Analysing Customer Sentiment to Forecast Business Success in Fast-Food Restaurants: Unveiling Insights from Online Reviews

Tracy Akowuah Acheampong

School of Computer Science

College of Science

University of Lincoln

Submitted in partial fulfilment of the requirements for the Degree of MSc Data Science and Applied Analytics

Supervisor: Dr. Lei Zhang

August 2024

Acknowledgements

I would like to express my sincere gratitude to those who have supported and encouraged me throughout this journey.

Firstly, I wish to thank my project supervisor, Dr. Lei Zhang, for his expert guidance and insightful advice. His continuous support and constructive feedback have been invaluable to the development of this thesis.

I am also extremely thankful to my partner, for all the sacrifices he made to support me throughout my studies, for being my biggest cheerleader and constantly believing in me.

Finally, I am profoundly grateful to my mother, for her persistent encouragement and for always inspiring me to persevere. Her faith in me has been a guiding light, helping me to stay determined and focused.

To all of you, your support has been fundamental to my success.

Abstract

The fast-food industry is a cornerstone of modern dining, known for its convenience, affordability, and widespread availability. In this competitive landscape, understanding and responding to customer feedback is essential for maintaining a strong market presence. Online reviews have become a vital source of real-time insights, offering candid evaluations of dining experiences and significantly impacting a restaurant's reputation and performance.

This research utilizes sentiment analysis to evaluate customer feedback from five fast-food restaurants: Village Burger, Lucky's Burger & Brew Marietta, AZN Sandwich Bar, Cheeseburger Bobby's, and Farm Burger Dunwoody. We implemented both Quantitative and Qualitative strategies in our analysis.

For the Quantitative Analysis, we employed a range of machine and deep learning models, including CNN, RNN, SVM, Random Forest, and Naive Bayes classifiers. Despite extensive efforts in model tuning, initial accuracy rates fell below the desired 80% threshold. We addressed this by implementing SMOTE with hyperparameter tuning, which resulted in marked performance improvements. Post-tuning, the CNN and RNN models achieved accuracies of 84.4% and 81.3%, respectively, while the Random Forest and SVM models also demonstrated enhanced performance, achieving accuracies of 93% and 95%, respectively.

On the Qualitative side, NLP techniques, including CountVectorizer and TF-IDF were applied to effectively transform customer reviews into structured data insights, and then LDA and NMF for topic modeling. These revealed critical themes within the online reviews centred around food quality, customer service, and dining experience.

From our findings we could tell that food quality is the key driver of customer satisfaction, with service efficiency and the overall dining experience also playing important roles. Most negative reviews were linked to inconsistent food quality, while positive feedback often praised the quality of service and the ambiance.

This research offers a practical framework for fast-food businesses to better understand and respond to customer sentiments. The insights gained can help businesses make targeted improvements, strengthen customer loyalty, and ultimately succeed in a fiercely competitive market.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The transformative power of data analytics is central to modern business strategies, especially in consumer-driven industries like fast food. This study focuses on utilising sentiment analysis to decode customer feedback from online reviews. The application of Natural Language Processing and machine learning techniques in this research aligns perfectly with the core principles of our degree program, which emphasizes using data-driven approaches to solve real-world problems and improve decision-making processes.

1.1 Background and Motivation

The restaurant industry is a dynamic and integral part of the global economy, encompassing a wide range of establishments from fine dining to fast food (The National Restaurant Association, 2019). Among these, the fast-food sector stands out due to its emphasis on convenience, low prices, and rapid service, catering to the fast-paced lifestyle of modern society. This industry has become a fundamental part of modern dining, providing a fast and cost-effective option compared to home-cooked meals and conventional dining establishments. (Warsi & Syeedun , 2005).

