Nova Information Management School
BSc in Data Science

Text Mining 2024-2025

Group Project: *"Solving the Hyderabadi Word Soup"*

Project Report

Group 10

Adriana Pinto | 20221921
David Duarte | 20221899
Maria Teresa Silva | 20221821
Marta Alves | 20221890
Miguel Nascimento | 20221876

Table of Contents

1. Introduction

1.1. Overview

This report addresses the primary and secondary information requirements for the "Solving the Hyderabadi Word Soup" project, as defined by the Hyderabad Tourism Board. By analyzing data from 105 restaurants in Hyderabad and 10 000 corresponding Zomato reviews, the project uses advanced text mining techniques to answer a set of information requirements. The primary requirements include multilabel classification and sentiment analysis, while the secondary requirements include topic modelling, co-occurrence analysis and clustering.

The objectives of this report are to provide comprehensive data understanding, data preparation, modelling and evaluation. Additionally, it aims to deliver actionable insights and predictive models capable of supporting the Hyderabad Tourism Board in understanding cuisine distributions, restaurant qualities, and related themes. The findings will be presented in a clear, concise, and visually engaging format, aligning with CRISP-DM methodology, a widely recognized framework for structuring data mining processes ensuring systematic exploration, integration, and analysis of the datasets.

2. Literature Review

Sunarko [1] explored multi-class text classification for restaurant customer reviews using BERT, demonstrating its effectiveness. They also discussed the use of deep learning approaches like LSTM, highlighting its potential for addressing similar challenges. This inspired the application of a LSTM, to address similar challenges.

N. Begum [2] explored Sentiment analysis on Zomato customer reviews using Random Forest. Although our goal was not to apply Random Forest on our sentiment analysis exploration, we found

this paper interesting as it used the same dataset that was provided for this project. We were able to take insights of the data pre-processing part of the paper.

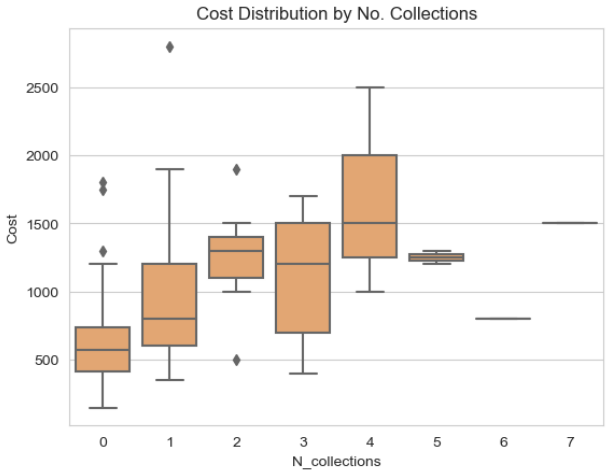
3. Data Understanding

This section of the project is very important as we take the first look into our data. The first impressions that we had about the reviews text were the emojis, urls, missing values and carriage return characters. It was also possible to detect informal expressions and sentences that had no meaning being just noise to our dataset.

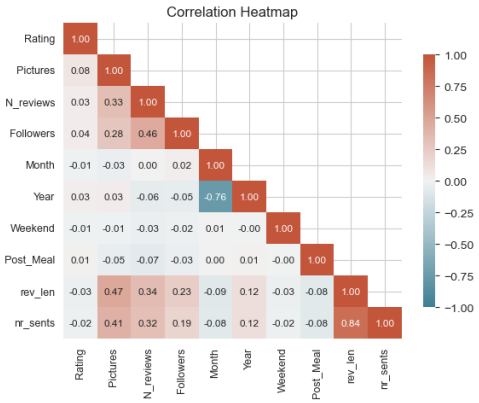
3.1. Exploration and Visualizations

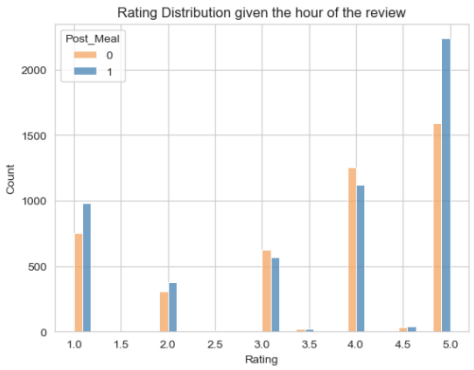
In order to get a deeper understanding of the data we were working on, multiple visualizations were performed, with just a few data cleaning steps found on this section. These were the main insights extracted:

- North Indian and Chinese are the most common cuisine types and Great Buffets and Food Hygiene Rated Restaurants in Hyderabad are the most common collections.
- Cost and N_collections have an interesting degree of correlation as it is expected the higher the number of collections a restaurant belongs, the higher its cost.



