Dining Diagnostics: Text Mining Methods for Review-based Restaurant Service Improvement

Șerpar Ariana-Andra
Department of Computer Science
West University of Timișoara
Timișoara, Romania
ariana.serpar00@e-uvt.ro

I. The importance of service improvement

The ultimate objective of every business is to meet and ideally surpass the expectations of its clients, as this is the only pursuit that leads to profitability. Satisfied customers are likely to frequent your restaurant and may recommend your services to friends and acquaintances, resulting in increased patronage. On the other hand, dissatisfaction may not only lead to the loss of a customer but may also yield scathing negative reviews that will keep away any other potential patrons.

While word of mouth traditionally influenced marketing (and it still does, but to a lesser degree), the birth of the Internet has greatly altered this landscape. Nowadays, we do not solely rely on acquaintances for service recommendations, but also on the readily available opinions of thousands of individuals who post them on social media and dedicated platforms. In the contemporary era, people prioritize reviews over firsthand experiences, because experiencing everything yourself requires vast financial resources. Reviews allow us to pick and choose what services are worth paying for.

Continuous service improvement is crucial for any company aiming to maintain client satisfaction. This is particularly pertinent in the fiercely competitive restaurant industry, where success hinges on providing exceptional dining experiences and standing out no matter the cost.

Supporting these statements, a study by a Harvard Business School graduate revealed that a one-star increase in Yelp ratings correlates with a 9% revenue boost [1]. Additionally, statistics indicate that 60% of individuals consult reviews before dining out, often trusting platforms like Trip Advisor, more than accredited food critics [2].

Given the overwhelming volume of reviews a restaurant can receive, handling them manually is time-consuming and extremely inefficient. To address this, our report focuses on testing various data analysis and Text Mining methods to compile a centralized list of improvements based on reviews, aimed at enhancing the overall dining experience.

The subsequent sections provide all the necessary details on the methodologies employed during the explorations and the results obtained. The improvements derived from the reviews through the application of various methods will be summarized in Section XIII.

II. Dataset selection and analysis

Typically, the initial phase in any implementation involves identifying suitable data tailored to our selected task. In our case, we are seeking datasets designed for **Natural Language Processing** (NLP) that contain both positive and negative reviews of restaurants. While the dataset employed in the implementation phase of this report is custom-made and imported from Google Drive [3], alternative resources can always be found on Kaggle.

Kaggle, a platform catering to Data Scientists and Machine Learning (ML) enthusiasts, is a repository offering not only free datasets but also Jupyter Notebooks that address diverse tasks using the same data. We recommend the "Restaurant Reviews" dataset shared by Arsh Anwar on Kaggle [4] if you would like to replicate our implementation, as it is almost identical to the dataset under consideration for this report.

Our dataset contains two columns: one featuring 1000 restaurant reviews labeled as "Review," and the other categorizing the reviews as either negative (0) or positive (1), denoted as "Liked". Samples from each category can be analyzed in Table I.

TABLE I
Dataset sample examples

Review	**Liked**
Server did a great job handling our large rowdy table	1
This place deserves one star and 90% has to do with the food	0
If you want healthy authentic or ethnic food, try this place	1
Left very frustrated	0
Definitely not worth the 3 dollars I paid	0

III. Exploratory Analysis and Preprocessing

For the execution of our code, we have chosen the Colab environment. This decision stems from the integration of most of our required libraries within this environment, which eliminates the risk of incompatibility. Before library imports, the libraries that are not already integrated need to be downloaded using the **!pip install** command. For this implementation, we have downloaded the *simpletransformers* [5], *summa* [6], and *apyori* [7] libraries, along with the *stopwords*, *punkt*, and *wordnet* packages from the *nltk* [8] library.

A comprehensive list of all the libraries, along with their respective versions, is available in Table II. It is worth mentioning that the substantial number of imported libraries reflects the diverse Text Mining methods employed in the implementation.

TABLE II
PACKAGE VERSIONS

Package	Version
google.colab	0.0.1a2
pandas	1.5.3
matplotlib	3.8.2
numpy	1.22.4
sklearn	1.2.2
nltk	3.7
seaborn	12.0b3
simpletransformers	0.60.3
torch	2.1.1
summa	1.2.0
string	1.0.0
sys	3.8
wordcloud	1.9.3
collections	0.3.0
gensim	4.3.2
apyori	1.1.2
spacy	3.7.2
requests	2.31.0

After importing the libraries, the next step involves loading the dataset into Colab. While it was previously mentioned that the dataset could also be imported from Kaggle, our preference is to load the dataset directly from Drive in the form of a dataframe. This approach is more straightforward and convenient than importing from Kaggle because we eliminate the need to share potentially sensitive passwords via Colab.

Once the dataframe is loaded, the exploratory analysis of the data may begin. This step is crucial for gaining an understanding of the dataset and determining the necessary measures for achieving optimal results. Using the *dtypes* attribute of the dataframe, it is observed that the "Review" column is of type *object* (representing the string data), while the "Liked" column is of type *int64*, indicating binary labels (0 or 1).

A search for missing values is also conducted, but the dataset appears well-processed, requiring no imputation methods. Additionally, we check the data for duplicates by grouping the dataframe based on the "Liked" column and utilizing the *describe()* attribute. This analysis reveals the presence of duplicates in the dataset, with two instances occurring twice: one negative review, *"The food was terrible"*, and one positive review, *"I love this place"*. Despite these two reviews not holding particular insights that could improve a restaurant, we decided to include the duplicates nonetheless. We also established that it is not far-fetched to assume that two separate reviewers used the same sentence to describe the restaurant, as the used phrases are quite general and not particularly complex.

The last step of the exploratory analysis is to check the balance of the dataset. If we used classifiers, the balance would help us avoid bias towards a certain class in our models. Even if that is not necessarily the case here, the number of favorable reviews compared to unfavorable ones can help us assess how dire the situation is. If there are overwhelmingly more positive reviews, the improvements would be minor, and if the reviews are mostly negative, then there is a desperate need for a change.

Unfortunately, we were not able to form a pertinent assessment of the situation, as the results show that the dataset is perfectly balanced, with exactly 500 instances in each class. Even though it would have been useful for us to have a clear starting point, we can still infer that most aspects of the restaurant are average. A visualization of the dataset balance can be seen in Figure 1.

