Jianmo Ni and provided in a JSON Lines (`.jsonl`) format, where each line represents a single review.

Initially, the dataset comprised approximately 494,121 reviews, each described by 10 distinct attributes. The core attributes for this project are the `rating`, the `title`, and the `text`. The `rating` is the numerical target variable, an integer from 1 (most negative) to 5 (most positive), representing the customer's expressed sentiment. The `title` (string) and `text` (string) fields contain the user-generated written content and are the primary sources for the predictive textual features. Other attributes included metadata such as image associations (`images`), product and user identifiers (`asin`, `parent_asin`, `user_id`), review timestamp (`timestamp`), helpfulness votes (`helpful_vote`), and purchase verification status (`verified_purchase`), though these were not directly utilized as features in the text-based rating prediction models developed in this study. Exploratory

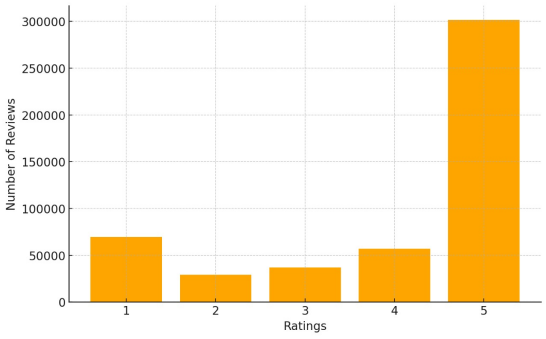

Fig. 1. Distribution of Reviews by rating

data analysis revealed several critical characteristics. Most importantly, the distribution of the target variable, `rating`, exhibited a significant class imbalance as shown in 1. The distribution was heavily skewed towards positive ratings, specifically 5-star reviews. The approximate distribution observed was:

- 5 Stars: 301,713 reviews (61.1%)
- 4 Stars: 57,000 reviews (11.5%)
- 3 Stars: 36,949 reviews (7.5%)
- 2 Stars: 28,895 reviews (5.8%)
- 1 Star: 69,564 reviews (14.1%)

Implication: This severe imbalance necessitates careful consideration during modeling. Standard accuracy metrics can be misleading, and models might develop a bias towards predicting the majority class (5 stars). Techniques such as class weighting or resampling, along with evaluation metrics robust to imbalance (e.g., Macro F1-score, per-class metrics, confusion matrix analysis), are required.

Analysis of the predictor features, `title` and `text`, confirmed they contain unstructured natural language expressing subjective opinions. The text varies significantly in length and style and includes informal language, potential typographical errors, and noise typical of user-generated content, such as inconsistent casing, punctuation, and special characters. To create a richer context for analysis, the `title` and `text` fields were concatenated. Regarding data quality, missing values were present, particularly in the `title` column; these were imputed with empty strings before concatenation. More critically, a small number of rows lacked the essential `text` or `rating` and were therefore removed from the dataset. The presence of noise in the text reinforced the need for a comprehensive preprocessing pipeline involving lowercasing, removal of punctuation and special characters, stop word elimination, and lemmatization prior to feature extraction. These initial findings established the basis for the subsequent data preparation and modeling stages, highlighting the challenges posed by class imbalance and noisy text data, and confirming the central role of the textual content in predicting review ratings.

C. Data Preparation

Following the insights gained during the Data Understanding phase, the Data Preparation phase focused on selecting, cleaning, transforming, and structuring the raw dataset into a format suitable for the various modeling techniques employed. This involved several key steps to handle data quality issues and extract meaningful features from the textual reviews.

First, relevant data was selected. For the task of predicting the review `rating` based on its content, the primary columns of interest were `rating`, `title`, and `text`. Other metadata columns such as `asin`, `user_id`, `timestamp`, `images`, `helpful_vote`, and `verified_purchase` were excluded from the feature set for the implemented models, as the focus was solely on leveraging the textual information.

Data cleaning addressed the quality issues identified earlier. Rows containing missing values in the essential `text` or `rating` columns were removed from the dataset to ensure

valid input for training and evaluation. Missing values within the `title` field were handled by imputing them with an empty string. Subsequently, to create a comprehensive textual context for each review, the potentially empty `title` was concatenated with the `text` column, creating a new feature, `review_full`, containing the complete written content.