- By analysing the rating distribution, it could be concluded that they are skewed towards the top. [Annex 1]
- As expected, the bigger the review, the more pictures they normally contain, and more followers and reviews published has the reviewer.
- Reviews published shortly after the meal had more extreme ratings (positives and negatives) than the ones published at other times.





- There are cases where the same reviewer reviewed the same restaurant multiple times and with different ratings.

- Some cuisine types reflect on word clouds specific foods but nothing that was not expected or different.

- No major insights could be taken from the wordclouds by collection.

4. Data Preparation

4.1. Initial Cleaning

Starting with the restaurant's dataset, the ‘Cost’ column was converted to integer type, the columns ‘Collections’ and ‘Cuisines’ were converted to lists and a new column ‘N_collections’ was created representing the number of collections a restaurant belongs. Moreover, the opening and closing time of the restaurants was extracted.
Regarding the Reviews dataset, missing reviews were deleted and a specific case where the review rating was set to ‘Like’ was substituted by 5, after reading the review content. We could also notice that some reviews had float values as such ‘4.5’ but were maintained like this as it would not harm any future work. On the matter of text normalization, the symbol ₹ was converted to ‘rupees’ and * were converted to ⭐ to be able to be used in sentiment analysis as stars. From the column ‘Metadata’ it was possible to extract the number of reviews and followers for each reviewer, and from the ‘Time’ column, extract the year and month of the review and create a flag named ‘Weekend’ that contained 1 if the review was written on a weekend and 0 otherwise. From the hour the review was posted it was possible to create a flag that checked if the review was most likely to be written after eating the meal, by defining a range of hours after meals. It was also extracted the length of each review and the number of sentences they contained.

4.2. Reviews Preprocessing

To be able to apply text preprocessing techniques on the reviews, the practical classes’ main pipeline was adapted in order to fit better the needs of the text we were working on. Those adaptions include:

- adapting the newline regex pattern to also remove carriage return characters (/r)
- updating the emoji pattern is order to remove all the situations we encountered in the reviews
- updating the url pattern to be able to extract all the hyperlinks in the text
- adding the argument no contractions that defines if the contractions will be removed from the text (example: isn’t -> is not) using the package contractions.
- updating the code for removing punctuation to fix cases like word1…word2 being converted to word1word2 (before) instead of word1 words2 (after).
- adding the argument exception_stopwords that allows the user to insert stopwords present in the nltk stopword dictionary that we don’t want to remove from the text.

Additionally, a Gibberish Classifier was implemented using the code from [7] in order to allow us to filter out reviews that contained letter spamming, or reviews that did not contain any real content. However, it was challenging to define a correct threshold where we would discard reviews.

5. Modelling

5.1. Multilabel Classification

The main goal for the multilabel classification is to classify a restaurant´s cuisine type based on content of their reviews.
Therefore, the first step was to clean the reviews dataset by removing gibberish and unnecessary characters and then discard any blank reviews and perform label encoding. After that, it was applied a stratified split to divide the dataset into training and test sets.

Following this, the text was transformed by using various word embeddings, including Bag of Words, TF-IDF, and Word2Vec. To each of them it was experimented different classification models, such as One-vs-Rest, Chain Classifier and Label Powerset, the last two with Logistic Regression and Random Forest. Despite experimenting with different hyperparameters, most of these models tended to overfit.
LSTM model (Long Short – Term Memory) [1], a deep learning approach that is commonly used in this type of problems, was also applied with Word2Vec, however the results were not good. Until this moment the best model achieved was Label Powerset of Random Forest with Bag of Words with an F1 weighted score of 0.546 in the train dataset and 0.515 in the test dataset.

