MSA 8050 Project Proposal - Team 9

Darsani Alapati, Dev Banerjee, Manikanta Surapathi, Sai Charanya Ponnala

Part 1: Discovering and Rating Best Sushi in Restaurants Based on Text Analysis of Yelp Reviews

1. Abstract

The hospitality industry has been revolutionized with the advent of online platforms that allow customers to leave reviews of their experiences with restaurants. By analyzing these customer reviews, businesses can identify their strengths and weaknesses and make improvements accordingly. In this project, we present a novel approach to compare the highlight features of restaurants using topic modeling with PySpark.

Sushi has become an increasingly popular cuisine in recent years, with many restaurants offering their own unique takes on the classic dish. To help sushi lovers find the best sushi in their area, we present a data-driven approach to discovering and rating the best sushi in restaurants using Yelp reviews. Our approach involves using topic modeling to extract the most frequently occurring topics for a businessi in Yelp reviews. We then use a rating system to score each topic for the restaurant based on the frequency of the topic and rating of the reviews.

To test our approach, we applied it to a dataset of Yelp reviews for sushi restaurants. Our analysis revealed the most commonly discussed topics in restaurant reviews, as well as the restaurant to have the best sushi in. Our results demonstrate the potential for using data-driven approaches to discover and rate the best sushi in restaurants. By analyzing large volumes of customer reviews, we can provide valuable insights into the strengths and weaknesses of different restaurants and help sushi lovers find the best dining experiences, while also providing businesses valuable insights from consumer reviews. Overall, our approach provides a powerful tool for restaurant owners and managers to improve their businesses by identifying areas that need improvement and promoting their strengths. It can also assist customers in making informed decisions about which service or product to purchase, ultimately improving customer satisfaction.

2. Introduction

In today's world, customer reviews are more important than ever before. With the rise of online review platforms such as Yelp, customers can easily share their experiences with others and businesses can gain valuable insights into their operations. The vast amount of data generated by customer reviews presents both an opportunity and a challenge for businesses. While this data can provide valuable insights into customer preferences and satisfaction levels, it can also be overwhelming to analyze and make sense of.

In this project, a data-driven approach to analyzing customer reviews using PySpark has been implemented. For scope of this project, analysis was focused on discovering and rating the best sushi in restaurants using Yelp review data. We utilized two datasets: the Yelp Business dataset and the Yelp Reviews dataset. The Yelp Business dataset provided information about each business, such as its id and description, while the Yelp Reviews dataset contained the actual reviews left by customers.

The data was filtered to contain the businesses that were relevant to our analysis, and had about 80,000 reviews before preprocessing. The filtered data contained reviews for only the businesses that had sushi in their description, as we were interested in identifying the best sushi restaurants for the scope of this project. For the scope of finding the positive highlights about the business, only the reviews labeled with 4 or 5 star rating were considered.

Once we had our filtered dataset, we performed text preprocessing tasks prerequisite to topic modeling, such as removing punctuation and stopwords and tokenization, following which, for each business, important topics talked about in its reviews were extracted.

Following topic extraction, a distributed calculation algorithm was made to assess how popular each topic was for the business, based on its term frequency and review ratings, which resulted in a comparable popularity score for each topic relevant to the business.

Overall, our approach provides a comparable metric for consumers to choose the business best rated specifically for the product or service they need, and also provides a powerful tool for businesses to uncover insights from customer reviews and make data-driven decisions to improve their operations. The approach has also been modified later to calculate the most unpopular aspects, or 'lowlights' of a business. Combining the information from both highlights and lowlights of their businesses, owners and managers can plan to improve overall customer satisfaction, resulting in increased popularity and profits. All our analysis is based on customer reviews, making it an accurate and reliable representation of the restaurant's performance. Overall, this project has the potential to improve customer satisfaction, boost business success, and provide a valuable resource for sushi lovers.

3. Motivation, Scope, and Resources

The restaurant industry is highly competitive, and customers often rely on online reviews to decide where to dine. However, with the abundance of information available on review sites, it can be overwhelming for customers to make a decision. Moreover, restaurants are also always looking for ways to improve their business and gain an edge over their competitors.

This project aims to address these challenges by analyzing customer reviews of sushi restaurants on Yelp and using topic modeling and rating aggregation to identify the strengths and weaknesses of each restaurant. By doing so, we can help customers make more informed decisions and help restaurants improve their business by focusing on their strengths and addressing their weaknesses. Additionally, by scoring the best sushi in restaurant based on their reviews, we can provide a valuable resource for sushi lovers and help restaurants attract new customers. **The project aims to help consumers decide which business to choose based on the specific product or service they need, and provides business owners and managers valuable insights to help boost the success of their business.** This scope of this project, however, is limited only to sushi restaurants, and provides sample analysis of where to get the best sushi among various restaurants that offer sushi.

Resources Used

The entire project was built in **Python** and hosted on **Google Cloud Platform**. Given the huge size of the dataset, it was pushed to the Google Cloud's **DataProc cluster** using google cloud command line sdk, and **distributed in HDFS** for faster computation.
The project utilizes **Pyspark** and its available machine learning libraries and sql functions for big data processing. Optimizations were made in the code to **aid parallel processing and memory utilization** at every step possible.

4. Dataset Description

The Yelp dataset is a comprehensive collection of customer reviews, ratings, and business information for various categories of businesses, including restaurants, bars, and cafes, among others. The dataset contains over 8 million reviews from more than 200,000 businesses across

many metropolitan areas in four countries. In this project, we utilized two specific datasets from Yelp - the **Yelp Business** dataset and the **Yelp Review** dataset.

