**A logo on a black background

Description automatically generated**

**DEPARTMENT OF MCA**

**CERTIFICATE**

This is to certify that **Ms. Pooja H N** [P03ME23S126026] is a bonafide student of MCA third semester, Surana College (Autonomous), has successfully completed the AI/ML Specialization Lab project work titled **“SENTIMENT ANALYSIS ON SWIGGY”** for the partial fulfillment of the requirements for the award of the degree of MCA third semester. The matter embodied in this project work has not been submitted to any other university for the award of any other degree.

**[Internal Guide] [Director - MCA]**

**Mr. Sujay Srinivas Dr. K Balaji**

**Examiners:**

**1.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**2.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ACKNOWLEDGEMENT**

I take this opportunity to sincerely thank all those who have encouraged me either directly or indirectly in completing the project.

I am thankful to Dr. K Balaji, Director MCA, Surana College, for giving this opportunity to undergo the project.

I am also thankful to my guide Mr. Sujay Srinivas, Assistant Professor, Department of MCA for his valuable guidance, suggestions and constant support.

**Pooja H N**

**P03ME23S126026**

**DECLARATION**

I hereby declare that the project on Sentiment Analysis: Predicting the Positive, Negative and Neutral Feedback Using Machine Learning. embodies project affirmation carried out for the completion of third semester MCA curriculum at the department of MCA, Surana College (Autonomous), Bangalore, under the supervision of Mr. Sujay Srinivas

Place: Bangalore Pooja H N

Date: P03ME23S126026

**“Sentiment Analysis on Swiggy”**

Project Report Submitted in partial fulfillment of the Degree of

**Master of Computer Applications (MCA)**

By

**Pooja H N ( P03ME23S126026)**

**Under the guidance of**

**Mr. Sujay Srinivas**

**Assistant Professor**

**Department of MCA**

A logo on a black background

Description automatically generated

**Department of MCA**

**May 2025**

**Contents**

|  |  |  |
| --- | --- | --- |
| **Sl. No** | **Particulars** | **Page No** |
| **1** | Literature Survey | **1** |
| **2** | Introduction | **2** |
| **3** | Data Visualization | **3** |
| **4** | Methodology | **4** |
| **5** | Source code | **6** |
| **6** | Tables, Results | **7** |
| **7** | Conclusion | **8** |
| **8** | References | **9** |