In a tentative to improve the results, two grid search experiments using the hold-out method were conducted. The first focused solely on tuning the model's hyperparameters, while the second combined hyperparameter tuning with optimizing the text cleaning process. Due to time constraints, we used only the latter approach. However, the hold-out method resulted in lower scores because it reduced the size of the training set.
To address this issue, the best-performing model from the grid search was selected and it was attempted a simple train-test stratified split. Unfortunately, this approach yielded worse results compared to our previous best model.

Finally, we evaluated our results using the best-performing model from the grid search.

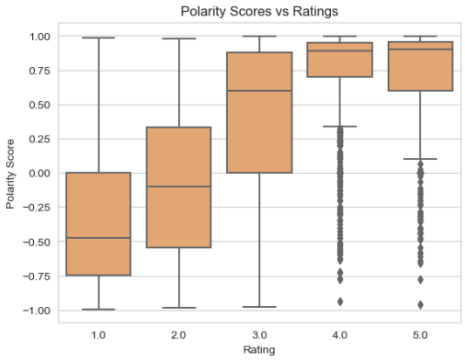
5.2. Sentiment Analysis

This section aims to explore people’s opinions and emotions towards restaurants in Hyderabad through their reviews. The first step involved preprocessing the data to obtain fully cleaned and tokenized sentences, ensuring that emojis and punctuation were preserved, as they provide valuable context for this task.

Next, to assess the sentiment polarity of the reviews, both VADER and TextBlob were tested using the same approach and then compared. It was found that applying the sentiment analysis to the entire review yielded better results than calculating the mean polarity of individual sentences within a review. Additionally, normalization was applied to both the polarity scores and ratings, allowing for easier comparison of the results.
To further understand this topic, some histograms, box plots and scatter plots were computed, to uncover the distributions of sentiment, according to the polarity. This confirmed that TextBlob is not as efficient as VADER for informal text, therefore, from this point, the analysis was conducted with VADER.

The next step involved generating WordClouds from all the reviews, starting with unigrams (single words) and then moving to bigrams (pairs of words). The goal was to identify patterns or recurring themes that could offer deeper insights into the emotions and priorities expressed within each sentiment category. This analysis was carried out specifically for positive, neutral, and negative sentiments. To enhance the visualization of other important words, common terms like ‘food’ and ‘good’ were excluded due to their high frequency.

To evaluate whether polarity can reliably predict ratings, a box plot was created.



After, a more in-depth analysis of the data was performed, in order to uncover additional patterns and potential relationships. The analysis explored whether cost, cuisine type, average rating, and specific review flags influenced polarity and ratings. For cost, the most expensive and cheapest restaurants were examined, while cuisine type analysis included polarity distributions and bar charts for the top 5 cuisine types by sentiment. The relationship between average rating and polarity was assessed using the top 10 best and worst restaurants through WordClouds, sentiment trends over time, and box plots to determine alignment. Lastly, the impact of Weekend and Post_Meal flags on polarity was analyzed using box plots and correlation calculations.

To end this section, a predictive model was developed to predict a Rating based on the polarity of the reviews.

5.3. Co-occurrence

In this section, the text was pre-processed creating two new columns: one tokenized and the other non-tokenized. This distinction is essential, as both versions are used to create co-occurrence matrices in subsequent steps.

Initially, a co-occurrence matrix was plotted, revealing insufficient food or dish related terms to analyse their co-occurrence in reviews effectively. To address this, it was decided to experiment with n-grams, as they better capture dish names often composed of two or more words.

Next, it was identified 3 words that appeared in numerous combinations but were not informative to our objective, so it was decided to remove them. To further refine the data, it was applied named entity recognition to exclude irrelevant entities, to improve the notoriety of food-related terms. Another approach was to exclude the 50 most common words in the top 10 cuisines with most reviews, as many of these words were unrelated to food.

The last approach involved using a pre-trained [10] model that identifies food related content in the reviews. Reviews identified as containing food-related content were then used to plot the co-occurrence matrix. With these approaches, the goal was to find the most effective co-occurrence matrix so that was possible to analyse the dishes that appeared more times together in the same review.

With the last approach a network graph was plotted, and we concluded that was a strong connection between “ice cream” and “cream stone”. Clustering was performed using BoW and TF-IDF. BoW performed slightly better than TF-IDF, but the results were still poor.

