NLP FINAL REPORT

Cuisine Compass: A Restaurant Recommendation System Utilizing Sentiment Analysis

Submitted By:
Lohith Srikanth Pentapalli (lpentapa@syr.edu)
Ritika Radhakrishnan (riradhak@syr.edu)
Lishel Aquinas (laquinas@syr.edu)
Dayasaagar Ramakrishnan (dkrishna@syr.edu)
Relina Vas (rvas@syr.edu)

Introduction

In today's digital era, the influence of customer reviews extends far beyond mere feedback. These reviews have transformed into a powerful tool that shapes public perception, influences consumer decisions, and plays a crucial role in determining the success or failure of businesses. This is especially pronounced in the highly competitive hospitality industry, where a single review can sway potential customers' choices. Reviews not only reflect customer satisfaction but also serve as a gauge for service quality and product excellence. They provide valuable insights that businesses can use to refine their offerings and enhance customer experiences.

Consequently, the strategic management of customer reviews has become an essential aspect of business operations, necessitating sophisticated analytics to monitor and respond to consumer sentiment effectively. This underscores the critical importance of harnessing and understanding customer feedback in shaping business strategies and maintaining a competitive edge in the marketplace.

Recognizing the transformative potential of user-generated content, our project, "Cuisine Compass," capitalizes on the vast reservoir of restaurant reviews available online to craft a sophisticated and dynamic restaurant recommendation system. This innovative system is meticulously designed not only to suggest dining establishments but also to significantly enhance the user experience by personalizing these suggestions based on an in-depth sentiment analysis of customer reviews.

The impetus for developing "Cuisine Compass" emerged from a clear discrepancy observed between customer expectations and their actual dining experiences. Traditional restaurant recommendation systems have typically relied heavily on numerical ratings alone, which, although useful, can often be misleading and fail to capture the full spectrum of diner experiences. These ratings do not provide insight into the specific attributes of dining that customers value, such as culinary creativity, service excellence, or the ambiance of an establishment.

To bridge this gap and offer a richer, more reliable guide to dining options, "Cuisine Compass" integrates advanced sentiment analysis into our recommendation process. This integration allows for a more comprehensive view of diner sentiments and preferences, enhancing the granularity with which we understand consumer behavior. By meticulously analyzing the textual data from reviews, we extract crucial insights into what truly resonates with diners in their culinary journeys—from the quality of the food and attentiveness of service to the overall atmosphere of the restaurant.

Sentiment analysis, a sophisticated branch of natural language processing (NLP), involves the computational study and interpretation of opinions, sentiments, subjectivity, and emotions expressed within text. In the realm of "Cuisine Compass," this technological tool is invaluable. It enables us to decode the latent semantic meanings of customer reviews, meticulously categorizing them into positive, negative, and neutral sentiments. This detailed approach allows us not only to identify the best-rated restaurants but also to deeply understand the reasons behind their ratings, thus providing a more transparent, nuanced, and reliable recommendation system.

Moreover, the utility of sentiment analysis in "Cuisine Compass" extends far beyond simple categorization. It enables us to perform a nuanced analysis that includes detecting sentiment intensity, understanding contextual nuances, and recognizing subtle differences in customer preferences that traditional methods might overlook. For instance, through techniques such as Named Entity Recognition (NER), we can extract specific dishes or services praised by customers. This capability allows us to recommend restaurants not just for their overall service but also for special menu items or unique features that may appeal to individual tastes, enhancing the personalization of our recommendations.

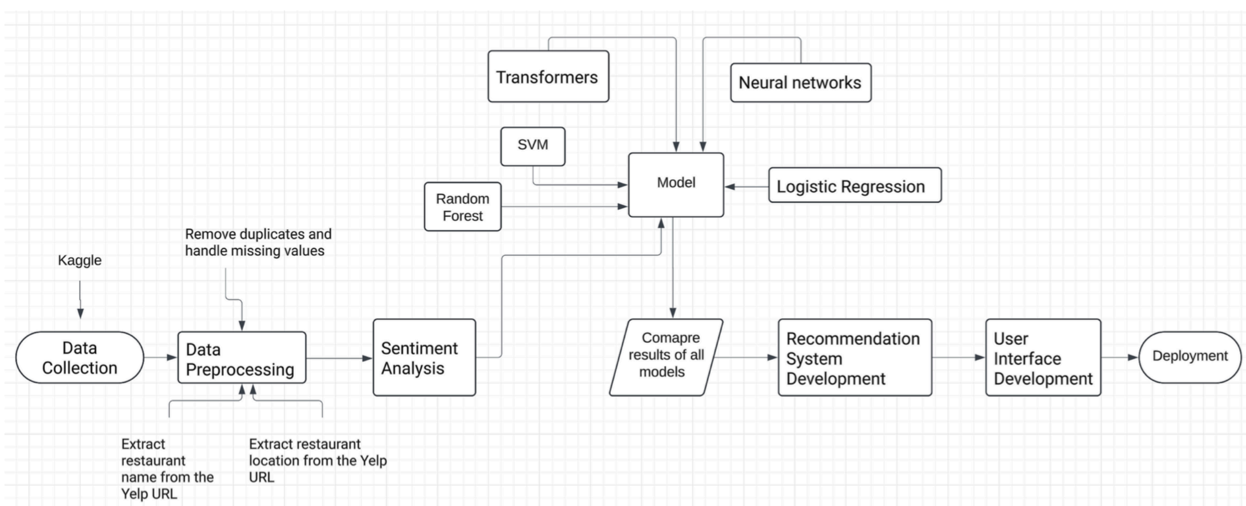
Our ultimate goal with "Cuisine Compass" is to revolutionize the way people choose where to eat by making the decision-making process more informed, enjoyable, and highly personalized. By leveraging the sophisticated capabilities of sentiment analysis, "Cuisine Compass" aims to offer a unique and innovative tool that significantly enhances customer satisfaction, aids business owners in better understanding their clientele, and ultimately shapes a more delightful and satisfying dining landscape.