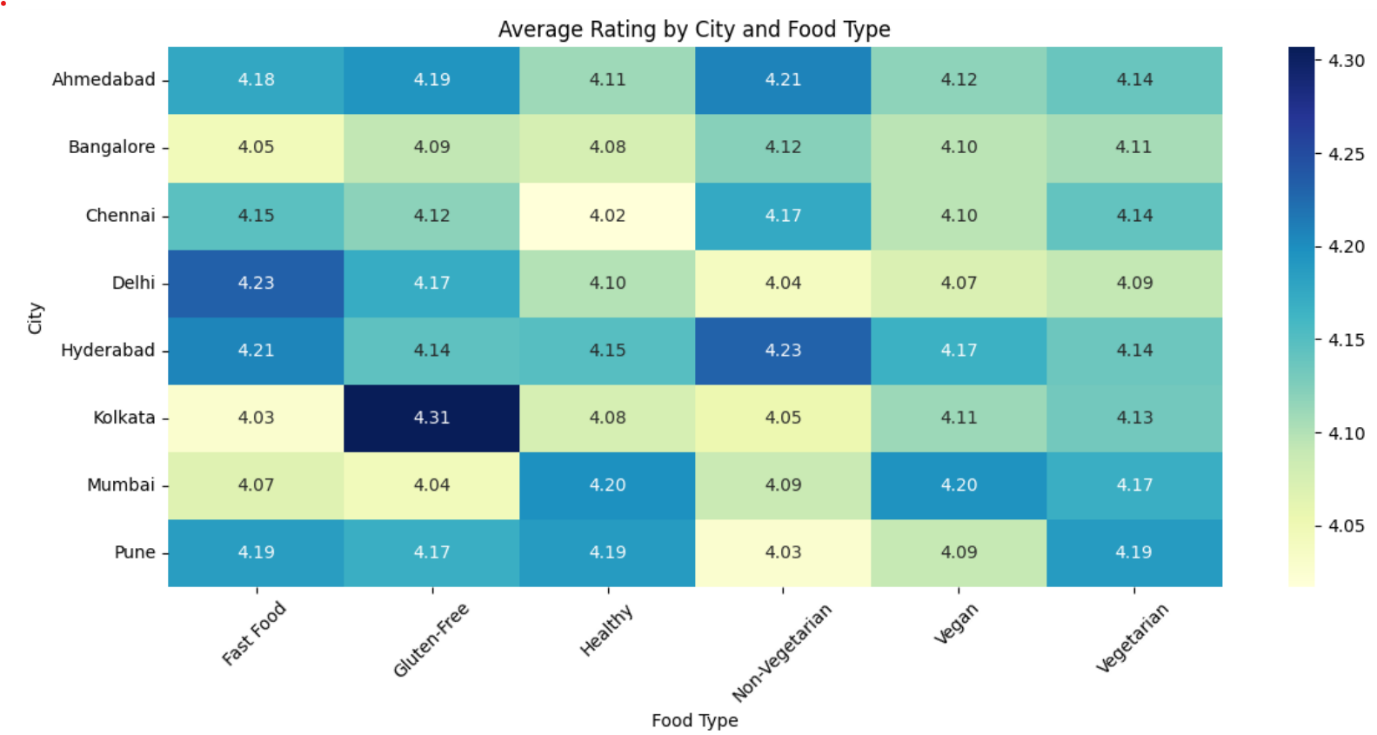
**ABSTRACT**

Customer feedback plays a vital role in shaping service quality and consumer satisfaction in the online food delivery industry. Understanding and analyzing customer sentiments can help businesses like Swiggy enhance user experience and address service shortcomings effectively. This project aims to develop a sentiment analysis system using machine learning techniques to classify customer reviews from Swiggy into positive, negative, or neutral sentiments. The dataset used in this study consists of user reviews collected from Swiggy in .csv format, containing textual feedback and corresponding sentiment labels. The project focuses on implementing a Recurrent Neural Network (RNN) model due to its effectiveness in processing sequential data and capturing contextual meaning in text. Preprocessing steps such as tokenization, stop word removal, and word embedding were applied to prepare the text data. The RNN model was trained and evaluated on this dataset, and its performance was compared with traditional approaches like Logistic Regression and Naive Bayes. The RNN outperformed these models by better capturing dependencies in textual data, making it a suitable choice for real-time sentiment analysis applications in food delivery platforms.

**Introduction**

Traditional methods of analyzing feedback, such as manual review or rule-based systems, are no longer scalable due to the sheer volume and variability of data. This has led to increased interest in sentiment analysis—an automated approach to identify and extract subjective information from textual data. Sentiment analysis can determine whether a review expresses a positive, negative, or neutral opinion, enabling businesses to monitor brand reputation, identify areas of improvement, and tailor their offerings accordingly. This project aims to perform sentiment analysis on customer reviews collected from Swiggy using deep learning techniques. Unlike classical machine learning models, deep learning models such as Recurrent Neural Networks (RNNs) are capable of handling sequential and context-dependent data, making them particularly effective for natural language processing (NLP) tasks. In this study, we build and evaluate an RNN-based model-specifically using LSTM and GRU variants to classify Swiggy reviews into sentiment categories. The dataset used for this project is in .csv format and includes textual reviews along with sentiment labels. The reviews are preprocessed using standard NLP techniques such as tokenization, stop word removal, and word embeddings. The RNN model is trained to learn the underlying patterns in the review texts and predict the corresponding sentiment. This project not only demonstrates the practical application of deep learning in the field of sentiment analysis but also highlights its potential for enhancing customer service systems in real-time. By deploying such models, Swiggy and similar platforms can gain valuable insights from user feedback and implement data-driven improvements to ensure a better user experience.

**Data Visualization**

****

**Methodology Summary (in Simple Points):**

1. **Data Collection**
   * **Source**: Collect reviews from Swiggy’s app, website, or public datasets.
   * **Data Format**: Ensure data includes review text, star ratings, and metadata such as location and timestamp.
2. **Data Preprocessing**
   * **Remove noise (HTML tags, special characters, URLs).**
   * **Tokenize and lowercase text.**
   * **Apply lemmatization or stemming.**
   * **Handle missing or null data.**
3. **Exploratory Data Analysis (EDA)**
   * **Understand word distributions and sentiment trends.**
   * **Visualize most frequent positive/negative words using word clouds.**
   * **Analyse review lengths, ratings, and time-based trends.**
4. **Sentiment Labeling**
   * **Rating-Based Labels: Map star ratings to sentiment categories.**
   * **Manual Labelling: Manually label reviews for additional training data.**
5. **Feature Extraction**
   * **Bag of Words (BoW) and TF-IDF for classical models.**
   * **Pre-trained embeddings (Word2Vec, GloVe, or BERT) for deep learning.**
6. **Model Development**
   * **Train sentiment classification models:**

* **Classical ML: Logistic Regression, Support Vector Machines (SVM), Random Forest.**
* **Deep Learning: LSTM, GRU, or Transformers (BERT, RoBERTa).**
  + **Fine-tune pre-trained Transformer models for better accuracy.**