The Yelp Business dataset includes information about businesses, such as their name, location, category, ratings, and review counts. We used this dataset to filter out the restaurants that offer sushi from the rest of the businesses. This dataset contains over 200,000 businesses across multiple categories and locations. A total of 303 businesses were found in the reviews dataset, which had ‘sushi’ in their description. This list of business id’s is stored in the file ‘intersection.csv’.

The Yelp Review dataset includes customer reviews for various businesses, including the filtered restaurants that offer sushi. This dataset contains over 8 million reviews and includes information about the user, the business, and the review itself. We used this dataset to analyze customer reviews for the selected restaurants and to extract information required to fulfill the objective. A total of **80000+ reviews** were available in the final dataset of the selected **303 business ids**.

5. Dataset Preprocessing

The data was cleaned with the help of PySpark libraries and NLTK libraries using the provided Yelp reviews dataset. The Word2Vec, Tokenizer, and StopWordsRemover classes were imported from the PySpark ML library, as well as the SparkContext and SparkSession classes from PySpark, and the NLTK library for natural language processing.

To invoke a spark session, a SparkContext and SparkSession were created, and a dataset of Yelp reviews was read from a CSV file. The "stars" column's data type was changed from a string to a double for more precise calculations, such as calculating the average star rating. Additionally, any rows with null or missing values were removed to avoid errors in further steps. Rows with blank values in the "label" and "text" columns were removed. The dataset was further filtered to include only rows with labels of 1.0, 2.0, 3.0, 4.0, or 5.0.

To clean and pre-process the data, a Tokenizer object was constructed and used to separate the text into individual words in the dataset's "text" column. Tokenization is a technique used in natural language processing to break down phrases and paragraphs into simpler language-assignable elements. The first stage in the NLP process involves gathering the data (a sentence) and breaking it down into intelligible components (words).

```
|This place is always so dirty and grimy, been there twice and will not be back.  Customer service is horrible!!!
|[this, place, is, always, so, dirty, and, grimy,, been, there, twice, and, will, not, be, back., , customer, service, is, ho
rrible!!!]
```

A StopWordsRemover object was built to remove stopwords from the text. Stopwords are a group of frequently used terms in any language. Examples of stop words in English include "the," "is," and "and." In NLP and text mining applications, stop words are used to filter out unnecessary words so that the key words can be the focus. The remover is applied to the tokenized text, and a user-defined function (UDF) is used to rejoin the list of words into a single string.

```
|service really slow here. waited 25 mins takeout. definitely worth wait. however, seem live music friday nites, looking ente
rtainment food sit-in probably lot faster
```

After processing the text data, the "clean_text" column along with the "text", "label", "words", and "clean_words" columns are shown below for comparison.

```
|Cycle Pub Las Vegas was a blast! Got a groupon and rented the bike for 11 of us for an afternoon tour. Each bar was more fun
than the last. Downtown Las Vegas has changed so much and for the better. We had a wide age range in this group from early 2
0's to mid 50's and everyone had so much fun! Our driver Tony was knowledgable , friendly and just plain fun! Would recommend
this to anyone looking to do something different away from the strip. You won't be disappointed!
|5.0  |[cycle, pub, las, vegas, was, a, blast!, got, a, groupon, and, rented, the, bike, for, 11, of, us, for, an, afternoon,
tour., each, bar, was, more, fun, than, the, last., downtown, las, vegas, has, changed, so, much, and, for, the, better., we,
had, a, wide, age, range, in, this, group, from, early, 20's, to, mid, 50's, and, everyone, had, so, much, fun!, our, driver,
tony, was, knowledgable, ,, friendly, and, just, plain, fun!, would, recommend, this, to, anyone, looking, to, do, something,
different, away, from, the, strip., you, won't, be, disappointed!]
|[cycle, pub, las, vegas, blast!, got, groupon, rented, bike, 11, us, afternoon, tour., bar, fun, last., downtown, las, vega
s, changed, much, better., wide, age, range, group, early, 20's, mid, 50's, everyone, much, fun!, driver, tony, knowledgable,
,, friendly, plain, fun!, recommend, anyone, looking, something, different, away, strip., disappointed!]
|cycle pub las vegas blast! got groupon rented bike 11 us afternoon tour. bar fun last. downtown las vegas changed much bette
r. wide age range group early 20's mid 50's everyone much fun! driver tony knowledgable , friendly plain fun! recommend anyon
e looking something different away strip. disappointed!
```

Average Review Length Calculation

The number of topics to be extracted for each business id was based on the average length of reviews. To calculate the average review length for each rating in the cleaned_df, the pyspark.sql.functions module's length and average functions were imported into the code. The length of each review was then determined by selecting the 'label' and 'clean_text' columns from cleaned_df and using the length() function to construct a new DataFrame named length_df. The resulting column was renamed to 'length' using the alias() method. The DataFrame was then grouped by ratings using groupBy('label'), and the average length of reviews for each rating was determined using the agg() function with the avg() function acting as the aggregation function. The outcome was a DataFrame with the name avg_length_df, which had average number of characters for each rating label.

```
+-----+------------------+
|label|       avg(length)|
+-----+------------------+
|  1.0|301.53846153846155|
|  4.0|251.02762430939225|
|  3.0| 256.5207100591716|
|  2.0| 280.7647058823529|
|  5.0|  259.749226006192|
+-----+------------------+
```

To determine the typical word count for each rating's reviews, a brand-new DataFrame called size_df was created that contained the rating, the clean text, and the word count for each review. The clean text was divided into an array of words using the split function, and the size function was used to determine how big the array should be.

After that, the size_df DataFrame was grouped by rating and the agg function was used with the input avg to determine the average review length (or average number of words) for each rating. With the rating and the typical size of reviews for each rating, a new DataFrame called avg_size_df was created.