This project addresses both a practical need for reliable, nuanced restaurant recommendations and demonstrates the remarkable potential of advanced analytics and machine learning in transforming the service industry. This report will detail the sophisticated methodologies employed, the comprehensive data analysis conducted, the insightful results of our experiments, and the profound overall impact of our recommendation system.

Through "Cuisine Compass," we not only envision but actively create a future where the process of choosing where to dine is as enjoyable and fulfilling as the dining experience itself. Our system is designed to enhance the cultural and social fabric of dining out, transforming it into a more interactive and personalized experience. By integrating cutting-edge NLP techniques and sentiment analysis, we empower users to make informed decisions that align closely with their preferences and expectations. Moreover, "Cuisine Compass" contributes to a broader dialogue about food, culture, and technology's role in enriching our social lives. Through this project, we aspire to set a new standard for how dining choices are made, elevating the act of eating out to a more informed, culturally rich, and satisfying part of everyday life.

Flow Chart

The flowchart for our project outlines the key phases of our project, from initiation to completion. It serves as a visual roadmap, detailing the critical tasks and decision points across stages. Designed to enhance understanding and coordination, this diagram is crucial for efficient project management and effective teamwork.



Step-by-Step Project Workflow Diagram

Data Acquisition and Preprocessing

1. Introduction:

In the "Cuisine Compass" project, the initial steps of data acquisition and preprocessing are crucial. They ensure the integrity and quality of the data which underpins all subsequent analyses and the effectiveness of the recommendation system. This section details the methodologies and processes implemented to prepare the Yelp restaurant reviews dataset for comprehensive analysis.

2. Data Acquisition:

The primary dataset for this project consists of 20,000 Yelp restaurant reviews collected from Kaggle (https://www.kaggle.com/datasets/farukalam/yelp-restaurant-reviews/data). Each entry includes detailed feedback from customers, which encompasses textual reviews, numerical ratings, and metadata such as user and restaurant identifiers. This dataset provides a rich source of information for extracting insights into customer preferences and restaurant performance.

3. Preprocessing Techniques:

Cleaning:
Initial preprocessing involved thorough cleaning procedures to ensure data quality. This included the removal of non-ASCII characters, standardization of text to a uniform case, and correction of misspelled words using automated scripting, which enhanced the clarity and usability of the review texts.
Handling Duplicates:
To maintain the integrity of the dataset, duplicate records were identified using a combination of review text, user ID, and date of review. Such duplicates were removed to ensure that each data entry was unique, thereby providing a more accurate analysis base.
Missing Values:
Missing values in key fields were addressed through a combination of deletion and imputation methods. For less critical fields where data was missing, entries were imputed using mode or mean, depending on the data type. Additionally, to further refine the dataset, URLs were processed to extract location information. Using Python functions, we removed any trailing numbers from the Yelp URLs and extracted the last word from the location part of the URL. This extraction process ensured uniformity and accuracy in location data. Specifically, for the 'las-vegas' case, we replaced it with 'lasvegas' to standardize the location naming convention. Entries with missing critical information were then removed to preserve the quality of the analysis

```python
# Function to extract the last word from the location part of the URL
def extract_last_word(url):
    parts = url.split('/')
    location = parts[-1]  # Get the last part of the URL
    location_parts = location.split('-')  # Split the location by hyphens
    last_word = location_parts[-1]  # Get the last part, which is the word we want
    return last_word

# Apply the function to create the 'Location' column
df['Location'] = df['Yelp URL'].apply(extract_last_word)

# Specifically handle the 'las-vegas' case
df['Location'] = df['Location'].str.replace('las-vegas', 'lasvegas', regex=False)
```

```
                                           Yelp URL Location
0      https://www.yelp.com/biz/sidney-dairy-barn-sidney   sidney
1      https://www.yelp.com/biz/sidney-dairy-barn-sidney   sidney
2      https://www.yelp.com/biz/sidney-dairy-barn-sidney   sidney
3      https://www.yelp.com/biz/sidney-dairy-barn-sidney   sidney
4      https://www.yelp.com/biz/sidney-dairy-barn-sidney   sidney
...                                                ...      ...
19891  https://www.yelp.com/biz/la-pasticceria-las-vegas    vegas
19892  https://www.yelp.com/biz/la-pasticceria-las-vegas    vegas
19893  https://www.yelp.com/biz/la-pasticceria-las-vegas    vegas
19894  https://www.yelp.com/biz/la-pasticceria-las-vegas    vegas
19895  https://www.yelp.com/biz/la-pasticceria-las-vegas    vegas

[19896 rows x 2 columns]
```