With around 200,859 fast-food restaurants in the United States alone, the industry is intensely competitive (IBIS World, 2024). This immense number of establishments makes it crucial for businesses to stand out and effectively cater to their customers' needs. In this context, customer feedback becomes a pivotal factor in shaping business strategies and operational practices. And unlike traditional methods like surveys and comment cards, online reviews offer real-time, unsolicited opinions from a diverse customer base (Lasek, et al., 2016). These reviews provide valuable insights into customer experiences and perceptions, enabling businesses to make informed decisions that can enhance service quality, improve menu offerings, and ultimately drive customer satisfaction and loyalty (Kushniruk & Rutynskyi, 2022). Understanding what customers are saying is not just beneficial but essential for fast-food restaurants looking to thrive in a crowded market. These reviews are accessible on various platforms, including Yelp, Google Reviews, and social media, where customers freely share their dining experiences (Beuscart, et al., 2016). Positive reviews can significantly enhance a restaurant’s reputation, attracting new customers and fostering loyalty among existing ones. Conversely, negative reviews can quickly tarnish a restaurant's image, deter potential patrons, and highlight areas that require immediate attention (Aureliano-Silva, et al., 2021).

However, the sheer volume of online reviews can be overwhelming, making manual analysis impractical. This is where sentiment analysis, a powerful tool that leverages Natural Language Processing, comes into play.

Sentiment analysis can systematically analyze customer reviews to extract valuable insights. By categorizing reviews as positive, negative, or neutral, and identifying key themes within the reviews, sentiment analysis can provide a comprehensive understanding of customer perceptions (Medhat, et al., 2014). This technological solution is particularly beneficial for fast-food restaurants, where operational efficiency and quick adaptation to customer preferences are crucial for success.

This research focuses on five fast-food restaurants: Village Burger, Lucky's Burger & Brew Marietta, AZN Sandwich Bar, Cheeseburger Bobby's, and Farm Burger Dunwoody. These establishments, though diverse in their branding and presentation, share several commonalities in terms of their menu offerings, infrastructure, meal prices, and location. The price range for single and combo meals across these restaurants typically falls between $5 and $15, appealing to a broad customer base seeking value for money.

Village Burger is known for its classic American burgers, fries, and milkshakes, providing a traditional fast-food experience made with quality ingredients (Village Burger, 2024). Similarly, Lucky's Burger & Brew Marietta offers a menu centred around burgers, complemented by a selection of craft beers (Lucky's Burger & Brew, 2024). AZN Sandwich Bar introduces a healthier twist to the fast-food concept, offering a fusion menu that includes both traditional and vegan-inspired sandwiches and smoothies (AZN Sandwich Bar, 2024). Cheeseburger Bobby's emphasizes fresh, made-to-order burgers with a range of customizable toppings, aiming to provide a gourmet experience within the fast-food framework (Cheeseburger Bobby's, 2024). Lastly, Farm Burger Dunwoody prides itself on sourcing ingredients from local farms, offering a farm-to-table experience that focuses on sustainability and quality (Farm Burger , 2024).

Figure 1: A picture of Village Burger from Gwinnett Post Daily.

Figure 2: A picture of Lucky's Burger & Brew Marietta from TripAdvisor.

Figure 3: A picture of AZN Sandwich Bar from Broken Rice Media.

Figure 4: A Cheeseburger Bobby's from TripAdvisor.

Figure 5: A picture of Farm Burger Dunwoody from the Farm Burger website.

Despite their unique selling points, these restaurants operate within the same market segment, characterized by quick service, moderate pricing, and a casual dining atmosphere. The similarities in their menus—primarily focusing on burgers and sandwiches—reflect a common strategy to appeal to a broad customer base seeking convenience and affordability. Their infrastructure typically includes a combination of dine-in and takeout options, catering to various customer preferences and dining habits. This means there is an urgent need to stand out.

A table showing some menu items from the five fast-food restaurants is displayed below:

Restaurant Name	Menu	Price	Retrieved from
Village Burger	Combo Meal	$11.29 - $13.19	*(Village Burger, 2024)*
Lucky's Burger & Brew Marietta	Combo Meal	$9 - $13	*(Lucky's Burger & Brew, 2024)*
AZN Sandwich Bar	Combo Meal	$10 - $15	*(AZN Sandwich Bar, 2024)*
Cheeseburger Bobby's	Combo Meal	$6.59 -$ 15	*(Cheeseburger Bobby's, 2024)*
Farm Burger Dunwoody	Combo Meal	$10.99 - $12.99	*(Farm Burger , 2024)*

Table 1: Here is a table showing the price range of common menu items from each restaurant.