Finally, a for loop iterated over the rows of avg_size_df and printed the outcomes in the desired manner. It took the rating and the average review length from each row, converted them into a string, and outputted the results.

```
Rating: 1; average number of words in review: 46.06
Rating: 4; average number of words in review: 39.16
Rating: 3; average number of words in review: 39.67
Rating: 2; average number of words in review: 41.84
Rating: 5; average number of words in review: 39.18
```

6. Topic Modeling- Extracting Highlights

For calculating ‘highlights’ of a business, only the 4 or 5 star rated reviews were considered, as it was assumed that only the best features about a business are mentioned in highly rated reviews. The ‘highlights’ or ‘topics’ for each sushi restaurant are then defined as the words most talked about in the top rated reviews for the restaurant. Therefore, for the task of topic modeling, the complete preporcessed dataframe, with stopwords removed, was subsetted to contain the reviews of only one business at a time. The reviews in this dataframe is now considered as the text corpus, and each review as a document. Based on this, two approaches were used to extract topics for each restaurant-

a. Term Frequency (TF)

A document term frequency matrix was calculated for each business using Pyspark ML’s CountVectorizer class, which creates a matrix of numbers, where each column represents a feature or term, each row corresponds to a document, and the value represents the frequency of the term in that document. After fitting the CountVectorizer object, a CountVectorizer ‘vocabulary’ is generated, within which the terms are ordered in descending order of term frequency, or how many times a term in used in the entire corpus.
Based on term frequency, the top ten words for each corpus were extracted from the vocabulary as the top ten topics. The number of topics for each business was decided based on the average number of stopwords-removed tokens in each review in the dataset, which was about 40, and an assumption was made that at least 5 topics are talked about in a review of 40 words.
However, the resulting list often contained insignificant words like adjectives - good, bad, amazing, etc., and verbs - never, always, love, etc., which did not contribute to meaningful highlights of the business, and were removed from the list.
This filtered list of words based on term frequency, ultimately had about **5 to 8 topics** for each business.

b. Term Frequency-Inverse Document Frequency (TF-IDF)

Topic modeling based on Term Frequency-Inverse Document Frequency is one of the most commonly used methods of topic modeling. This method assigns higher weightage to terms which are frequent within individual documents, but penalizes weightage from terms which are present in many documents. This has advantage of being able to discard common adjectives and actions from being considered as important topics.
However, in our case of each individual review being a document, and the entire set of the business’ highly rated reviews being the text corpus, a topic being talked about in most reviews would be considered less important by TF-IDF, but in reality would be the most important topic. For example, if most highly-rated reviews mention ‘sushi’, then it implies that ‘sushi’ is an important highlight for the business, which would otherwise be lower in the importance calculated by the TF-IDF method.

Extracting Topics Using Term Frequency Alone

Term Frequency alone was finalized as the metric for topic modeling, because a business’s highlight is supposed to be talked about in most of its highly rated reviews. The TF-IDF method in contrast, penalizes the document frequency, which is not appropriate in our use case. The resulting dataframe with the calculated topics had schema as shown below, with the list of topics for the business mapped across all reviews for ease of topic popularity score calculation.

business_id	review_id	label	clean_words	clean_text	topics_list
huZ1fY8x0-9l5Mo-l...	URoxdWVKMWbGcBBPd...	5.0	[went, today, lun...	went today lunch ...	[roll, sushi, bow...
huZ1fY8x0-9l5Mo-l...	uUSnz0uN5RO_TydXr...	4.0	[good, quick, goi...	good quick going ...	[roll, sushi, bow...

7. Comparable Normalized Popularity Scoring

7.1 Concept

The objective of this project is to provide a comparison metric to consumers, based on which they can decide which restaurant should they go to have the best sushi in. The restaurant's overall rating must have no bearing on this comparison, and the entire analysis must be done solely on the text reviews received by the restaurant, and the rating associated with those reviews. The lack of consideration of overall restaurant rating will provide a fair comparison for all restaurants with regards to sushi alone, despite the range of dishes it offers. For example, if a restaurant serves three dishes - burrito, ramen, and sushi - it's overall rating might not be representative of the quality of sushi it offers, but rather of the average quality of all the dishes it offers. Whereas if the same restaurant offers only good sushi, but average burrito and ramen, its highly rated reviews are more likely to mention sushi. Similarly, if a highly rated restaurant offers good burritos and ramen, but only average sushi, considering its overall rating in assigning a comparable score for sushi would be a misrepresentation of quality of sushi available at that restaurant.

Therefore, the following challenges were presented in assigning a Comparable Normalized Popularity Score for the topics calculated for each restaurant-

1. The score must be independent of the restaurant's overall rating
2. Each factor that makes up the score must be individually quantifiable and comparable across different businesses
3. The score must be representative of how popular a topic *(sushi)* is for that restaurant
4. The score must be representative of an absolute comparable metric directly derived from the business reviews

The Comparable Normalized Popularity Score can now be defined as-

$$Topic\ Popularity\ score\ =\ F\ *\ R$$

Where
F = Fraction of 4 or 5 Star reviews on the business that mention the topic
R = Average rating of the 4 or 5 star reviews on the business that mention the topic

The score factors in **how many 4 or 5 star reviews** on the business actually **mention the topic** and the **actual rating of the reviews which mention the topic.** Its effectiveness can be understood by a simple example-

- If a business has 'sushi' mentioned in 70% of its highly rated reviews, it has better 'sushi' than the business which has sushi mentioned in 20% of its highly rated reviews
- If a business has average rating of reviews that mention sushi as 4.7, it has better sushi than the business that has average rating of reviews that mention sushi as 4.2.