Feature construction then focused on preprocessing the combined text data (`review_full`) to reduce noise and standardize the content. A text cleaning pipeline was implemented using libraries such as NLTK, involving the following sequential steps:

- Conversion to lowercase to ensure case-insensitivity.
- Removal of HTML tags, which might be present in scraped web data.
- Removal of punctuation and special characters using regular expressions, retaining only alphabetic characters and whitespace.
- Tokenization, splitting the cleaned text into individual words (tokens).
- Removal of common English stop words (e.g., "a", "the", "is") that typically do not carry significant sentiment information.
- Lemmatization, reducing words to their base or dictionary form (e.g., "running" becomes "run") to consolidate related terms, using the WordNet lemmatizer.

The processed tokens for each review were then joined back into a single string, resulting in a `review_cleaned` feature. Any reviews that became empty strings after this extensive cleaning process were removed.

The next critical step was feature extraction, converting the cleaned textual data (`review_cleaned`) into numerical representations suitable for machine learning algorithms. Several approaches were explored:

- **TF-IDF (Term Frequency-Inverse Document Frequency):** This technique represents each review as a vector where each dimension corresponds to a word in the vocabulary, and the value represents the importance of that word within the review relative to the entire corpus. It captures keyword importance but ignores word order and semantic similarity. The `TfidfVectorizer` from `scikit-learn` was used, with experiments potentially involving parameters like `max_features` to limit vocabulary size. This approach was used as input for classical machine learning models like Logistic Regression and Linear SVM.
- **Word Embedding Averaging:** To capture semantic meaning for input into classical machine learning models, pre-trained word embeddings were utilized. Specifically, FastText embeddings (`fasttext-wiki-news-subwords-300`, loaded via `gensim`) were chosen due to their ability to handle out-of-vocabulary (OOV) words using subword information, a significant advantage for user-generated text. For each review, the embedding vectors for its constituent words (present in the FastText model) were retrieved and averaged to produce a single, dense vector representation for the entire review. This averaged vector was used as input for models like Logistic Regression, Linear SVM, and Random Forest.
- **Sequence Preparation (for RNNs):** For sequence models like Bidirectional LSTMs (BiLSTMs), a different preparation workflow was necessary to preserve word order. Text was tokenized using Keras' `Tokenizer`, mapping words to integer indices based on frequency (limited by `MAX_NUM_WORDS`). These integer sequences were then padded or truncated to a fixed length using `pad_sequences` to create uniform input tensors for the neural network. To initialize the Keras `Embedding` layer with semantic information, different pre-trained word embedding models were experimented with. Specifically, embedding matrices were constructed using vectors from Word2Vec (e.g., Google News 300d), GloVe (e.g., Common Crawl 300d), and FastText (`fasttext-wiki-news-subwords-300` and `cc.en.300.bin`), corresponding to the Keras tokenizer's vocabulary. This allowed the BiLSTM models to process sequences of semantically rich word vectors.

Finally, the processed dataset (containing either TF-IDF features, averaged embeddings, or padded sequences, along with the corresponding ratings) was split into training and testing sets using `scikit-learn`'s `train_test_split` function. An 80% training and 20% testing split was used. Crucially, the `stratify=y` parameter was employed during splitting to ensure that the proportion of each rating class was preserved in both the training and testing sets, mitigating potential biases introduced by the significant class imbalance identified during data understanding. A fixed `random_state` was used to ensure the reproducibility of the split.

This preparation process resulted in distinct training and testing datasets, formatted appropriately for input into the various machine learning and deep learning models developed in the subsequent modeling phase.

D. Modeling

Following data preparation, the modeling phase involved selecting, configuring, and applying various machine learning and deep learning techniques to predict the review `rating` based on the processed textual features. The goal was to evaluate the effectiveness of different modeling paradigms and feature representations for this sentiment-based classification task. A diverse range of models was explored, encompassing both classical machine learning algorithms and more complex deep learning sequence models.

The first set of models explored involved established classical machine learning algorithms, known for their interpretability and computational efficiency relative to deep learning methods. These served as important baselines. Specifically, Logistic Regression (LR), Linear Support Vector Classifier (LinearSVC), and Random Forest (RF) were implemented using the `scikit-learn` library. These models were trained

using two primary types of numerical features derived from the cleaned review text:

- **TF-IDF Vectors:** Sparse vectors capturing word frequency and importance, generated using `TfidfVectorizer`.
- **Averaged Word Embeddings:** Dense vectors created by averaging the pre-trained FastText (`fasttext-wiki-news-subwords-300`) word vectors for the words present in each review.