Sentiment analysis leverages Natural Language Processing, machine learning, and computational linguistics to interpret and classify the sentiments expressed in text data (Liu, 2012) so in this case, online reviews. The primary use of sentiment analysis is to gauge public opinion on various topics, products, or services. In the business context, sentiment analysis is widely used to monitor customer feedback, manage brand reputation, and inform marketing strategies. For instance, companies can analyze social media posts, online reviews, and survey responses to understand customer sentiments toward their products or services. This information helps businesses identify areas of improvement, tailor their offerings to meet customer needs and enhance overall customer satisfaction (Cambria, et al., 2017).

This research not only aims to apply sentiment analysis to online reviews but also to demonstrate how these insights can drive tangible improvements in business practices. By focusing on the intersection of technology and consumer feedback, this study aspires to contribute to a deeper understanding of how fast-food establishments can better align with customer expectations, enhance their competitive edge, and ultimately, sustain long-term success in a rapidly changing market landscape. The implications of this research extend beyond the five restaurants studied, offering a model that can be applied broadly across the industry to foster innovation and customer-centric growth.

1.2 Aims & Objectives

This research aims to develop a comprehensive model for conducting sentiment analysis on online reviews of five fast-food restaurants. The focus is on evaluating key business performance indicators, including overall customer satisfaction, potential for customer retention, and business viability. The restaurants in this study include Village Burger, Lucky's Burger & Brew Marietta, AZN Sandwich Bar, Cheeseburger Bobby's, and Farm Burger Dunwoody. We will analyse customer feedback to develop actionable strategies that can improve business performance in the competitive fast-food market. The research is guided by the following specific objectives:

1.2.1 Evaluate Customer Sentiment

The first objective is to assess customer sentiment as expressed in online reviews. This involves analysing perceptions of three critical aspects: food quality, service, and the restaurant setting. By using sentiment analysis techniques, the research aims to quantify these perceptions and identify patterns in customer feedback. This analysis is essential for understanding the factors that drive customer satisfaction and loyalty. As the fast-food industry continues to evolve, businesses must stay attuned to consumer expectations. This study will highlight how these expectations vary across different restaurants and how they impact customer experiences. The insights gained will help businesses understand what aspects of their offerings resonate positively with customers and what areas require improvement.

1.2.2 Predict Business Viability

The second objective focuses on predicting the business viability of the five fast-food restaurants based on the sentiment data collected. By developing predictive models, this research seeks to forecast key factors influencing customer retention and overall business success. These models will incorporate sentiment analysis data to predict potential market trends, customer preferences, and areas of concern. The predictive capability is crucial for fast-food businesses, as it enables them to anticipate changes in the market and customer behaviour. This foresight can lead to better resource allocation, improved operational efficiency, and strategic planning. For instance, if a particular aspect of service consistently receives negative feedback, businesses can proactively address these issues before they significantly impact customer retention and business viability.

1.2.3 Inform Strategic Decisions

The final objective is to integrate quantitative sentiment analysis with qualitative insights to provide comprehensive, actionable recommendations. The research will go beyond merely identifying positive or negative sentiments; it will delve into the underlying themes and topics that customers discuss in their reviews. This qualitative analysis will help uncover deeper insights into customer needs and preferences. By synthesizing these findings, the study aims to inform strategic decision-making processes within fast-food businesses. The recommendations will be tailored to each restaurant's unique challenges and opportunities, providing specific guidance on improving service offerings, enhancing customer experiences, and fostering long-term customer relationships. These strategies are vital for ensuring sustained growth and profitability in a sector characterized by intense competition and rapidly changing consumer preferences.

1.3 Thesis Structure

This dissertation is meticulously organised to present the research process in a clear and structured manner, from the initial background and rationale to the final conclusions and recommendations. It is divided into six chapters, each serving a distinct purpose in addressing the research questions and objectives.

- **Chapter 1: Introduction**

This chapter lays the foundation for this dissertation by presenting the background of the study, its significance, and the research problem. Additionally, it introduces the scope of the study and provides an overview of the structure of the dissertation.

- **Chapter 2: Literature Review**

The literature review offers a critical evaluation of existing research related to sentiment analysis, machine learning, and their real-world applications in but not limited to the fast-food industry. It identifies gaps in the literature that this research aims to address and positions the study within the broader academic context.