7.2 Implementation and Results

The topic popularity score was calculated for each of the topics/highlights calculated using term frequency of review texts for each business. For each review, each topic was first mapped to a key

value pair where the topic was the key and value was either 1 or 0, indicating whether the topic was mentioned in the review or not-

```
+-----+--------------------+--------------------+--------------------+
|label|         clean_words|         topics_list|        topic_counts|
+-----+--------------------+--------------------+--------------------+
|  5.0|[went, today, lun...|[roll, sushi, bow...|[{roll, 1}, {sush...|
|  4.0|[good, quick, goi...|[roll, sushi, bow...|[{roll, 0}, {sush...|
+-----+--------------------+--------------------+--------------------+
```

Another similar operation was performed to convert the value of each key-value pair into a tuple of values , containing the review rating as well-

```
+-----+--------------------+--------------------+--------------------+--------------------+
|label|         clean_words|         topics_list|        topic_counts| topic_counts_rating|
+-----+--------------------+--------------------+--------------------+--------------------+
|  5.0|[went, today, lun...|[roll, sushi, bow...|[{roll, 1}, {sush...|[{roll, {1, 5.0}}...|
|  4.0|[good, quick, goi...|[roll, sushi, bow...|[{roll, 0}, {sush...|[{roll, {0, 4.0}}...|
+-----+--------------------+--------------------+--------------------+--------------------+
```

For further calculation, only the ‘topic_counts_rating’ column was relevant. In order to perform RDD operations to aggregate average presence of topic in reviews and average rating of reviews in which topic was present, the lists within the ‘topic_counts_rating’ column were exploded such that each cell contained only one key-value pair-

```
+------------------+
|topic_count_rating|
+------------------+
|  {roll, {1, 5.0}}|
| {sushi, {0, 5.0}}|
+------------------+
```

This column was then stored in an RDD, on which a series of paired RDD operations were performed to calculate the required topic popularity score for each topic within the business’ reviews. The resulting RDD was collected and stored in a csv file with a column for business, and another for a list of topics with their comparable, normalized, popularity scores.

bupMXFUaZfranBLda\	[('sushi', 0.7393015873015873), ('fresh', 0.2794920634920635), ('cleveland', 0.2614603174603175), ('ginko', 0.252444
0xfVubbU3z8O2NcuE\	[('sushi', 0.6284023668639053), ('hour', 0.37529585798816567), ('happy', 0.37529585798816567), ('service', 0.314201
nrahyQyopCtajDqUtV\	[('sushi', 0.6498269896193771), ('service', 0.32491349480968856), ('rolls', 0.2978373702422145), ('like', 0.2436851211
nrOxXGd3Vx6iz5xuIJT\	[('sushi', 0.5746556473829202), ('fresh', 0.49256198347107444), ('sashimi', 0.32837465564738294), ('lunch', 0.301010
pyqnGlIfP9Zw8LLxBjyA	[('food', 0.4078890734171245), ('nobu', 0.3509045705132615), ('service', 0.3449062017865391), ('sushi', 0.323911911

Model Results for Business Highlights

8. Insights

Insights can be drawn from the results in two perspectives, from the perspective of a consumer, and that of the business owner. Consider this sample of results from the model below for both perspectives-

	business	highlights
1	9UVkAmy(	[('food', 0.3207100591715976), ('sushi', 0.30414201183431955), ('bowl', 0.3017751479289941), ('poke', 0.2899408284023669), ('burrito', 0.25325
2	11fdZ98T5	[('sushi', 0.6), ('try', 0.6), ('food', 0.4666666666666667), ('nice', 0.4333333333333333), ('way', 0.3), ('fast', 0.3), ('burrito', 0.26666666666666666)]
3	Mbat8t9X(	[('food', 0.5471698113207547), ('sushi', 0.46037735849056605), ('hibachi', 0.29811320754716986), ('service', 0.2528301886792453), ('staff', 0.169
4	X88w0XrU	[('sushi', 0.4803738317757009), ('fish', 0.44859813084112143), ('fresh', 0.42803738317757006), ('pittsburgh', 0.2672897196261682), ('food', 0.24
5	ghVhlFpNh	[('sushi', 0.7097435897435898), ('rice', 0.3476923076923077), ('fish', 0.3456410256410256), ('service', 0.3415384615384615), ('food', 0.31487179
6	eWer42jE(	[('food', 0.568), ('hibachi', 0.488), ('sushi', 0.4159999999999999), ('saga', 0.368), ('service', 0.36), ('ive', 0.168)]
7	LWgaRojE	[('poke', 0.5690476190476191), ('fresh', 0.33809523809523806), ('bowl', 0.3047619047619048), ('food', 0.3), ('friendly', 0.2928571428571428), ('g
8	DkZuVzYu	[('sushi', 0.6483582089552239), ('service', 0.30388059701492537), ('fresh', 0.28776119402985073), ('fish', 0.24238805970149255), ('food', 0.2417

Business Highlights

8.1 Consumer Perspective

Building on the example of sushi that has already been introduced, let us assume that a consumer wants to have sushi at a restaurant, and has a list of 8 available sushi restaurants in their area to chose from. From the results of our popularity scoring model, the consumer can look at the highlights for each restaurant and decide to go to the restaurant which has sushi in its highlights. This excludes restaurant number 7 from their choices. Now, based on the remaining restaurants and their highlights, the model suggests that according to the reviews, restaurant number 5, 'ghVhlFpNhfBwWDFGSlt2JA', has the **highest popularity score for sushi (0.709), and therefore is the best choice for sushi among the restaurants in consideration.**
However, if the consumer also plans to have some other dish, like hibachi, it can be seen that restaurant number 6, 'eWer42jEQGG1o8mNQUk4Kg' would be the ideal choice.