Crucially, to address the significant class imbalance identified in the data, the `class_weight='balanced'` parameter (or `class_weight='balanced_subsample'` for Random Forest) was consistently utilized during the training of these scikit-learn models. This adjustment helps prevent the models from being overly biased towards the majority 5-star rating class.

The second set of models focused on deep learning techniques, specifically sequence models designed to capture contextual information and word order dependencies, which are often lost in TF-IDF or simple averaging approaches. Bidirectional Long Short-Term Memory (BiLSTM) networks were chosen as the primary sequence modeling architecture. These models process the input sequence in both forward and backward directions, allowing them to learn context more effectively. The BiLSTM models were implemented using TensorFlow and its Keras API. The input pipeline for these models involved:

- Processing the cleaned text into sequences of integer indices using Keras' `Tokenizer`.
- Padding these sequences to a uniform length (`MAX_SEQUENCE_LENGTH`).
- Utilizing a Keras `Embedding` layer as the first layer of the network. This layer was initialized with weights derived from various pre-trained word embedding models to inject semantic knowledge. Experiments were conducted using:
  - Word2Vec (e.g., Google News 300d vectors)
  - GloVe (e.g., Common Crawl 300d vectors)
  - FastText (both `fasttext-wiki-news-subwords-300` and `cc.en.300.bin` models)
- The typical architecture involved one or more BiLSTM layers (with varying numbers of units, e.g., 100, 200), followed by `Dropout` layers for regularization, and a final `Dense` output layer with 5 units and `softmax` activation to predict probabilities for the five rating classes.

Similar to the classical models, class imbalance for the BiLSTM training was addressed by calculating and supplying appropriate class weights to the Keras `model.fit` method. Given the computational intensity of training deep sequence models on this dataset size, GPU acceleration available via Google Colab was leveraged.

The selection of this diverse set of models and feature representations allowed for a comprehensive comparative analysis. This facilitated evaluation of traditional frequency-based features (TF-IDF) against semantic embeddings, the effectiveness of simple embedding averaging versus sequential processing (BiLSTM), and the performance differences between various pre-trained word embedding types within the deep learning framework. The trade-offs between model complexity, computational resource requirements, and predictive performance could thus be assessed.

IV. Evaluation and Results

This section details the evaluation methodology used to assess the performance of the various models developed for predicting Amazon review ratings. It discusses the rationale behind parameter choices, presents the comparative results of the models, and analyzes their implications regarding the feasibility and effectiveness of sentiment-based rating prediction for this dataset.

A. Evaluation Methodology

The primary objective of the evaluation was to determine how accurately each modeling approach could predict the 5-point star rating (`rating`) based on the review text (`title` and `text`). Given the significant class imbalance observed during data understanding (with ˜61% of reviews being 5-star), relying solely on overall accuracy would be misleading. Therefore, a combination of metrics was employed to provide a comprehensive assessment:

- **Overall Accuracy:** The standard measure of correctly classified instances across all classes, serving as a baseline.
- **Weighted Averages (Precision, Recall, F1-Score):** These metrics calculate the per-class score and average them, weighted by the number of true instances (support) for each class. This gives an overall performance score that accounts for class imbalance. Weighted Recall is mathematically equivalent to Overall Accuracy.
- **Macro Averages (Precision, Recall, F1-Score):** These metrics calculate the per-class score and take the unweighted average. This treats all classes equally, providing insight into how well the model performs across both majority and minority classes. The Macro F1-score is particularly crucial for evaluating performance on imbalanced datasets.
- **Classification Report:** Provides the precision, recall, and F1-score for each individual class (1 through 5 stars), allowing for detailed analysis of performance on specific rating levels.
- **Confusion Matrix:** Visualizes the model's predictions, showing the counts of reviews for each actual class versus each predicted class. This helps identify specific misclassification patterns (e.g., which ratings are most often confused).

The combination of weighted and macro averages, alongside the detailed per-class metrics and confusion matrix, allows for a robust evaluation that addresses the challenges posed by the imbalanced nature of the review rating data.