- **Chapter 3: Methodology**

This chapter details the research design, data collection methods, and analytical approaches employed in the study. It provides a rationale for the choice of methods and discusses how they align with the research objectives. The chapter also addresses ethical considerations and potential limitations of the methodology.

- **Chapter 4: Implementation**

The implementation chapter describes the practical application of the research methods. It includes a comprehensive discussion of the data preprocessing techniques, the development and tuning of machine learning models, and the qualitative categorisation of customer reviews. Each step of the implementation process is detailed, including the use of software tools, algorithms, and evaluation metrics.

- **Chapter 5: Results and Discussion**

This chapter presents the findings of the study, both quantitative and qualitative. The results of the sentiment analysis and thematic categorisation are discussed in relation to the research objectives. The chapter also provides a comparative analysis of the machine and deep learning models used and discusses the implications of the findings for the fast-food industry.

- **Chapter 6: Conclusion**

The conclusion summarises the key findings of the research, reflecting on how the objectives were met. It discusses the contributions of the study to the academic field and practical implications for the industry. The chapter also offers recommendations for future research and potential areas for further investigation.

- **References**

This section lists all the academic sources and literature referenced throughout the dissertation, following the Harvard referencing style.

Chapter 2

Literature Review

The literature review is a fundamental component of any academic research, serving as the foundation upon which the entire study is built. It provides a thorough exploration of existing knowledge, theories, methodologies, and findings relevant to the research topic. By critically evaluating and synthesising previous studies, the literature review establishes the context for the current research, identifying gaps in the existing body of knowledge that the study seeks to address (Webster & Watson, 2002). This section will delve into the key areas of sentiment analysis, machine learning, deep learning models, word embeddings, and preprocessing techniques employed for this study. The insights gained from the literature will not only inform the research methodology and analytical framework but also guide the development of all procedures and theories we will apply (Boell & Cecez-Kecmanovic, 2015). Through a comprehensive review of contemporary studies, this section aims to position the current research within the broader academic discourse, ensuring that it contributes meaningfully to the field of sentiment analysis.

Sentiment Analysis for Business

Sentiment analysis, often referred to as opinion mining, is a vital tool in understanding customer feedback across various industries (Liu 2012). It involves the use of Natural Language Processing to determine the sentiment expressed in textual data, whether it be positive, negative, or neutral. This process is increasingly essential for businesses as they seek to understand public opinion, improve services, and develop more effective business strategies.

Liu (2012) seminal work on sentiment analysis provides a foundational understanding of how businesses can leverage NLP to analyse customer opinions. It discusses the theoretical underpinnings of sentiment analysis, emphasising its significance in understanding consumer behaviour and informing business decisions. Liu's work also explores various methodologies for sentiment classification, such as lexicon-based approaches and machine learning techniques, highlighting the strengths and limitations of each. It is noted that while lexicon-based methods are straightforward and easy to implement, they often lack the sophistication to handle the nuances of human language, such as sarcasm or idiomatic expressions. Machine learning methods, on the other hand, offer more flexibility and accuracy but require large annotated datasets for training, which can be resource-intensive to obtain. Liu's discussion on the balance between simplicity and accuracy in sentiment analysis is applied in our study. The choice to employ machine learning models, such as SVM and Random Forest, reflects an effort to harness the accuracy and flexibility that Liu champions, while our use of hyperparameter tuning addresses the challenges associated with fine-tuning these models to improve performance.

Farhadloo and Rolland (2016) also emphasise the importance of sentiment analysis as a tool for organizations to gauge public opinion and customer satisfaction through the analysis of unstructured data, such as reviews. They outline that sentiment analysis can be conducted at various levels of granularity, including document, sentence, and aspect levels, which are all relevant to our project's focus on extracting actionable insights from customer reviews. The authors also explore the challenges associated with sentiment analysis, such as dealing with synonymy, polysemy, and the detection of sarcasm, which are crucial considerations when processing natural language data.

This is essential to our project as we are interested in the dynamic nature of customer opinions and the necessity of handling large volumes of data efficiently. Their discussion on the use of probabilistic models like Latent Dirichlet Allocation for discovering themes and sentiments within unstructured text data resonates with our approach, where we aim to identify and analyze key aspects that influence customer satisfaction in the fast-food industry.

