Exploring E-Commerce Reviews on Amazon using Bidirectional Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) For Sentiment and Recommendation Analysis

Sharmin Hossain
sharmin.hossain@ndus.edu
North Dakota State University
Fargo, ND, USA

ABSTRACT

In today's marketing strategies, it is very important for companies to comprehend the sentiments of their customers. This knowledge about customers allows them to understand how their products and/or services are perceived and gives them insight into how to enhance their services. The objective of this project is to investigate the connection between various variables in customer reviews of Amazon's products and classify each review based on whether it recommends the product and whether it has a positive, negative, or neutral sentiment. I have used univariate and multivariate analyses on the dataset features and utilized a bidirectional recurrent neural network (RNN) with a long-short-term memory unit (LSTM) for recommendation and sentiment classification. I have also tried to find the differences between unidirectional & bidirectional analysis on the same dataset. The results showed that a recommendation is a reliable indicator of a positive sentiment score, and vice versa. I have also found that the bidirectional LSTM achieved an F1-score of 0.96 for recommendation classification and 0.94 for sentiment classification.

CCS CONCEPTS

• **Information systems → Data Analytics; Recommentation Analysis; Sentiment Analysis; Business Intelligence; Data Cleaning**; • **Computing Methodologies → Natural Language Processing; Supervised Learning by Classification; Neural Networks**.

KEYWORDS

e-commerce reviews, recurrent neural network, longshort term memory, sentiment analysis, recommendation analysis, artificial intelligence, artificial neural networks, supervised learning, natural language processing,classification, data analytics, data science, data visualization, deep learning

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Sharmin Hossain. 2023. Exploring E-Commerce Reviews on Amazon using Bidirectional Recurrent Neural Network (RNN) and Long-Short Term Memory (LSTM) For Sentiment and Recommendation Analysis. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Many companies have adopted social media monitoring as a valuable tool to gain insights into their customers' preferences and needs, which can help them refine and enhance their offerings. This has led to increased interest in text analysis, which is an active field of research in computational linguistics and natural language processing. One of the most prevalent issues in this field is text classification, which aims to classify documents into one or more categories either manually or computationally. In recent years, sentiment analysis has gained significant attention in this area, particularly in classifying sentiments of statements found in social media, review sites, and discussion groups. This involves using natural language processing techniques and statistics to identify and categorize opinions expressed in text, specifically, to determine the writer's attitude (positive, negative, or neutral) towards a product or topic. Companies now widely employ this process to understand their clients through customer support on social media or review boards online.

In my project, I analyzed customer reviews of Amazon's products specifically electronics products using statistical analysis and sentiment classification. I first analyzed the non-text review features to understand any potential connections between them and customer recommendations for the product. I then employed a bidirectional recurrent neural network (RNN) with long-short term memory (LSTM) to classify whether a review text recommends the purchased product and to classify the user's sentiment toward the product. I also compared this bidirectional analysis with unidirectional and tried to understand why the bidirectional approach worked better with this particular type of dataset of reviews.

2 METHODOLOGY

2.1 Machine Learning Library

For the data preprocessing and handling, the numpy[8] and pandas[10] python libraries were used. To implement bidirectional recurrent neural network with long-short term memory (LSTM)[9],I have used Keras[4] with Google Tensor Flow[1]. For data visualization, the matplotlib[7] and seaborn[6] python libraries were used.

2.2 The Dataset

This dataset consisted of 16,087 rows and 9 columns.The columns are Customer ID, Title, Review Text, Rating, Recommended IND, Helpful review count, Product Type, Primary Category & Category.

Column Name	Mean	Std	Min	Max
Rating	4.56	0.74	1	5
Recommended IND	0.95	0.21	0	1
Helpful Review Count	0.32	3.06	0	130
Label	0.97	0.15	0	1

Table 1: Statistical Summary of Some Columns

2.3 Data Analysis

2.3.1 Word Count Analysis. Review texts were split into words and added to a new column known as "Word Count" to see a general summary of reviewed text columns. Here the Total Word Count is 489374.