B. Parameterization Choices

Parameter settings can significantly influence model performance. The following outlines the key parameterization choices made for the different modeling approaches:

a) Classical Machine Learning Models:: For Logistic Regression (LR), Linear Support Vector Classifier (LinearSVC), and Random Forest (RF) implemented via `scikit-learn`:

- **Class Imbalance Handling:** The crucial `class_weight='balanced'` parameter (`'balanced_subsample'` for RF) was consistently used. This automatically adjusts weights inversely proportional to class frequencies, forcing the models to pay more attention to minority classes.
- **LR/LinearSVC:** Solvers like `'liblinear'` were used, suitable for the data size and feature types. `max_iter` was increased (e.g., to 2000) to ensure convergence. `dual=False` was used for LinearSVC when applicable (samples ¿ features).
- **Random Forest:** Configured with 200 trees (`n_estimators=200`) and no maximum depth limit (`max_depth=None`) as per experimental requirements, using the `'gini'` criterion. `n_jobs=-1` was set to utilize all available CPU cores.
- **Feature Extractors:** For TF-IDF, default settings or specific `max_features` (e.g., 10,000) were explored. For embeddings, the averaging approach was applied to pre-trained FastText or GloVe vectors (300 dimensions).

These parameters represent a combination of standard practices (like class weighting for imbalance) and specific experimental setups (like RF depth). Default settings were often used for other parameters to establish baseline performance.

b) Deep Learning Models (BiLSTM):: For the BiLSTM models implemented using Keras/TensorFlow:

- **Architecture:** Experiments involved two stacked BiLSTM layers (typically with 100 and 200 units, respectively) with `return_sequences=True` for the first layer. A `Dropout` rate of 0.1 was applied after each BiLSTM layer for regularization.
- **Embedding Layer:** Initialized with pre-trained weights from Word2Vec, GloVe, or FastText (300 dimensions). The weights were kept frozen (`trainable=False`) during initial experiments to leverage the pre-trained knowledge effectively. `MAX_NUM_WORDS` (e.g., 20,000) limited the vocabulary size, and sequences were padded/truncated to `MAX_SEQUENCE_LENGTH` (e.g., 200).
- **Output Layer:** A `Dense` layer with `softmax` activation was used. For most experiments, this had 5 neurons corresponding to the 5 rating classes. (Note: One specific Word2Vec experiment inadvertently used 9 output neurons, impacting the direct interpretability of its per-class metrics).
- **Training Parameters:** The `Adam` optimizer was used with a learning rate of `1e-04`. Training was conducted for 10 epochs (`EPOCHS=10`) with a `BATCH_SIZE` of 200. `EarlyStopping` (monitoring `val_loss` with patience 2) was used to prevent significant overfitting and save computational resources. Class weights, calculated similarly to the classical models, were supplied to the `model.fit` method to address imbalance. GPU acceleration was utilized due to the computational demands.

These parameters reflect choices aimed at building reasonably complex sequence models capable of capturing context, while managing resources and mitigating overfitting through standard techniques like dropout and early stopping.

C. Results

The performance of the various models on the held-out test set is summarized in Tables I & II.

TABLE I
MODEL PERFORMANCE COMPARISON: WEIGHTED METRICS AND ACCURACY

Model	Feature/Embedding	F1	Prec.	Recall	Acc.
Log. Regression		0.74	0.74	0.74	0.74
LinearSVC	TF-IDF	0.73	0.73	0.73	0.73
Random Forest		0.68	0.76	0.74	0.73
Log. Regression		0.68	0.68	0.69	0.69
LinearSVC	FastText	0.68	0.68	0.69	0.69
Random Forest		0.68	0.68	0.69	0.69
Log. Regression		0.65	0.67	0.71	0.70
LinearSVC	GloVe	0.67	0.67	0.68	0.68
Random Forest		0.57	0.69	0.67	0.66
	FastText	0.70	0.73	0.68	0.67
BiLSTM	GloVe	**0.75**	**0.75**	**0.77**	**0.77**
	Word2Vec	0.74	˜0.75	˜0.77	˜0.77

TABLE II
MODEL PERFORMANCE COMPARISON: MACRO AVERAGE METRICS

Model	Features	F1	Prec.	Recall
Log. Regression		0.55	0.55	0.55
LinearSVC	TF-IDF	0.54	0.54	0.54
Random Forest		0.46	0.80	0.42
Log. Regression		0.48	0.48	0.48
LinearSVC	FastText	0.48	0.48	0.48
Random Forest		0.48	0.48	0.48
Log. Regression		0.41	0.57	0.40
LinearSVC	GloVe	0.46	0.47	0.46
Random Forest		0.31	0.72	0.30
	FastText (Seq)	0.53	0.51	0.55
BiLSTM	GloVe (Seq)	**0.55**	**0.64**	**0.53**
	Word2Vec (Seq) (Note 1)	˜0.54	˜0.65	˜0.53

Detailed classification reports and confusion matrices for each model provided further insights into per-class performance and error patterns (refer to Appendix or specific outputs if included).

