VOICE OF THE CUSTOMER

(Sentiment Analysis using Yelp Data)

Final Project Work for
MSA 8050: Scalable Data Analytics
Instructor: Dr. Kai Zhao

April 23, 2024

Contributors:
Naveen Seelam
Surya Kiran Varma Vegesna
Dipak Bhattarai
Shangze Li
Shakira Robinson

Table of Contents

1. Cloud Setup

The initial step of the analysis involved setting up a cloud environment to prepare the Yelp review dataset for subsequent analysis using Spark. We utilized GSU (Georgia State University) Cluster for this phase. This section details the process undertaken to establish the requisite cloud infrastructure ensuring data accessibility and maintaining data integrity.

1. **Project setup in cloud**: A dedicated project folder was created within the GSU cluster to organize analysis files. This included the compressed dataset and all relevant scripts needed for processing.

```
D:\gsu_projects>ssh dbhattarai1@arc.insight.gsu.edu
dbhattarai1@arc.insight.gsu.edu's password:
Web console: https://compute-11.insight.gsu.edu:9090/ or https://10.230.100.224:9090/

Last login: Fri Apr 19 00:47:20 2024 from 131.96.47.236
[dbhattarai1@compute-11 ~]$ mkdir sda_final_project
[dbhattarai1@compute-11 ~]$ cd sda_final_project/
[dbhattarai1@compute-11 sda_final_project]$ pwd
/home/dbhattarai1/sda_final_project
[dbhattarai1@compute-11 sda_final_project]$
```

2. **Uploading data and scripts:** The compressed dataset and script files were uploaded to the designated project folder in the cloud storage. The dataset was then unzipped within the cloud environment for further processing.

```
D:\gsu_projects>scp yelp-dataset.zip dbhattarai1@arc.insight.gsu.edu:/home/dbhattarai1/sda_final_project
dbhattarai1@arc.insight.gsu.edu's password:
yelp-dataset.zip                                              100% 2569MB   2.7MB/s   15:45

D:\gsu_projects>
```

```
D:\gsu_projects>scp yelp_review_preprocess.py dbhattarai1@arc.insight.gsu.edu:/home/dbhattarai1/
sda_final_project
dbhattarai1@arc.insight.gsu.edu's password:
yelp_review_preprocess.py                                      100%  782    21.1KB/s   00:0
0

D:\gsu_projects>
```

3. **Transferring and verifying data in HDFS:** We transferred data files from cloud storage to the HDFS (Hadoop Distributed File System). Before running the Spark script, we verified that the data files were successfully transferred and accessible with HDFS.

```
[dbhattarai1@compute-11 sda_final_project]$ hadoop fs -ls
Found 1 items
-rw-r--r--   3 dbhattarai1 supergroup 3791120545 2024-04-21 22:40 yelp_review.csv
[dbhattarai1@compute-11 sda_final_project]$
```

4. **Running Spark Script**: Spark scripts were written to utilize Spark DataFrames for processing the data. These scripts involved downsizing the dataset for further analysis such as filtering rows, selecting specific columns.

```
[dbhattarai1@compute-11 sda_final_project]$ nohup spark-submit yelp_review_preprocess.py
nohup: ignoring input and appending output to 'nohup.out'
[dbhattarai1@compute-11 sda_final_project]$
```

5. **Downloading compressed data for further analysis:** Finally, the processed and downsized data was compressed again and downloaded back to a local computer for further analysis using Google Colab.

```
D:\gsu_projects>scp dbhattarai1@arc.insight.gsu.edu:/home/dbhattarai1/sda_final_project/
yelp_review_cleaned.zip d:\gsu_projects\
dbhattarai1@arc.insight.gsu.edu's password:
yelp_review_cleaned.zip                       100%  699MB   9.3MB/s   01:14

D:\gsu_projects>
```

2. Data Cleaning

Preprocessing and cleaning downsized dataset via Colab:

The data cleaning process was a critical step in our analysis of the yelp review dataset. The dataset originally contained a sizable number of rows, which we downsized to 3 million reviews using cluster computing. After being limited to 3 million reviews, we moved to Google Colab and set up Apache Spark environments for further refinement.