Data Preprocessing: Location Extraction and Standardization

Feature Extraction:

A significant enhancement in feature extraction was achieved by parsing Yelp URLs to retrieve restaurant names and geographical locations. A meticulously crafted script was developed, adept at deciphering the varied URL structures within the dataset. Leveraging a combination of regular expressions and string manipulation techniques, this script efficiently extracted essential data by pinpointing specific segments of the URLs. Moreover, to enrich the dataset and facilitate multifaceted analysis, additional attributes such as the review date, sentiment scores derived from review text, and categorical data from user profiles were also extracted. This comprehensive approach not only broadened the dataset's dimensions but also paved the way for more nuanced insights into restaurant reviews and user preferences.

```python
# Drop the 'Url' and 'Date' columns
df.drop(['Yelp URL', 'Date'], axis=1, inplace=True)

# Reorder the columns to Location, Rating, Review
df = df[['Company name','Location', 'Rating', 'Review Text']]

# Display the modified DataFrame to verify changes
print(df.head())

# Optionally, save the modified DataFrame
df.to_csv('Final Yelp Restaurant Reviews.csv', index=False)
```

```
       Company name Location  Rating  \
0  sidney dairy barn   sidney       5
1  sidney dairy barn   sidney       4
2  sidney dairy barn   sidney       5
3  sidney dairy barn   sidney       4
4  sidney dairy barn   sidney       5

                                         Review Text
0  All I can say is they have very good ice cream...
1  Nice little local place for ice cream.My favor...
2  A delicious treat on a hot day! Staff was very...
3  This was great service and a fun crew! I got t...
4  This is one of my favorite places to get ice c...
```

Data Refinement and Output Preview

4. Data Transformation:

Data transformation processes included the normalization of review scores to align them across a standard scale and the transformation of categorical data into numerical formats through one-hot encoding. This was critical for subsequent machine learning applications which require numerical input for generating predictive models.

5. Tooling and Technologies:

Extensive use of Python and its libraries facilitated the data acquisition and preprocessing phases. Libraries such as pandas were instrumental for data manipulation tasks, while natural language processing tasks were handled by SpaCy, which provided tools for text normalization and feature extraction from textual data.

6. Challenges and Solutions:

A particular challenge was the variability in URL formats, which complicated the extraction of restaurant names and locations. The solution was a custom-built parsing algorithm tailored to recognize and adapt to the nuances of Yelp's URL schema, ensuring accurate data extraction.
Addressing data sparsity, particularly in demographic data fields, required the use of advanced statistical imputation techniques, which allowed for the estimation of missing values based on existing patterns and distributions within the dataset.

7. Summary:

The thorough and meticulous data preprocessing described herein is pivotal for the success of the "Cuisine Compass" project. This cleaned and well-structured dataset not only supports the project's analytical tasks but is also crucial for developing a robust recommendation system. By ensuring data quality at this stage, the project lays a solid foundation for generating insightful, reliable analyses and personalized dining recommendations.

Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an essential step in understanding the structure, patterns, and potential insights that can be derived from a dataset. In this section, we delve into the exploration of Yelp Restaurant Reviews dataset obtained from Kaggle, focusing on key aspects such as location extraction from URLs and sentiment analysis based on ratings.

A pivotal aspect of our EDA was the sentiment analysis of the reviews, primarily focusing on the ratings provided by users. To simplify the sentiment classification, we mapped the numerical ratings into three categories: negative (ratings 1-2), neutral (rating 3), and positive (ratings 4-5). This categorization allowed us to gauge the overall sentiment distribution across the reviews and assess the prevalent attitudes of users towards the restaurants.

```python
#Replacing the ratings with positive and negative values (1-3 negative) (4-5 postive)
sentiment = []
for i in range(len(df['Rating'])):
    if df['Rating'].iloc[i] < 3:
        sentiment.append('negative')
    elif df['Rating'].iloc[i] == 3:
        sentiment.append('neutral')
    else:
        sentiment.append('positive')
```

Normalizing rating values for Sentiment Analysis

Visualization plays a crucial role in conveying insights effectively. Throughout our EDA process, we employed various visualization techniques, including histograms, box plots, and pie charts.. These visual representations helped in elucidating patterns, outliers, and geographic clusters within the dataset, enhancing our understanding of the underlying dynamics.

Below are the few visualizations that we performed.