Count	16086
Mean	30.42
Std	39.81
Min	1
25%	14
50%	20
75%	35
Max	1539

Table 2: Summary of Word Count Columns

2.3.2 Frequency by Product Type and Primary Category. Here, frequencies grouped by product type and Primary category were shown to understand product-wise data frequencies in this dataset. We can also see how significantly the electronics category and Tablet product dominated in this dataset.

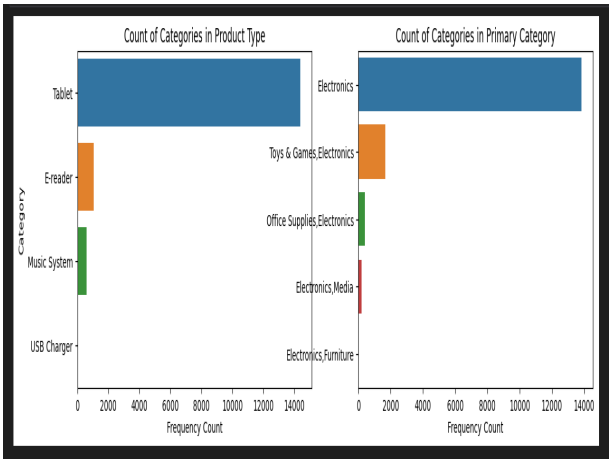

Figure 1: Frequencies by Product Type & Primary Category

2.3.3 Frequency by Rating, Recommended IND & Label. Label is basically another column which is taking the value 1 when the rating is greater or equal to 3 and 0 otherwise. Here, frequencies were grouped by rating, recommended IND and labels were given to see how biased the general dataset is in terms of recommended IND (1- people who recommended the product) and rating 4 & 5.

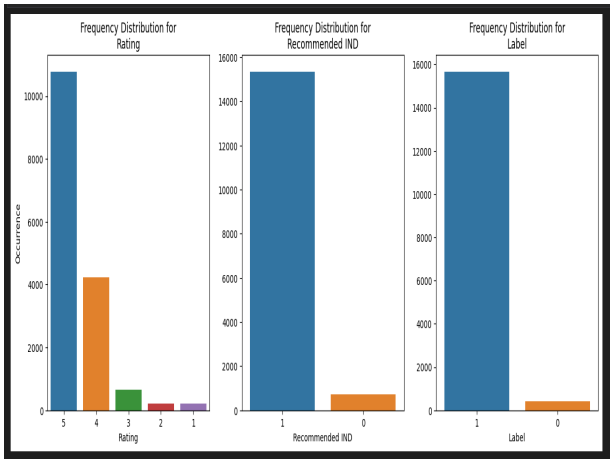

Figure 2: Frequencies by Rating, Recommended IND & Label

2.3.4 Percentage Frequency by Product Type of Rating & Recommended IND. Figure 3 exhibited consistent rating & recommended IND distribution across Product types.

2.3.5 Frequencies of Rating by Recommended IND. Figure 4 provided evidence supporting the hypothesis that the rating of a review aligns with its recommendation status. Specifically, a higher rating indicated a recommendation, while a lower rating indicated the absence of a recommendation.

2.3.6 Occurrence of Sentiment by Recommended IND. Sentiment processing was conducted using a threshold approach, where ratings higher than 3 were considered positive, while the remaining ratings were categorized as negative. But it had a problem identifying the neutral sentiments. That's why NLTK[5] was used. Figure 5 showed that positive sentiment had a much more percentage in recommending products than other sentiments.

2.3.7 Standardized Percentage of Rating &Sentiments. Figure 6 showed that a higher rating demonstrated more positive sentiments than others.

2.3.8 Correlation of All Variables. Based on Figure 7, it is evident that the recommended indicator and rating exhibited a stronger correlation compared to other variables. Additionally, there is a high correlation between the polarity score and the positive score.

