Exploring Review Sentiments with AI: A Strategic Analysis Project

Hema Deepika Thanigai Arasu, Kritthika Shanmugam, SathiyaShivani Sathish Kumar and VarunPrakash Shanmugam (Team Number: 2, DS510-Artificial Intelligence for Data Science, School of Technology and Computing, City University of Seattle)

[thanigaiarasuhemade@cityuniversity.edu](mailto:thanigaiarasuhemade@cityuniversity.edu)

[shanmugamkritthika@cityuniversity.edu](mailto:shanmugamkritthika@cityuniversity.edu)

[sathishkumarsathiya@cityuniversity.edu](mailto:sathishkumarsathiya@cityuniversity.edu)

[shanmugamvarunpraka@cityuniversity.edu](mailto:shanmugamvarunpraka@cityuniversity.edu)

**ABSTRACT**

This project develops a sentiment analysis model utilizing Naive Bayes classification to interpret customer sentiments from restaurant reviews. The primary objective is to automate the categorization of customer feedback into positive or negative sentiments based on their textual reviews. The methodology begins with data manipulation and analysis using the pandas library to load and preprocess the dataset, which includes handling structured data and cleaning missing values. The project employs the scikit-learn library's Count Vectorizer for text data transformation, converting reviews into a matrix of token counts, facilitating the application of machine learning algorithms. The Multinomial Naive Bayes classifier is then trained on this numerical data to predict sentiment labels. The process involves splitting the dataset into training and testing sets to validate the model's performance, ensuring the accuracy and reliability of the sentiment analysis. Evaluation metrics, including accuracy score and classification report from scikit-learn, are utilized to assess the model's effectiveness in categorizing sentiments. The classification report provides detailed insights into the precision, recall, f1-score, and support for each class, enabling a comprehensive understanding of the model's performance. This sentiment analysis model stands as a valuable tool for businesses, especially in the hospitality sector, enabling them to derive insights from customer feedback and make informed decisions to enhance service quality and customer satisfaction.

**Keywords:** Sentiment analysis, Artificial intelligence, Online customer reviews, Deep learning, Natural language processing, Business intelligence.

1. **INTRODUCTION**

In the digital era, online reviews have become a major influence on consumer behavior, serving as a vital source of customer feedback for businesses. Particularly in the hospitality sector, restaurant reviews play a crucial role in shaping potential customers' perceptions and decisions. Recognizing the importance of efficiently analyzing this vast amount of data, our project focuses on developing a sentiment analysis model tailored specifically for restaurant reviews. This model aims to automate the classification of reviews into binary sentiment categories: positive and negative.

The foundation of our project is built on robust data manipulation and analysis techniques, utilizing the powerful Python library and pandas. This library aids in handling structured data, enabling us to organize and prepare the reviews dataset for the analytical process. We begin by loading the restaurant reviews from a CSV file into a pandas Data Frame, which offers a comprehensive view of the data and allows for efficient manipulation and cleaning.

The next pivotal step involves transforming the raw review ratings into binary sentiment labels, a process crucial for the sentiment analysis model. We classify ratings of 4 and above as positive sentiments and ratings below 4 as negative sentiments. This binary classification simplifies the analysis and provides a clear framework for assessing customer feedback.

To ensure the accuracy and reliability of our model, we employ the Count Vectorizer from scikit-learn's feature extraction module. This tool converts text documents into a matrix of token counts, transforming the textual data into a format suitable for machine learning algorithms. Subsequently, we leverage the Multinomial Naive Bayes algorithm, implemented via the MultinomialNB class in scikit-learn, renowned for its effectiveness in handling classification with discrete features, such as word counts in text classification.

Our project not only focuses on developing a sentiment analysis model but also emphasizes the importance of preprocessing and data cleaning. By eliminating rows with missing values in crucial columns, we ensure the integrity and quality of our dataset, paving the way for more accurate and meaningful analysis.

We employ Word Clouds to visually represent the most frequent words in positive and negative reviews. Additionally, our project extends to include k-Nearest Neighbors (kNN) as an alternative classification algorithm. Leveraging scikit-learn's KNeighborsClassifier, we explore the effectiveness of this model in comparison to Naive Bayes. By splitting the dataset, vectorizing the text data, and training the kNN classifier, we broaden our analytical approach.