Count of Unique Restaurants per City

```python
# printing the count of resturants in each city.
# Remove duplicates based on 'Company name' and 'Location'
unique_restaurants = df.drop_duplicates(subset=['Company name', 'Location'])

# Count the number of unique restaurants in each city
city_counts = unique_restaurants['Location'].value_counts()

plt.figure(figsize=(10, 8))
ax = city_counts.plot(kind='bar', color='skyblue')
plt.title('Count of Unique Restaurants per City')
plt.xlabel('City')
plt.ylabel('Number of Unique Restaurants')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)

# Add numbers to the y-axis
for i in ax.patches:
    plt.text(i.get_x() + i.get_width() / 2, i.get_height(), str(int(i.get_height())), ha='center', va='bottom')

plt.show()
```

Code snippet for Count of Unique Restaurants per City

Bar graph depicting the Count of Unique Restaurants per City

Count and Percentage of Individual Sentiments

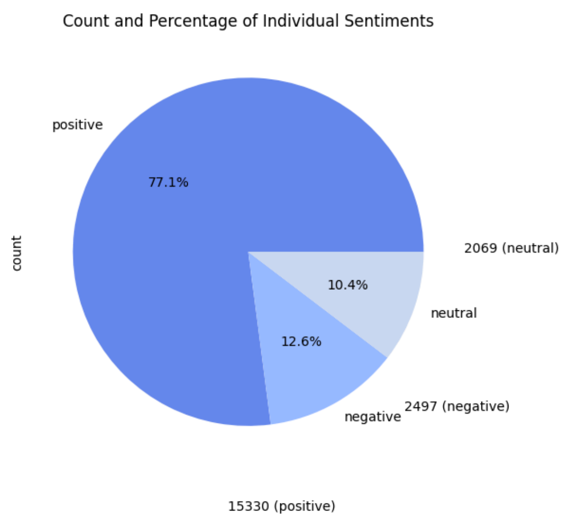
```python
import numpy as np
# Vizualizing the total number of positive, negative, and neutral reviews
reviewCount = df['Sentiment'].value_counts()

plt.figure(figsize=(10,6))
pie = reviewCount.plot(kind='pie', colors=sns.palettes.color_palette('coolwarm'), autopct='%1.1f%%')

# Add numbers and percentages to each pie slice
for i, (count, label) in enumerate(zip(reviewCount.values, reviewCount.index)):
    angle = (reviewCount.values[i] / sum(reviewCount.values)) * 360
    x = 1.5 * np.cos(np.pi * (angle + pie.patches[i].theta1) / 180)
    y = 1.5 * np.sin(np.pi * (angle + pie.patches[i].theta1) / 180)
    plt.text(x, y, f'{count} ({pie.patches[i].get_label()})', ha='center', fontsize=10)

plt.title("Count and Percentage of Individual Sentiments")
plt.show()
```

Code snippet to count and percentage of Individual Sentiments



Pie chart depicting the count and percentage of Individual Sentiments

Average rating by location

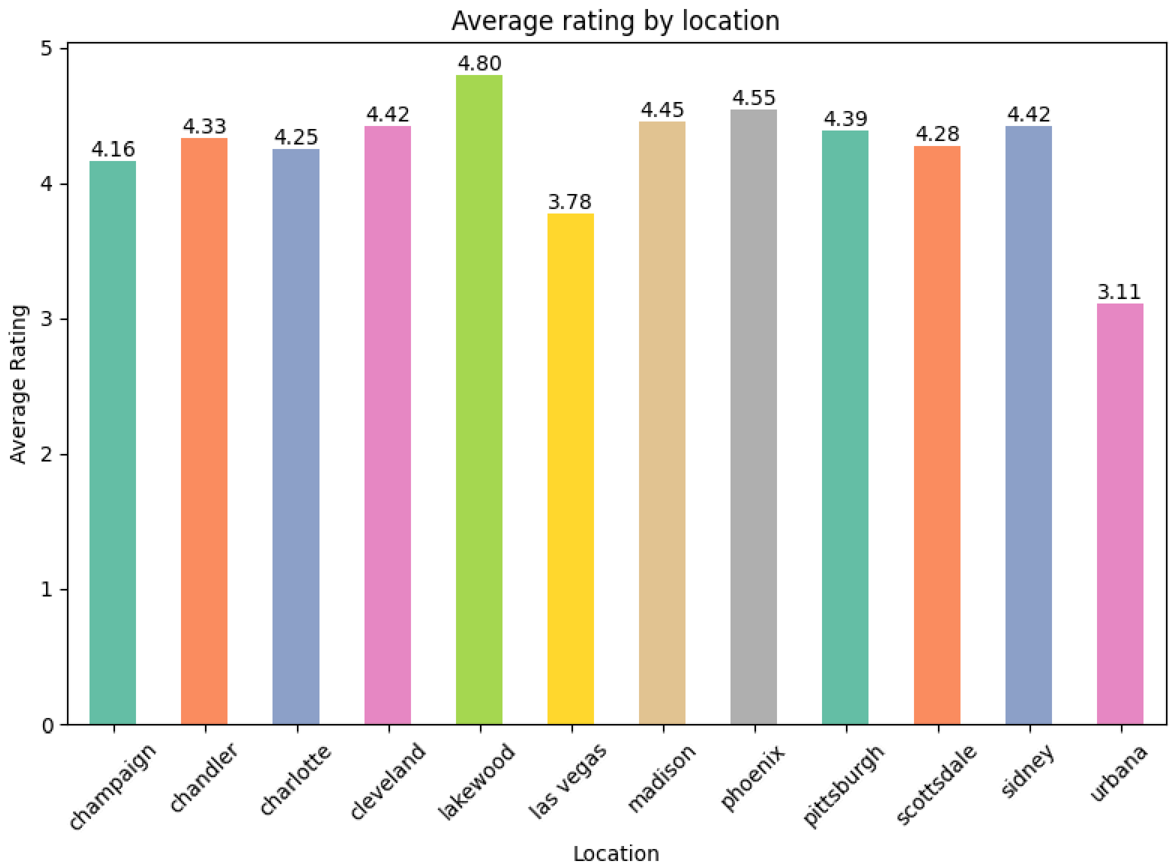
```
import numpy as np
# Vizualizing the total number of positive, negative, and neutral reviews
reviewCount = df['Sentiment'].value_counts()

plt.figure(figsize=(10,6))
pie = reviewCount.plot(kind='pie', colors=sns.palettes.color_palette('coolwarm'), autopct='%1.1f%%')

# Add numbers and percentages to each pie slice
for i, (count, label) in enumerate(zip(reviewCount.values, reviewCount.index)):
    angle = (reviewCount.values[i] / sum(reviewCount.values)) * 360
    x = 1.5 * np.cos(np.pi * (angle + pie.patches[i].theta1) / 180)
    y = 1.5 * np.sin(np.pi * (angle + pie.patches[i].theta1) / 180)
    plt.text(x, y, f'{count} ({pie.patches[i].get_label()})', ha='center', fontsize=10)

plt.title("Count and Percentage of Individual Sentiments")
plt.show()
```