Similarly, Doan and Kalita's (2022) work on sentiment analysis aligns closely with the goals of this project, particularly in terms of understanding and responding to customer feedback in real-time.

In our project, we focus on the sentiment analysis of customer reviews, aiming to provide actionable insights that can help improve customer satisfaction. Just as Doan and Kalita (2022) emphasise the limitations of traditional, static sentiment analysis models, our work addresses these challenges by incorporating more adaptive techniques. While their study suggests the use of incremental learning to continuously update sentiment analysis models, our approach involves optimizing machine learning models through hyperparameter tuning and using SMOTE to balance the dataset, ensuring that the models remain effective over time.

Furthermore, Doan and Kalita (2022) highlight the significant impact that negative sentiment can have on a business, which is a crucial consideration in our project as well. By analysing sentiment from customer reviews, we aim to identify areas where fast-food restaurants can improve, particularly in aspects such as food quality, service, and ambiance. This proactive approach allows businesses to address issues before they result in significant customer dissatisfaction.

Preprocessing Techniques in NLP

Preprocessing is a critical step in any NLP task, ensuring that the text data is clean and ready for analysis. Jones (2018) highlighted the importance of preprocessing steps such as tokenization, normalization, and lemmatization, which help reduce noise in the data and improve model performance. These steps were essential in preparing the fast-food reviews for sentiment analysis.

Tokenization, which involves splitting text into individual words or tokens, is a fundamental step in NLP. Sennrich et al. (2016) discussed how effective tokenization is crucial for converting text into numerical features that can be used by machine learning models. This process was applied in this study to ensure that the text data was in the right format for further analysis.

Normalization, including lowercasing and the removal of special characters, was another important step in our preprocessing pipeline. Uysal and Gunal (2014) emphasized that normalization is necessary to ensure consistency in text representation, which is particularly important in sentiment analysis where even small variations can affect model predictions. In our study, normalization helped standardise the text data, making the sentiment analysis process more reliable.

Lemmatization, which reduces words to their base forms, further refined the text data. Siddhartha (2021) discussed the benefits of lemmatization in reducing the dimensionality of text data while retaining its meaning. By implementing lemmatization, we were able to ensure that different forms of a word were treated as a single entity, simplifying the data and improving the accuracy of sentiment classification.

Word Embeddings

Word embeddings play a critical role in Natural Language Processing by providing a way to represent words as dense vectors that capture their meanings in context. Mikolov et al. (2013) introduced Word2Vec, a technique that has since become a staple in NLP tasks, including sentiment analysis. Word2Vec transforms words into vectors based on their context in large text corpora, allowing models to grasp not just the words themselves but their relationships with other words (Suri, 2022).

Pennington et al. (2014) later developed GloVe, which improved on Word2Vec by incorporating global word co-occurrence statistics into the embedding process. GloVe has proven effective in various NLP tasks, including sentiment analysis, by capturing both the local and global relationships between words (Hui, 2019). The choice of embedding model can significantly impact the performance of sentiment analysis models, as these embeddings form the foundation upon which the models are built.

In this study, we utilised SpaCy's en_core_web_md embeddings, which strikes a balance between capturing the necessary context and being computationally efficient. While more advanced embeddings like BERT (Ruder et al., 2019) offer dynamic word representations that adjust based on the context, they are also more resource-intensive. The choice of SpaCy's embeddings was reinforced by Shen (2024), who explored SpaCy, particularly its effectiveness in text preprocessing tasks such as tokenization, lemmatization, and stop word removal. The study utilised a dataset of 50,000 movie reviews from IMDB to evaluate the performance of different machine learning algorithms, including Logistic Regression, Decision Trees, and Multilayer Perceptron. It stresses that the choice of preprocessing tools, like spaCy, significantly impacts the quality of the sentiment analysis, as it helps in reducing noise and standardising the text for better model performance. Shen's findings demonstrate its efficiency in handling large text corpora and improving the accuracy of sentiment analysis models. Specifically, SpaCy's robust lemmatization and tokenization capabilities make it a suitable choice for preprocessing in sentiment analysis tasks, leading to more accurate and reliable results.