5.4. Topic Modelling

The text used for this section was pre-processed removing stopwords but maintaining the words ['no','not','nor','very','few','all','again','but'] so that the topics do not lose the true meaning of the reviews.

Firstly, Latent Semantic Analysis, Latent Dirichlet Analysis and BERTopic were tested using basic hyperparameters in order to compare which one achieved better performance. Even though by computing the coherence score for these 3 approaches the highest score was for LSA, by analysing each topic's content we could conclude that the most meaningful topics content-wise were the topics created by BERTopic. Furthermore, BERTopic allows a more diverse range of hyperparameters to tune so it allows as it will be proven more margin to improve this process. Therefore, it was chosen to continue Topic Modelling the restaurants reviews with BERTopic.

While testing different combinations of hyperparameters, though we could reach similar coherence scores as LSA and LDA, when humanly analysing how well the segmentation was performed, we could see that only some of the 'clusters' were well formed and truly reflected the vast majority of the reviews. Therefore, we have decided to follow this approach:

- 1st Step: Apply BERTopic with HDBSCAN while testing hyperparameters to better define the topics.
- 2nd Step: Identify the well define topics and label the corresponding reviews.
- 3rd Step: Repeat the 1st step until all reviews are labeled.

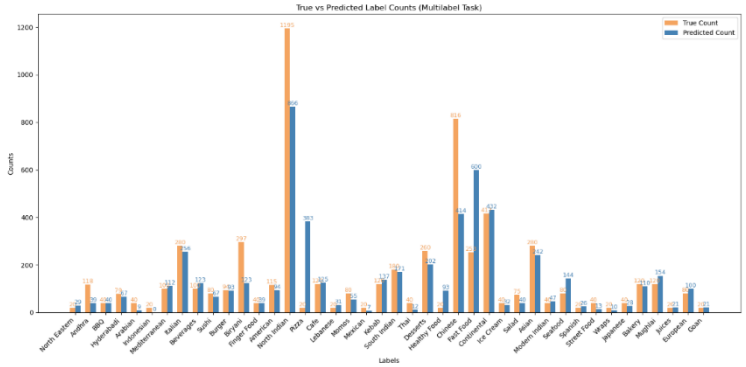
Note that the word 'good' was added as a custom stopword as in the first tests, most of the topics were dominated by this word and this way we could go deeper in topic modelling the contents of the reviews, by removing this word.

Still regarding BERTopic, a grid search pipeline was implemented before following the mentioned approach to test more easily multiple combinations of hyperparameters and variations on the data preprocessing phase. However, due to the amount of time it took to run even for a small parameter grid and as it was only focused on the coherence score, it did not allow the human part of analysing how well the topics reflected the reviews. Therefore, it was not used as part of the final strategy to define the topics and the preprocessing steps were determined by intuition according to what we thought would work better to extract the topics.

6. Results

6.1. Multilabel Classification

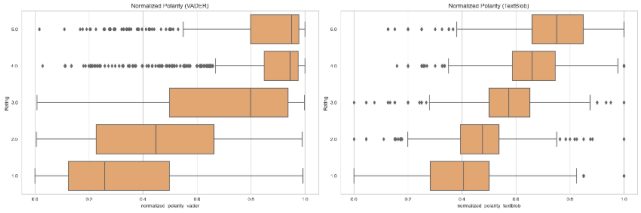
In relation to Multilabel Classification, the weighted F1 scores of 0.546 (train) and 0.515 (test). In the end some cuisines were being overpredicted while another's underpredicted. The reasons why this might happen will be addressed in the conclusion.



6.2. Sentiment Analysis

The comparison between normalized VADER and TextBlob polarities allowed to reach the conclusion that values of RMSE translate to approximately 1.4 stars on the original scale, indicating that, on average, the predicted polarity of each user review is about 1.4 stars away from the actual rating, reflecting a moderate level of error. The Pearson correlation coefficient of 0.705 suggests a strong positive relationship between polarity and rating, making polarity a meaningful predictor despite the moderate error indicated by the RMSE and MAPE.