D. Discussion of Results and Implications

The evaluation results presented in Tables I & II reveal several key insights into the effectiveness of different approaches for sentiment-based rating prediction on this dataset.

a) Overall Performance Comparison:: The models exhibited a wide range of performance. The deep learning approaches leveraging sequence modeling (BiLSTM) generally outperformed the classical machine learning models trained on either TF-IDF or averaged embeddings, particularly when considering the weighted F1-score and overall accuracy. The BiLSTM model utilizing GloVe embeddings achieved the highest overall accuracy (0.7730) and the best balance according to Weighted F1 (0.75) and Macro F1 (0.55) scores among the evaluated configurations.

b) TF-IDF vs. Averaged Embeddings:: For the classical models (LR, LinearSVC, RF), TF-IDF features consistently yielded slightly better or comparable results compared to averaged FastText or GloVe embeddings, particularly in terms of Macro F1-score. TF-IDF + Logistic Regression (Macro F1: 0.55) slightly outperformed TF-IDF + LinearSVC (Macro F1: 0.54). Averaged embedding models often lagged, with Macro F1 scores around 0.41-0.48. This suggests that for these simpler models, the keyword importance captured by TF-IDF might be a stronger signal than the diluted semantic information obtained from simple averaging, which loses word order and context. The poor performance of Random Forest with averaged GloVe embeddings (Macro F1: 0.31) was particularly notable, potentially indicating difficulty handling the dense input or specific interactions.

c) Impact of Sequence Modeling (BiLSTM):: The introduction of BiLSTMs, which process sequences of embeddings rather than single averaged vectors, demonstrated clear benefits. The BiLSTM + GloVe model significantly outperformed all classical models and the BiLSTM + FastText configuration. This highlights the importance of word order and contextual information, which LSTMs are designed to capture, for accurately discerning sentiment nuances required for multi-class rating prediction. While BiLSTM + FastText did not perform as well as anticipated in this run (Macro F1: 0.53), it still outperformed most averaged embedding models.

d) Embedding Type Influence (BiLSTM):: Within the BiLSTM framework, the choice of pre-trained embedding significantly impacted results. GloVe embeddings led to the best performance (Accuracy: 0.77, Weighted F1: 0.75, Macro F1: 0.55), surpassing FastText (Accuracy: 0.68, Weighted F1: 0.70, Macro F1: 0.53). The Word2Vec results (Accuracy: 0.77) were comparable to GloVe in accuracy but are less reliable for detailed comparison due to the incorrect output layer configuration during that specific experiment. The superior performance with GloVe might suggest that the co-occurrence statistics it captures were particularly beneficial for this dataset, or perhaps there was better vocabulary overlap or alignment with the task compared to FastText in this instance.

e) Persistent Challenge of Class Imbalance:: Despite employing class weighting, a consistent pattern across all models is the difficulty in predicting the minority rating classes (2, 3, and 4 stars). While the models achieve high precision and recall for the majority 5-star class and reasonable performance for the 1-star class, the F1-scores for the intermediate ratings remain low (often in the 0.20-0.42 range even for the best models). This is reflected in the Macro F1-scores, which plateau around 0.55 even for the top-performing BiLSTM+GloVe model, indicating only moderate success in handling all classes equally well. Confusion matrices (not shown here) typically reveal significant confusion between adjacent classes and a tendency to misclassify intermediate ratings, often predicting them as the majority 5-star class.

f) Implications and Limitations:: The results demonstrate that predicting fine-grained ratings (1-5) from review text is challenging, especially for intermediate sentiments. While overall accuracy above 77% is achievable using sequence models like BiLSTM with appropriate embeddings (GloVe), the models struggle significantly with differentiating between 2, 3, and 4-star reviews. This suggests that while strong positive and negative sentiments are relatively easier to detect, capturing the nuances of mixed or moderate sentiment requires more sophisticated approaches or perhaps features beyond just the text. The classical models with TF-IDF provide a strong, efficient baseline, outperforming simple averaged embeddings in this case. The findings highlight the importance of sequence modeling for this task but also underscore the limitations of current models in fully overcoming the class imbalance problem for nuanced sentiment categories. Further improvements might require more advanced architectures (e.g., Transformers like BERT), fine-tuning of embeddings, incorporating other features, or more sophisticated techniques for handling imbalance or focusing on difficult classes. The presented results are specific to the chosen parameters and architectures; extensive hyperparameter tuning could potentially yield further improvements for any given model type.