2.3.9 Word Cloud For Review Titles. Figure 8 provided a word cloud displaying the most frequently used words in review titles. Among these words, only "Disappointed" appeared to indicate a negative sentiment. However, it is important to note that the presence of this word does not necessarily imply that the entire product review expresses a negative sentiment. It is worth mentioning that the

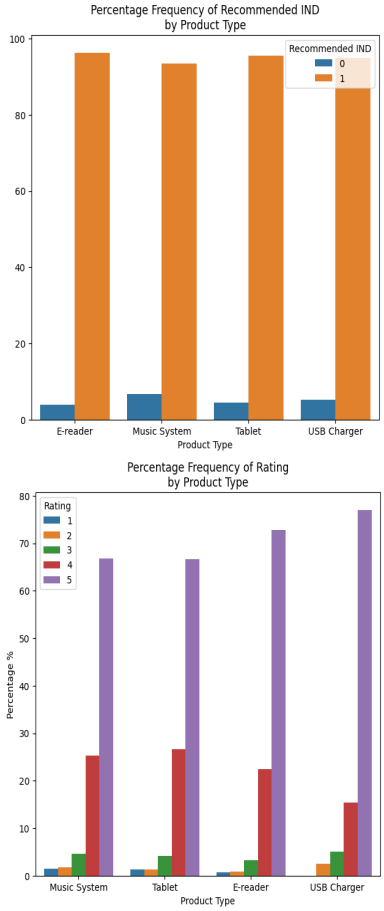

Figure 3: Percentage Frequency by Product Type of Recommended IND & Rating

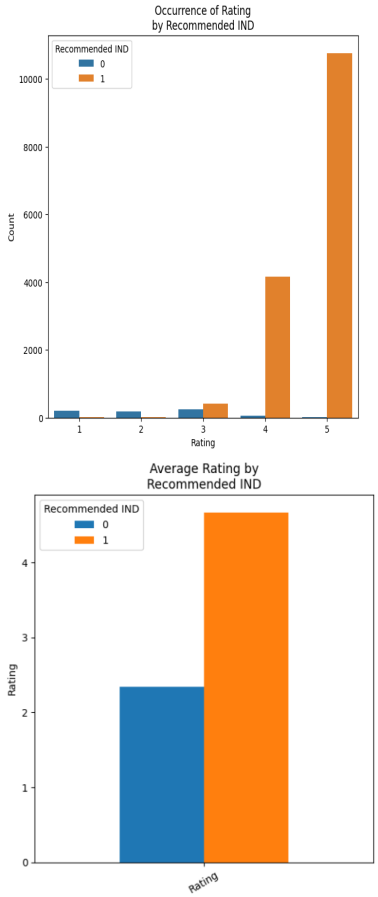

Figure 4: Frequencies of Rating by Recommended IND

word cloud solely considers the frequency of individual words in titles and does not account for phrases. Therefore, there may exist contrasting words that counteract negative word indicators but have not been included in the word cloud. Similarly, positive indicators in the word cloud may not include negators if they exist.

2.3.10 Most Frequent Words For Highly Rated Comments. Assuming that the words in Figure 9 reflected the content of their respective reviews, it can be inferred that this word cloud represented reviews with high ratings.

2.3.11 Most Frequent Words For Low Rated Comments. Given that Figure 10 represented a word cloud for reviews with low ratings, it can be presumed that the words depicted in this figure reflected the content of their corresponding reviews.

2.4 Dataset Processing

2.4.1 Text Cleaning. Text cleaning was performed on the user review texts to remove delimiters, such as "" and "", that were present in the texts.

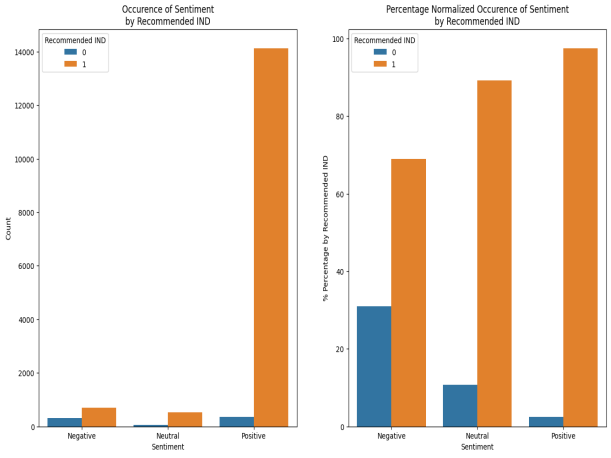

Figure 5: Occurrence of Sentiment by Recommended IND

2.4.2 Sentiment Analysis. To automate the process of tagging review texts, the sentiment analyzer from NLTK [5] was utilized