The computed histograms revealed that VADER polarity scores are skewed towards positive sentiment, while TextBlob exhibits a more balanced distribution, capturing a mix of neutral, positive, and slightly negative sentiments. The generated box plot further demonstrated that VADER captures a wider range of sentiment, which is beneficial for this task, as demonstrated below.

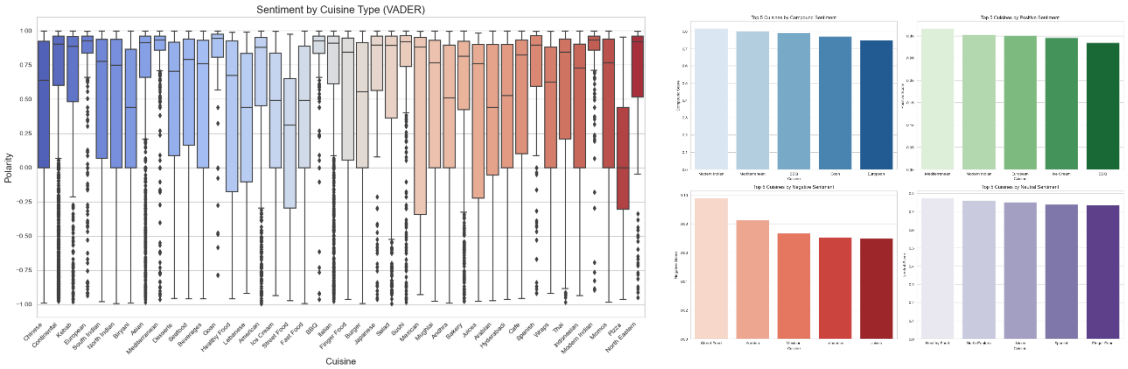


The WordCloud analysis revealed key themes in customer feedback across different sentiment categories. Positive and really positive sentiments were dominated by terms like "great," "taste," "service," and "ambience," highlighting satisfaction with food quality and the overall atmosphere. Negative and really negative sentiments centered on words like "bad," "worst," and "quality," indicating dissatisfaction with food or service. Neutral sentiment was characterized by more factual descriptions, with words like "ordered," "delivered," and "chicken" reflecting a focus on the experience without strong emotional tones. This is possible to observe in the following image:

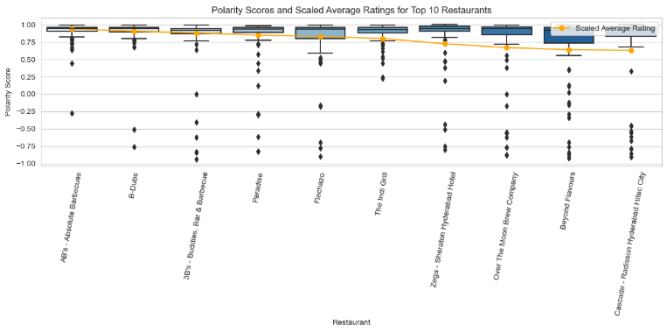
The box plot Polarity Scores VS Rating revealed that polarity can predict the rating with some degree of accuracy, as it displayed a noticeable linear trend: low scores correspond to lower polarity, while high scores are associated with higher polarity. Additionally, the plot highlighted the presence of some exceptions, indicating that existed reviews with low polarity that have a high rating.

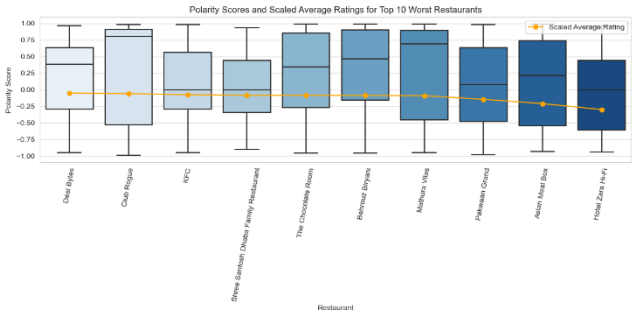
The analysis of the cost revealed that customers tend to complain more about cheaper restaurants than expensive ones, indicating that price, as expected, influences restaurant quality, which impacts review polarity and ratings.