1. **LITERATURE REVIEW**

Sentiment analysis, particularly within the realm of consumer reviews, has garnered extensive focus in the academic and commercial sectors due to its significant implications for business intelligence and customer relationship management. The burgeoning field has evolved from mere polarity detection to sophisticated analyses that decipher nuanced sentiments from textual data. In this regard, the application of sentiment analysis to restaurant reviews represents a critical area of study, as evidenced by the growing body of literature exploring the influence of consumer feedback on dining experiences and business outcomes (Liu, 2012; Pang & Lee, 2008).

Central to the development of sentiment analysis tools is the implementation of machine learning algorithms capable of classifying text into predefined categories. The transformation of raw text into a structured format suitable for algorithmic interpretation involves techniques such as tokenization, vectorization, and feature extraction, as outlined in seminal works by Salton and McGill (1983) and further refined in recent studies (Wang & Manning, 2012). The Count Vectorizer, a common tool in text analysis, exemplifies this process by converting text documents into matrices of token counts, facilitating the subsequent application of machine learning models.

Among the various algorithms employed in sentiment analysis, the Naive Bayes classifier has been particularly noted for its efficacy in handling discrete data and its suitability for text classification tasks (McCallum & Nigam, 1998). Its application to sentiment analysis is well-documented, with researchers praising its simplicity, computational efficiency, and satisfactory performance across diverse datasets (Rennie et al., 2003; Taddy, 2013). This probabilistic model, despite its assumption of feature independence, has proven effective in distinguishing between positive, neutral, and negative sentiments, especially in contexts where the feature space is vast and the relationship between words is complex.

**3. METHODOLOGY**

This project employs a structured approach to develop a sentiment analysis model, focusing on classifying restaurant reviews into binary sentiment labels: positive and negative. The methodology is grounded in established machine learning practices and utilizes Python’s popular libraries for data handling and model development.

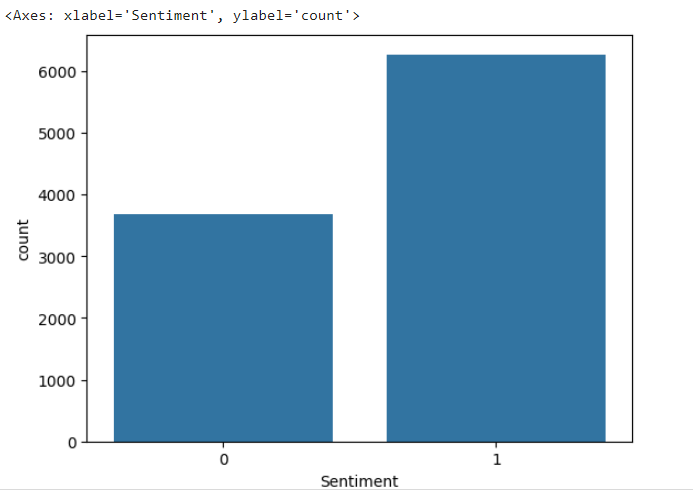
The methodology commences with library installation, employing pandas, scikit-learn, and Matplotlib. Text preprocessing utilizes CountVectorizer for sentiment analysis preparation. Multinomial Naive Bayes classifies sentiments, and KNN introduces a comparative model analysis. Evaluation metrics provide performance insights, while visualization tools like Seaborn and WordCloud enhance textual data understanding.

The dataset comprising restaurant reviews is loaded into a pandas Data Frame from a CSV file, a standard format for structured data. This step is crucial for organizing the raw data, enabling efficient manipulation and analysis. The DataFrame structure facilitates a clear overview of the dataset, including essential attributes such as review texts and ratings.

A screenshot of a test

Description automatically generated

After data loading, the methodology involves preprocessing the dataset to convert numerical ratings into binary sentiment labels. This conversion is based on a predefined threshold: ratings of 4 and above are classified as positive (1), while ratings below 4 are deemed negative (0). This binary classification simplifies the sentiment analysis, making it more focused and manageable. Additionally, the dataset is cleansed of missing values, particularly in the 'Review' and 'Sentiment' columns, to ensure data quality and integrity. Cleaning the data is a fundamental step to prevent the machine learning model from learning from incomplete or skewed data, thereby enhancing the accuracy and reliability of the analysis.



In preparation for sentiment analysis, we systematically converted all reviews to lowercase, promoting uniformity. Simultaneously, we removed punctuation to eliminate irrelevant noise and standardize the text further. These preprocessing steps enhance the model's accuracy by ensuring consistent interpretation of words.