Machine Learning Models for Sentiment Analysis

Machine learning models have long been instrumental in the field of sentiment analysis. Samal and Panda (2017) provides an insightful evaluation of various supervised machine learning models for sentiment analysis, particularly focusing on movie reviews. The authors explore several popular supervised learning algorithms, including Naive Bayes (Multinomial and Bernoulli variants) and Support Vector Machines (SVM), comparing their effectiveness in classifying sentiments from large datasets.

The paper's analysis reveals that Linear SVM outperforms other models with an impressive accuracy of 100% on larger datasets, making it the most suitable for sentiment analysis in this study. The study by Huda et al. (2019) equally provides a comprehensive examination of various machine learning models applied to sentiment analysis of restaurant reviews. The authors implement and compare several classification algorithms, including Naïve Bayes and Support Vector Machines, to determine which model most effectively predicts sentiment in a dataset of restaurant reviews from Bangladesh. In this study, SVM, combined with the TF-IDF vectorizer, outperforms other models, achieving the highest accuracy at 95%. These finding align with our project, where we also utilise SVM for its robustness in handling high-dimensional feature spaces, which is common in text data. The study's emphasis on feature extraction techniques such as TF-IDF underscores the importance of vectorization methods, which are crucial in our project for transforming text data into a format suitable for machine learning models.

Moreover, the authors suggest that more advanced models, like RNN, could improve sentiment analysis performance in future work. This suggestion influenced our project's direction, where exploring deep learning approaches such as RNN provided significant improvements in handling context and sequential dependencies in customer reviews.

Sarker et al. (2019) pointed out that Random Forests are particularly effective in managing imbalanced datasets, which is a common issue in sentiment analysis where certain sentiments may be underrepresented. Their findings resonate with the approach taken for this study, where Random Forests, enhanced by the use of Synthetic Minority Over-sampling Technique (SMOTE), help mitigate class imbalance and improve model performance in analysing fast-food reviews.

Yadav and Vishwakarma (2021) compared the performance of different machine learning models and concluded that model efficacy is often dependent on the dataset's characteristics. This notion guided the decision to experiment with various models, including SVM, Random Forest, and Naive Bayes, to determine the best fit for the sentiment analysis task at hand. The choice of models in this study reflects this understanding, ensuring that the selected model is the most appropriate for the dataset's nuances.

Deep Learning Models for Sentiment Analysis

Deep learning has brought significant advancements to sentiment analysis, particularly through models like Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The study by Çano and Morisio (2018) introduces NgramCNN, a deep learning architecture tailored for sentiment analysis, particularly focusing on long text documents. This model leverages pre-trained word embeddings and employs a CNN to capture essential features through convolution and pooling layers. The research highlights the CNN's strength in handling both short and long texts, outperforming traditional linear models and Recurrent Neural Networks (RNNs) across various sentiment analysis tasks.

The NgramCNN's effective use of pre-trained word embeddings and its emphasis on capturing critical features through convolutional layers directly informs our approach. The study reinforces the advantages of CNNs, particularly in processing complex and lengthy customer reviews, and provides valuable insights into optimizing CNN architectures to enhance sentiment prediction accuracy.

In the paper "Deep learning-based sentiment analysis for social media: A focus on multimodal and aspect-based approaches", Feng (2023) discusses several deep learning models relevant to sentiment analysis, many of which align with the methods used in our study. Notably, the paper explores the application of Convolutional Neural Networks and their extensions, such as TextCNN and Deep Pyramid Convolutional Neural Networks (DPCNNs), which are foundational in text classification tasks. These models are particularly effective for processing and extracting features from text data, which is a key aspect of our approach.

The discussion on TextCNN, introduced by Kim (2014), highlights the model's effectiveness in capturing sentiment from text data, a technique we also employ in our project. Furthermore, the paper mentions the use of the DPCNN model, which enhances text classification by incorporating deeper convolutional layers, allowing for better processing of longer texts—a challenge we address in our project as well.