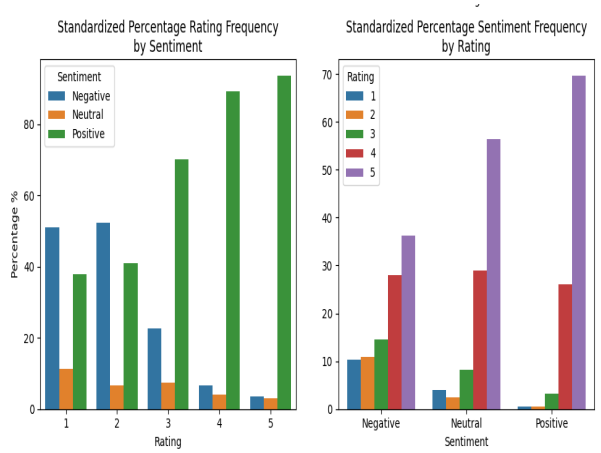

Figure 6: Standardized Percentage of Rating &Sentiments

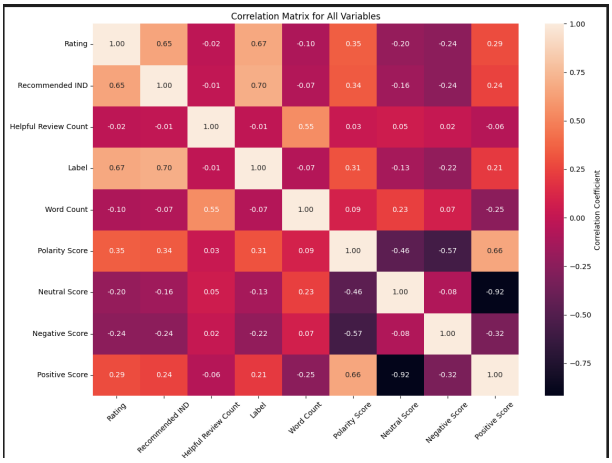

Figure 7: Correlation of All Variables

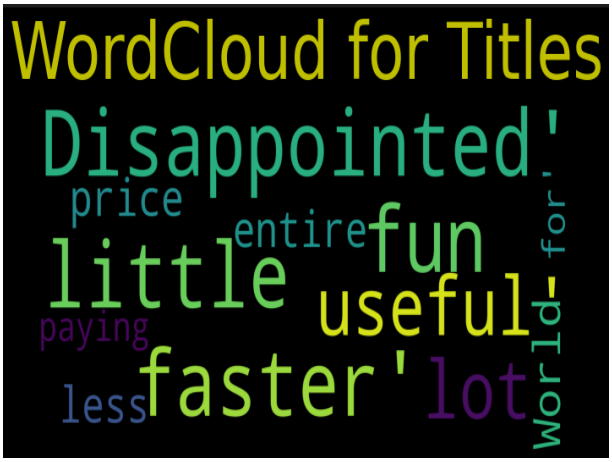

Figure 8: Word Cloud For Review Titles

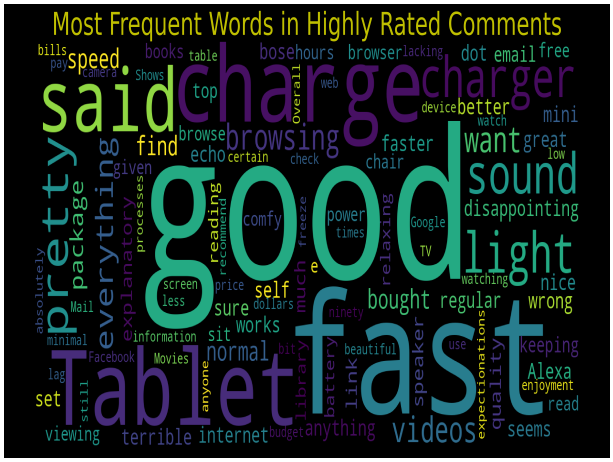

Figure 9: Most Frequent Words For Highly Rated Comments

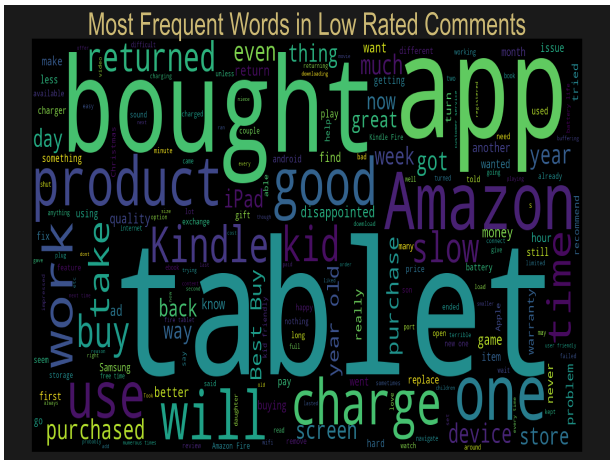

Figure 10: Most Frequent Words For Low Rated Comments

instead of manual tagging. This approach eliminated the previous intuitive tagging method that classified reviews with a rating threshold of 3 as positive feedback and the rest as negative feedback. The manual tagging approach had the drawback of not considering neutral sentiments. Therefore, the NLTK[5] sentiment analyzer was employed.

2.4.3 Word Embeddings. GloVe word embeddings[3] were employed to convert the words in the review texts into numerical vectors, enabling their representation in a vector space.

2.5 Machine Learning Models

A bidirectional neural network is a type of architecture that processes input data in two directions: forward and backward. It combines two separate recurrent neural networks (RNNs) to capture both past and future context. During the forward pass, the network processes the input sequence sequentially, capturing dependencies in the forward direction. In the backward pass, the input sequence is reversed, allowing the network to capture dependencies in the