**Word Cloud**

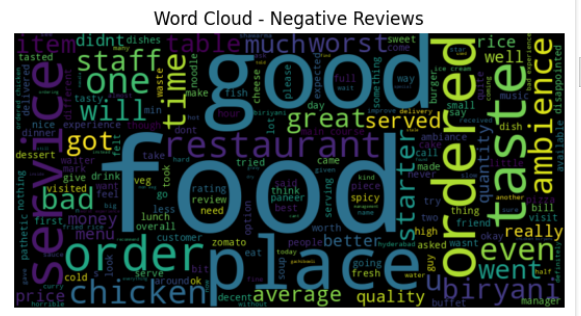
Word Clouds visually highlight frequently occurring words, offering an immediate overview of key themes. By representing words in varying sizes based on frequency, they emphasize prominent terms. This graphical tool simplifies complex textual data, aiding in quick pattern recognition. Word Clouds serve as a user-friendly method for understanding data sentiments and prevalent topics.

The Word Cloud visualizations for positive and negative reviews based on sentiment labels is generated. It aggregates positive and negative review texts separately and creates Word Clouds to visually represent the most frequent words in each sentiment category. The Word Clouds provide quick visualizations into prevalent themes and sentiments within the dataset, aiding in the understanding of customer perceptions and preferences.

**Positive review-word cloud**



**Negative review-word cloud**



The core of the methodology lies in feature extraction and model training. Using the Count Vectorizer from scikit-learn, the textual data is transformed into a format suitable for machine learning: a matrix of token counts. This process converts the raw text into numerical data, enabling the application of statistical methods for sentiment analysis. The choice of Count Vectorizer is motivated by its effectiveness in handling text data and its compatibility with Naive Bayes algorithms, known for their suitability in text classification tasks.

**Naive Bayes**

Following feature extraction, the Multinomial Naive Bayes (MultinomialNB) algorithm is applied to the processed data. This choice is informed by the algorithm’s proficiency in handling discrete features, such as word counts, making it ideal for text classification problems. The dataset is divided into training and testing sets to evaluate the model's performance accurately, employing metrics such as accuracy score and classification report from scikit-learn. These metrics provide a comprehensive understanding of the model's effectiveness in classifying sentiments, guiding further refinements and adjustments to enhance performance.

Naive Bayes is a probabilistic classification algorithm widely utilized in machine learning for various tasks, including sentiment analysis. Its simplicity and effectiveness make it particularly suited for text classification problems. Despite its "naive" assumption of feature independence, Naive Bayes excels in practice, especially with discrete features like word counts in text data.

The algorithm calculates the probability of a document belonging to a particular class based on the observed frequencies of words. This makes it well-suited for sentiment analysis, where the occurrence of specific words can strongly indicate positive or negative sentiments. Naive Bayes has proven to be robust, efficient, and requires minimal training data, making it a popular choice for tasks involving natural language processing.

In sentiment analysis, Naive Bayes categorizes text into predefined sentiment classes, providing a foundation for automated assessment of sentiment in reviews, comments, or other textual data. Its simplicity, speed, and reliable performance contribute to its widespread adoption in sentiment analysis applications across various domains.

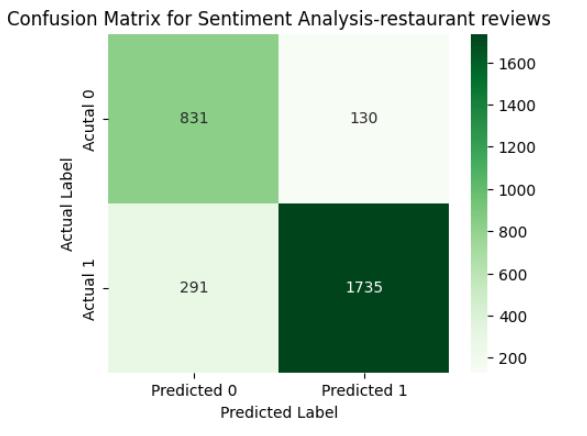
**K-Nearest Neighbors Classifier**

Analyzing with k-Nearest Neighbors (kNN), the scikit-learn KNeighborsClassifier is used as an alternative classification strategy for restaurant reviews. The dataset underwent division into training and testing subsets, and text data underwent vectorization using the CountVectorizer. The kNN model, configured with a parameter k=3, underwent training on the vectorized training data. Following this, predictions were generated on the test set, and the model's efficacy was assessed through accuracy metrics and a detailed classification report. Visual aids, including the confusion matrix and heatmap, were employed to illustrate the kNN classifier's performance. This enriches our analytical framework, allowing for a comparative assessment alongside the Naive Bayes model and contributing to a more comprehensive comprehension of our sentiment analysis methodology.