After configuring the Spark environment and loading the necessary libraries. We use simple commands to validate the structure of our dataset. In summary, it comprises 9 columns, each initially of the string data type: ‘review_id’, ‘user_id’, ‘business_id’, ’stars’, ‘date’, ‘text’, ’useful’, ’funny’ and ’cool’. We also observed several mismatched issues that some text misplaced in different columns, resulting the overall size of dataset exceed 3 million rows. Then we identified ‘user_id’, ‘text’ and ‘stars’ as essential columns for our sentiment analysis and proceeded to eliminate rows that contained null value. We made the strategic decision to simplify our dataset by dropping the columns ‘useful’, ‘funny’, and ‘cool’. These fields were not aligned with the core objectives of our analysis.

The ‘stars’ column, which displays the user’s rating for their reviews, contained some values outside the expected range of 1 to 5. We opted to remove any rows that did not have values that fall inside this anticipated range. On the other hand, the ‘user_id’ column has fixed 22 characters in length, the possible input contains from a to z, capital a to capital a, 0 to 9, underscores and dashes. We refined our dataset by removing any rows with ‘user_id’ that did not match our specified pattern.

After the data cleaning process, the resulting validated dataset has 2484513 rows, which have been retained for our sentiment analysis.

3. Data Preprocessing

Processing User Categories with RDDs

We employed Resilient Distributed Datasets (RDDs), a fundamental data structure of Apache Spark, to distribute the processing load and enable a scalable analysis. The RDDs were populated with yelp data, which represents the dataset after preprocessing and validation.

We classified users into categories such as 'generous', 'normal', and 'picky', reflecting their rating behavior. Leveraging Spark's RDDs allowed us to manage the data efficiently. This classification allowed us to understand the general disposition of users towards the businesses they reviewed on Yelp. By assigning users to 'generous', 'picky', or 'normal' categories, we aimed to discern patterns in their rating behaviors.

Computing Average Ratings:

We began by calculating the average star ratings for each user, utilizing the groupBy and agg functions to aggregate ratings by ‘user_id.’ The resulting average ratings were then merged with the original dataset, ensuring that each review could be associated with the reviewer's typical rating behavior.

Defining Rating Categories:

With the average ratings in place, we set thresholds to define our categories:

- **Generous**: Users with an average rating of 4 stars or higher were defined as generous reviewers.
- **Normal**: Users whose average rating fell between 2 and 4 stars were categorized as normal reviewers.
- **Picky**: Users with an average rating of 2 stars or lower were deemed picky reviewers.

Sentiment Analysis on Text Reviews

Implementation of Sentiment Analysis:

To dissect the sentiment embedded in Yelp reviews, we constructed a Natural Language Processing (NLP) pipeline, complemented by sentiment analysis functions. User-defined functions (UDFs) played a crucial role in this process, evaluating the polarity of review text as a metric of sentiment.

Polarity and Sentiment Classification:

We introduced get_polarity, a UDF that computes the sentiment polarity of review text, with a higher polarity indicating a more positive sentiment. Subsequently, classify_sentiment was employed to categorize the polarity into 'Positive', 'Negative', or 'Neutral' sentiments.

NLP Pipeline Configuration:

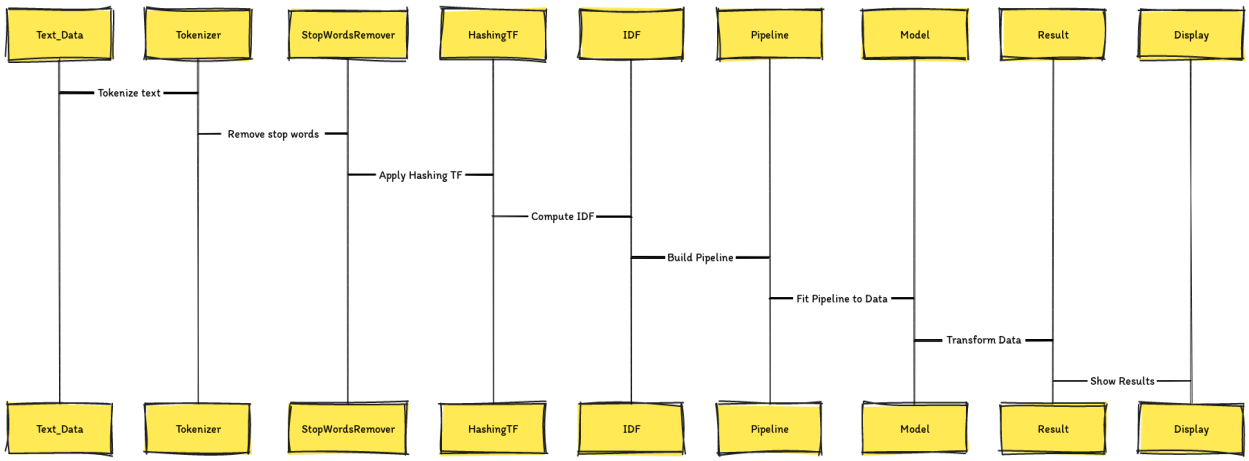
Our NLP pipeline comprised the following stages:

- **Tokenization**: Segmenting text into individual words.
- **Stop Word Removal**: Filtering out common words that contribute little to sentiment.
- **Feature Hashing**: Converting text into numerical representation suitable for machine learning.
- **Inverse Document Frequency (IDF)**: Weighing the importance of each word within the corpus of documents.

Upon constructing and fitting the model to our dataset, each review was transformed through the pipeline, and the sentiment polarity and category were appended to the resultant DataFrame.

Insight Extraction:

The integration of sentiment polarity and classification into our data model enriches the dataset with direct measures of sentiment, paving the way for in-depth sentiment analysis. This enhancement enables a granular examination of the emotional undertones in Yelp reviews and allows us to discern overarching sentiment trends among Yelp users.



Sentiment Patterns & Counts Amongst Diverse Yelp Reviewers:

In the pursuit of unpacking the sentiments expressed by different types of reviewers on Yelp, we deployed a function that intricately plots sentiment counts. This function was instrumental in illustrating the sentiment distribution for 'generous', 'normal', and 'picky' reviewers, providing a visual representation of how each group's ratings align with sentiment polarities.

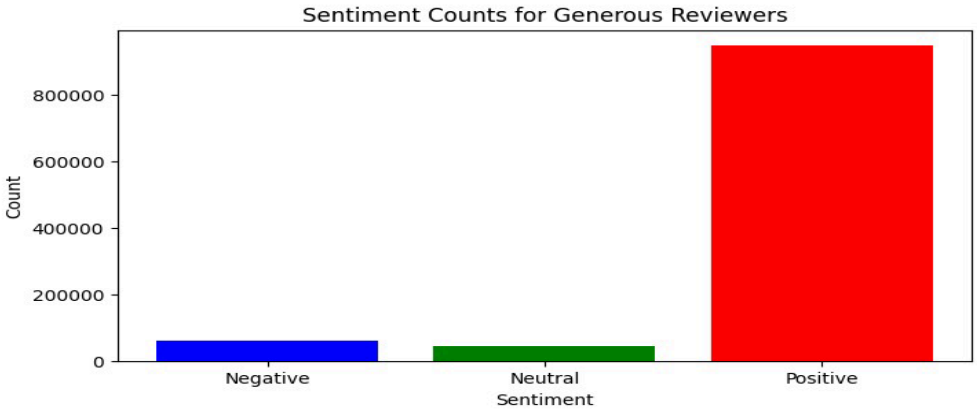
Analytical Approach:

A methodical plotting function was designed to filter the classified dataset based on user categories and then group the data by sentiment. This aggregation yielded the number of reviews associated with each sentiment classification within the respective user category. The sentiments were encoded as 'Negative', 'Neutral', and 'Positive'.

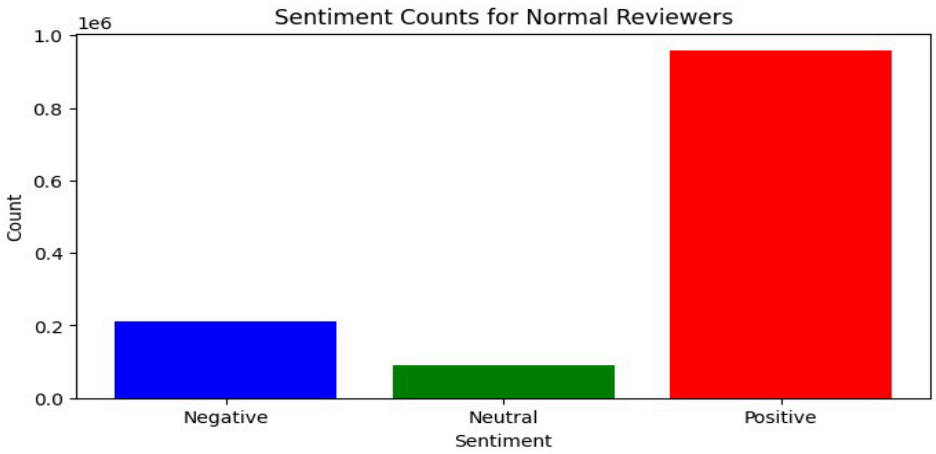
Visual Insights:

The resulting plots present a stark contrast among the three categories:

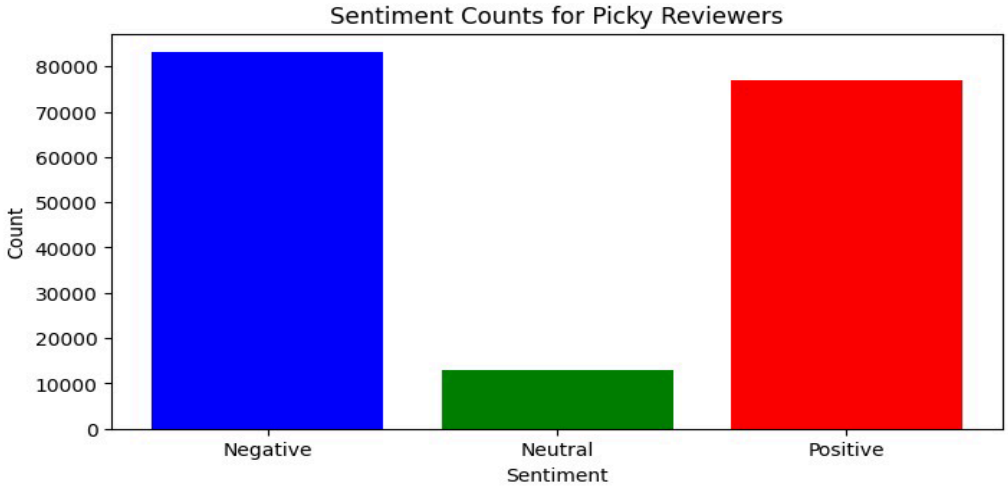
- **Generous Reviewers**: Dominantly positive sentiments, with a remarkable number of reviews celebrating the high end of sentiment polarity.



- **Normal Reviewers**: A more balanced sentiment distribution, yet with a clear skew towards positivity, which might indicate a satisfactory experience.



- **Picky Reviewers**: A significant presence of negative sentiments, substantiating their categorization, but not without a substantial count of positive sentiments as well.



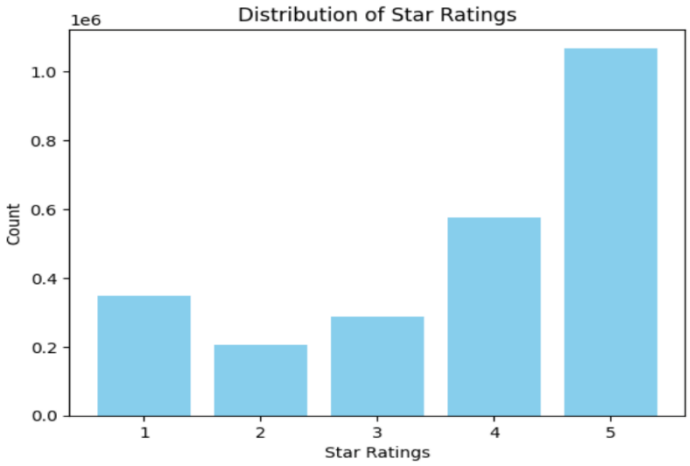
Conclusive Remarks:

These findings eloquently display the diverse perspectives within the Yelp ecosystem. 'Generous' reviewers are overwhelmingly positive, which could suggest either a bias towards favorable reviews or genuinely pleasant experiences. 'Normal' reviewers exercise moderation in their sentiments, and 'picky' reviewers, despite their critical nature, do not shy away from praising when deemed fit. This sentiment analysis enriches our understanding of consumer behavior and preferences, indicating that even among the most critical users, positive experiences can resonate and be reflected in their reviews.

