Green University of Bangladesh

Department of Computer Science and Engineering (CSE)
Semester: (Fall, Year: 2025), B.Sc. in CSE (Day)

Restaurant Recommendation System

Course Title: Artificial Intelligence Lab
Course Code: CSE 316
Section: D3 (223)

Students Details

Name	ID
ANISUR RAHAMAN MARUF	222902078
SAKIBUL HASAN	222902083

Submission Date: December 29, 2025
Course Teacher's Name: Mr. Mozdaher Abdul Quader

[For teachers use only: Don't write anything inside this box]

Lab Project Status	
Marks:	**Signature:**
Comments:	**Date:**

Contents

1 Overview **3**

1.1 Abstract 3
1.2 Introduction 3
1.3 Project Overview 4
1.4 Motivation 4
1.5 Objectives 4

2 System Design **5**

2.1 Dataset Description 5
2.2 Feature Information 5
2.3 Data Statistics 5
2.4 Data Preprocessing and Cleaning 6
2.5 Exploratory Data Analysis (EDA) 6
2.6 Visualizing Distribution of Ratings 6
2.7 City-wise Restaurant Distribution 6
2.8 Sentiment Distribution (Positive, Negative, Neutral) 7
2.9 Correlation and Data Insights 7

3 Workflow **8**

3.1 Proposed Methodology 8
3.2 System Architecture 8
3.3 Sentiment Analysis using BERT 8
3.4 Content-Based Filtering 9
3.5 Similarity Measurement and Ranking 9

4 Implementation of the Project **10**

4.1 Implementation Details 10
4.2 Development Environment and Tools 10

4.3 Technology Stack (Libraries) 10
4.4 Step-by-Step Execution Logic 11

5 Results and Discussion **14**
5.1 Analysis of the Recommendation Output 14
5.2 System Authentication and Data Access 14
5.3 Initial Data Loading and Preprocessing Results 14
5.4 Sentiment Analysis Implementation 15
5.5 Distribution of Restaurant Ratings 15
5.6 Top 10 Restaurant List 16
5.7 Sentiment Distribution 16
5.8 Most Famous Restaurants 17
5.9 Recommend Restaurant by Locations 17

6 Challenges Limitations **18**
6.1 Computational Complexity of BERT 18
6.1.1 Data Sparsity and Missing Values 18
6.1.2 Cold Start Problem 18
6.1.3 Language Nuances in Reviews 18

7 Conclusion **19**
7.0.1 Conclusion 19
7.0.2 Future Work 19

Chapter 1

Overview

1.1 Abstract

This project presents an intelligent Restaurant Recommendation System developed using a comprehensive Foodpanda dataset. The primary goal of the system is to enhance the user experience by providing personalized restaurant suggestions based on geographical relevance and customer sentiment.
The methodology integrates advanced Natural Language Processing (NLP) techniques, specifically utilizing the **BERT (Bidirectional Encoder Representations from Transformers)** model to perform high-accuracy sentiment analysis on user reviews. By analyzing the sentiment scores, the system can distinguish between restaurants that are highly favored by customers and those with less satisfactory feedback.
Furthermore, the system employs Content-based Filtering powered by Cosine Similarity to match user preferences with restaurant attributes such as location and city. The implementation demonstrates an effective way to handle large-scale food delivery data, offering a robust solution for location-based and quality-driven restaurant discovery. The results indicate that combining transformer-based sentiment analysis with similarity metrics significantly improves the precision of recommendations compared to traditional rating-based methods.

1.2 Introduction

The rapid growth of online food delivery platforms has transformed how consumers interact with the culinary industry. With thousands of dining options available at their fingertips, users often face "choice paralysis," making it difficult to select a restaurant that aligns with their specific tastes, location, and quality expectations. To address this, automated recommendation systems have become essential tools for platforms like Foodpanda to provide personalized and relevant suggestions.