reverse direction. The outputs from both passes are then combined to provide a more comprehensive understanding of the input, making bidirectional neural networks particularly effective for tasks that require context from both directions, such as natural language processing.

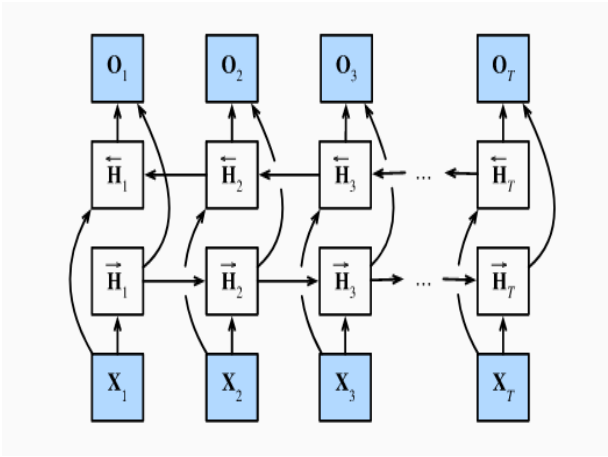

Figure 11: Bidirectional Recurrent Neural Network

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that overcomes the vanishing gradient problem in traditional RNNs. It is designed to capture long-term dependencies in sequential data. LSTMs utilize a memory cell that can retain information over long periods and selectively forget or update information based on input signals. The architecture consists of three main components: an input gate that regulates the flow of information into the memory cell, a forget gate that controls which information to discard, and an output gate that determines the output based on the memory cell's content. This unique design enables LSTMs to effectively model and process sequential data with long-range dependencies, making them well-suited for tasks like natural language processing and speech recognition.