The results of the Cuisine type analysis showed variations in sentiment across cuisines, with Modern Indian, BBQ and Mediterranean receiving the highest and most consistent positive sentiment, reflecting strong customer satisfaction. In contrast, Fast Food, Street Food, and Japanese show more polarized sentiments, indicating mixed experiences and potential inconsistencies. Cuisines like Chinese, Kebab, and Sushi exhibit mostly neutral to slightly positive sentiment but have outliers indicating occasional dissatisfaction. Meanwhile, Bakery, Cafe, and Arabian maintain steady moderate positivity, positioning them as reliable options, as showed by these plots:



For the top 10 best restaurants, reviews highlighted appreciation for food, service, and ambiance but also critique similar aspects, showing mixed sentiments. Polarity scores were mostly positive with small variability, though outliers reveal instances of low-polarity high-rating reviews and high-polarity low-rating reviews. Reviews for the 10 worst restaurants showed word variety, with frequent mentions of dissatisfaction using terms like "bad" and "worst." Club Rogue displays diverse opinions with a wide polarity range, while restaurants like "Mathura Vilas" show consistent sentiments. Scaled average ratings align closely with sentiment polarity, reflecting a general consistency between review sentiments and customer ratings.





For the defined flags Weekend and Post_Meal, it was possible to infer that none of them have influence in the polarity.

The model predicted Rating = 0.889Polarity + -0.003, demonstrating a positive relationship between polarity and ratings, with reasonable predictive accuracy (RMSE: 0.264, MAPE: 14.4%), though other factors likely influence ratings beyond polarity alone.

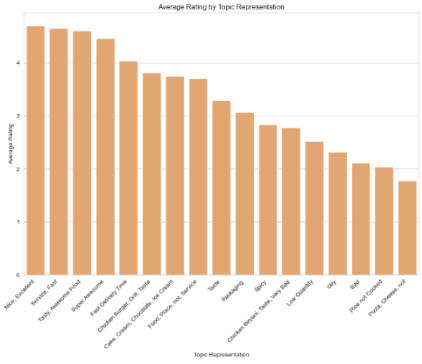
6.3. Co-occurrence and Clustering

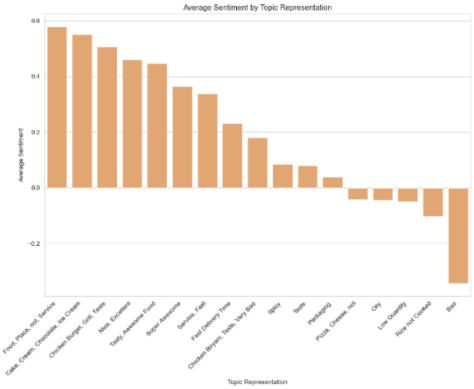
The results in this section were not satisfactory as despite trying different approaches the co-occurrence matrices did not demonstrate many situations where dishes appeared together in the same review. Despite, this results the approach that yield the better co-occurrence matrix was the

one with the pre-trained model. Regarding clustering although they appear well-defined visually, it was challenging to determine the specific cuisine type represented by each cluster.

6.4. Topic Modelling

After 5 rounds of applying BERTopic to the restaurant reviews, it was possible to form 17 topics. Even though some of them are constituted by only a small sample of reviews, they were considered really specific and with potential to extract meaningful insights. Based on the top 10 most frequent words for each topic, the topics were labelled to improve understandability. By plotting an heatmap with the topics representation per cuisine type it was possible to verify topics deeply correlated as expected to a cuisine such as 'Chicken Burger, Grill, Taste' in the type 'Burger'. However, when doing the same for the different collections it was possible to identify more interesting relations like the topics 'Oily' and 'Packaging' on the Veggie Friendly Collection. [Annex 2] Furthermore, it can be seen below the variation of the reviews average rating and VADER polarity for each of the topics with that labelling. Furthermore, it can be seen below the variation of the reviews average rating and VADER polarity for each of the topics with that labelling.





7. Conclusions

7.1. Multilabel

In multilabel classification, the results indicate moderate effectiveness in identifying cuisine types from reviews. This can be due to the challenges in achieving ideal stratification and the possible variability and subjective nature of the content. To get around this problem, one possible solution is to apply Cross-validation in the grid search process, however it is computationally expensive, even with a 5-fold setup. Therefore, the exploration of the hyperparameter space was prioritized, using Hold Out method.