1.3 Project Overview

This project focuses on building a specialized Restaurant Recommendation System using a real-world dataset from Foodpanda. The system is designed to analyze restaurant data, including locations, cities, and customer reviews, to recommend the best dining options. A key highlight of this implementation is the integration of **BERT (Bidirectional Encoder Representations from Transformers)** for sentiment analysis, allowing the system to understand the nuances of customer feedback and prioritize restaurants with genuine positive sentiments.

1.4 Motivation

The motivation behind this project stems from the need for more qualitative recommendation engines. Most traditional systems rely solely on numerical ratings, which can sometimes be misleading. By incorporating sentiment analysis from textual reviews, we can capture a more accurate representation of customer satisfaction. Furthermore, providing location-specific recommendations ensures that the suggestions are practically useful for the user.

1.5 Objectives

The core objectives of this project are:

- To perform **Exploratory Data Analysis (EDA)** on the Foodpanda dataset to identify trends in restaurant distributions and ratings.
- To implement a **Sentiment Analysis** pipeline using the BERT model to categorize user reviews into positive, negative, and neutral sentiments.
- To develop a **Content-based Filtering** mechanism using Cosine Similarity to recommend restaurants based on geographical similarity (City and Location).
- To create a **Hybrid Scoring System** that combines location relevance with sentiment scores to rank the top recommended restaurants.

Chapter 2

System Design

2.1 Dataset Description

The dataset used in this project is sourced from **Foodpanda**, one of the leading online food delivery platforms. It contains detailed information about various restaurants, their locations, and customer experiences, which serves as the foundation for the recommendation engine.

2.2 Feature Information

The dataset consists of several key features that are essential for analyzing restaurant characteristics and user preferences:

- **Restaurant Name:** The unique identity of each dining establishment.
- **City:** The geographical city where the restaurant is located (e.g., Dhaka, Chittagong).
- **Location:** Specific area or neighborhood information within the city.
- **Rating:** Numerical feedback provided by users, reflecting the general popularity and quality of the restaurant.
- **Reviews/Comments:** Textual feedback from customers, which is utilized for advanced sentiment analysis using the BERT model.

2.3 Data Statistics

Before processing, the dataset underwent a structural analysis to understand its dimensions:

- `Total Entries:` – The dataset contains a significant number of restaurant records and user reviews to ensure diverse recommendations.

- `Data Consistency:` – Initial checks were performed to identify the number of unique cities and the distribution of restaurants across different regions.

2.4 Data Preprocessing and Cleaning

To ensure the accuracy of the recommendation system, the raw dataset was cleaned and prepared:

- **Handling Missing Values:** Null values in critical columns such as 'Ratings' or 'Location' were identified and either removed or imputed to maintain data integrity.
- **Text Normalization:** Review texts were preprocessed by removing special characters and standardizing the format for the BERT tokenizer.
- **Feature Engineering:** A composite feature was created by combining 'Location' and 'City' to facilitate more precise geographical matching during the recommendation process.

2.5 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) plays a critical role in understanding the underlying patterns and distributions within the Foodpanda dataset. This phase involves statistical summaries and visual representations to gain insights into restaurant popularity, geographical trends, and customer sentiment

2.6 Visualizing Distribution of Ratings

An analysis of the 'Rating' feature was conducted to understand the general quality of services provided by the restaurants. By plotting a distribution of ratings, we observed the concentration of high-performing versus low-performing dining establishments. This visualization helps in identifying whether the majority of restaurants maintain a standard rating or if there are significant outliers that could affect recommendation accuracy.

2.7 City-wise Restaurant Distribution

To understand the geographical spread of the data, a distribution analysis was performed based on the 'City' attribute. This visualization highlights the cities with the highest density of restaurants (e.g., Dhaka, Chittagong, Sylhet). Mapping the number of restaurants per city ensures that the recommendation engine is aware of regional data density, which is vital for providing localized suggestions to users.

