IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Sentiment Analysis of Swiggy Customer Reviews Using Recurrent Neural Network and LSTM

Ms. Jyoti Mohansingh Tomar[1] and Ms. Asha Mane[2]
Sadhu Vaswani Institute of Management Studies for Girls, Pune[1-2]
Savitribai Phule Pune University
tomarjyoti883@gmail.com1 and ashamane90@gmail.com2

Abstract: *Sentiment analysis is a Natural Language Processing (NLP) task that identifies opinion polarity in text. This study applies recurrent neural networks to Swiggy mobile-application customer reviews to classify feedback as Positive or Negative. The dataset used in the final experiment contains 200,791 review records with review text and rating information. After cleaning, ratings greater than 3.5 are labelled Positive and ratings of 3.5 or below Negative. Text is tokenized, converted into integer sequences, and padded or truncated to 100 tokens. A SimpleRNN baseline is compared with a Long Short-Term Memory (LSTM) model using an 80:20 stratified train-test split. SimpleRNN achieved 92.14% test accuracy, while LSTM achieved 94.22% accuracy and 0.1666 test loss, with macro and weighted F1-scores of approximately 0.94. An interactive prediction interface was also developed for new reviews. The results suggest that LSTM performed better under the selected experimental configuration; however, labels derived from ratings are proxies for sentiment rather than manually verified annotations..*

Keywords: Sentiment analysis; Natural Language Processing; Swiggy; customer reviews; SimpleRNN; LSTM; deep learning.

I. INTRODUCTION

Online food-delivery services receive large volumes of user feedback about food quality, delivery time, pricing, app usability, and customer support. Reviews are useful for understanding satisfaction and recurring service problems, but manual analysis becomes difficult at scale. Sentiment analysis offers an automated way to classify text according to expressed polarity and can help summarize customer feedback.

Traditional sentiment-classification approaches include sentiment lexicons, bag-of-words features, n-grams, and supervised machine-learning algorithms. Neural approaches learn representations from data and can model word order. A Simple Recurrent Neural Network (SimpleRNN) processes a sequence one token at a time, while Long Short-Term Memory (LSTM) networks use gated memory mechanisms designed to preserve useful information over longer sequences.

This study compares SimpleRNN and LSTM for binary sentiment classification of Swiggy reviews. Both models use the same prepared dataset and test split, allowing a direct comparison of the reported results. The work also includes a reusable prediction pipeline and an interactive interface for entering new reviews.

1.1 Objectives and research problem

The research problem is to classify a large volume of customer-review text consistently and to evaluate whether an LSTM model improves performance over a simpler recurrent baseline. The objectives are to:

- Prepare and preprocess Swiggy review data for text classification.
- Generate binary sentiment labels from customer ratings.
- Implement SimpleRNN and LSTM models using a consistent data split.
- Compare accuracy, loss, precision, recall, and F1-score.

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-40337

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

• Build a small interface for predicting sentiment for a new review.

1.2 Scope

The study focuses on English-text review processing as implemented in the project notebook and binary sentiment classification. The rating is used to create labels; the rating itself is not a model input. The comparison is limited to SimpleRNN and LSTM, and the reported results reflect the selected preprocessing, model settings, and train-test split.

II. RELATED WORK

Sentiment analysis has developed from semantic-orientation and supervised text-classification methods to distributed word representations and neural sequence models. Turney [1] explored semantic orientation for review classification, while Pang and Lee [2, 3] examined supervised sentiment classification and rating-oriented sentiment categories. Go et al. [4] studied Twitter sentiment using distant supervision, and Maas et al. [5] connected learned word vectors with sentiment analysis.

Neural methods extended this work through recursive composition, word embeddings, convolutional sentence models, and attention. Socher et al. [6] introduced recursive modelling with the Stanford Sentiment Treebank. Mikolov et al. [7] proposed efficient word representations, and Pennington et al. [8] introduced GloVe. Kim [9] demonstrated convolutional neural networks for sentence classification. Attention-based approaches, including work on aspect-level sentiment [10], provide additional ways to focus on informative words or phrases. Transformer models such as BERT [11] later established contextual language representations that can be adapted to classification tasks.

The literature motivates a comparison of recurrent architectures on a domain-specific review dataset. In this project, SimpleRNN provides a lightweight baseline and LSTM provides a gated recurrent alternative. The experiment does not claim that these models outperform transformer-based approaches; rather, it compares the two selected architectures under the same project configuration.