V. Conclusions and Future Work

This study demonstrated the prediction of multi-class Amazon review ratings from text, comparing classical machine learning and deep learning approaches. Our findings indicate that sequence models significantly outperform methods relying on TF-IDF or averaged word embeddings. Specifically, a Bidirectional LSTM (BiLSTM) network using pre-trained GloVe embeddings yielded the best performance, achieving approximately 77.3% accuracy and a Macro F1-score of 0.55. While classical models like Logistic Regression with TF-IDF provided strong baselines (˜74% accuracy), they struggled more with nuanced sentiment compared to the BiLSTM.

A key limitation across all models, however, was the difficulty in accurately classifying the minority intermediate rating classes (2, 3, and 4 stars), despite using class weighting techniques. This highlights the persistent challenge of handling subtle sentiment expression and class imbalance simultaneously, as reflected in the modest Macro F1 scores.

Future work should focus on improving minority class performance, potentially through advanced imbalance techniques (e.g., focal loss, sophisticated resampling). Exploring state-of-the-art Transformer architectures (e.g., BERT, RoBERTa) fine-tuned on this data represents a promising direction. Further gains might also be realized through rigorous hyperparame-

ter optimization of the BiLSTM, fine-tuning the pre-trained embeddings, or incorporating non-textual features.

In summary, this work provides a comparative benchmark for sentiment-based rating prediction on this dataset, confirming the superiority of sequence modeling while identifying the key challenge of classifying intermediate sentiment levels, thereby contributing insights into the practical application of ML/DL for nuanced review analysis.

VI. Source Code & Dataset Availability

The source code for this project is available on GitHub and can be accessed using the following link:
https://github.com/Data-Mining-Assignment/rating-predictor

The dataset used in this study is publicly available and can be accessed here:
https://amazon-reviews-2023.github.io

Acknowledgment

We acknowledge the use of Gen AI tools, such as Chatgpt, which were used to improve clarity and readability. The reliability of the content was verified. Reviewing of the papers, synthesis, and arriving at conclusions were solely done by the authors. Any errors introduced by the use of these tools are sole responsibility of the authors.

References

[1] A. S. Rathor, A. Agarwal, and P. Dimri, "Comparative study of machine learning approaches for amazon reviews," *Procedia computer science*, vol. 132, pp. 1552–1561, 2018.

[2] S. AlZu'bi, A. Alsmadiv, S. AlQatawneh, M. Al-Ayyoub, B. Hawashin, and Y. Jararweh, "A brief analysis of amazon online reviews," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019, pp. 555–560.

[3] D. Shukla and S. K. Dwivedi, "The study of the effect of preprocessing techniques for emotion detection on amazon product review dataset," *Social Network Analysis and Mining*, vol. 14, no. 1, p. 191, 2024.

[4] N. Asghar, "Yelp dataset challenge: Review rating prediction," *arXiv preprint arXiv:1605.05362*, 2016.

[5] C. S. C. Reddy, K. U. Kumar, J. D. Keshav, B. R. Prasad, and S. Agarwal, "Prediction of star ratings from online reviews," in *TENCON 2017-2017 IEEE Region 10 Conference*. IEEE, 2017, pp. 1857–1861.

[6] T. U. Haque, N. N. Saber, and F. M. Shah, "Sentiment analysis on large scale amazon product reviews," in *2018 IEEE international conference on innovative research and development (ICIRD)*. IEEE, 2018, pp. 1–6.

[7] P. K. Roy, "Deep ensemble network for sentiment analysis in bi-lingual low-resource languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–16, 2024.

[8] X. Lei, X. Qian, and G. Zhao, "Rating prediction based on social sentiment from textual reviews," *IEEE transactions on multimedia*, vol. 18, no. 9, pp. 1910–1921, 2016.

[9] B. Nguy, "Evaluate helpfulness in amazon reviews using deep learning," in *Stanford University*, 2016.