Therefore, the model effectively suggests which restaurant to chose for a particular dish, while also providing other highlights of the restaurant to consumers.

8.2 Business Perspective

The same results which allow a consumer to choose which restaurant to go to for a certain dish, can also be used by the business owners to assess which of their business' aspects are actually highly rated by consumers. For example, for business number 7, 'poke' and 'bowl' are the reviewers' favorite, and must be focussed more within the restaurant's offerings to increase the restaurant's popularity. Also, for the same business, operational aspects like 'friendly' staff are well appreciated by the reviewers, and the business must solidify amicability of the staff within its set of best practices.

While the model discussed yet provides highlights for a business based on the topics extracted from its highly rated (4 and 5 star) reviews, the same model can be slightly modified to extract the 'lowlights' of the business, or topics most talked about in its 1 star or 2 star rated reviews, and assign a similar score for unpopularity. A sample result for business' lowlights is shown below-

	business	lowlights
1	9UVkAmy(	[('like', 0.3), ('burrito', 0.28888888888888886), ('poke', 0.26666666666666666), ('food', 0.2333333333333333)]
2	11fdZ98T5	[('paid', 0.6), ('turned', 0.6), ('sign', 0.6), ('look', 0.6), ('price', 0.6), ('12', 0.6), ('went', 0.6), ('sushi', 0.6), ('sour', 0.6), ('taking', 0.6)]
3	Mbat8t9X(	[('food', 0.5058823529411766), ('service', 0.48235294117647065), ('cook', 0.24705882352941178), ('sushi', 0.2235294117647059), ('ev
4	X88w0XrU	[('tasted', 0.4), ('fish', 0.375), ('food', 0.35), ('time', 0.35), ('sushi', 0.30000000000000004), ('eat', 0.275), ('tacos', 0.2), ('want', 0.075)]
5	ghVhlFpNh	[('order', 0.4375), ('service', 0.3625), ('food', 0.3375), ('sushi', 0.30000000000000004), ('never', 0.2375)]
6	eWer42jE(	[('sushi', 0.475), ('food', 0.325), ('sauce', 0.3), ('hibachi', 0.22499999999999998), ('service', 0.15), ('spicy', 0.15), ('never', 0.075)]
7	LWgaRojE	[('like', 0.4571428571428572), ('poke', 0.2857142857142857), ('smell', 0.2857142857142857), ('bowl', 0.2857142857142857), ('sushi', (
8	DkZuVzYu	[('sushi', 0.4130434782608696), ('service', 0.2956521739130435), ('order', 0.291304347826087), ('food', 0.2782608695652174), ('get',

Business Lowlights

Taking the example of restaurant number 5, 'ghVhlFpNhfBwWDFGSlt2JA', which was deemed as the restaurant for best sushi, the lowlights for that restaurant suggest that 'service' is a topic which was highly talked about in its negative reviews, and that the management must put into place an

action plan that prohibits bad service experience for its customers. Similar inferences can be drawn for other restaurants.

However, if for a restaurant, the **same topic occurs with a high score in both its highlights and lowlights**, it could be suggestive of either of the two scenarios-

- **Inconsistent quality**- Some consumers are offered high quality food, while some are not, with the latter giving low ratings about the food. In this case, the restaurant must stick to a standard operating procedure for certain dishes and services so that all customers are offered the same quality of food and services.

- **Mismatch in consumer expectation and actual product**- The restaurant offers the same quality of food and services to all customers, however some customers expect a particular kind of taste which might differ from the restaurant's standard offering. The restaurant must take into consideration the preferred taste and other details by the customers and convey the details of the offered standard in order to better match consumer expectation.

9. Conclusion

In a highly competitive market like that of restaurants, it is imperative to stay in touch with consumer feedback. Oftentimes, restaurants indulge in seeking on-the-spot feedback from their customers, however, some customers provide just a customary feedback in this setting, and would rather provide a verbose, detailed feedback in online reviews. Therefore, online reviews can be regarded as a trustworthy source of feedback and information about businesses, which must be capitulated on for insights to-

1. **Consumers,** for selecting a business for the product or service that they seek, and
2. **Businesses**, for keeping up with their customers' needs and tending to their feedback.

Our model effectively transforms an expansive dataset of business reviews into a crowdsourced survey to inform consumers which business to choose for a product or service, or which product/service to choose from a business.
Further, by calculating highlights and lowlights for the business, the business owners and managers can draw action plans to solidify and promote their areas of popularity and to capture the pain points of customers and improve their overall experience.

Therefore, the model adequately serves to match customers with the business or product they are most likely to appreciate, and help the businesses improve their profits and the customer experience they have to offer.

10. Discussion and Future Scope

By combining these scores with other features such as price range and location, search engines like Google can suggest the best restaurants in order of quality of desired product or service based on quality perceived from review texts, which would be most relevant to the consumer's geographical, or other preferences.

Further, the model can be further callibrated to provide improvement suggestions to business owners based on how their highlight scores compare to other businesse in the same geographical or price range, in a generic way, abiding with the data privacy and security laws of the region.

Part 2 (Bonus): Live Streaming Classification of Positive and Negative Reviews

1. **Introduction**

This code implements real-time processing of data received via a network connection using **Apache Spark Streaming**. It examines each line of text received across the connection while listening to a socket on port 9999 and categorizes the text into "good" and "bad" based on the existence of specific keywords. The fundamental abstraction offered by Spark Streaming is called a **Discretized Stream**, or **DStream**, which represents a continuous stream of data that can be either the input stream obtained from the source or the processed stream produced by processing the input stream.