2.1 Research gap

Customer reviews from a specific service domain provide a practical setting for evaluating sentiment models. This project addresses that setting by comparing a basic recurrent network and LSTM on the same Swiggy review dataset. Important challenges remain: ratings are imperfect proxies for textual sentiment, class distribution is not balanced, and informal language, negation, spelling variation, and mixed opinions may be difficult for basic models to interpret.

III. MATERIALS AND METHODS

3.1 Research design and dataset

The study follows an experimental, quantitative design. The dataset contains Swiggy mobile-application reviews collected from a Google Play Store dataset used in the project. The fields used are review_description, which contains customer-written text, and rating, which provides the label source. After removing records with missing review text or rating, 200,791 records were used in the final experiment.

Dataset field	Description	Use
review_description	Customer-written review text	Input to the text-processing pipeline
rating	Customer rating	Source for Positive/Negative label

Sentiment class	Rating rule	Label	Count	Share
Negative	Rating ≤ 3.5	0	119,681	59.60%
Positive	Rating > 3.5	1	81,110	40.40%
Total	—	—	200,791	100%

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

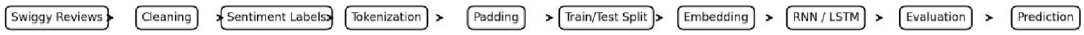

Figure 1. Overall workflow used for Swiggy review sentiment classification.

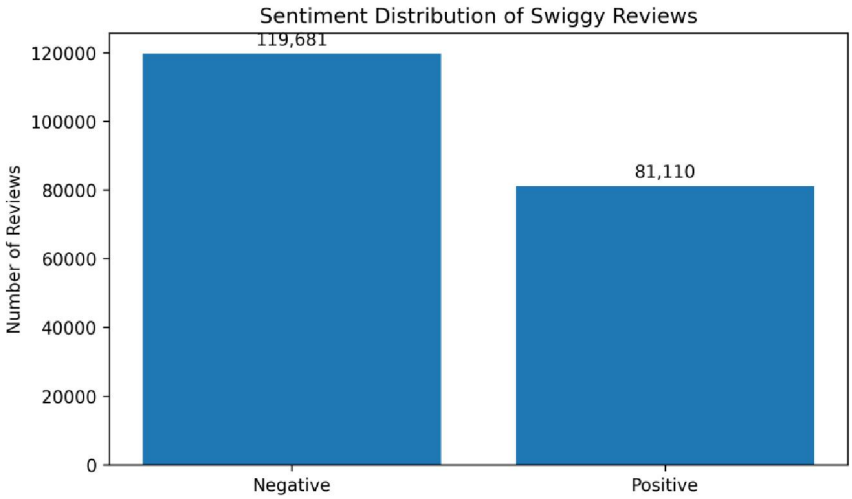

Figure 2. Distribution of rating-derived sentiment labels in the dataset.

3.2 Preprocessing and text representation

The review and rating fields are selected, rating values are converted to numeric form, and rows missing either field are removed. Review text is converted to lowercase and unnecessary special characters and punctuation are removed.

Sentiment labels are then generated using the threshold in Table 1. Because these labels originate from ratings, they may not always match the sentiment expressed in the written review.

A Keras Tokenizer converts words to integer token IDs, with a vocabulary limit of 10,000. Each review is represented as a sequence and padded with zeros or truncated to a maximum length of 100 tokens. The tokenizer is saved with the trained model so the same mapping can be used for later predictions.

3.3 Train-test split

The dataset is divided into training and testing subsets using an 80:20 split, random_state 42, and stratification. The final split contains 160,632 training records and 40,159 test records. Stratification preserves approximately the same class proportions across the two subsets

Subset	Records	Share
Training	160,632	80%
Testing	40,159	20%
Total	200,791	100%

.

3.4 Model architectures

Both networks receive token sequences of length 100. An embedding layer maps token IDs to dense vectors of dimension 64. The recurrent layer contains 64 units, followed by a one-unit Dense layer with sigmoid activation to produce the probability of the Positive class. The SimpleRNN layer serves as the baseline; the LSTM layer uses gated memory to model sequence information.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

Component	SimpleRNN configuration	LSTM configuration
Input	100-token sequence	100-token sequence
Embedding	64 dimensions	64 dimensions
Recurrent layer	SimpleRNN, 64 units	LSTM, 64 units
Output	Dense, 1 unit, sigmoid	Dense, 1 unit, sigmoid
Training	Adam; binary cross-entropy; 5 epochs	Adam; binary cross-entropy; 5 epochs; batch size 64

The reported LSTM configuration uses a vocabulary size of 10,000, one sigmoid output unit, and 673,089 trainable parameters. The project notebook records the trained models and tokenizer for reuse. Model comparisons are reported from the final experiment outputs supplied with the project.