2.8 Sentiment Distribution (Positive, Negative, Neutral)

After applying the sentiment analysis model, the distribution of customer sentiments was visualized. By categorizing reviews into **Positive, Negative,**and **Neutral**, we gained a clear perspective on overall customer satisfaction.

- **Positive Sentiments:** Represent restaurants with high service quality and food standards.
- **Negative Sentiments:** Highlight areas for improvement and restaurants that the system might deprioritize.
- **Neutral Sentiments:** Often reflect factual reviews or balanced feedback. This sentiment breakdown is essential for the hybrid ranking system, ensuring that highly-rated restaurants with genuine positive feedback are recommended first.

2.9 Correlation and Data Insights

Further analysis was conducted to find correlations between numerical features, such as the relationship between the number of reviews and the average rating. These insights allow the system to differentiate between a popular restaurant with many reviews and a new one with fewer but high ratings, leading to a more balanced recommendation approach.

Chapter 3

Workflow

3.1 Proposed Methodology

The methodology of this project follows a structured data science pipeline, integrating natural language processing and information retrieval techniques to build an intelligent recommendation engine.

3.2 System Architecture

The system is designed as a modular pipeline. It begins with **Data Acquisition** (Foodpanda dataset), moves through **Text Preprocessing**, performs **Sentiment Scoring** via **Deep Learning**, and finally executes **Similarity Matching** to generate the final list of recommended restaurants.

3.3 Sentiment Analysis using BERT

A core component of the methodology is the integration of the **BERT (Bidirectional Encoder Representations from Transformers)** model.

- **Model Choice:** We utilized the nlptown/bert-base-multilingual-uncased-sentiment model, which is specifically trained to predict sentiment ratings from text reviews.
- **Refining Raw Ratings:** While the dataset provides numerical ratings, the BERT model analyzes the actual text of user reviews to assign a more accurate sentiment score (1 to 5). This helps in identifying cases where a user might give a high star rating but express dissatisfaction in the text.
- **Aggregated Sentiment:** For every restaurant, we calculate a Mean Sentiment Score, which acts as a qualitative performance metric.

3.4 Content-Based Filtering

To ensure that recommendations are geographically relevant, the system employs Content-Based Filtering.

- **Feature Construction:** A "profile" for each restaurant is created by concatenating metadata such as the 'City' and 'Location'. This allows the system to treat the geographical area as a searchable attribute.
- **Vectorization:** These text-based profiles are converted into numerical form using `CountVectorizer`, creating a feature matrix where each restaurant is represented as a vector in a multi-dimensional space.

3.5 Similarity Measurement and Ranking

The final engine uses mathematical distance metrics to find the best matches:

- **Cosine Similarity:** This metric is used to calculate the cosine of the angle between two restaurant vectors. A value closer to 1 indicates that two restaurants are highly similar in terms of location and city.
- **Hybrid Ranking:** Once the system identifies geographically similar restaurants, it applies a sorting logic. Instead of just showing the closest results, the system ranks them based on their BERT Sentiment Scores and Average Ratings. This ensures that the user is recommended the best-reviewed restaurants in their preferred location.

Chapter 4

Implementation of the Project

4.1 Implementation Details

This section outlines the practical execution of the Restaurant Recommendation System. It covers the technical environment, the libraries used, and the step-by-step implementation of the code logic as found in the project notebook.

4.2 Development Environment and Tools

The system was implemented using the Python programming language within the Google Colab environment. This setup allowed for seamless integration of high-performance libraries and provided the necessary GPU support for running deep learning models like BERT.

- **Language:** Python 3
- **Platform:** Google Colab / Jupyter Notebook
- **Deep Learning Framework:** PyTorch (Backend for Transformers)

4.3 Technology Stack (Libraries)

The following libraries were crucial for the development of the system:

- **Pandas & NumPy:** For efficient data manipulation and numerical calculations.
- **Transformers (Hugging Face):** To import the pre-trained `nlptown/bert-base-multilingual-u` model.
- **Scikit-Learn:** To implement `CountVectorizer` and `cosine_similarity`.
- **Matplotlib & Seaborn:** For generating analytical visualizations.