Code snippet for averaging the rates across locations



Bar depicting the average rating by location

In conclusion, our exploratory data analysis of the Yelp Restaurant Reviews dataset has provided valuable insights into the structure, sentiment, and geographical distribution of user reviews. By leveraging techniques such as location extraction and sentiment analysis, we have gained a deeper understanding of the dataset, laying the foundation for further analysis and modeling in our NLP endeavors.

Sentiment Analysis

Sentiment analysis, a subfield of natural language processing (NLP), aims to computationally identify and categorize the sentiment expressed in textual data, such as reviews or social media posts. In this report, we present a comparative analysis of various sentiment analysis techniques employed in our NLP project, focusing on their performance metrics and underlying methodologies.

We utilized five distinct approaches for sentiment analysis, each leveraging different algorithms and techniques:

Transformers: Leveraging cutting-edge pre-trained models such as BERT, Transformers excel at capturing intricate contextual nuances within textual data. By contextualizing words in relation to their surrounding context, these models achieve superior sentiment classification accuracy compared to traditional approaches.

Logistic Regression: Esteemed for its simplicity and computational efficiency, Logistic Regression remains a formidable contender in sentiment analysis. By modeling the probabilities of different sentiment classes based on TF-IDF features, Logistic Regression provides a straightforward yet effective means of sentiment classification.

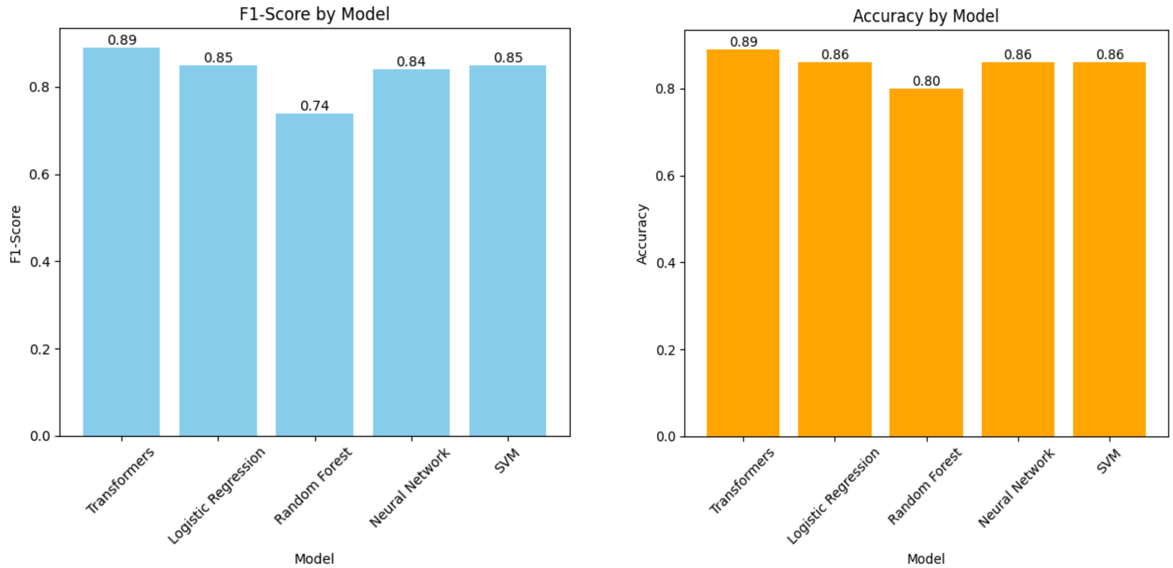
Random Forest: Dealing with the high dimensionality and sparsity inherent in TF-IDF vectors extracted from review texts, Random Forest employs an ensemble approach to mitigate overfitting. Its ability to handle complex feature spaces while maintaining robustness makes it a valuable asset in sentiment analysis tasks.