7.2. Sentiment Analysis

There is a moderate relationship between the polarity of a restaurant's reviews and its Zomato score. The polarity, tends to align with the rating, meaning that higher polarity usually corresponds to higher scores and vice versa. While this suggests that polarity can serve as a useful input for predicting a restaurant's Zomato score, the presence of exceptions indicates that other factors beyond sentiment play a role in determining the rating. Factors like price and cuisine type, as we saw, also influence the polarity of reviews, impacting the overall rating.

7.3. Co-occurence and Clustering

In relation to co-occurrence and clustering, it was not possible to provide answers to all the information requirements. The only dishes mentioned together included "thai curry" with "chicken tikka", and "chole bhature" with "chicken tikka". These were the few relevant dishes combinations identified. Due to the limited dishes appearing together, as seen in the co-occurrence matrix, we

were unable to form clusters and, consequently, could not identify cuisine types based on those clusters.

7.4. Topic Modelling

In topic modelling, the results showed topics capable of capturing distinct aspects of restaurant experiences such as food quality, service efficiency, ambiance, and delivery, aligning with the average ratings and polarity scores retrieved from the Sentiment Analysis phase.

By providing insights into specific attributes, they reflect critical aspects that influence customer satisfaction in restaurants and can help identify common customer concerns (e.g., oily food, poor packaging, delayed delivery) and strengths (e.g., excellent taste, efficient service).

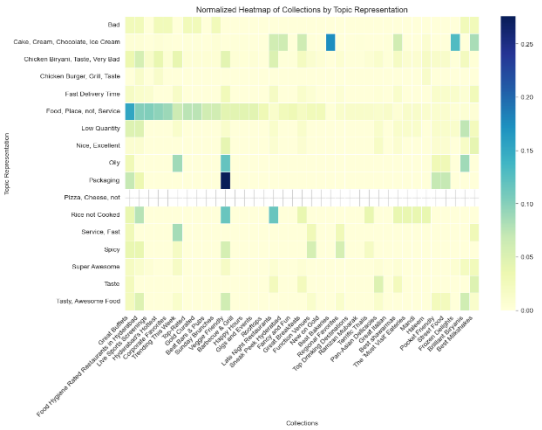
In summary, the topics are interpretable and offer actionable insights for enhancing customer experiences and addressing pain points.

8. References

1. B. Sunarko, U. Hasanah and S. Hidayat, "Enhancing Restaurant Customer Review Analysis: Multi-Class Text Classification with BERT," *2023 6th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Batam, Indonesia, 2023, pp. 501-506, doi: 10.1109/ISRITI60336.2023.10467438
2. N. Begum, M. Ruthika, N. L. Deepika, M. S. Sucharitha and V. P. Rao, "Sentiment analysis on Zomato customer reviews using Random Forest Classifier," *2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC)*, Chennai, India, 2023, pp. 225-230, doi: 10.1109/ICRTAC59277.2023.10480757
3. NLTK. *Natural Language Toolkit (NLTK) API Documentation*. Access https://www.nltk.org/api/nltk.html.
4. Scikit-learn. *Scikit-learn API Documentation*. Access https://scikit-learn.org/stable/api/index.html.
5. Rehurek, Radim. *Gensim API Documentation*. Access https://radimrehurek.com/gensim/apiref.html.
6. Grootendorst, Maarten. *BERTopic: A Topic Modeling Technique Using BERT Embeddings*. Access https://maartengr.github.io/BERTopic/index.html.
7. Tanvircr7. (n.d.). *learn_hf_food_not_food_text_classifier-distilbert-base-uncased*. Hugging Face. Access https://huggingface.co/tanvircr7/learn_hf_food_not_food_text_classifier-distilbert-base-uncased
8. Daniels, Thomas. *GibberishClassifier-Python*. GitHub. Acess https://github.com/thomas-daniels/GibberishClassifier-Python.
9. Practical Classes Notebooks
10. Tanvircr7. (2024). *learn_hf_food_not_food_text_classifier-distilbert-base-uncased*. Hugging Face. Accesshttps://huggingface.co/tanvircr7/learn_hf_food_not_food_text_classifier-distilbert-base-uncased.

Annexes

Annex 2



Annex 1