2. **Motivation:**

The goal of this project is to provide a data-driven solution for real-time processing of business reviews received via a network connection using Spark Streaming. By categorizing the text into "good" and "bad" based on specific keywords, businesses can identify areas that need improvement and promote their strengths, ultimately improving customer satisfaction.

3. **Implementation:**

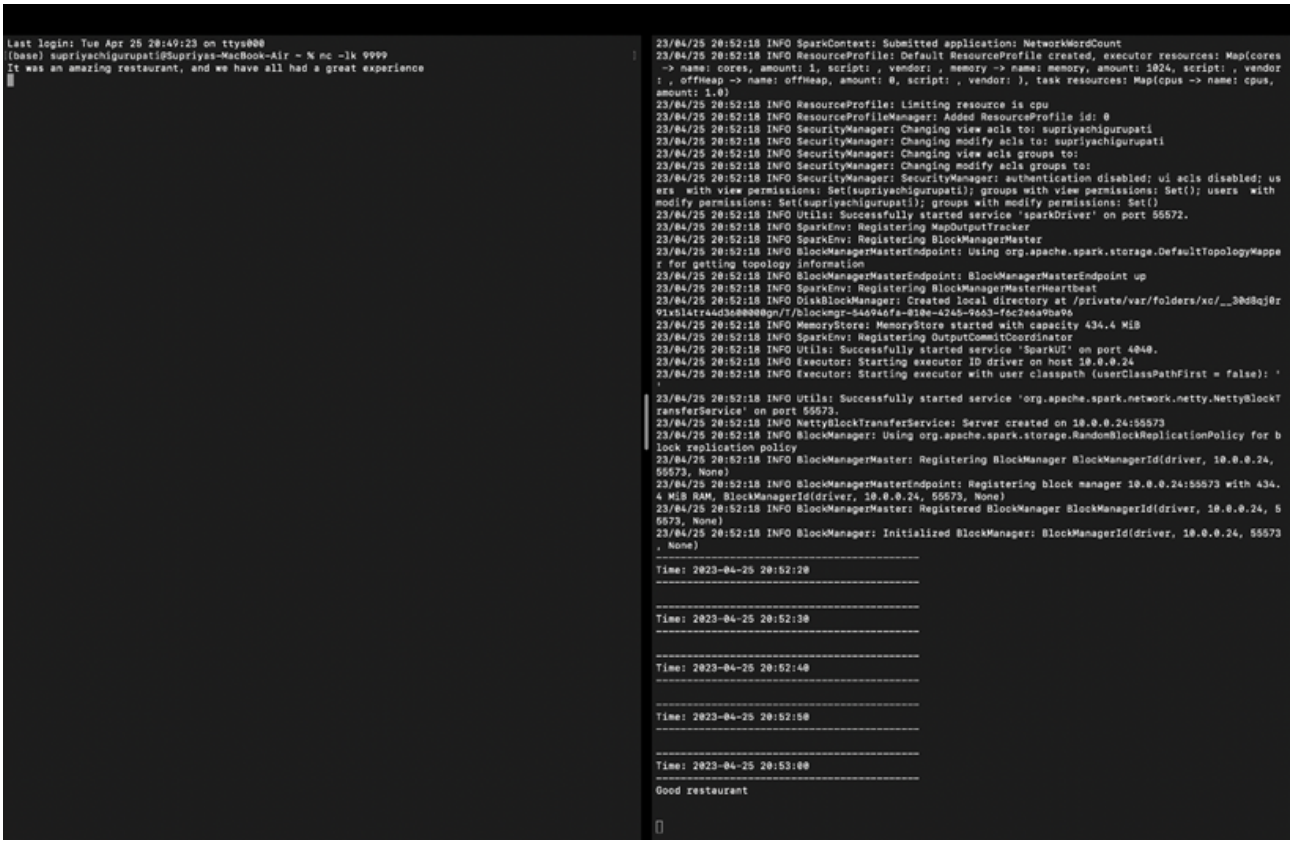
The implementation begins by creating the required libraries, including re, SparkContext, and StreamingContext, and creating the SparkContext object sc to leverage Spark's features. The StreamingContext object ssc is created as the key entry point for Spark Streaming capability, with a batch interval of 10 seconds, specified along with the SparkContext object that the StreamingContext should use. Using the socketTextStream() function on ssc, it produces a DStream object called lines that listens to incoming data on port 9999. The flatMap() transformation is used to break up each line of text into words, and a new DStream of distinct words is created as a result of this modification. Each word is categorized as "good" or "bad" using the map() transformation based on whether it appears in a list of positive or negative keywords. Finally, the program's output is printed to the console using the pprint() method to the map() transformation's output.

Netcat, a simple program that is present in most Unix-like operating systems, was launched as a first step by the data server. Since a MAC environment was used, the command `nc -lk 9999` was directly typed into a new terminal to connect to Netcat.

Afterwards, the below examples were started in a separate terminal by inputting the sample texts-

Example 1: -
Review: - It was an amazing restaurant, and we have all had a great experience!!