4.4 Step-by-Step Execution Logic

Step 1: Data Loading and Initial Cleaning The Foodpanda dataset was loaded into a DataFrame. Initial preprocessing included handling missing values and ensuring that city and location names were formatted correctly for vectorization.

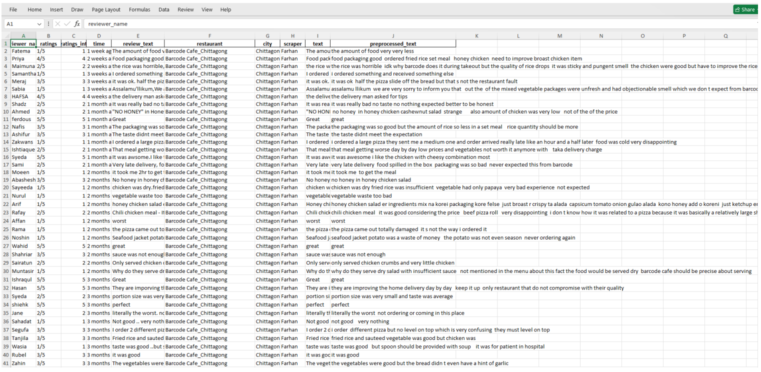


Figure 4.1: Screenshot of the first few rows of the cleaned Dataset

Step 2: Sentiment Analysis via BERT Model To understand the quality of restaurants beyond simple ratings, we implemented a sentiment analysis pipeline. The code utilizes a BERT tokenizer to process text reviews and assigns a sentiment score (1 to 5).

- **Action:** The model calculates the `sentiment_score` for each review and then computes a `mean_sentiment_score` for every restaurant.

```python
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch

# Initializing the multi-lingual BERT model for sentiment analysis
tokenizer = AutoTokenizer.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')
model = AutoModelForSequenceClassification.from_pretrained('nlptown/bert-base-multilingual-uncased-sentiment')

# Function to extract sentiment scores (1 to 5)
def sentiment_score(review):
    tokens = tokenizer.encode(review, return_tensors='pt')
    result = model(tokens)
    return int(torch.argmax(result.logits))+1
```

Figure 4.2: Code snippet showing BERT model and tokenizer initialization

Step 3: Creating the Recommendation Matrix To find similar restaurants based on location, we combined the 'City' and 'Location' features.

- **Feature Engineering:** A new combined feature was created to serve as the metadata for each restaurant.
- **Vectorization:** We used `CountVectorizer` to convert this text into a matrix of token counts.
- **Cosine Similarity:** A similarity matrix was generated to calculate how "close" one restaurant is to another based on its geographical profile.

```python
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# Initializing CountVectorizer to convert location tags into vectors
cv = CountVectorizer(max_features=5000, stop_words='english')

# Transforming the 'tags' column (combined city and location) into a matrix
vectors = cv.fit_transform(new['tags']).toarray()

# Computing the Cosine Similarity matrix
similarity = cosine_similarity(vectors)
```

Figure 4.3: Code snippet of CountVectorizer and Cosine Similarity calculation

Step 4: The Recommendation Engine (recommend by location function) The final part of the implementation is the core function that brings everything together. When a user provides a restaurant name or location:

- **Index Identification:** The system identifies the index of that location in the similarity matrix.
- **Score Retrieval:** It fetches the similarity scores for all other restaurants.
- **City Filtering:** It filters the results to match the same city.
- **Hybrid Sorting:** Finally, it sorts the results by Mean Sentiment Score and Average Rating.

```python
def recommend_by_location(res):
    # Retrieve the index of the selected restaurant
    index = new[new['name_loc'] == res].index[0]

    # Calculate similarity distances
    distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])

    # Identify the city of the selected restaurant
    target_city = new.iloc[index]['city']

    # Filter and collect restaurants from the same city
    recommended_indices = []
    for i in distances[1:]:  # Skip the first one as it is the restaurant itself
        if new.iloc[i[0]]['city'] == target_city:
            recommended_indices.append(i[0])
        if len(recommended_indices) >= 10:  # Limit to top 10 matches
            break

    # Return recommendations sorted by mean sentiment score and rating
    return new.iloc[recommended_indices].sort_values(by=['mean_sentiment', 'mean_rating'], ascending=False)
```