RNNs, and more specifically Long Short-Term Memory (LSTM) networks, have shown exceptional performance in processing sequential data (Zhou et al., 2015). This ability to model the sequence of words and understand long-term dependencies is particularly beneficial when analysing customer reviews, where the order of words can significantly alter the sentiment conveyed. Tang et al. (2015) further explored the application of RNNs in sentiment analysis, emphasising their strength in understanding context over time however, as noted by Young et al. (2018), deep learning models are not without their challenges—they require extensive tuning and are computationally demanding. In this study, we addressed these challenges by implementing hyperparameter tuning for CNN and RNN models, ensuring that they are optimally configured for sentiment analysis in the fast-food sector. This tuning process was crucial to enhancing the models' ability to accurately classify sentiments within the reviews.

SMOTE and Hyperparameter Tuning

Class imbalance is a common challenge in sentiment analysis, where one sentiment class may dominate the dataset. Chawla et al. (2002) introduced SMOTE, a technique that generates synthetic samples for the minority class to address this imbalance. SMOTE has been widely adopted in sentiment analysis to ensure that models are trained on a balanced dataset, leading to more accurate predictions.

Bergstra and Bengio (2012) also highlight the importance of hyperparameter tuning in optimizing model performance. Techniques like GridSearchCV systematically search through hyperparameter spaces to find the best configuration for a given model. This process is crucial in sentiment analysis, where the right combination of hyperparameters can significantly enhance model accuracy.

Buda et al. (2018) explored a different approach by combing both SMOTE and hyperparameter tuning to improve model performance on imbalanced datasets. Their findings support the approach taken in our study, where SMOTE was used to balance the dataset, and hyperparameter tuning was applied to fine-tune the machine learning models. This combination helped ensure that the models were not only accurate but also generalised well to new data.

The literature reviewed underscores the complexity and importance of sentiment analysis in understanding customer feedback, particularly within the fast-food industry. The integration of machine learning models with traditional lexicon-based approaches has emerged as a powerful method to navigate the intricacies of customer sentiments, addressing challenges such as the subjective nature of language and the context-dependent interpretation of reviews. Additionally, the role of hyperparameter tuning has been highlighted as critical in refining these models to enhance their accuracy and robustness.

By synthesising insights from recent studies, this project builds upon established methodologies, applying advanced machine learning techniques to effectively categorise sentiments expressed in online reviews. This comprehensive approach not only aligns with the challenges identified in existing research but also pushes the boundaries of sentiment analysis by incorporating state-of-the-art techniques such as deep learning models and word embeddings.

Chapter 3

Methodology

This research employs an experimental approach to explore and evaluate various machine and deep learning models for sentiment analysis. The study focuses on three key dimensions: the ambiance of the restaurant, the quality of the food, and the level of service provided. These elements are crucial for assessing business performance indicators such as overall customer satisfaction, the likelihood of customer retention, and the long-term viability of the business.

To achieve a comprehensive understanding, the study adopts both qualitative and quantitative methodologies. On the qualitative side, the analysis seeks to uncover underlying themes within the customer reviews, providing deeper insights into customer sentiments. The quantitative aspect, on the other hand, involves the use of advanced machine learning models to categorize and quantify the sentiment expressed in these reviews, offering a data-driven perspective on customer feedback.

3.1 Data Collection

The dataset used in this study was sourced from Kaggle.com, a well-known platform for data science and machine learning datasets. It initially comprised 5,000 rows of online reviews from five fast-food restaurants — Village Burger, Lucky's Burger & Brew Marietta, AZN Sandwich Bar, Cheeseburger Bobby's, and Farm Burger Dunwoody — and five columns, each representing a different aspect of the reviews:

- **Restaurant Name**: The name of the fast-food restaurant where the review was made.

- **Restaurant Type**: The type of establishment, which in this case was uniformly "Fast Food."
- **Star Rating**: The numerical rating given by the customer, ranging from 1 to 5.
- **Sentiment**: The sentiment label derived from the review text, categorised as "Negative," "Neutral," or "Positive".
- **Reviews**: The textual content of the customer's review.

3.2 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is the initial step in understanding the dataset and identifying any potential issues before proceeding with more complex analyses. This involves checking for missing or null values, generating a statistical summary of the data columns, and visualization (Tukey, 1977).