Neural Networks: Harnessing the power of layers and non-linear transformations, Neural Networks excel at learning intricate patterns embedded within textual data. Techniques like dropout and batch normalization enhance the network's generalization capabilities, enabling it to effectively capture the subtle nuances of sentiment expression.

SVM (Support Vector Machine): Operating in a transformed feature space, SVM constructs hyperplanes to delineate different sentiment classes. Its emphasis on maximizing the margin between classes ensures optimal separation, making it a reliable choice for sentiment analysis tasks.

We evaluated the performance of each sentiment analysis technique using two key metrics: F1-score and accuracy. The F1-score provides a balance between precision and recall, while accuracy measures the overall correctness of sentiment predictions.

Here are the performance scores achieved by each model:



F1- Score and accuracy

Transformers outperformed other models with the highest F1-score and accuracy. Their ability to capture intricate contextual information from textual data led to superior sentiment classification accuracy.

Logistic Regression and SVM demonstrated competitive performance, leveraging simpler algorithms and feature representations.

Random Forest exhibited lower performance compared to other models, possibly due to its limitations in handling high-dimensional and sparse feature spaces.

Neural Networks showcased strong performance, highlighting their capability to learn complex patterns inherent in sentiment analysis tasks.

In conclusion, our comparative study of sentiment analysis techniques reveals the effectiveness of advanced pre-trained models like Transformers in capturing nuanced sentiment expressions. However, traditional methods such as Logistic Regression and SVM continue to be viable options, especially for scenarios where computational resources are limited. Moving forward, further exploration and fine-tuning of these techniques could enhance their applicability and performance in real-world NLP applications. This comparative analysis serves as a foundation for selecting appropriate sentiment analysis techniques in NLP projects, considering factors such as accuracy, computational complexity, and interpretability.

Recommendation System

Location based recommendations

Since we want our system to recommend the restaurants from highest ranked to lowest ranked for a particular location, we need to choose a metric to decide the ranking of each of the restaurants.

To rank restaurants based on sentiment analysis, we calculated a cumulative rating for each restaurant using the sentiment scores of all its reviews. This cumulative rating was derived from the sentiment analysis results, where positive reviews contributed positively to the rating, negative reviews contributed negatively, and neutral reviews had little to no impact. By summing up the ratings of all the reviews for each restaurant, we obtained a cumulative rating that reflects the overall sentiment towards that establishment. Restaurants were then ranked from highest to lowest based on their cumulative rating, allowing us to recommend the most positively perceived ones first.

To determine the ranking of a restaurant, we did the sentiment analysis of every review for a restaurant using the SentimentIntensityAnalyzer from the nltk.vader module. The SentimentIntensityAnalyzer takes a text as input and returns a dictionary in the following format:

{'neg': 0.158, 'neu': 0.124, 'pos': 0.891, 'compound': 0.983}

This dictionary gives us the negative, neutral, positive and compound scores for the input text. The negative, neutral and positive scores range from 0 to 1, where if the score is closer to 1, the more intensely it represents that particular sentiment. The compound score ranges from -1 to 1 where a score closer to -1 represents a negative sentiment and a score closer to 1 represents a positive sentiment.

Since we know what the scores represent, we took the compound scores for all the reviews of a restaurant and made sure to map the polarity/compound score to a rating. This mapping was done using a function named map_score_to_rating() and the mapping was done in the following manner:

- Compound score in range -1 to -0.2 was given a rating of 1.
- Compound score in range -0.2 to 0.4 was given a rating of 2.
- Compound score in range 0.4 to 0.8 was given a rating of 3.
- Compound score in range 0.8 to 0.9 was given a rating of 4.
- Compound score greater than 0.9 was given a rating of 5.

The reason these ranges were chosen is that most of the reviews in our dataset were ranging from mildly positive to extremely positive leading to a lot of the restaurants having very similar cumulative ratings. By choosing the above ranges, we were able to clearly discern between the ratings and quality of every restaurant.

Once a rating was assigned to every review, we obtained a cumulative rating for these restaurants which would eventually be used to rank the restaurants from the most favorable ones to the least favorable ones.

```python
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer

nltk.download('vader_lexicon')
# Function to map polarity score to a rating
def map_score_to_rating(score):
    if score <= -0.2:
        return 1
    elif score <= 0.4:
        return 2
    elif score <= 0.8:
        return 3
    elif score <= 0.9:
        return 4
    else:
        return 5

# Instantiate the SentimentIntensityAnalyzer
sid = SentimentIntensityAnalyzer()

# Calculate the polarity score of every review in the dataset
df['Polarity_Score'] = df['Review Text'].apply(lambda x: sid.polarity_scores(x)['compound'])

# Map the polarity score to a rating on a scale of 1-5
df['Calculated_Rating'] = df['Polarity_Score'].apply(map_score_to_rating)

df.head()
```

Python

```python
# Aggregate the Calculated_Ratings for every restaurant to get the mean Calculated_Rating
mean_scores = df.groupby('Company name')['Calculated_Rating'].mean().reset_index()
```

Code snippets showing the mapping of sentiment scores to a rating and aggregating the ratings for a restaurant

Food based recommendations

Alongside providing the top rated restaurants to try, we also intend to provide must-try foods at each of these restaurants. To achieve this, we performed Named Entity Recognition (NER) on only the positively rated reviews to get the names of the foods. Since some of the positive reviews also contain the names of the foods that the reviewer particularly liked, we extract them and present them to the user so that they can make an informed choice on the best rated menu items for each restaurant.