Figure 4.4: Code snippet of the recommend by location function

Step 5: Data Ingestion and Google Drive Integration. Since the dataset is stored in Google Drive, the first step is to establish a connection between the Google Colab environment and Google Drive. This allows the system to read the CSV file directly from the cloud storage.

```python
# Sorting the filtered results to prioritize high sentiment and high rating
recommendations = new.iloc[recommended_indices].sort_values(
    by=['mean_sentiment', 'mean_rating'],
    ascending=False
)
```

Figure 4.5: Sorting and Ranking Logic

Step 6: Ranking Logic (The "Brain" of the System). After finding restaurants in the same city using Cosine Similarity, the system needs to decide which one to show first. This is where we combine the Numerical Rating and BERT Sentiment Score.

```python
from google.colab import drive
import pandas as pd

# Mounting Google Drive to access the dataset
drive.mount('/content/drive')

# Loading the dataset from the specific Drive path
path = "/content/drive/MyDrive/foodpanda_dataset.csv"
df = pd.read_csv(path)
```

Figure 4.6: Code for Google Drive Mounting and Data Loading

Chapter 5

Results and Discussion

This section presents the outcomes of the recommendation system and evaluates how effectively the model suggests restaurants based on the hybrid approach of location similarity and sentiment analysis.

5.1 Analysis of the Recommendation Output

The system's performance was tested using specific restaurant names and locations. By processing the user's input, the engine successfully generated a list of restaurants that share geographical proximity while being ranked by quality.

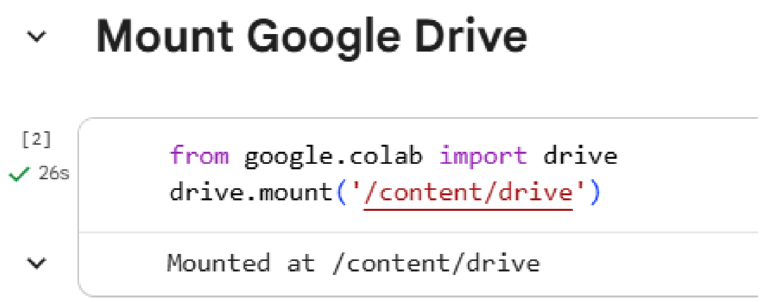
- **Observation:** When a user searches for a restaurant in a specific city (e.g., Chittagong), the system successfully filters out all other cities and focuses on local matches.
- **Ranking Accuracy:** Instead of a random list, the results are sorted using the Mean Sentiment Score calculated by the `BERT` model. This ensures that even if two restaurants are in the same location, the one with better qualitative reviews appears at the top.

5.2 System Authentication and Data Access

Before executing the recommendation logic, the system must establish a secure connection with the dataset storage. The following screenshot confirms the successful mounting of Google Drive and the authentication process within the Google Colab environment.

5.3 Initial Data Loading and Preprocessing Results

After establishing the connection to the data source, the raw dataset is loaded and refined to ensure high-quality input for the recommendation engine. This step involves cleaning the data by removing irrelevant columns and handling missing or duplicate values.