3.5 Evaluation metrics

Accuracy is the fraction of all predictions that are correct. Precision measures the fraction of predicted instances of a class that are correct, while recall measures the fraction of actual instances of a class that are identified correctly. The F1-score is the harmonic mean of precision and recall. Test loss is reported using binary cross-entropy. The confusion matrix provides class-level counts of correct and incorrect predictions.

IV. RESULTS AND DISCUSSION

4.1 Overall model comparison

Under the reported configuration, LSTM achieved 94.22% test accuracy compared with 92.14% for SimpleRNN, an increase of 2.08 percentage points. LSTM test loss was 0.1666, compared with 0.2276 for SimpleRNN. The macro and weighted F1-scores were reported as approximately 0.94 for LSTM and 0.92 for SimpleRNN.

Metric	SimpleRNN	LSTM
Test accuracy	92.14%	94.22%
Test loss	0.2276	0.1666
Macro F1-score	0.92	0.94
Weighted F1-score	0.92	0.94

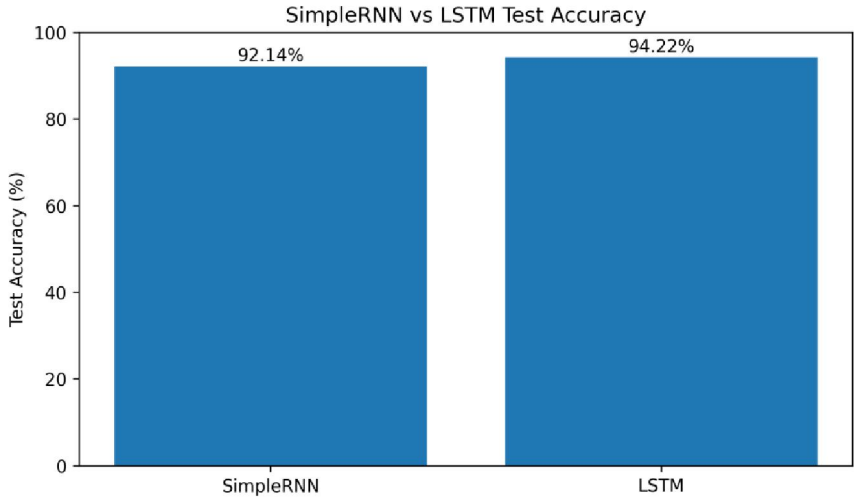

Figure 3. Test accuracy comparison reported for SimpleRNN and LSTM.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

The measured results indicate that the LSTM configuration performed better than the SimpleRNN baseline on this held-out test set. This outcome is consistent with the motivation for testing a gated recurrent architecture, but it should be interpreted within the limits of the experiment. The models were evaluated using rating-derived labels, and the results alone do not establish performance on manually annotated sentiment or on other datasets.

4.2 LSTM classification results

The LSTM classification report shows similar precision and recall for the two sentiment classes. The report gives a precision, recall, and F1-score of approximately 0.95 for Negative reviews and 0.93 for Positive reviews. The overall accuracy is 0.94 on 40,159 test records.

Class	Precision	Recall	F1-score	Support
Negative	0.95	0.95	0.95	23,937
Positive	0.93	0.93	0.93	16,222
Accuracy	—	—	0.94	40,159
Macro average	0.94	0.94	0.94	40,159
Weighted average	0.94	0.94	0.94	40,159