We extracted the names of the foods from positive reviews using the “Dizex/InstaFoodRoBERTa-NER” transformer model. Once the names were extracted, only dishes containing multiple words were chosen to be recommended to the user since a lot of common singular words such as “bread”, “donuts”, “cakes” etc. are not considered to be valid food recommendations.

Therefore, we get a list of foods to try for every restaurant and compile it into a dictionary. In the end, we have multiple dictionaries where the key is the name of the restaurant and the values in these dictionaries represent the rating, location, list of top reviews and list of top foods. We compile each of these dictionaries into a single dictionary so that we can write it as a json file. This json file would then be used by the User Interface to render the top rated restaurants along with their rating, their top rated reviews and the top foods to try in each of them.

```python
restaurants = []
import pickle
import json

with open('food_recommendations.pkl', 'rb') as f:
    food_recommendations = pickle.load(f)

for name, rating in scores_dict.items():
    location = restaurant_location_dict[name]
    reviews = reviews_dict[name]
    foods = food_recommendations[name]
    restaurants.append({
        'name': name,
        'location': location,
        'rating': rating,
        'reviews': reviews,
        'foods': foods
    })

pprint.pprint(restaurants)

with open('restaurants.json', 'w') as json_file:
    json.dump(restaurants, json_file, indent=4)
```

Code snippet showing the compilation of multiple dictionaries to get a single list of dictionary objects dumped into a json file

Evaluation of Recommendation System

While our system provides recommendations of restaurants, it also provides the top 5 rated reviews for every restaurant. We get the top 5 rated reviews based on the polarity/compound score generated by the SentimentIntensityAnalyzer. It is safe to assume that the reviews with the highest scores will inherently be positive since the scores that are closest to 1 are considered to be the most positive. Hence, the gold label attached to all the top reviews will be 'positive'.

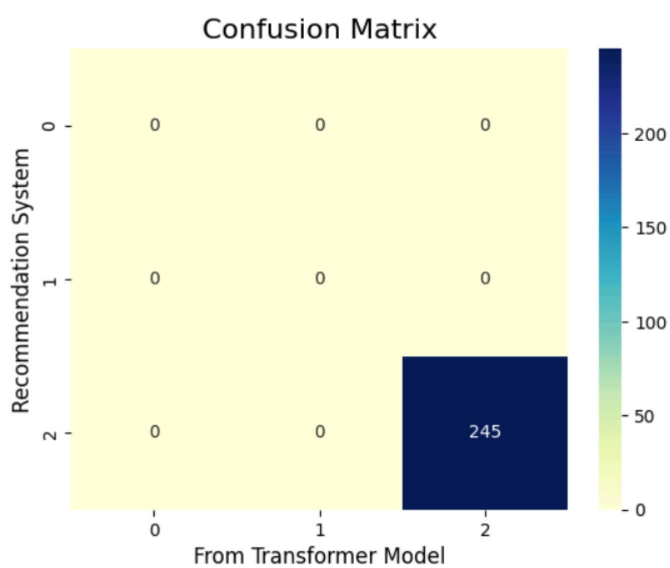
```
# Group by 'Company name' and get the top 5 reviews for each group
top_reviews = df.groupby('Company name').apply(lambda x: x.nlargest(5, 'Polarity_Score')).reset_index(drop=True)
top_reviews_for_restaurant = top_reviews.groupby('Company name')['Review Text'].apply(list).reset_index()
reviews_dict = top_reviews_for_restaurant.set_index('Company name')['Review Text'].to_dict()
```

Code snippet showing that the top 5 reviews are generated by taking the polarity (compound) scores into account

Previously, we trained our transformer model to do the sentiment analysis on every review to classify them as 'positive', 'neutral' or 'negative'. We attach label 0 to a negative review, label 1 to a neutral review and label 2 to a positive review. To evaluate our recommendation system and see if only the top reviews are being recommended to the user, we extracted all the top reviews of all the restaurants and performed sentiment analysis on each review using our trained transformer model. Since we have 49 restaurants in our dataset and we extracted 5 top reviews for each restaurant, we evaluated a total of 245 reviews. After evaluating these reviews with our trained transformer model, the following is the result of our evaluation:

- Accuracy: 1.0
- Precision: 1.0
- Recall: 1.0
- F1 score: 1.0

The following image shows the confusion matrix:



From the confusion matrix, we can see that all the recommended reviews have been classified as positive by our trained transformer model. This indicates that only the most positive reviews for a restaurant are being recommended to the user and the fact that they are indeed positive is being validated by our trained transformer model.

User Interface

The user interface was created using Streamlit. For the user interface, we are displaying the restaurants from the highest rated ones to the lowest rated ones. Once the user selects a location and clicks on the "Show Recommendations" button, the top rated restaurants are displayed for that particular location as indicated in the following screenshot:

By clicking on the arrow icon next to each restaurant, the user will be able to preview the top rated reviews as well as the top foods to try at that restaurant.

The json file that was created before by compiling all the dictionaries together is used by the UI to render the top rated restaurants for each location along with its rating, top reviews and top foods to try.