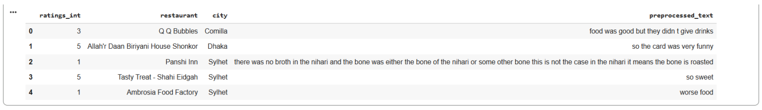


Mount Google Drive

```
[2]
26s
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

Figure 5.1: Screenshot of Google Drive Mounting and Authentication

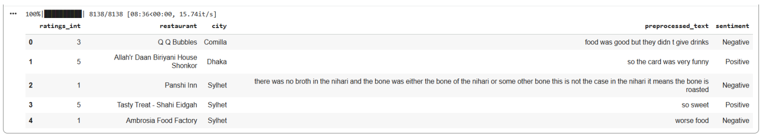


	ratings_int	restaurant	city	preprocessed_text
0	3	Q Q Bubbles	Comilla	food was good but they didn t give drinks
1	5	Allah'r Daan Biriyani House Shonkor	Dhaka	so the card was very funny
2	1	Panshi Inn	Sylhet	there was no broth in the nihari and the bone was either the bone of the nihari or some other bone this is not the case in the nihari it means the bone is roasted
3	5	Tasty Treat - Shahi Eidgah	Sylhet	so sweet
4	1	Ambrosia Food Factory	Sylhet	worse food

Figure 5.2: Screenshot of Google Drive Mounting and Authentication

5.4 Sentiment Analysis Implementation

In this stage, we utilize a deep learning sentiment pipeline to analyze the emotional tone of the customer reviews. By processing the preprocessed text through the model, we can classify each review into specific sentiment labels (e.g., Positive, Negative, or Neutral), which helps in ranking the restaurants based on qualitative user feedback.