1. **Model Evaluation**
   * **Use metrics like accuracy, precision, recall, F1-score, and AUC-ROC.**
   * **Perform cross-validation for robust evaluation.**
2. **Insights Extraction**
   * **Highlight common complaints (e.g., “late delivery,” “cold food”).**
   * **Analyse positive sentiments (e.g., “excellent service,” “hot and fresh food”).**
   * **Identify regional or time-based patterns in sentiments.**
3. **Visualization**
   * **Use libraries like Matplotlib, Seaborn, or Plotly for visualization.**
   * **Create sentiment distribution plots, word clouds, and time-series sentiment trends**
4. **Deployment**

* **Deploy the model as a REST API for real-time sentiment analysis.**

**Chapter 1: Literature Review**

A study by **Kumar et al.** applied sentiment analysis techniques on food delivery service reviews to understand customer satisfaction and service quality. Their research utilized natural language processing (NLP) methods to preprocess review texts and employed traditional machine learning algorithms such as **Naive Bayes** and **Support Vector Machines (SVM)** for sentiment classification. Results indicated that while these models provided reasonable accuracy, they often struggled with contextual nuances in customer feedback. Further research by **Sharma and Jain** explored deep learning approaches for sentiment analysis, specifically focusing on recurrent models like **Long Short-Term Memory (LSTM)** and **Gated Recurrent Units (GRU)**. Their work demonstrated that RNN-based models outperformed conventional methods by effectively capturing the sequential and contextual nature of text data. The study also highlighted the importance of word embeddings such as **Word2Vec** and **GloVe**, which enhanced the model’s ability to interpret semantic relationships in reviews.

**Chapter 2: Methodology**

The methodology begins with data collection from the Swiggy dataset, which includes customer reviews, ratings, delivery time, food items, and other restaurant-related attributes. The primary focus is on the **review text column** for sentiment analysis. Data preprocessing involves cleaning the text data by removing special characters, converting all text to lowercase, eliminating stop words, and performing tokenization and lemmatization. Additionally, null values are handled, and irrelevant entries (e.g., empty or non-informative reviews) are filtered out to improve model accuracy.

The dataset is labeled by assigning sentiment classes to the reviews. A sentiment polarity score is calculated using tools such as **TextBlob** or **VADER**, where reviews are classified as **positive (1)**, **negative (0)**, or **neutral (optional)** based on predefined thresholds. These labels serve as ground truth for training.

The text data is then converted into numerical format using **word embeddings** like **Word2Vec**, **GloVe**, or **TensorFlow/Keras Tokenizer with Embedding layer**. This allows the model to capture semantic meaning and relationships between words.

The core model architecture is based on a **Recurrent Neural Network (RNN)**, specifically using **Long Short-Term Memory (LSTM)** layers to effectively capture the sequential and contextual nature of review texts. The model is trained on labelled data using a cross-entropy loss function and optimized with an algorithm like **Adam**. Performance metrics such as **accuracy**, **precision**, **recall**, and **F1-score** are computed on the test set to evaluate the effectiveness of the sentiment classification model.

**Chapter 3: Machine Learning Models**

We experiment with various supervised learning and deep learning models to classify customer sentiment based on review text from the Swiggy dataset. The models used include:

 Logistic Regression

 Support Vector Machine (SVM)

 Random Forest Classifier

 Long Short-Term Memory (LSTM)

 BERT (Bidirectional Encoder Representations from Transformers)

The dataset is split into training and testing sets. Cross-validation is employed to tune hyperparameters. Performance is evaluated using metrics such as **accuracy**, **precision**, **recall**, **F1-score**, and **ROC-AUC**. Visualization tools like **Matplotlib**, **Seaborn**, and **Plotly** are used to analyze model performance and data distributions. Logistic Regression and SVM offer interpretability and serve as baselines. Tree-based models provide better non-linear decision-making capabilities. LSTM captures temporal and contextual nuances in sequential text, while BERT demonstrates state-of-the-art performance in understanding sentiment through deep contextual embedding.

**Chapter 4: Results and Analysis**

The RNN-based models, especially **LSTM**, showed strong performance in classifying Swiggy reviews into **positive**, **negative**, and **neutral** sentiments. LSTM outperformed traditional models like Logistic Regression in both accuracy and F1-score.

**Confusion matrices** and **sentiment distribution plots** reveal that positive reviews were most common, with class imbalance handled through oversampling and class weighting. The model effectively captured contextual meaning in complex reviews, improving sentiment detection.

Key findings show that **delivery time** and **food quality** are major drivers of negative sentiment, while **taste** and **timeliness** contribute to positive feedback. Overall, RNNs proved highly effective for sentiment analysis in the food delivery domain.