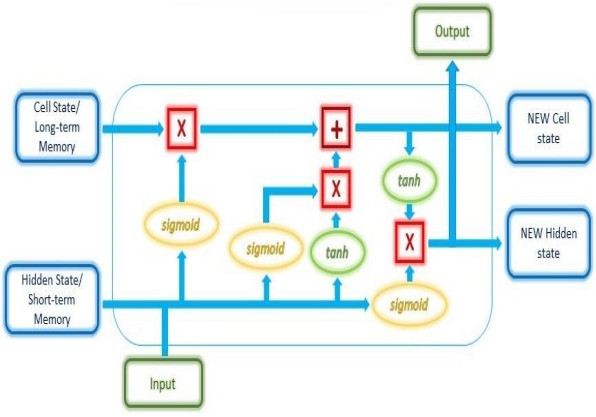

Figure 12: Long Short Term Memory Network

Below are the LSTM gate equations [2] which i have implemented using Google TensorFlow[1]

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * tanh(C_t)$$

In this context, the forget gate (denoted as 'f') serves the purpose of disregarding non-essential information for the model. The input gate ('i') accepts new data input at each time step ('s_t'). The candidate cell state value ('$\tilde{C}$') represents the potential cell state of each LSTM cell. The cell state ('C') is the value that gets passed on to the next RNN LSTM cell. The output gate ('o') determines what the cell state will output, and the output of the cell state value and the decided output is denoted as 'h'.

I applied this machine learning model to address two text classification tasks using a specific dataset. The first task is recommendation classification, where I determined whether a review text recommends the reviewed product. The second task is sentiment classification, where I determined the tone of the review text toward the purchased product.

2.5.1 Recommendation Classification. A product review can be categorized into two recommendation states: recommended or not recommended. This classification problem is binary in nature.

2.5.2 Sentiment Classification. A product review can be assigned one of three sentiment states: negative, neutral, or positive. This classification problem involves multiple categories.

3 RESULTS AND DISCUSSION

The dataset was divided into three parts following a 60/20/20 split, where 60% of the data was used for training, 20% for validation, and 20% for testing purposes. The hyper-parameters employed by the Bidirectional RNN-LSTM in the experiments are listed in Table 3. It is important to note that these hyper-parameters were chosen arbitrarily, as performing hyper-parameter tuning would require additional computational resources. The test accuracy and test loss of the Bidirectional RNN-LSTM for both recommendation classification and sentiment classification experiments are presented in Table 4.

Hyper-Parameter	Value
Batch Size	256
Cell Size	256
Dropout Rate	0.50
Epochs	10
Learning Rate	1e-3

Table 3: Hyperparameter Used in RNN-LSTM

Task	Test Accuracy	Loss
Recommendation Classification	0.960534	0.116165
Sentiment Classification	0.943132	0.165056

Table 4: Test Accuracy & Tess Loss Using Bidirectional LSTM

Nevertheless, it is important to consider that both the recommendation and sentiment classes exhibit imbalanced frequency distributions. Specifically, there are more instances of recommended classes compared to not recommended classes, and there are more positive sentiments compared to the combined count of negative and neutral sentiments. This creates a challenge as the model may develop a biased classification tendency towards the class with the highest frequency distribution. Therefore, it is necessary to examine the statistical report on recommendation classification provided in Table 5.

3.1 Recommendation Analysis

The predictive performance for the negative class in the recommendation classification problem is comparatively weaker, as indicated by Table 5 and further supported by the confusion matrix in Figure 13. In the confusion matrix, 0 represents the not recommended class, while 1 represents the recommended class. To evaluate the model's performance in a more equitable manner, I examined the ROC curve for the results, which is depicted in Figure 14. An ROC AUC (Area Under the Curve) score of 0.75 indicated that the model has a moderate level of discrimination ability to distinguish between the positive and negative classes. The model is performing better than random chance but there is still room for improvement.

	Precision	Recall	F1-Score	Support
(0) Not Recommended	0.60	0.51	0.55	154
(1) Recommended	0.98	0.98	0.98	3064
Accuracy			0.96	3218
Macro Avg	0.79	0.75	0.77	3218
Weighted Avg	0.96	0.96	0.96	3218

Table 5: Bidirectional_Recommendation Classification

3.2 Sentiment Analysis

Table 6 reinforced the earlier findings regarding the biased classification towards the class with the highest frequency distribution, which is supported by the confusion matrix shown in Figure 15. In this confusion matrix, 0 represents the negative class, 1 represents the neutral class, and 2 represents the positive class. The report indicated that the model exhibited relatively weaker predictive performance for the negative and neutral sentiments.