4.3 Confusion matrix

Actual / predicted	Negative	Positive
Negative	22,746	1,191
Positive	1,129	15,093

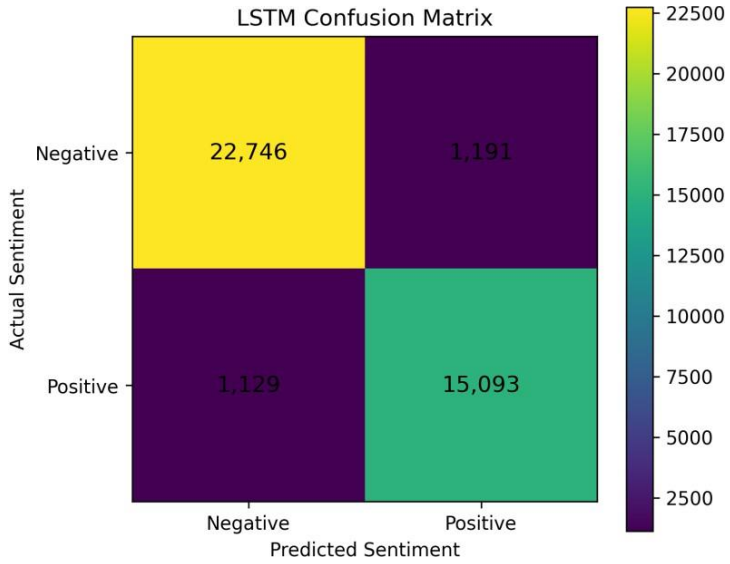

Figure 4. Confusion matrix for LSTM predictions on the test set.

The confusion matrix records 22,746 correctly predicted Negative reviews and 15,093 correctly predicted Positive reviews. It also records 1,191 Negative reviews classified as Positive and 1,129 Positive reviews classified as Negative. The diagonal values represent correct classifications, while off-diagonal values represent errors. These counts sum to the 40,159 records in the test set.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

4.4 Interactive prediction interface

An interactive sentiment analyzer was implemented using IPyWidgets in Google Colab. A user enters a review, after which the text is lowercased and cleaned, converted into token IDs by the saved tokenizer, and padded or truncated to length 100. The LSTM returns a probability; values of 0.5 or above are interpreted as Positive, and lower values as Negative. The interface displays the predicted class and confidence. This interface demonstrates how the trained model can be used for new text, although its confidence should not be treated as a calibrated probability without further validation.

V. CONCLUSION

This study developed a binary sentiment-classification pipeline for 200,791 Swiggy customer reviews using SimpleRNN and LSTM architectures. The process included data cleaning, rating-based label generation, tokenization, sequence padding, stratified splitting, model training, evaluation, and an interactive prediction interface. SimpleRNN achieved 92.14% test accuracy, while LSTM achieved 94.22% accuracy and 0.1666 test loss. The reported results show stronger performance for LSTM within this experiment.

The study illustrates the practical use of recurrent neural models for summarizing customer feedback. The findings should be interpreted carefully because star ratings are proxies for sentiment, the chosen threshold influences the class distribution, and preprocessing removes punctuation and special characters. The experiment is also limited to two recurrent architectures and a binary classification task.

5.1 Limitations and future work

- Validate a sample of labels through manual annotation and test alternative rating thresholds.
- Examine errors involving negation, spelling variation, emojis, informal language, and mixed sentiment.
- Compare Bidirectional LSTM and GRU models, attention-based recurrent networks, and transformer models such as BERT.
- Explore multilingual and code-mixed reviews and aspect-level sentiment for delivery, food quality, pricing, and customer service.
- Evaluate generalization on an independent dataset and report repeat-run variability.

REFERENCES

[1] Turney, P. D. (2002). Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews.
[2] Pang, B., & Lee, L. (2004). A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts.
[3] Pang, B., & Lee, L. (2005). Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales.
[4] Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision.
[5] Maas, A. L., et al. (2011). Learning Word Vectors for Sentiment Analysis.
[6] Socher, R., et al. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.
[7] Mikolov, T., et al. (2013). Efficient Estimation of Word Representations in Vector Space.
[8] Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global Vectors for Word Representation.
[9] Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification.
[10] Wang, Y., et al. (2016). Attention-based LSTM for Aspect-level Sentiment Classification.
[11] Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
[12] Vaswani, A., et al. (2017). Attention Is All You Need.
[13] Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification.
[14] Zhang, W., Wang, Y., & Liu, X. (2018). Deep Learning for Sentiment Analysis: A Survey.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, October 2026
Impact Factor: 8.2

[15] Wang, Y., et al. (2018). Sentiment Analysis by Capsules.
[16] He, Y., et al. (2018). Effective Attention Modeling for Aspect-Level Sentiment Classification.
[17] Liu, Y., et al. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.
[18] Zhang, W., et al. (2024). Sentiment Analysis in the Era of Large Language Models: A Reality Check.
[19] Kastrati, Z., et al. (2024). Leveraging Distant Supervision and Deep Learning for Twitter Sentiment and Emotion Classification.