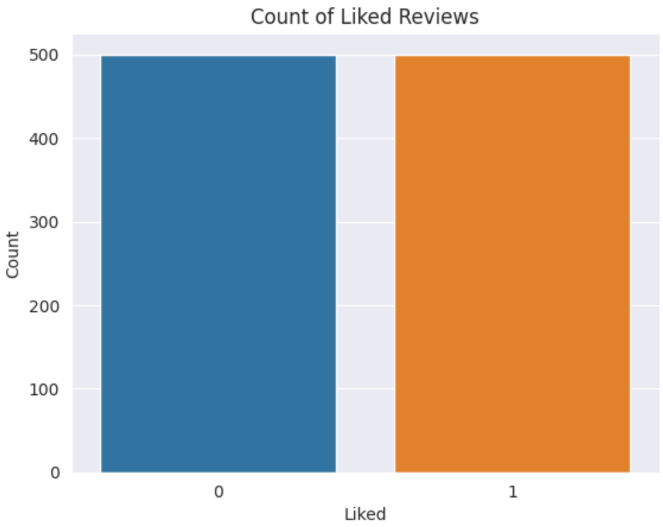

Fig. 1. Count of negative and positive reviews

With the exploratory analysis now concluded, we can proceed to the preprocessing stage. It is important to mention that, while some Text Mining approaches utilize raw data, methods like Word Frequency necessitate the use of preprocessed data to yield relevant results.

The initial task is to separate the reviews from their corresponding labels, focusing on preprocessing the textual content exclusively. The first operation involves **tokenization**, which entails breaking down sentences into words and paragraphs into sentences. Given the concise nature of each review, tokenization primarily involves dividing phrases into individual words.

Following tokenization, the words are converted to lowercase to facilitate subsequent **stopword removal**. Stopwords, commonly used words devoid of significant meaning, serve mainly as connectors between other, more telling structures. As they contribute no insights, they are eliminated by comparing each tokenized word with the stopword dictionary from the *nltk* library. Consequently, words such as "is," "in," "they," or "not" are removed from all reviews. Although the resulting "review" may lack grammatical coherence, it retains the more meaningful words.

After stopword removal, punctuation is also discarded as it does not offer discernible insights or trends for analysis. Numbers and extra spaces are also eliminated, ensuring thorough data cleaning. Notably, after manual inspection, there have been quite a few instances in which grammatical errors were encountered. We did not find that the existence of misspelled words affects the selected Text Mining methods, so no measures were taken to correct the mistakes.

The final step in review preprocessing involves **lemmatization**, a process in which words are reverted to their base structure. Typically, lemmatization removes plural forms of nouns and other similar transformations. During this process, it is important to note that the part of speech remains unchanged.

After lemmatization, a new dataframe is created containing the preprocessed reviews and their corresponding labels, marking the completion of the preprocessing step. As a result, we possess all the required elements to commence the application of Text Mining methods.

IV. Length Analysis

Our initial hypothesis is that negative reviews exhibit greater length compared to positive ones. If true, this theory would not only facilitate the task of improving the restaurant (because longer negative reviews would mean more feedback) but also enable the discovery of a correlation for rapid classification. The underlying idea behind this theory stems from the observation that individuals leaving highly negative reviews often seek to vent and express their frustration, while those with positive experiences tend to provide more concise details.

The first step in this analysis is to compute the length of each review. We used the raw reviews for this calculation, because we considered that the results would not be particularly relevant otherwise. The *len* function is applied to the "Review" column of the dataframe, and the results are appended to a new column named "Length."

Utilizing the *describe()* attribute, various statistical values, including count, minimum value, maximum value, standard deviation, and quantiles, are computed. The analysis reveals that the average review has 58 characters, with a standard deviation of 32. The longest review spans 149 characters, while the shortest contains just 11.

To illustrate, we present the longest and shortest reviews. The lengthiest review is *"The problem I have is that they charge 11.99 dollars for a sandwich that is no bigger than a Subway sub (which offers better and more amount of vegetables)"* . We may notice that it is indeed a negative review, as our initial theory would suggest.

Upon inspecting the shortest review, we find more than one. In fact, there are five reviews, each with only 11 characters. Examples include *"DELICIOUS!!"*, *"I LOVED it!"*, *"Over rated."*, *"Great food."*, and *"Both great!"*. Notably, four out of the five shortest reviews are positive.

At this juncture, it appears that our theory holds validity, with negative reviews generally displaying more characters. However, the observed pattern deviates from the initial expectations upon visualizing the histograms. The histograms shown in Figure 2 represent the distribution of the number of characters for positive and negative reviews.

As depicted in the illustration, the majority of positive reviews exhibit approximately 40 to 60 characters, while most negative reviews are limited to around 20 to 50 characters. This variation in length suggests that extracting a rule based on length alone may not effectively distinguish between the two classes.

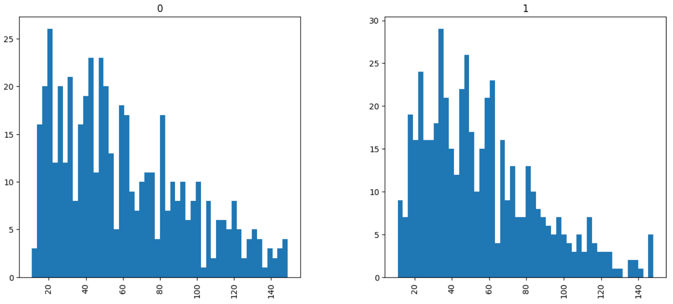

Fig. 2. Histograms for the number of characters

Despite this, we deemed it imperative to conduct a correlation analysis to conclusively dismiss our initial theory. Consequently, we generated a heatmap using the *seaborn* library, showcased in Figure 3. The correlation coefficient of -0.075 between length and the two classes indicates an insignificant connection, hence unfortunately discrediting our initial hypothesis.

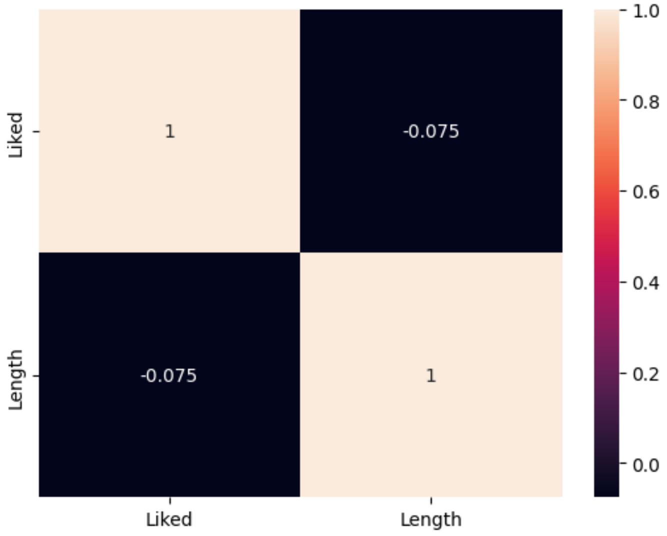

Fig. 3. Correlation between length and positive/negative reviews

V. Word Frequency and Wordclouds

We suppose that identifying the most prevalent words within the reviews may hold certain insights about what could be improved. Naturally, we conduct this assessment using preprocessed reviews to ensure the absence of stopwords, as the presence of such words would distort the results. Utilizing preprocessed reviews ensures a more insightful outcome, as the most frequent words will not be stopwords.