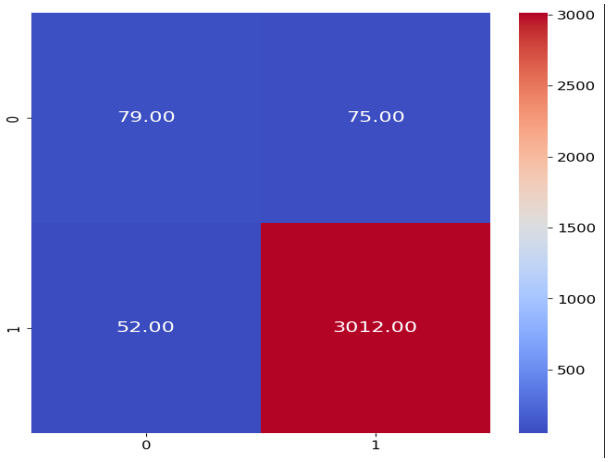

Figure 13: Confusion Matrix on Recommendation Classification

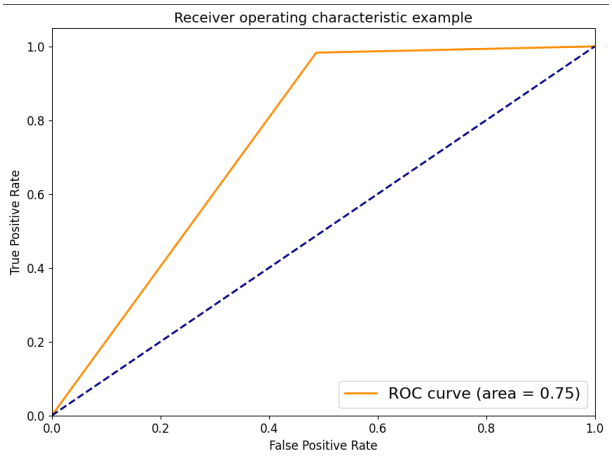

Figure 14: ROC Curve For Binary Classification on Recommendation Indicator

The empirical evidence presented in this paper demonstrates a relatively high-performing predictive performance for both recommendation classification and sentiment classification, despite the imbalanced class frequency distribution in the dataset. This outcome supports the assertion that utilizing the Bidirectional RNN-LSTM model better captures the context of review texts, resulting in improved predictive performance. However, to further validate this claim, I employed a unidirectional RNN-LSTM model for the same classification problems to facilitate a fair comparison.

	Precision	Recall	F1-Score	Support
(0) Negative Class	0.66	0.61	0.63	204
(1) Neutral Class	0.76	0.73	0.75	115
(2) Positive Class	0.97	0.97	0.97	2899
Accuracy			0.94	3218
Macro Avg	0.80	0.77	0.78	3218
Weighted Avg	0.94	0.94	0.94	3218

Table 6: Bidirectional_Sentiment Classification

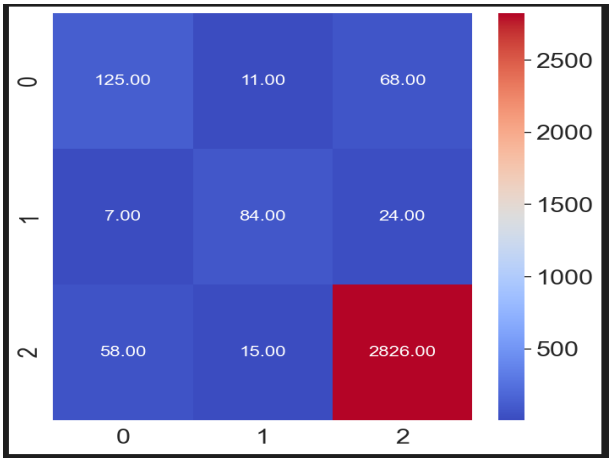

Figure 15: Confusion Matrix on Sentiment Classification

3.3 Comparison With Unidirectional RNN-LSTM

Based on the data presented in Table 7, it is clear that while the accuracy of recommendation and sentiment classification is comparable in both unidirectional and bidirectional RNN-LSTM models, the unidirectional model exhibits significantly higher loss.