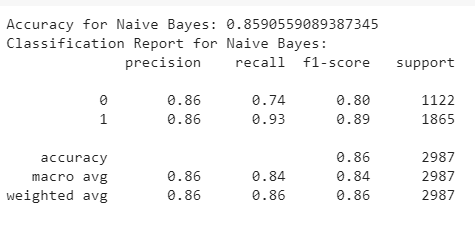
1. **RESULTS**

**Naive Bayes**

The provided results indicate an accuracy of approximately 86% for the Naive Bayes classifier. The confusion matrix illustrates the model's performance with a breakdown of true positive, false positive, true negative, and false negative values:



The overall accuracy is calculated as 0.8590559089387345.



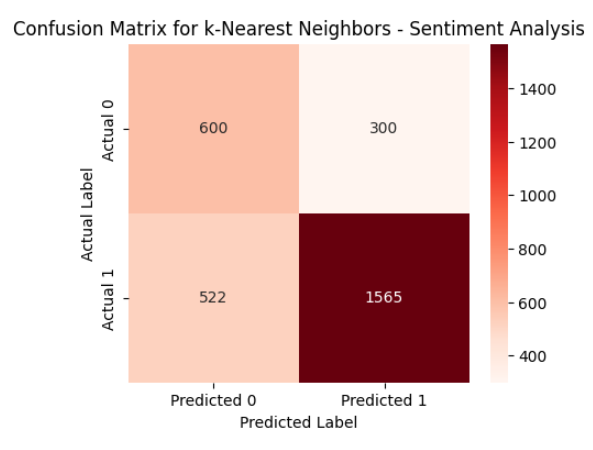
In the analysis using the Naive Bayes classifier, an accuracy of 85.91% was achieved, indicating the model's overall effectiveness in predicting sentiment from restaurant reviews. The precision for negative reviews (class 0) was 86%, with a recall of 74% and an F1-score of 80%. Conversely, for positive reviews (class 1), the precision, recall, and F1-score were notably higher at 86%, 93%, and 89%, respectively.

In specific terms, out of 1122 negative reviews, the model correctly classified 831, while 291 were misclassified as positive. For positive reviews, out of 1865 instances, the model accurately predicted 1735, with 130 misclassified as negative. These results demonstrate the model's ability to better discern positive sentiment, as evidenced by its higher recall and precision scores in classifying positive reviews. The weighted average metrics provide a comprehensive view, with an overall F1-score of 86%, reaffirming the model's reliability in sentiment analysis.

**KNN Classifier**

The K-Nearest Neighbors (KNN) classifier achieved an accuracy of 72.48%, showcasing its competency in sentiment prediction. Delving deeper into its performance metrics, we find distinct evaluations for negative (class 0) and positive (class 1) reviews.

For negative sentiments, the model displayed a precision of 67%, signifying that 67% of reviews predicted as negative were accurate. However, a lower recall of 53% implies a challenge in capturing all actual negative sentiments, with 47% being missed. This discrepancy highlights the model's struggle in comprehensively identifying negative sentiments.

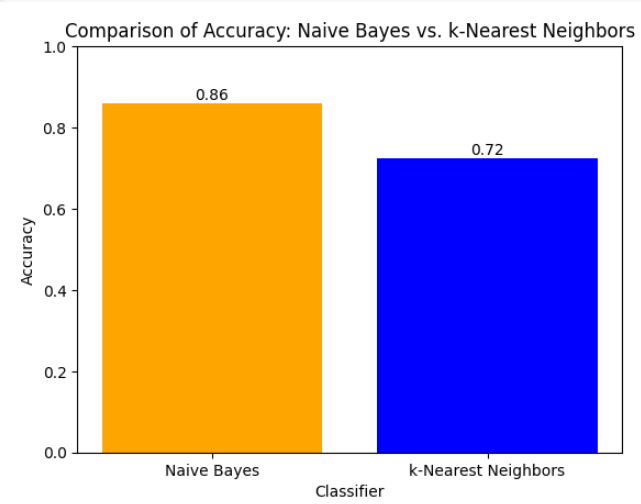


Conversely, positive sentiments demonstrated more favorable results. The model exhibited a higher precision of 75%, indicating that 75% of reviews predicted as positive were indeed positive. The recall for positive sentiments was even more robust at 84%, suggesting effective capture of 84% of actual positive sentiments. These metrics culminated in a commendable F1-score of 79%, reflecting an overall balanced performance for positive sentiments.