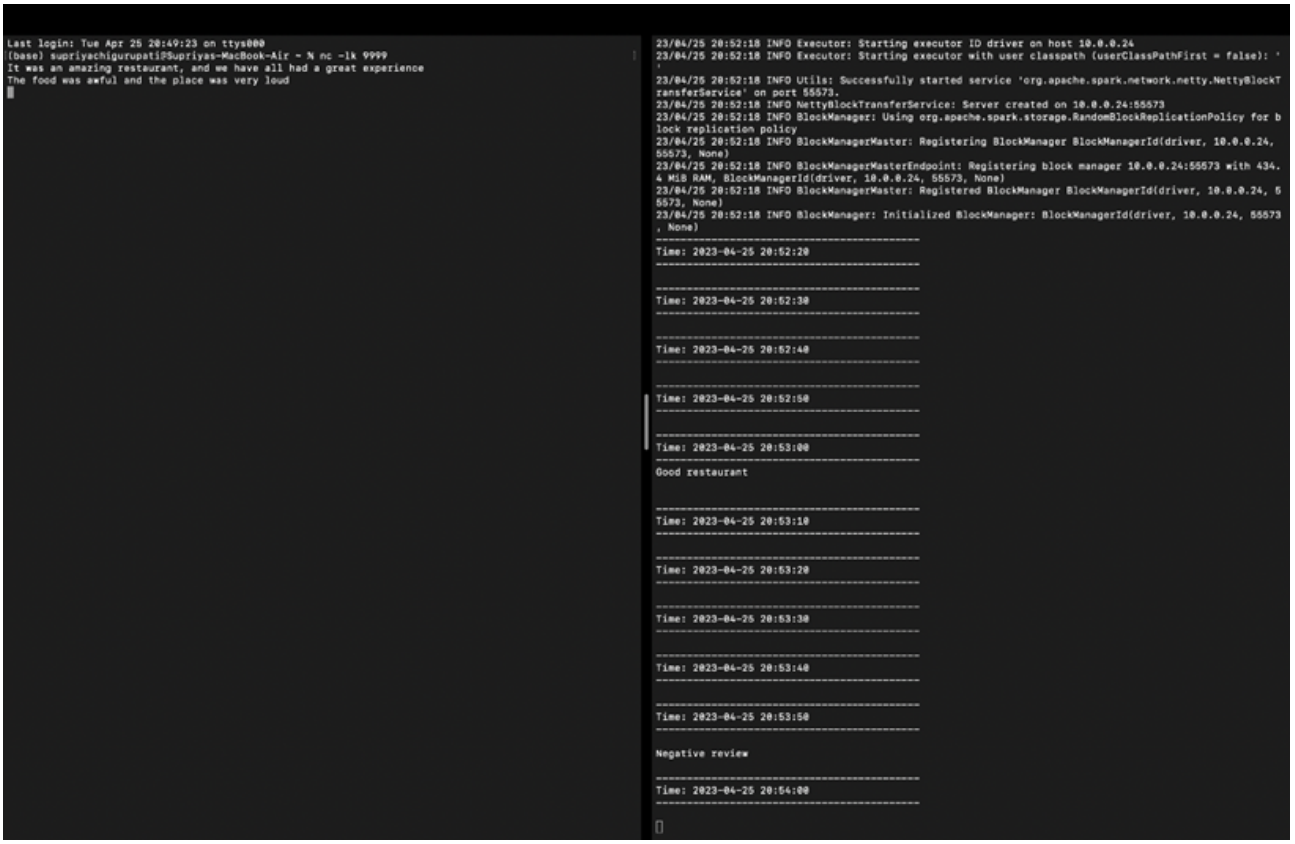
Result- Good restaurant



Example 2: -

Review: - The food was awful and the place was very loud.