We computed the word frequency for three distinct categories: the most frequent words across all reviews, and separately for positive and negative reviews. This approach allows for a comprehensive analysis from different perspectives.

Initially, the data is merged into a single string variable by concatenating all rows from the "Review" column. Subsequently, the string is split into words, akin to the tokenization process. With the words separated, frequency calculation is accomplished using the *Counter* function from the *collections* library. For our implementation, we opted to display the first 50 most frequent words, because we wanted the ability to sift through them ourselves. Additionally, we employed the *matplotlib* [9] library to aggregate the results into a word cloud, where word size indicates frequency.

The word cloud for all reviews is visualized in Figure 4.

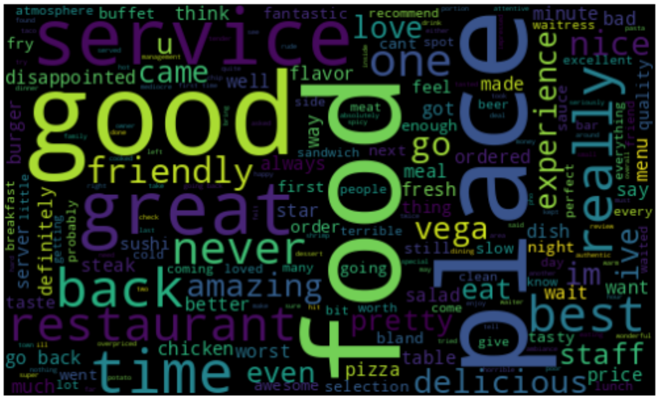

Fig. 4. Most frequent words in all reviews

From the aforementioned figure, several conclusions can be derived. The most frequently occurring word across all reviews is *"food"*, with a frequency of 125. This is unsurprising, given that food constitutes the most important aspect of any restaurant and is the primary consideration for clients. The term *"place"* is also notably prevalent, closely followed by *"good"*. While we could potentially associate the adjective with the two nouns, implying that the food and place are described as *"good"*, we realistically have no way of knowing at this point. *"Service"* is another commonly mentioned aspect across all reviews, as is *"time"*. Given the inclusion of all reviews, the sentiment conveyed by these words, whether positive or negative, remains unknown.

An intriguing component worth discussing is the inclusion of the word *"vega"* among the most common words in all reviews. While this word does not exist, this apparent error can be attributed to the lemmatization algorithm. The lemmatizer considers *"Vegas"* to be a plural form, reducing it to the base form *"vega"*. Despite being an error, this information remains valuable for various reasons. Firstly, it underscores that proper nouns can pose challenges in lemmatization. Secondly, it implies that many reviews may mention the restaurant's location, specifically the city of Las Vegas.

Given the previous results, it becomes imperative to analyze the most common words from positive and negative reviews separately. This approach enables us to identify which aspects of the restaurant are already commendable and which areas may require improvement. Employing the same algorithm as previously described, we computed the most frequent words from reviews with "Liked" values of 1 or 0.

Let's first examine the words from positive reviews, visualized in Figure 5 as a word cloud.

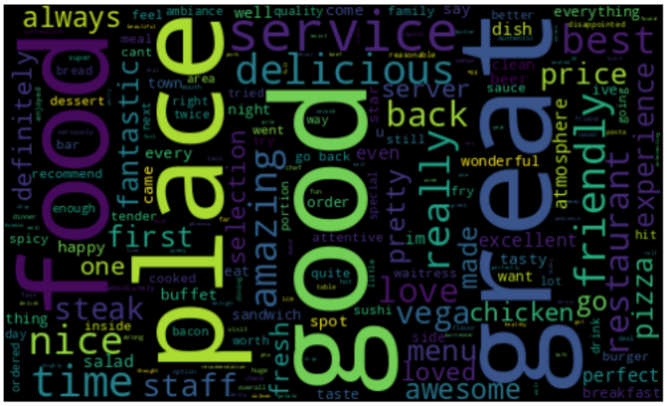

Fig. 5. Most frequent words in positive reviews

The terms *"good"* and *"great"* are encountered in both the current word cloud and the previous one, alongside various other words. Additionally, the term *"food"* reappears, and it is anticipated to also feature in the word cloud dedicated to negative reviews.

Upon examination of the existing word clouds, it becomes evident that the most prevalent words are rather generic and offer limited insights into more specific aspects of the restaurant that garnered positive or negative sentiments. To provide more targeted recommendations for the restaurant's improvement, we manually selected words that we deemed most relevant and insightful. The chosen words for the most relevant common terms in positive reviews can be found in Table III.

TABLE III
RELEVANT FREQUENT WORDS IN POSITIVE REVIEWS

Word	Frequency
service	46
friendly	23
steak	14
price	13
pizza	12
selection	10
chicken	10
atmosphere	10
salad	9

The initial essential term, *"service"*, indicates that some customers express satisfaction with the performance of the waitstaff. However, considering that this word is also present in the word cloud for all reviews, it is reasonable to infer that opinions about the service are somewhat divided.

Another term remotely linked to service is *"friendly"*, implying that the staff exhibited hospitality toward customers.

The term *"price"* is also frequently mentioned in positive reviews, suggesting that the services offered are reasonably priced, which is a favorable aspect. Additionally, customers included the terms *"selection"* and *"atmosphere"* in their positive reviews, indicating that the variety of food offered is suitable, and clients perceive the ambiance favorably.

We find the common words associated with specific dishes offered by the restaurant to be particularly informative. Positive reviews often reference terms like *"pizza"*, *"steak"*, and *"salad"*, highlighting the significance of these dishes in the restaurant's menu.

Transitioning to the negative reviews, our focus will initially be on the word cloud presented in Figure 6.

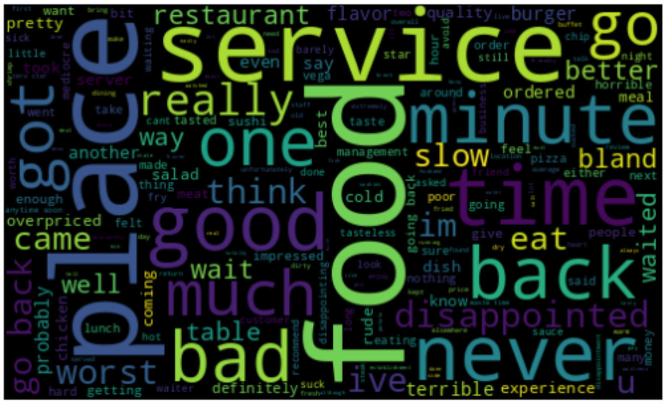

Fig. 6. Most frequent words in negative reviews

As previously suspected, the term *"food"* emerges once again as one of the most frequently used words, alongside *"service"*. This potential inconsistency in the waiting staff's efforts may indicate that some employees deviate from the established standard. Addressing this issue would necessitate an internal investigation or, at the very least, closer monitoring of the waiting staff and their performance.