Examining the confusion matrix, we observe that out of 1122 negative reviews, 593 were accurately classified, while 529 were misclassified as positive. On the positive sentiment front, out of 1865 instances, the model correctly predicted 1564, but 301 were misclassified as negative. This outcome underscores the model's relative strength in identifying positive sentiments but reveals a limitation in accurately capturing negative sentiments.

**Comparison between Naive Bayes and KNN**

Our journey into sentiment analysis for restaurant reviews prompted a comprehensive examination of two prominent machine learning models: Naive Bayes and K-Nearest Neighbors (KNN). Naive Bayes emerged as a formidable contender, boasting noteworthy precision, recall, and an impressive F1-score. Its adaptability in distinguishing both positive and negative sentiments was evident, and the model secured an impressive 85.91% accuracy, positioning it as a robust choice for nuanced sentiment analysis tasks.



Conversely, KNN showcased a distinctive strength in identifying positive sentiments, unveiling a commendable precision of 75% and a robust recall of 84%. However, the model encountered challenges in achieving a harmonious performance, notably struggling in capturing negative sentiments. Despite a respectable overall accuracy of 72.48%, KNN exhibited a lower recall for negatives, thus impacting its precision-recall equilibrium.

Naive Bayes excels in sentiment analysis for restaurant reviews due to its adept handling of text data, modeling of word probability distributions, and robustness with limited datasets. The model's balanced performance across positive and negative sentiments, reflected in commendable precision, recall, and F1-score metrics, underscores its reliability. In contrast, K-Nearest Neighbors (KNN) faces challenges in achieving balance, particularly in capturing negative sentiments, influencing its precision-recall equilibrium. Naive Bayes' suitability for binary classification aligns seamlessly with our task of identifying positive and negative sentiments, making it the preferred choice for our sentiment analysis project. Overall, Naive Bayes provides a nuanced and efficient approach to discerning sentiments in restaurant reviews.

1. **CONCLUSION**

Sentiment analysis for restaurant reviews has been insightful, revealing valuable lessons and recognizing certain limitations inherent in the paper. Throughout the study, we have learned the importance of robust data preprocessing techniques, feature extraction methodologies, and the selection of appropriate machine learning algorithms for effective sentiment classification.

Our study underscores the criticality of data quality and integrity in sentiment analysis endeavors. Through cleaning and preprocessing of the dataset, we significantly elevated the accuracy and dependability of our sentiment analysis model. Moreover, employing feature extraction techniques such as Count Vectorization and TF-IDF proved instrumental in transforming textual data into a format conducive to machine learning algorithms. One significant limitation of our sentiment analysis approach lies in the intricacies of cleaning and preprocessing data, particularly reviews. The time-intensive nature of these tasks may pose challenges in handling large-scale datasets, affecting the scalability of our sentiment analysis process.

The learning includes various machine learning models, notably Naive Bayes and K-Nearest Neighbors (KNN), to discern their respective strengths and limitations. While Naive Bayes demonstrated robust performance in sentiment classification, leveraging its adept handling of textual data, KNN encountered challenges in achieving balanced performance across different sentiment categories.

1. **REFERENCES**

Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media, Inc.

JoeBeachCapital. (n.d.). Restaurant Reviews Dataset. Kaggle. https://www.kaggle.com/datasets/joebeachcapital/restaurant-reviews/data

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

McKinney, W. (2010). Data structures for statistical computing in Python. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51-56).

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1–2), 1-135.

Manning, C. D., Raghavan, P., & Schütze, H. (2008).

1. **WORKLOAD ASSIGNMENT**

Hema Deepika Thanigai Arasu: Craft a concise summary of our project's goals in the abstract.

Take care of compiling and organizing our reference section.

Kritthika Shanmugam: Dive into the real-world impact of our project, explaining its practical value in methodology. Research and pinpoint the tools we will be using for our project. Work closely with the team on coding tasks.

SathiyaShivani Sathish Kumar: Condense the key elements of our project to make them easily understandable for any reader within the literature review. Join forces with the team in coding activities.

VarunPrakash Shanmugam: Create a detailed and comprehensive project description, providing insights into the project's intricacies. Lead the charge in training and deploying our sentiment analysis model. Collaborate actively with the team on coding task