4. Exploratory Data Analysis

Star Rating Distribution Analysis

The analysis aimed to discern the frequency of different star ratings, which are indicative of customer satisfaction levels. Star ratings were mapped from their original floating-point representation to integers, establishing consistency and facilitating the aggregation process. We counted the occurrences of each rating level by mapping each to a key-value pair and applying a reduceByKey operation. The results were subsequently sorted based on the star rating values.
The analysis underscores a predominant inclination towards higher star ratings among Yelp users, with 5 stars being the most frequently assigned rating. This observation may reflect a positive bias in user reviews or indicate satisfactory experiences by most reviewers. Our subsequent sentiment analysis will delve deeper, beyond numerical ratings, into the textual feedback provided by users to understand the nuances and contexts of these ratings.




```
1 star: 346793
2 star: 207072
3 star: 287633
4 star: 574508
5 star: 1068507
```

Unique User Identification in Yelp Reviews

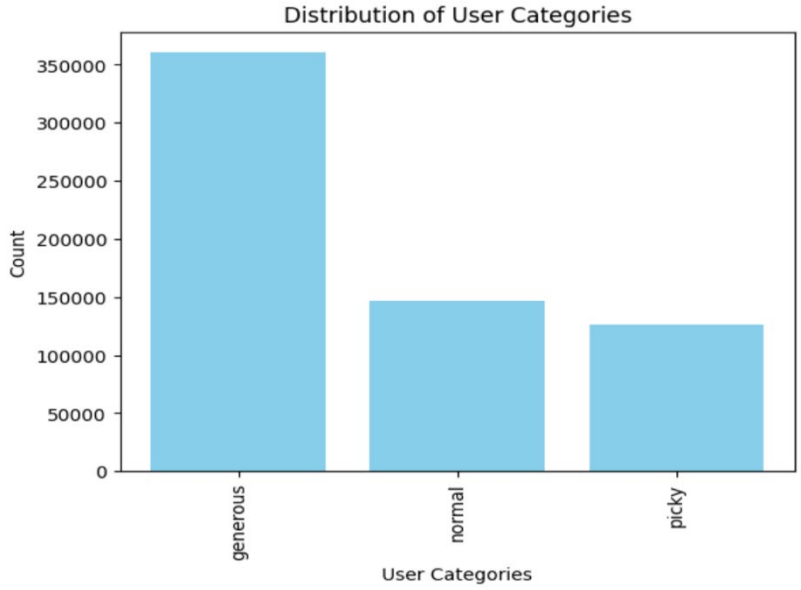
To quantify the individual contributions to Yelp's review dataset, it was imperative to ascertain the count of unique users. This ensures that each user's perspective is considered singularly, providing an equitable basis for further sentiment analysis.
The final count of unique users was determined to be: 633771

By providing a clear count of unique users, we lay the groundwork for subsequent analyses that will delve deeper into the nature of the reviews themselves, ensuring that each user's voice is given equal weight in the overall sentiment analysis.

Distribution of Yelp Users

Analysis of User Category Distribution:

By mapping each user to their respective category and applying a count operation, we aggregated the total number of users in each category. This count was achieved through a reduceByKey operation, which summed up the occurrences of each category. The categories were then ordered by count in descending order to identify which category had the highest representation.




```
generous user: 360455
normal user: 146832
picky user: 126484
```

This categorization reveals the propensity of Yelp users to give ratings. The higher number of 'generous' users suggests that a massive portion of the Yelp community may tend to give favorable

ratings. On the other hand, 'picky' users, while fewer, indicate the presence of a critical audience that is harder to please.

Word Cloud Analysis of Yelp Reviewers' Sentiment

Through the visual artistry of word clouds, we have illustrated the most frequent terms used by different categories of Yelp reviewers. These word clouds were generated by aggregating the text of reviews from each respective category: generous, normal, and picky.

Generous Reviewers

In the word cloud representing 'generous' reviewers, positive descriptors like "**amazing,**" "**love**," and "**best**" dominate the visual space, underscoring the positive nature of their reviews.

Normal Reviewers

For 'normal' reviewers, the word cloud shows a balanced use of positive terms like "**good**" and "**great**," while also including moderate terms such as "**nice**" and "**delicious**," reflecting a balanced perspective in their reviews.