Similar to our approach with positive reviews, we have also curated a selection of the most pertinent frequent words in negative reviews. This enables us to pinpoint the primary weaknesses of the restaurant more precisely. The summarized results of our selection are presented in Table IV.

TABLE IV
RELEVANT FREQUENT WORDS IN NEGATIVE REVIEWS

Word	Frequency
food	65
service	38
time	29
slow	11
wait	11
bland	11
burger	10
flavour	10

The term *"food"* appears frequently with negative connotations, with 65 out of 125 instances occurring in negative reviews. This prevalence suggests that the food falls short of meeting the needs and preferences of the customers. Supporting this notion are other relevant words, such as *"bland"* and *"flavor"*. The inclusion of these terms underscores the possibility that customers desire food with more spices and condiments, highlighting an area for consideration by the chefs and their assistants. Notably, among the entire menu selection, the burger appears to be a particular source of dissatisfaction, featuring prominently in unfavorable reviews.

In addition to the negative common words associated with food, another recurrent theme pertains to the performance of the waitstaff. The term *"service"* is mentioned 38 times. While the staff was characterized as friendly in positive reviews, negative reviews reveal that the primary complaint revolves around the time spent waiting for dishes to be cooked and served. Customers emphasize the slow service, a crucial aspect that warrants attention from the management. Implementing strategies such as preparing some ingredients in advance could help improve waiting times and enhance overall customer satisfaction.

VI. Keyword Extraction

The next Text Mining technique applied to the dataset involves the extraction of keywords from the reviews. While it might appear that we already accomplished this task by identifying the most frequent words, this assumption is not entirely accurate. Even though the words presented previously may be common, they are not necessarily keywords, because a keyword captures the essence of a particular subject. Therefore, not every frequent word is a keyword, and not every keyword needs to be frequent.

To extract keywords from the reviews, we employed two distinct algorithms, each described in the following paragraphs along with their respective outcomes. The rationale behind utilizing two algorithms lies in the desire to assess the consistency of the results.

The initial algorithm employed is **Term Frequency-Inverse Document Frequency**, abbreviated as TF-IDF. As the name implies, this approach is rooted in frequency but calculates a specific score instead of merely tallying common words. The methodology can be dissected into distinct steps: Term Frequency and Inverse Document Frequency.

The **Term Frequency** (TF) aspect of this algorithm involves calculating the weight of a term (word) based on its frequency. The primary principle underlying this step states that, as the length of a document increases, a specific word is more likely to appear in that document. Consequently, the frequency of a word needs to be standardized, or divided by the length of the document. As a result, TF is computed using the formula from Equation 1.

$$TF = \frac{\text{Word Frequency in document}}{\text{Total number of document words}} \qquad (1)$$

Unlike TF, **Inverse Document Frequency** (IDF) is calculated across all documents. As the word's frequency increases across all documents, its weight decreases, facilitated by the logarithmic operation. Much like TF, IDF is also a ratio that can be computed using Equation 2.

$$IDF = \log \frac{\text{Total number of documents}}{\text{Documents where the word is mentioned}} \tag{2}$$

With the TF and IDF values already calculated, the final step is to compute the TF-IDF. This entails a straightforward multiplication process, as illustrated in Equation 3.

$$\text{TF-IDF} = TF \cdot IDF \tag{3}$$

For our implementation, we utilized the *TfidfVectorizer()*, a feature extraction tool available in the *sklearn* library. This tool simplifies the application of the TF-IDF algorithm, requiring minimal code to achieve the desired results. The vectorizer yields keywords for each review along with their corresponding scores. However, our interest extended beyond individual reviews, as we aimed to identify keywords representing the entire collection of reviews. Therefore, we opted to display the most common keywords across all documents and calculate the mean of their scores. The top 10 keywords obtained through TF-IDF are succinctly presented in Table V.

TABLE V
TF-IDF KEYWORDS

Keyword	Score
food	0.0333
good	0.0300
place	0.0291
service	0.0269
great	0.0237
back	0.0231
time	0.0152
go	0.0149
like	0.0129
really	0.0109

As evident from the keywords, there is a notable resemblance between them and the most frequent words identified for all the reviews in the preceding section. Naturally, we anticipated words like “food,” “place,” and “service” to be included in this list, as these words would also be considered keywords through human inspection. However, we find that the tool did not meet our expectations, given that many of the present words lack significant meaning outside of their original sentences.

Considering the suboptimal results of the TF-IDF algorithm, we deemed it necessary to apply an alternative method for keyword extraction, hoping for more satisfactory performance. The second approach employed for keyword extraction was **TextRank**, a graph-based tool introduced in 2004 [10]. Functioning similarly to Google’s PageRank system, TextRank can be utilized for both keyword and sentence extraction.

Although there are limited resources offering detailed insights into how the approach functions, we will endeavor to explain the methodology to the best of our abilities. Initially, a graph is constructed by connecting words that frequently appear together in the corpus. For instance, we might expect to find the word *“good”* associated with *“food”*. However, *“good”* could also be linked to *“service”*, *“atmosphere”*, and other related nouns. In this context, *“good”* is considered a keyword for the analyzed document.

After all words are integrated into the graph, keywords are selected by evaluating which words are consistently used across various formulations. The resulting score essentially represents the weight of the graph vertex.

In the implementation context, the TextRank approach was applied using the *keywords* package from the *summa* library. Once again, the required code is minimal, as passing the reviews in string format to the *keywords()* function serves as the primary step.

The keywords obtained through TextRank, along with their respective weights, are presented in Table VI.

TABLE VI
TEXTRANK KEYWORDS

Keyword	Score
place	0.32
service	0.23
loved place	0.22
food amazing	0.21
liked	0.17
good tasty	0.15
love	0.12
lovely	0.12

As observed, there are keywords extracted by both TF-IDF and TextRank, specifically *“place”* and *“service”*. Similar to the *TF-IDF* algorithm, some of the other words do not accurately represent the content of the reviews as one might expect. Notably, TextRank appears to favor words like *“loved,” “love,”* and *“lovely,”* as indicated by their prevalence in the top 10 keywords.

We consider that the words *“food,” “place,”* and *“service”* serve as keywords for the reviews, offering insights into the elements that restaurant customers perceive as crucial.

VII. N-GRAM ANAYSIS

We also tried gaining insights by examining the frequent combinations of words to understand how clients describe the restaurant’s services. To achieve this, we conducted an **N-gram** analysis, focusing on bigrams and trigrams, which are associations of two and three words, respectively. Increasing the N-gram size beyond three did not yield relevant results suitable for presentation in this report.