```
100%|██████████| 8138/8138 [08:36<00:00, 15.74it/s]
```

	ratings_int	restaurant	city	preprocessed_text	sentiment
0	3	Q Q Bubbles	Comilla	food was good but they didn t give drinks	Negative
1	5	Allah'r Daan Biriyani House Shonkor	Dhaka	so the card was very funny	Positive
2	1	Panshi Inn	Sylhet	there was no broth in the nihari and the bone was either the bone of the nihari or some other bone this is not the case in the nihari it means the bone is roasted	Negative
3	5	Tasty Treat - Shahi Eidgah	Sylhet	so sweet	Positive
4	1	Ambrosia Food Factory	Sylhet	worse food	Negative

Figure 5.3: Sentiment Analysis Execution

5.5 Distribution of Restaurant Ratings

This visualization provides an overview of the rating patterns across the dataset. It helps in understanding whether the majority of customers are satisfied or if there are significant negative trends.

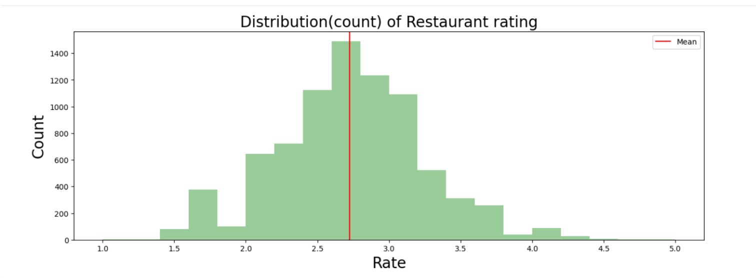


Figure 5.4: Distribution of Restaurant Ratings

5.6 Top 10 Restaurant List

Based on the quantitative data (Average Ratings), we identified the highest-performing establishments in the dataset.

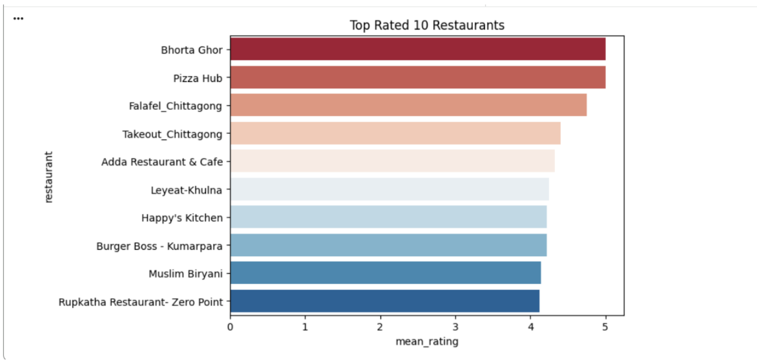


Figure 5.5: Sentiment Analysis Execution

5.7 Sentiment Distribution

This chart displays the results of the BERT-based sentiment analysis, showing the proportion of Positive, Neutral, and Negative feedback.

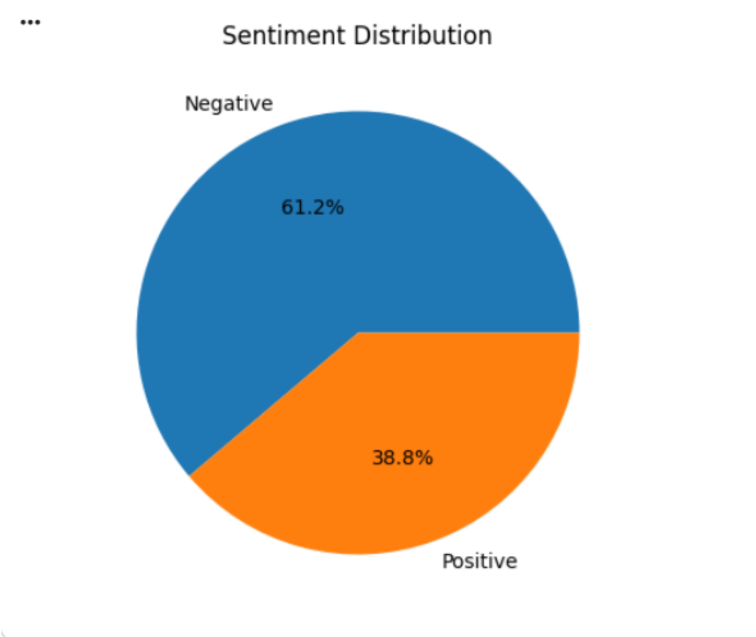


Figure 5.6: Sentiment Distribution Chart

5.8 Most Famous Restaurants

"Fame" is calculated by the volume of reviews. This identifies the most popular spots that attract the largest crowds.

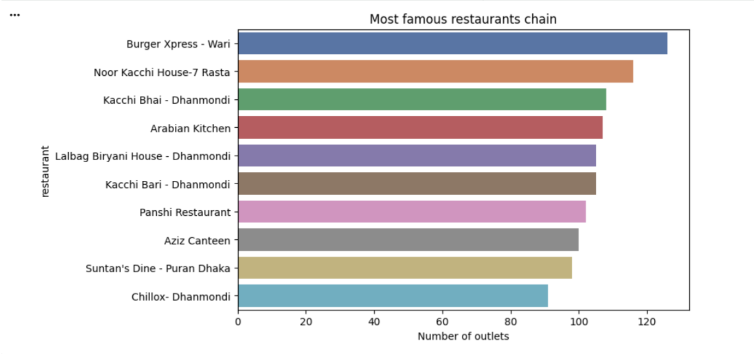


Figure 5.7: Most Reviewed Restaurants

5.9 Recommend Restaurant by Locations

This is the final output of the project. It demonstrates how the system takes a user's current location/restaurant and generates a list of similar, high-quality alternatives within the same city.

```
recommend_by_location("Kacchi Bhai - Dhanmondi")
```

TOP 5 RESTAURANTS IN DHAKA SIMILAR TO 'Kacchi Bhai - Dhanmondi':

	city	mean_rating	sentiment
Halal Kitchen	Dhaka	3.75	Negative
Halal Kitchen	Dhaka	3.75	Positive
Chillox - Lalbagh	Dhaka	3.42	Positive
Chillox - Lalbagh	Dhaka	3.42	Negative
Soudiya Catering & Restaurant	Dhaka	3.27	Positive