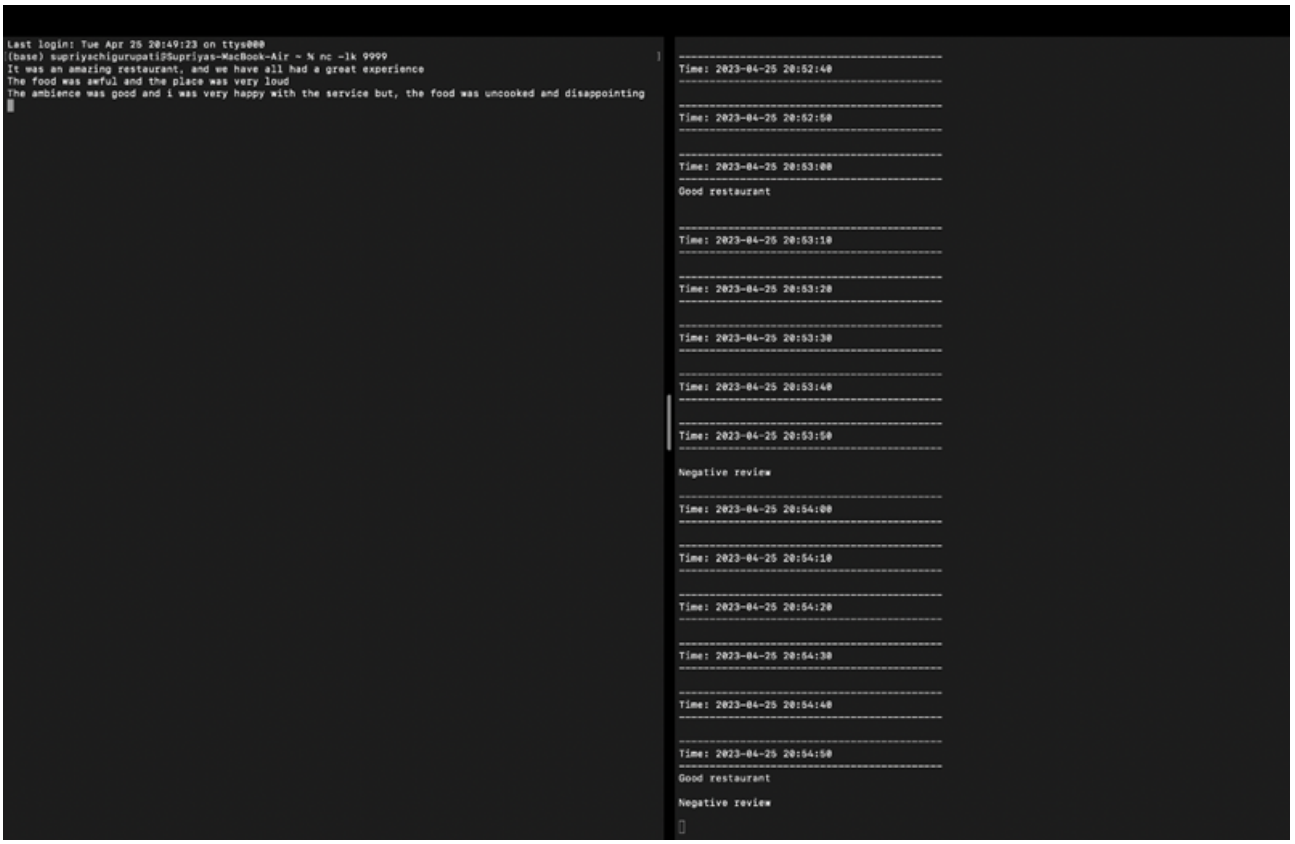
Result- Negative review



Example 3: -

Review: - The ambience was good and I was very happy with the service but, the food was uncooked and disappointing.

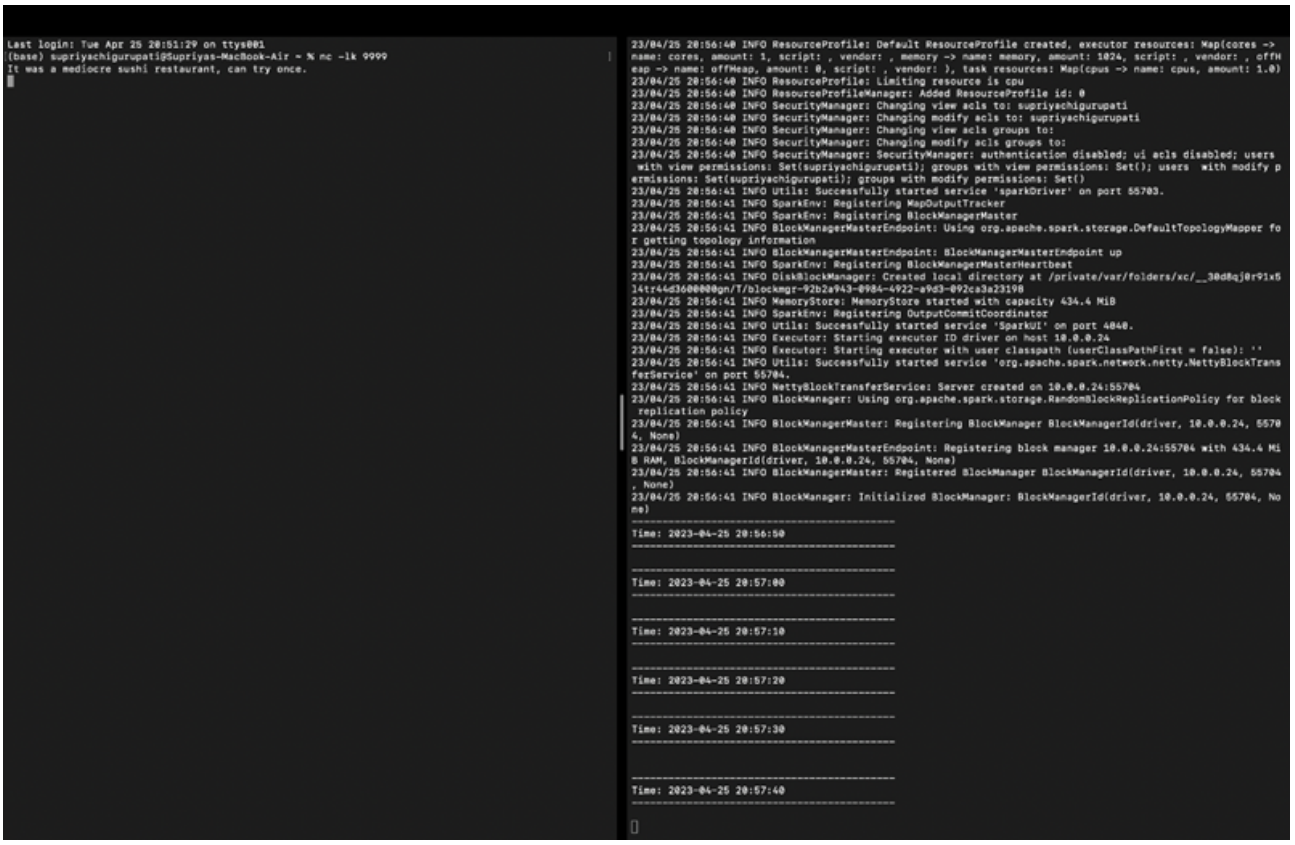
Result- Good restaurant. Negative review.



Example 4: -

Review: - It is a mediocre sushi restaurant. Can try once.

Result- No output, neutral review.



4. Conclusion and Future Scope

This live steaming review classification model performs useful labeling of incoming reviews, and can be a powerful tool in shortlisting negative or positive **reviews about businesses**, or **live social media comments about a business** for further analysis.

However, this review classification model currently works on a limited vocabulary, and matches single words with its vocabulary of good and bad words in order to assign the ‘Negative review’ or ‘Good restaurant’ labels. As a result, the model does not recognize phrases like ‘not good’ or ‘not tasty’ as negative reviews, and same for double negative phrases. This can be worked upon by incorporating pre-built sentiment analysis models like Vader, which use N-grams to discern multiple word phrases as positive or negative.

Further, incorporating topic modeling in the reviews, based on text available in rest of the reviews for the business like in part 1 of this project, will definitely add value to the live streaming insights from this review analysis.