To identify the N-grams, we began by breaking down the reviews into individual words. Utilizing the *ngram* function from the *nltk* library, we generated lists of bigrams and trigrams. The only modification involved changing the function’s parameter from 2 to 3. We also calculated the frequency of each bigram and trigram using the *Counter* from the *collections* library. For bigrams, we presented the 50 most frequent structures, while for trigrams, we displayed the top 20. It is worth noting that trigrams are less common than bigrams, given that associations of three words are not as prevalent as those of two words.

The most pertinent bigrams, along with their frequencies, are outlined in Table VII.

TABLE VII
RELEVANT BIGRAMS

Bigram	Frequency
go back	18
good food	9
great service	6
food delicious	5
friendly staff	4
bad food	4
waste time	4
slow service	3
good price	3

The most prevalent bigram, *"go back"*, suggests that many customers convey whether or not they would consider seeking the restaurant's services again in their reviews. Additionally, notable bigrams include *"good food"* and *"bad food"* indicating divided opinions on the dining experience. It's noteworthy that the positive association is twice as common as the negative one. Contradictory sentiments are expressed through *"great service"* and *"slow service"* with the negative bigram further supported by the association *"waste time"*.

However, a majority of the encountered bigrams convey positive connotations, such as *"food delicious"*, *"friendly staff"*, and *"good price"*. This observation implies that negative reviews may employ more distinctive associations when describing the services, while positive reviews tend to utilize similar descriptors.

Moving on to trigrams, a less extensive category compared to bigrams, we can refer to the findings presented in Table VIII.

TABLE VIII
RELEVANT TRIGRAMS

Trigram	Frequency
back anytime soon	3
service extremely slow	2
running around like	2
give zero star	2
first vegas buffet	2

Upon trigram examination, it becomes apparent that the majority of them convey negative sentiments. The trigram *"back anytime soon"* indicates that some reviewers express their dissatisfaction with the restaurant's services to the extent that they would not consider revisiting. Once again, complaints related to the staff are evident, as inferred by the trigram *"service extremely slow"*. Given the recurrence of this shortcoming across various Text Mining methods, we can confidently assert that the sluggish service speed stands out as one of the restaurant's major issues.

It's also noteworthy to highlight the trigrams *"running around like"* and *"give zero star"*, both of which reflect a considerable amount of frustration from customers concerning the restaurant. As previously mentioned, the primary concern for the restaurant appears to be the subpar performance of its staff.

VIII. ASSOCIATION RULES

While we've delved into the prevalent word associations derived from the reviews, the limitation of N-gram analysis lies in its focus on words that are adjacent in a sentence. Our hypothesis asserts that certain words may be connected by virtue of being present in the same review, irrespective of their proximity. This is precisely where association rules methods come into play.

Association rules mining is a type of Data Mining task that is specifically designed for the analysis of transactions [11]. The main idea of this approach is to pinpoint the co-occurrence of items or sets of items in lists of transactions. This concept is the foundation of the "Frequently Bought Together" sections that exist on e-commerce websites.

As the name suggests, the final results of this technique involve a set of rules that contain the items of the transactions. The quality of the established rules is quantified using three specific measures: support, confidence and lift [12].

The **support** metric shows the reliability of an association rule, by taking into consideration how many times the items are indeed bought together out of all transactions. To put it plainly, the support is the frequency of an item set.

Let X and Y be two items that exist in a list of transactions. In that case, the support of X and Y will be calculated using the following formula:

$$\mathit{Support}\{X,Y\} = \frac{\mathit{Nr.\ of\ transactions\ with\ both\ X\ and\ Y}}{\mathit{Total\ nr.\ of\ transactions}}$$

Confidence is calculated through the support metric and is meant to present how probable the co-occurrence of the items in a transaction truly is. It is worth mentioning that the confidence threshold must be supplied by the user, as he decides the minimum value that is still considered acceptable.

Taking into consideration the notations previously used in the case of the support metric, confidence is computed using the following equation:

$$\mathit{Confidence}\{X,Y\} = \frac{\mathit{Support}\{\text{X, Y}\}}{\mathit{Support}\{\text{X}\}}$$

Also called "interest", the **lift** metric is supposed to quantify the novelty of a rule. If a rule is interesting and can be used to gain insights, its lift would not be close to 1. Therefore, in order for the lift to be considered satisfactory, its value must be either lower or higher than 1.

Like confidence, lift is also calculated by using support and by applying the formula:

$$\mathit{Lift}\{X,Y\} = \frac{\mathit{Support}\{\text{X, Y}\}}{\mathit{Support}\{\text{X}\}*\mathit{Support}\{\text{Y}\}}$$

One of the most common algorithms employed for generating rules for frequent item sets is the **Apriori algorithm** [13]. The first step of the algorithm is to compute the support of each item individually. After analyzing the results, a support threshold must be established, as the items that have values lower than the threshold value cannot be considered frequent.

Subsequent to the removal of the infrequent items, the support of each item set is computed. An item set is constructed by making all possible combinations between the frequent items. It is best practice to calculate the support of itemsets with two items first and then move on to three items. Therefore, the last item set will be the largest, containing all frequent items.

Upon performing all the necessary computations, what is left to do is to generate the association rules and compute the confidence for each rule. After confidence is calculated, the last step is to compute lift.

For the implementation, we employed the *apyori* library, which encompasses the apriori algorithm. However, before applying the algorithm, we need to structure all the reviews into a nested list, or a list of lists. Initially, we tokenize the reviews to differentiate them. Subsequently, using the *tolist()* function, we aggregate all the lists containing the reviews into a larger list.

Following this, using the apriori algorithm, we compute rules that exhibit a minimum support of 0.0022, a minimum confidence of 0.20, and a minimum lift of 3. By configuring these parameters, we were able to identify 172 distinct rules that meet the specified criteria. The sets are then displayed along with the support of each rule. The most relevant rules are presented in Table IX.

TABLE IX
RELEVANT ASSOCIATION RULES

Set	Support
{'rude', 'management'}	0.003
{'area', 'place'}	0.003
{'atmosphere', 'service'}	0.003
{'clean', 'friendly'}	0.003
{'selection', 'beer'}	0.004
{'service', 'food', 'great'}	0.006
{'going', 'anytime', 'soon', 'back'}	0.003

The third rule implies that individuals mentioning the service may also reference the atmosphere of the restaurant. Customers who prioritize the cleanliness of a place are likely to be concerned about both the staff's attitude and the quality of food, often featuring both aspects in the same review. Once again, it is a common practice to disclose whether or not patrons intend to revisit a restaurant, as highlighted by the rule *'going', 'anytime', 'soon', 'back'*.