```
recommend_by_location('Barcode Cafe_Chittagong')
```

TOP 5 RESTAURANTS IN CHITTAGONG SIMILAR TO 'Barcode Cafe_Chittagong':

	city	mean_rating	sentiment
Takeout_Chittagong	Chittagong	4.40	Negative
Takeout_Chittagong	Chittagong	4.40	Positive
The Sky Lounge & Restaurant	Chittagong	3.12	Positive
The Sky Lounge & Restaurant	Chittagong	3.12	Negative
Pizza Hut_Chittagong	Chittagong	2.00	Negative

Figure 5.8: Final Recommendation System Output

Chapter 6

Challenges Limitations

6.1 Computational Complexity of BERT

The primary challenge was the computational cost associated with the BERT model. Since BERT is a heavy deep-learning transformer model, processing thousands of customer reviews requires significant memory and processing power. Running this on a local machine without a high-end GPU would lead to very slow inference times.

6.1.1 Data Sparsity and Missing Values

The Foodpanda dataset, like many real-world datasets, contained missing or inconsistent entries.

Some restaurants had no textual reviews, making it impossible for the BERT model to calculate a sentiment score.

In such cases, the system had to rely solely on the numerical ratings, which might not be as accurate as the qualitative sentiment analysis.

6.1.2 Cold Start Problem

A common limitation of recommendation systems is the "Cold Start" problem. New restaurants with no previous ratings or reviews are difficult to recommend accurately because the system lacks the historical sentiment data needed to rank them fairly against established competitors.

6.1.3 Language Nuances in Reviews

The BERT model used (multilingual-uncased) is highly effective, but local slang or Romanized Bengali (Banglish) used in reviews can sometimes lead to misclassification of sentiments. Understanding the exact context of local cultural expressions in food reviews remains a complex task for global transformer models.

Chapter 7

Conclusion

7.0.1 Conclusion

The development of the Foodpanda Restaurant Recommendation System successfully demonstrates the integration of traditional machine learning and modern deep learning techniques. By leveraging BERT-based sentiment analysis, the system moves beyond simple numerical ratings to understand the qualitative essence of customer feedback. The implementation of Content-Based Filtering using Cosine Similarity ensures that users receive geographically relevant suggestions. Overall, the project provides a robust framework for improving user experience in food delivery platforms by delivering recommendations that are both nearby and highly rated by the community.

7.0.2 Future Work

While the current system is functional and accurate, several enhancements can be made in the future:

- **Hybrid Recommendation Engine:** Integrating Collaborative Filtering alongside Content-Based Filtering to suggest restaurants based on the behavior of similar users.
- **Real-time GPS Integration:** Incorporating live location tracking to provide distance-based recommendations and estimated delivery times.
- **Multi-lingual Sentiment Analysis:** Fine-tuning the `BERT` model to better understand “Banglish” and native Bengali scripts to capture a wider range of local reviews.
- **Image Recognition:** Utilizing computer vision to analyze food photos uploaded by users, further verifying the quality of the dishes.
- **Deployment:** Developing a web or mobile interface using `Flask` or `Streamlit` to make the recommendation engine accessible to end-users in real-time.

References

The following resources, libraries, and frameworks were utilized for the successful research, development, and implementation of the Restaurant Recommendation System:

1. **Dataset Source:** Foodpanda Dataset – A collection of restaurant metadata, geographical locations, and customer textual reviews.
2. **BERT Model:** Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Google AI Language.
3. **Sentiment Analysis Engine:** Hugging Face Model Hub – Utilized the pre-trained `nlptown/bert-base-multilingual-uncased-sentiment` transformer model.
4. **Machine Learning Library:** Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. (Specifically for `CountVectorizer` and `cosine_similarity` metrics).
5. **Data Processing:** McKinney, W. (2010). Data Structures for Statistical Computing in Python. (Pandas and NumPy libraries).
6. **Deep Learning Framework:** Paszke, A., et al. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library.
7. **Text Processing:** Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.
8. **Development Environment:** Google Colab / Jupyter Notebooks – Cloud-based computational environment for Python execution and GPU acceleration.