The F-1 score also indicates that both unidirectional and bidirectional models performed well for recommendation classification. However, when it comes to sentiment classification, bidirectional RNN-LSTM models outperformed the others significantly. Also because of the imbalanced dataset, the calculation of precision were not trustworthy for unidirectional analysis.

Task	Test Accuracy	Loss
Recommendation Classification	0.954008	0.186537
Sentiment Classification	0.907085	0.370119

Table 7: Test Accuracy & Tess Loss Using Unidirectional LSTM

	Precision	Recall	F1-Score	Support
(0) Not Recommended	1.00	0.00	0.00	148
(1) Recommended	0.95	1.00	0.98	3070
Accuracy			0.95	3218
Macro Avg	0.98	0.50	0.49	3218
Weighted Avg	0.96	0.95	0.93	3218

Table 8: Unidirectional_Recommendation Classification

	Precision	Recall	F1-Score	Support
(0) Negative Class	1.00	0.00	0.00	200
(1) Neutral Class	1.00	0.00	0.00	99
(2) Positive Class	0.91	1.00	0.95	2919
Accuracy			0.91	3218
Macro Avg	0.97	0.33	0.32	3218
Weighted Avg	0.92	0.91	0.86	3218

Table 9: Unidirectional_Sentiment Classification

4 CONCLUSION

Although there were limitations in the experimental setup for this project, it can be deduced that the Bidirectional RNN-LSTM model demonstrated excellent performance, achieving an F1-score of 0.96 for recommendation classification and 0.94 for sentiment classification. Moreover, the statistical measures applied to the classification problem can be considered satisfactory.

Based on the dataset used, the results obtained from the bidirectional model were significantly superior and more reliable compared to the results obtained from the unidirectional model.

5 RECOMMENDATION

In order to enhance the effectiveness of the model, it is necessary to conduct hyper-parameter tuning. However, due to computational constraints, this project was limited to an arbitrary selection of hyper-parameters. While the bidirectional recurrent neural network with long short-term memory used in this project is a powerful tool for sentiment analysis, other models such as support vector machines or decision trees could be evaluated to determine if they provide better results. To further improve the model's accuracy, the dataset could be augmented with additional reviews or data from other sources. While the focus of this project was on accuracy, it would be useful to provide more insight into why the model makes certain predictions. Techniques such as attention mechanisms could be used to visualize the most important words or phrases in a review for a given sentiment classification. Moreover, to gain more comprehensive insights into the model's predictive performance, the implementation of k-fold cross-validation can be considered.

6 ACKNOWLEDGMENT

A sincere appreciation is given to the website data.world for the dataset of reviews of Amazon Products.

REFERENCES

[1] Google. 2022. *Introduction to TensorFlow.* https://developers.google.com/machine-learning/crash-course/first-steps-with-tensorflow/toolkit

[2] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[3] Christopher D. Manning Jeffrey Pennington, Richard Socher. 2014. *GloVe: Global Vectors for Word Representation.* https://nlp.stanford.edu/projects/glove/

[4] Keras. 2015. *Keras: Deep Learning for humans.* https://github.com/keras-team/keras

[5] Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics.* Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 63–70. https://doi.org/10.3115/1118108.1118117

[6] Shubham Singh. 2019. *Become a Data Visualization Whiz with this Comprehensive Guide to Seaborn in Python.* https://www.analyticsvidhya.com/blog/2019/09/comprehensive-data-visualization-guide-seaborn-python/

[7] Thetechwriters. 2021. *Introduction to Matplotlib using Python for Beginners.* https://www.analyticsvidhya.com/blog/2021/10/introduction-to-matplotlib-using-python-for-beginners/

[8] Stéfan van der Walt, S. Colbert, and Gael Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science Engineering* 13 (05 2011), 22 – 30. https://doi.org/10.1109/MCSE.2011.37

[9] Yugesh Verma. 2021. *Complete Guide To Bidirectional LSTM (With Python Codes).* https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/

[10] Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). 56 – 61. https://doi.org/10.25080/Majora-92bf1922-00a