A particularly intriguing set is the *'selection', 'beer'*, suggesting various possibilities. On one hand, it might indicate that the diversity of beer offerings is a noteworthy topic in reviews, irrespective of the sentiment. On the other hand, it could imply that customers who value the food selection also express interest in beer, although this is less likely.

In summary, this section primarily served to reinforce our insights for restaurant improvement, largely affirming aspects and shortcomings we had previously suspected.

IX. REVIEW CLASSIFICATION WITH TRANSFORMERS

Although our primary focus revolves around immediate enhancements for the restaurant, it is essential to recognize that ongoing reviews will continue to present new concerns requiring attention. As a result, we found it pertinent to implement a classification model on the dataset, a valuable tool for businesses to swiftly discern client preferences and grievances within reviews.

For the task of classifying future reviews into negative and positive categories, we utilized the *ClassificationModel* from the *simpletransformers* library, known for delivering excellent results in various NLP tasks. The documentation emphasizes the significance of using the text as it is, with minimal preprocessing. Our only preprocessing step involved dividing the dataset into a training set (75% of reviews) and a validation set (25% of reviews).

To enhance the reliability of our results, we opted for K-Fold cross-validation with 5 folds. BERT served as the base transformer due to its promising results, despite the longer training time compared to RoBERTa or Electra. The chosen performance metric was accuracy, and the training process extended over 3 epochs.

It's worth noting that the training process was time-intensive, lasting nearly 5 hours. While transformers generally demand substantial training time, this limitation was exacerbated by the unavailability of Cuda in Google Colab.

In terms of results, the first, third, and fifth folds achieved 96% accuracy each, while the second and fourth folds demonstrated higher accuracy at 98% and 97%, respectively. The mean accuracy reached an impressive 97%, signifying robust performance for this specific task. In the testing phase, the restaurant can input new reviews into the classifier, which will automatically categorize them as either negative or positive.

X. LATENT DIRICHLET ALLOCATION

Latent Dirichlet Allocation (LDA) is a topic modeling tool, characterized as a "three-level hierarchical Bayesian model," specifically tailored for handling textual data. The fundamental concept underlying LDA is the notion that a document can be separated into distinct topics, each characterized by a unique word distribution [14].

The implementation of LDA in Python heavily relies on the *gensim* library, leveraging its *LdaModel()* for actual topic categorization and *CoherenceModel()* for assessing how well the words align with the established topics. A higher coherence score indicates a more effective topic split.

In our specific exploration, we experimented with various potential topic splits, and the outcomes can be characterized as moderately satisfactory. The most successful split, comprising 9 topics, achieved a coherence score of 0.52, marking the highest score attainable.

Visual representations of the 9 topics are presented through word clouds in Figure 7, and each topic will be thoroughly examined in the ensuing paragraphs.

Topic 1 appears to center around the performance of the employees, featuring terms like *"service"*, *"staff"*, and

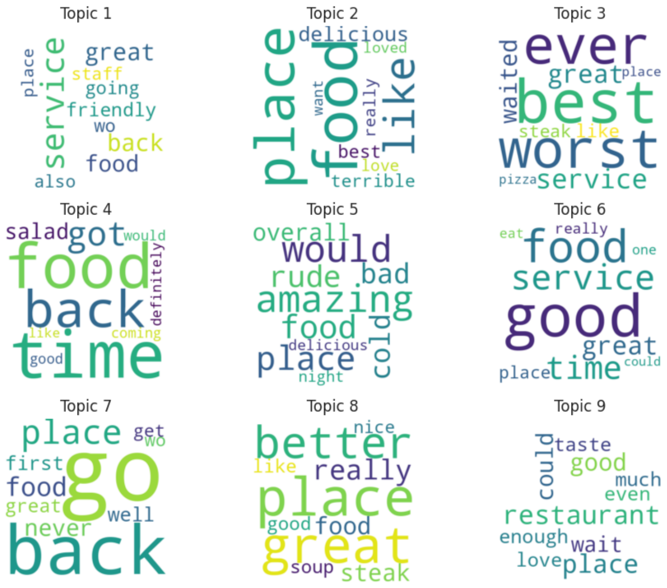

Fig. 7. Most relevant words per topic

"friendly". Particularly noteworthy is the term *"service"*, which holds the highest weight in this topic with a value of 0.049. Topic 3 also delves into the realm of service but extends its focus to include aspects related to food, incorporating words like *"pizza"* and *"steak"*. Similarly, topic 6 explores service but from the perspective of the ambiance and overall atmosphere.

Topics 2, 6, and 4 prominently emphasize food, with the first two adopting a positive tone, featuring words like *"delicious"* and *"great"*, while topic 4 appears to take a less favorable approach, that criticizes the prolonged wait time.

Topics 7 and 8 revolve around the restaurant's physical space and the likelihood of patrons returning. Notably, topic 9 stands out as an outlier, housing words that resist easy classification into any specific category.

While the division into 9 topics yielded the highest coherence score, we find the results challenging to interpret, with topics lacking clear distinction. This difficulty may be attributed to factors like the briefness of reviews, disjointed themes, and repetitive expressions. Despite these limitations, we acknowledge LDA as a valuable tool for uncovering overarching themes in restaurant-related content. However, the information derived lacks novelty, given the application of previously employed methods.

XI. Named Entity Recognition

Utilizing **Named Entity Recognition** (NER), our objective is to pinpoint proper nouns referenced in both negative and positive reviews. This approach allows us to identify the names of employees frequently cited with positive or negative sentiments, as well as the names of dishes featured in the reviews.

To execute this task, we leverage the *spacy* library along with its tools. Although there is an option to apply NER using the model from the *simpletransformers* library, we find its performance unsatisfactory, coupled with significant training time. That model exhibits a notable flaw of either labeling all structures as "O" (indicating a non-entity) or incorrectly classifying non-entities as entities.

Commencing with the first step, we import the *spacy* and *requests* libraries, adjusting the *pandas* display to 200 to prevent truncated output. Employing the raw data, we apply the *ent* attribute, extracting the word, start character, end character, and the assigned label. The text and label values are then stored in a dataframe for subsequent queries.

NER offers an array of labels for classification, each of which will be presented and discussed in the subsequent sections.

Firstly, the **GPE** label, denoting Geopolitical Entities, identifies the locations mentioned in the reviews. Additionally, we calculate the frequency of these terms to gauge their relevance. The list of pertinent locations along with their corresponding frequencies is detailed in Table X.

TABLE X
Relevant Entities with label GPE

Location	**Frequency**
Vegas	16
Phoenix	3
Edinburgh	1
Philadelphia	1
Tucson	1

Given that a majority of reviews reference Las Vegas as a location, it is reasonable to infer that the restaurant is likely situated in that area. Moreover, the inclusion of other locations such as Phoenix, Edinburgh, and Philadelphia might suggest that reviewers are drawing comparisons with restaurants in different geographical locations. This supposition gains support from the notable discrepancy in frequency between Las Vegas and the other places labeled GPE.