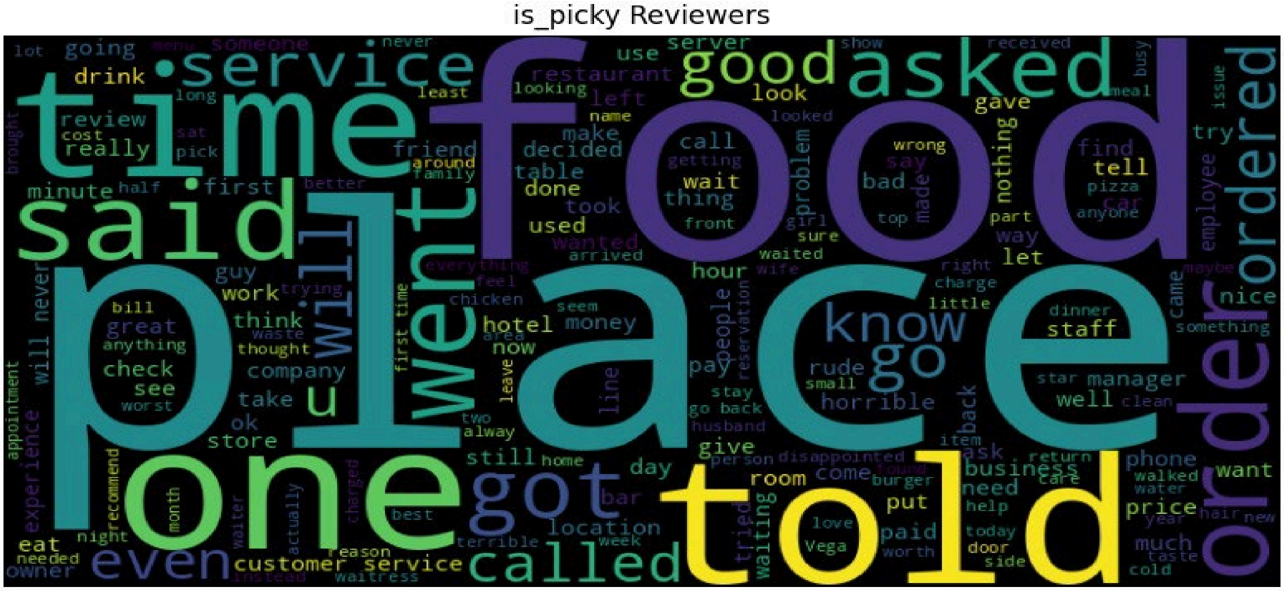
is_normal Reviewers

Picky Reviewers

Conversely, the word cloud for 'picky' reviewers highlights a different sentiment. Words such as "**asked,**" "**told**," and "**bad**" are prominent, which may suggest the critical nature of their feedback.

These word clouds not only add a visual dimension to the data but also quickly convey the prevailing sentiments within each reviewer category, providing an at-a-glance understanding of the textual data from thousands of reviews.

is_picky Reviewers



5. Sentiment Classification with Logistic Regression and Naive Bayes Models

Model Selection and Setup

For our sentiment analysis task, we chose two distinct machine learning models: Logistic Regression and Naive Bayes. These models were selected for their suitability in binary classification problems and their ability to handle large datasets efficiently.

- **Logistic Regression**: A robust statistical method that predicts the probability of a binary outcome. It is particularly known for its interpretability and was configured with regularization to prevent overfitting.
- **Naive Bayes**: A probabilistic classifier that assumes independence between predictors. Naive Bayes is known for its simplicity and speed in making predictions.

Train the Models

A pipeline was crafted to preprocess the text data, involving tokenization, stop word removal, term frequency counting, and TF-IDF transformations. This prepared dataset was then split into training and testing sets, with 70% of the data used for training and the remaining 30% for evaluation purposes.

Each model was trained using the features vector derived from the pipeline, with Logistic Regression employing an iterative process limited to 10 iterations, while Naive Bayes utilized a smoothing parameter set to 1.0.