By clicking on the "Results of Comparative Study" tab, all the visualizations involved in Exploratory Data Analysis, comparative analysis of different models on sentiment analysis and the confusion matrix to evaluate the effectiveness of our recommendation system.

Future Scope

To expand the scope of our project in the future, the following are some of the novel ideas we intend to implement:

1. Dynamically adjust restaurant suggestions based on the time of day and user preferences

- Using existing data on user behavior and preferences, suggest appropriate options based on the time of day.
- Consider factors such as user location, historical dining patterns, and time of day to dynamically adjust recommendations.
- Employ machine learning algorithms to continuously learn and adapt to user preferences and trends.

2. Highly personalized suggestions with integration of user-specific data:

- Integrate user-specific data such as dietary preferences (e.g., vegan, gluten-free), past dining experiences, favorite cuisines, and preferred ambiance.
- Permit users to input their preferences directly into the system or leverage existing data.
- Utilize filtering techniques to recommend restaurants similar to those the user has enjoyed in the past while aligning new suggestions with their preferences.

3. Real-time feedback mechanism to continuously refine and improve recommendations:

- Implement a feedback feature where users can rate and provide feedback on recommended restaurants after dining experiences.
- Evaluate user comments in real time to see where improvements can be made and how effective the recommendations are.
- Utilize natural language processing (NLP) and sentiment analysis tools to glean information from user reviews and comments.

4. Make recommendations relevant and personalized over time:

- Recommendation algorithms can be enhanced over time by using reinforcement learning approaches in conjunction with user input and interactions.
- Update user preferences and profiles on a regular basis in response to new information and changes in behavior.
- In order to improve personalization and relevance, the recommendation engine should be regularly reviewed and updated to include new features, data sources, and algorithms.

Conclusion

In the rapidly evolving landscape of the hospitality industry, the significance of customer feedback cannot be overstated. "Cuisine Compass," our innovative restaurant recommendation system, leverages advanced sentiment analysis techniques to revolutionize the way diners explore dining options. Through meticulous data acquisition, preprocessing, and exploratory analysis, we have laid a robust foundation for the development of a highly personalized and dynamic recommendation platform.

Our approach transcends the capabilities of traditional recommendation systems by incorporating comprehensive sentiment analysis. This enables a deeper understanding of user reviews, transforming raw data into actionable insights that align perfectly with individual preferences. By leveraging cutting-edge natural language processing techniques, "Cuisine Compass" delivers recommendations that are not only accurate but also reflect the nuanced tastes and dietary requirements of users.

Additionally, our comparative analysis of various models for sentiment analysis has played a crucial role in identifying the most effective algorithms. By evaluating models like Transformers, SVM, and Neural Networks, we have determined which technologies best capture the subtle nuances of consumer sentiment, thereby enhancing the precision and reliability of our recommendations.

In conclusion, "Cuisine Compass" not only addresses a practical need for reliable and nuanced restaurant recommendations but also demonstrates the transformative potential of advanced analytics and machine learning in the service industry. By empowering users to make informed dining decisions and enriching their culinary experiences, our system contributes to shaping a more enjoyable and satisfying dining landscape. Through this project, we envision a future where the act of choosing where to dine is as delightful and fulfilling as the dining experience itself, fostering a more vibrant and interconnected dining community.

References

1. Bordoloi M, Biswas SK, “Sentiment analysis: A survey on design framework, applications and future scopes”, Artif Intell Rev, 2023 Mar 20:1-56, doi: 10.1007/s10462-023-10442-2, Epub ahead of print, PMID: 37362892; PMCID: PMC10026245
2. S. Dhar, S. Pednekar, K. Borad and A. Save, "Sentiment Analysis Using Neural Networks: A New Approach," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1220-1224, doi: 10.1109/ICICCT.2018.8473049
3. Duong, HT., Nguyen-Thi, TA. A review: preprocessing techniques and data augmentation for sentiment analysis. Comput Soc Netw 8, 1 (2021). https://doi.org/10.1186/s40649-020-00080-x
4. Barbhuiya, Towfiqu. “Sentiment Analysis of Amazon Reviews Using Natural Language Processing.”Medium,8 February 2024, https://medium.com/@eulene.tech/sentiment-analysis-of-amazon-reviews-using-natural-language-processing-6ecb3e253a5e. Accessed 3 May 2024
5. “Getting Started with Sentiment Analysis using Python.” Hugging Face, 2 February 2022, https://huggingface.co/blog/sentiment-analysis-python. Accessed 3 May 2024.
6. Mariana, Aqila Fadia. “Natural Language Processing(NLP): Restaurant Review Sentiment Analysis with NLTK.” Artificial Intelligence in Plain English, 9 October 2023, https://ai.plainenglish.io/natural-language-processing-nlp-restaurant-review-sentiment-analysis-with-nltk-dcf1986a4989. Accessed 3 May 202
7. Reddy, Vasista. “Sentiment Analysis using SVM. Sentiment Analysis is the NLP technique, ScrapeHero.” Medium, 11 November 2018, https://medium.com/scrapehero/sentiment-analysis-using-svm-338d418e3ff1. Accessed 3 May 2024.