In addition, insights into individuals mentioned in the reviews, such as waiters, chefs, and sous-chefs, can be garnered. To achieve this, we examine the outcomes of NER for the **PERSON** label. The relevant individuals identified are outlined in Table XI.

TABLE XI
Relevant Entities with label PERSON

Person	**Frequency**
Rick Steve	1
Maria	1
Gordon Ramsey	1
Jeff	1
Otto	1

Individuals mentioned solely by their first names in the reviews (*Maria*, *Jeff*, *Otto*) are likely members of the restaurant staff, considering that the alternative scenario involves reviewers naming their dining companions, which is very unlikely. Therefore, it is plausible that certain reviews specifically mention the server's name.

Regarding the two names that include both the first and last names, identification is straightforward. *Gordon Ramsey* is a renowned chef and restaurateur widely recognized for his role in the Hell's Kitchen reality TV cooking show [15]. This suggests a potential connection between the restaurant and Gordon Ramsey, who is known to own prominent establishments on the Las Vegas Strip.

As for *Rick Steve*, the closest match found is the TV personality and travel book author Rick Steves [16]. It is reasonable to assume that the missing "s" is due to the lemmatization process. Furthermore, the association with this travel expert implies that some reviewers may have visited the restaurant based on his recommendation.

Moving on to the **NORP** label, which stands for "Nationalities, religious or political groups," it is evident from Table XII that the majority of words classified under this category pertain specifically to nationalities.

TABLE XII
RELEVANT ENTITIES WITH LABEL NORP

Nationalities	Frequency
Thai	3
Greek	3
Mexican	2
Indian	2
Italian	2
Japanese	1
Jamaican	1

As evident from the preceding table, a diverse array of nationalities is mentioned in the reviews, which raises some eyebrows. While acknowledging that in a restaurant context, these may refer to specialties, the sheer number and diversity are noteworthy. This could potentially imply two scenarios: either the restaurant attempts to encompass a broad range of cuisines and, as hinted by earlier methods, often falls short, or the reviews are not exclusively related to a single restaurant but rather to various ones. It is essential to note that the dataset description was brief and vague, omitting this aspect, which cannot be definitively ruled out. At this point, the most prudent approach is to bear this information in mind.

Transitioning to the outcomes under the **ORG** label, we will now examine the organizations mentioned in the reviews. Table XIII provides insights in this regard.

TABLE XIII
RELEVANT ENTITIES WITH LABEL ORG

Organizations	Frequency
Bachi	2
MGM	1
Nobu	1
Google	1
Costco	1
Pizza Hut	1
Chipotle	1
Miraje	1
Subway	1

The table features entities not directly tied to dining, such as *"MGM"*, *"Google"*, and *"Costco"*. While one might argue that Costco is food-related as it sells edible products, it still doesn't qualify as a restaurant.

Fast-food chains like *"Subway"*, *"Pizza Hut"* and *"Chipotle"* can be identified in the reviews. Notably, since *"Subway"* is mentioned only once, we can ascertain that the review related to it was analyzed in the preceding section. In that instance, the reviewer compared the sandwiches of the restaurant with those from Subway, portraying the latter in a more positive light. We can reasonably infer that a similar scenario exists for the other two fast-food chains.

Certain restaurants are explicitly mentioned in the reviews, including *"Bachi"*, *"Nobu"* and *"Mirage"*. Any culinary enthusiast would promptly recognize these names as among the most renowned establishments on the Las Vegas Strip. Although The Mirage is technically a hotel, it also provides exquisite fine dining services, much like Nobu. Bachi Burgers, referred to as Bachi, is a now-closed restaurant in Las Vegas. The inclusion of these restaurants in the reviews supports the hypothesis that the dataset incorporates reviews from various restaurants.

The last pertinent label for our NER implementation is the **TIME** one, which can be examined in Table XIV.

TABLE XIV
RELEVANT ENTITIES WITH LABEL TIME

Organizations	Frequency
an hour	3
20 minute	3
40 minute	2
another 35 minute	1
45 minute	1
2 hour	1

Incorporating entities labeled as TIME holds significance in light of our earlier conjecture about waiting times. As observed, customers frequently mention waiting for an hour, with many instances indicating extended durations in the table concerning service. When patrons visit a restaurant, it's reasonable to assume they are hungry. In such cases, prompt service becomes crucial, as people tend to be highly impatient and prone to frustration when hunger is a factor. Additionally, the phrase "another 35 minutes" implies a gradual decline in service speed.

The NER method has yielded valuable insights, which will be leveraged in the final stage of our implementation to offer more targeted suggestions. It's important to note that the results presented in this report are selected from NER output, which may not always be entirely accurate. We've observed instances where words containing grammatical errors and **Out-of-Vocabulary** (OOV) words are mistakenly classified as entities. This is one of the considerations when applying NER.

XII. CATEGORY-BASED REVIEW FILTERING

Having thoroughly examined the overarching themes and frequently mentioned words in the reviews, we must now formulate relevant improvements for the restaurant. To achieve this, we conducted targeted queries on the reviews based on various aspects highlighted in the preceding sections of this report.

Upon querying the dataframe, it became evident that the burger was among the most criticized items on the menu, drawing predominantly negative reviews. The burger was often characterized as *"bland"*, *"overcooked"* and *"cold"*, leaving patrons dissatisfied. Our recommendation is to either eliminate the burger from the menu entirely or enhance its recipe. Notably, one reviewer expressed a desire for the burger to have *"more charcoal flavor"*, which could be a valuable consideration.

Customers also exhibited a degree of dissatisfaction with dishes containing chicken, including chicken Pho, roasted chicken, and chicken tenders. Despite positive reviews, the negative ones were strongly critical, citing issues such as *"dry and soggy"*, *"undercooked"* and *"unbalanced"*. While removing these dishes from the menu may not be practical, our advice is to improve the recipes and ensure all meals are served hot.

Another contentious food category is the salad, receiving mixed reviews ranging from descriptions like *"tasty"* and *"refreshing"* to frequent characterizations as *"bland"*. Numerous reviews expressed frustration about the extended wait times for salads, with some customers even reporting that the ordered salads never arrived. Our suggestion is to add more spices and condiments to the salad, accompanied by increased focus on order management to expedite delivery.

On a positive note, the steak and pizza garnered widespread acclaim from customers, often described as *"perfectly cooked"*. As a key takeaway, we recommend maintaining the recipes for steak and pizza, as altering them could risk losing valued customers.