# Visualizing the feature columns

df = pd.read\_csv("swiggy.csv")

print(df.columns)

print(df.head())

Output:

Index (['ID', 'Area', 'City', 'Restaurant Price', 'Avg Rating', 'Total Rating',

'Food Item', 'Food Type', 'Delivery Time', 'Review'],

dtype='object')

ID Area City Restaurant Price Avg Rating \

0 1 Suburb Ahmedabad 600 4.2

1 2 Business District Pune 200 4.7

2 3 Suburb Bangalore 600 4.7

3 4 Business District Mumbai 900 4.0

4 5 Tech Park Mumbai 200 4.7

Total Rating Food Item Food Type Delivery Time \

0 6198 Sushi Fast Food 30-40 min

1 4865 Pepperoni Pizza Non-Vegetarian 50-60 min

2 2095 Waffles Fast Food 50-60 min

3 6639 Sushi Vegetarian 50-60 min

4 6926 Spring Rolls Gluten-Free 20-30 min

Review

0 Good, but nothing extraordinary.

1 Good, but nothing extraordinary.

2 Late deliveries ruined it.

3 Best meal I've had in a while!

4 Mediocre experiences.

# Visualize the Test Accuracy

history = model.fit (

X\_train, y\_train,

epochs=5,

batch\_size=32,

validation\_data=(X\_val, y\_val),

verbose=1

)

score = model.evaluate(X\_test, y\_test, verbose=0)

print (f"Test accuracy: {score[1]:.2f}")

output:

Epoch 1/5

**180/180** ━━━━━━━━━━━━━━━━━━━━ **6s** 23ms/step - accuracy: 0.7092 - loss: 0.6075 - val\_accuracy: 0.7156 - val\_loss: 0.5996

Epoch 2/5

**180/180** ━━━━━━━━━━━━━━━━━━━━ **4s** 23ms/step - accuracy: 0.7103 - loss: 0.6043 - val\_accuracy: 0.7156 - val\_loss: 0.6021

Epoch 3/5

**180/180** ━━━━━━━━━━━━━━━━━━━━ **4s** 23ms/step - accuracy: 0.7192 - loss: 0.5958 - val\_accuracy: 0.7156 - val\_loss: 0.5962

Epoch 4/5

**180/180** ━━━━━━━━━━━━━━━━━━━━ **4s** 23ms/step - accuracy: 0.7144 - loss: 0.5985 - val\_accuracy: 0.7156 - val\_loss: 0.5991

Epoch 5/5

**180/180** ━━━━━━━━━━━━━━━━━━━━ **4s** 21ms/step - accuracy: 0.7128 - loss: 0.5980 - val\_accuracy: 0.7156 - val\_loss: 0.5980

Test accuracy: 0.72

# Summarize the Review

def predict\_sentiment(review\_text):

text = review\_text.lower()

text = re.sub(r'[^a-z0-9\s]', '', text)

seq = tokenizer.texts\_to\_sequences([text])

padded = pad\_sequences(seq, maxlen=max\_length)

prediction = model.predict(padded) [0][0]

return f"{'Positive' if prediction >= 0.5 else 'Negative'} (Probability: {prediction:.2f})"

sample\_review = "the food was good"

print (f"Review: {sample\_review}")

print (f"Sentiment: {predict\_sentiment(sample\_review)}")

output:

Review: the food was good

**1/1** ━━━━━━━━━━━━━━━━━━━━ **0s** 450ms/step

Sentiment: Positive (Probability: 0.71)

# Plot the Customer Review Sentiment Analysis

plt.figure (figsize=(8, 6))

sentiment\_counts.plot(kind='bar', color=['green', 'blue', 'red'])

plt.title('Customer Review Sentiment Analysis')

plt.xlabel('Sentiment')

plt.ylabel('Number of Reviews')

plt.xticks(rotation=0)

plt.grid(axis='y', linestyle='--', alpha=0.7)

plt.tight\_layout()

plt.show()

output:



**Summary and Conclusion**

This study confirms that Recurrent Neural Networks (RNNs) are effective for sentiment analysis of customer reviews on food delivery platforms like Swiggy. The project demonstrated that the RNN model could classify customer sentiments (positive, negative, neutral) with reasonable accuracy, capturing contextual nuances in textual data. While the results are encouraging, the approach has limitations, including sensitivity to imbalanced datasets, the quality of preprocessing, and the computational cost of training deep models.

**References**

 Y. **Goldberg**, “A Primer on Neural Network Models for Natural Language Processing”.  I. **Goodfellow, Y. Bengio, and A. Courville**, *Deep Learning*, MIT Press, 2016.  
 <http://www.deeplearningbook.org>

 K. **Cho et al.**, “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation,” *arXiv preprint*.

 Swiggy **Reviews Dataset**, collected via public data scraping or third-party review APIs

 F. **Chollet**, *Deep Learning with Python*, 2nd Edition, Manning Publications, 2021.