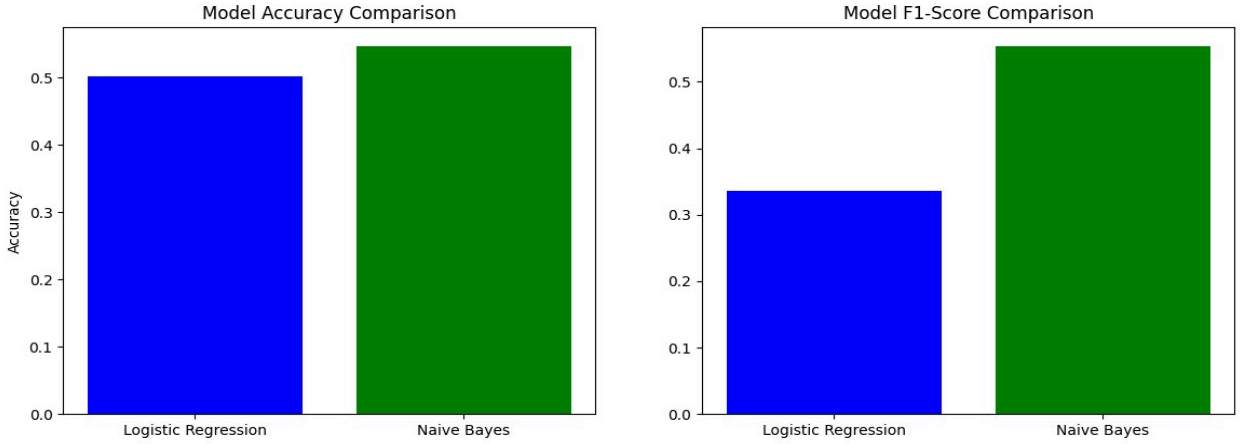
6. Model Evaluation & Result

Post-training, both models were applied to the test data to predict sentiments. The performance of each model was gauged using two metrics:

- **Accuracy**: The proportion of correctly predicted instances to the total instances in the dataset.
- **F1-Score**: The weighted average of precision and recall, providing a more comprehensive look at the model's performance, especially when the classes are imbalanced.

Results Visualization

The comparative results were displayed in a bar chart format, facilitating a visual comparison between the models:



- **Model Accuracy Comparison**: The Naive Bayes model achieved a higher accuracy compared to the Logistic Regression model, indicating a more reliable predictive performance.
- **Model F1-Score Comparison**: Consistent with the accuracy results, the Naive Bayes model also registered a higher F1-score, suggesting it maintains a better balance between precision and recall.

The analysis culminated in the conclusion that the Naive Bayes model, with its superior accuracy and F1-score, stands out as the more effective choice for classifying sentiments in Yelp reviews within our dataset.

7. Conclusion

Our analysis focuses on analyzing Yelp reviews to understand the sentiment of customers towards businesses. The various aspects of the analysis, including data cleaning, data preprocessing, sentiment analysis, exploratory data analysis, and sentiment classification using machine learning models.

Data cleaning involved downsizing the dataset to 3 million reviews using cluster computing. Data preprocessing involved categorizing users into 'generous', 'normal', and 'picky', based on their rating behavior, and computing average ratings. Sentiment analysis was conducted using Natural Language Processing techniques to analyze the sentiment polarity and classify reviews as 'Positive', 'Negative', or 'Neutral'.

Exploratory data analysis included analyzing star rating distributions, identifying unique users, and analyzing user category distributions. Word clouds were used to visually represent the most frequent terms used by different categories of reviewers. Sentiment classification was performed using Logistic Regression and Naive Bayes models, with Naive Bayes showing superior accuracy and F1-score in predicting sentiment.

The sentiment analysis not only uncovers the sentiment trends among different types of reviewers but also sheds light on the diverse perspectives and behaviors within the Yelp community. By integrating sentiment polarity and classification into the data model, the analysis provides a comprehensive understanding of consumer sentiment and preferences, indicating that positive experiences can resonate even among the most critical users. The exploration of sentiment through text reviews, visualization of sentiment patterns, and sentiment classification using machine learning models collectively contribute to an insightful analysis of the voice of the customer on Yelp.

Overall, the analysis provided insights into the sentiment patterns among different types of Yelp reviewers, highlighting the diversity of perspectives within the Yelp ecosystem. The findings enriched the understanding of consumer behavior and preferences, displaying a range of sentiments across different user categories. The sentiment analysis enhanced the dataset with direct measures of sentiment, allowing for a detailed examination of emotional undertones in Yelp reviews.