Transitioning from the discussion of food, it is evident that customer attention frequently centers on the quality of service. While opinions vary, a prevailing sentiment among reviewers is that the service is *"slow"*, *"lacking"*, *"sub-par"*, *"atrocious"*, *"terrible"* and *"beyond bad"*. Addressing the primary concern of slow service is crucial, as it consistently generates dissatisfaction among customers.

Within the same realm, we sought to explore customers' opinions on the management, and the sentiments echoed those expressed about the service. According to reviewers, the management is perceived as *"rude"*, *"of poor quality"*, and *"with attitudes that grow rapidly"*. We recommend exercising caution in handling upset customers and prioritizing training to de-escalate situations rather than exacerbate them.

On a positive note, the staff received overwhelmingly favorable reviews, being described as *"pleasant and enthusiastic"*, *"very friendly and efficient*, *"helpful"*, *"courteous"* and *"attentive"*.

Further investigation involved querying the dataframe for names of wait staff identified through NER. Maria and Jeff were identified as waiters, with positive reviews commending their exceptional service. However, Otto emerged as the name of a restaurant, supporting the theory that the dataset includes reviews for multiple establishments. Notably, the review mentioning Otto began with *"The staff at Otto always makes us feel welcome"*, indicating that Otto is the restaurant under review. In contrast, another review mentioning *"Bachi"* stated, *"I went to Bachi Burger on a friend's recommendation and was not disappointed"*. In conclusion, the dataset indeed encompasses reviews for multiple restaurants, explaining the divergent opinions. However, for the purpose of this analysis, we will continue to treat them as pertaining to a single establishment.

We conducted searches in the dataframe for mentions of Gordon Ramsay and Rick Steves, and the reviews corroborated our suspicions. Customers expressed sentiments like,*"Sadly, Gordon Ramsey's Steak is a place we shall sharply avoid during our next trip to Vegas"* and *"Stopped by during the late May bank holiday off Rick Steve recommendation and loved it"*.

Considering that atmosphere and ambiance are vital aspects of a restaurant, we also queried the dataframe for insights into these elements. Opinions about the ambiance were evenly divided, with some expressing dissatisfaction and others commending it. However, when it came to the atmosphere, references were solely positive, with descriptors like *" modern and hip"*, *"exquisite"*, and *"fun"*. Consequently, we have no specific suggestions for alterations to the atmosphere.

While frequently mentioned positively, opinions about the pricing are also divided. Some customers express dissatisfaction, attributing it to inflation, while others find the restaurant affordable. We acknowledge that pricing is subjective and varies from person to person.

Price often intertwines with portion size, as assessing the price-quality (and quantity) ratio is crucial. Some reviews describe portions as *"huge"*, while others lament that portions are diminishing as prices increase.

We also explored whether music is a component of the restaurant experience, uncovering reviews that mention it. The majority of customers speak positively about the music, describing it as *"wonderful"* and *"pleasant"*. We advocate for incorporating live music at the restaurant, as it enhances the overall ambiance. It is essential, however, for the volume to be suitable, as excessively loud music can impede conversations.

Decor is occasionally mentioned in favorable terms, albeit infrequently. Regarding the cleanliness of the establishment, customers generally agree that it is well-maintained. Even negative reviews acknowledge the restaurant's cleanliness, with concerns often centered around the freshness of the food.

XIII. Conclusions and suggestions

The purpose of this report was to employ diverse data analysis and Text Mining methods to offer constructive feedback for a restaurant, aiming at its enhancement. Several methods were utilized, including keyword extraction, Latent Dirichlet Allocation, and Named Entity Recognition.

Throughout this investigation, the inconsistency in the reviews became evident, with customers expressing divergent opinions on various aspects of the restaurant. It was later discerned that this divergence stems from the fact that the reviews pertain to a variety of restaurants, differing in locations and cuisines. We attribute this diversity to the nature of the dataset, which would have yielded more cohesive results with a more homogenous collection of reviews.

Notwithstanding this, we will proceed to discuss the compiled suggestions as if they pertain to a singular restaurant, aligning with the report's objective. The primary recommendations are presented in a list format for clarity:

1) Food Quality Enhancement
   - Drastically improve food quality, avoiding the service of cold or stale items, as customers readily discern such issues.
   - Consider altering recipes for burgers and dishes containing chicken, addressing common customer dissatisfaction. Recipe refinement is favored over permanent removal, given the importance of these dishes.
2) Management and Service Improvements
   - Implement changes in management and service, targeting issues of perceived rudeness and inefficiency.
   - Monitor employee performance closely, considering the removal of those not dedicated to their roles, with the ultimate goal of addressing slow service.
3) Price and Portion Considerations
   - Ensure that prices remain affordable, maintaining a competitive edge, while also balancing portion sizes.
   - Strive for portions that are neither wasteful nor insufficient, aiming for customer satisfaction and a reasonable profit margin.

References

[1] M. Luca, "Reviews, reputation, and revenue: The case of yelp. com," *Com (March 15, 2016). Harvard Business School NOM Unit Working Paper*, no. 12-016, 2016.

[2] O. Table, "The restaurant industry, by the numbers."

[3] T. O. Gallaway and J. Starkey, "Google drive," *The Charleston Advisor*, vol. 14, no. 3, pp. 16–19, 2013.

[4] A. Anwar, "Restaurant reviews."

[5] V. Wagh, S. Khandve, I. Joshi, A. Wani, G. Kale, and R. Joshi, "Comparative study of long document classification," in *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)*, pp. 732–737, IEEE, 2021.

[6] K. Stevenson, S. Elsegood, D. Seaman, C. Pawlek, and M. P. Nielsen, "Next generation library catalogues: reviews of encore, primo, summon and summa," *Serials: The Journal for the Serials Community*, vol. 22, no. 1, pp. 68–82, 2009.

[7] C. Aflori and M. Craus, "Grid implementation of the apriori algorithm," *Advances in engineering software*, vol. 38, no. 5, pp. 295–300, 2007.

[8] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural language processing: python and NLTK*. Packt Publishing Ltd, 2016.

[9] S. Tosi, *Matplotlib for Python developers*. Packt Publishing Ltd, 2009.

[10] R. Mihalcea and P. Tarau, "Textrank: Bringing order into text," in *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404–411, 2004.

[11] T. A. Kumbhare and S. V. Chobe, "An overview of association rule mining algorithms," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 927–930, 2014.

[12] F. Bao, L. Mao, Y. Zhu, C. Xiao, and C. Xu, "An improved evaluation methodology for mining association rules," *Axioms*, vol. 11, no. 1, p. 17, 2021.

[13] R. Agarwal, R. Srikant, *et al.*, "Fast algorithms for mining association rules," in *Proc. of the 20th VLDB Conference*, vol. 487, p. 499, 1994.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[15] B. E. Schreider, "Gordon ramsay: Scottish chef and restaurateur."

[16] R. S. Europe, "About rick steves."