3.3 Data Cleaning and Preprocessing

This step is crucial to ensure the dataset is suitable for analysis so several preprocessing techniques will be applied. These will include:

1. **Handling Missing Data**: Any missing data we found during EDA, will be removed as they can introduce bias and reduce the validity of statistical analysis (Little & Rubin, 2019).
2. **Removing Duplicates**: Duplication in the dataset can skew the results, particularly in sentiment analysis, where repeated sentiments can falsely inflate the prevalence of certain opinions. Removing duplicates ensures that each review represents a unique customer perspective, leading to more accurate sentiment analysis (Aggarwal & Zhai, 2012).
3. **Text Normalization**: Text normalization is the process of converting text into a consistent format, which helps in reducing variability in the data that does not contribute to the semantic meaning. Common steps in text normalization include converting text to lowercase, removing punctuation, and eliminating

stop words (words that are common but carry little meaningful content, such as "and," "the," etc.) (Bird, et al., 2009).

4. **Tokenization**: Tokenization is the process of splitting text into individual units called tokens, which can be words, phrases, or subwords (Webster & Kit, 1992). Tokenization is essential for breaking down the text into manageable parts which allows it to be analysed at the word level, enabling the model to understand the context and relationships between the words in a sentence (Pai, 2024).
5. **Lemmatization**: Lemmatization is another critical preprocessing step that involves reducing words to their base or root form, known as the lemma (Siddhartha 2021). This process is vital in sentiment analysis because it ensures that different forms of a word (e.g., "running," "ran," "runs") are treated as a single entity ("run"). This reduction in word forms helps in lowering the dimensionality of the text data and enhances the model's ability to generalize by focusing on the semantic content rather than surface-level variations (Saumya, 2024)

3.4 Sentiment Analysis and Categorisation

Sentiment analysis will then be performed using a combination of traditional Natural Language Processing techniques and Machine Learning models. This process involves converting the text data into numerical representations using word embeddings before classifying with machine learning algorithms.

3.4.1 Word Embeddings

The first step in the sentiment analysis process is converting the raw text data into a format suitable for machine learning models. This is achieved through the use of word embeddings. They represent words as dense vectors in a continuous vector space, capturing their semantic relationships (NSS, 2023).

Word embeddings were chosen because they provide a more sophisticated representation of text data compared to traditional methods like Bag of Words (BoW) or Term Frequency-Inverse Document Frequency (TF-IDF). Unlike these methods, which rely on word frequency and are often sparse, word embeddings capture the semantic meaning and context of words, allowing models to better understand and differentiate between different uses of the same word (Mikolov, et al., 2013). The use of pre-trained embeddings also allows the models to leverage knowledge from large, diverse datasets, improving their ability to generalise to new data.

3.4.2 Model Selection and Training

With the text data converted into numerical embeddings, the next step is to select appropriate machine learning models for sentiment classification. Five models have been chosen for this task: Support Vector Machine (SVM), Random Forest, Naïve Bayes, Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN). These models were selected based on their established effectiveness in text classification tasks, particularly in handling high-dimensional data like word embeddings (Scikit-learn Documentation, 2024).

Support Vector Machine

Support Vector Machines are a set of supervised learning methods used for classification and regression (Saini, 2024). The primary goal of SVM is to find the optimal hyperplane that separates different classes in the feature space (Scikit-learn Documentation, 2024). They are particularly well-suited for high-dimensional spaces and are effective in cases where the number of dimensions exceeds the number of samples (Scikit-learn Documentation, 2024). There are two types of SVM, namely Support Vector Classification (SVC) for classification tasks and Support Vector Regression (SVR) for regression tasks (Scikit-learn Documentation, 2024). In this study, we will be implementing SVC to classify customer reviews based on sentiment.

Support Vector Classification (SVC) is a specific implementation of SVM tailored for classification tasks (Scikit-learn Documentation, 2024). It focuses on finding the best separating hyperplane for binary or multi-class classification (Huda, et al. 2019) SVC is particularly powerful in situations where the data is not linearly separable, as it can use kernel functions to study the data into a higher-dimensional space where a linear separation is possible (Scikit-learn Documentation, 2024). It solves the following primal problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$\text{subject to} \quad y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \quad \zeta_i \geq 0, i = 1, \ldots, n \tag{1}$$

The objective is to minimise the norm of the weight vector w, which corresponds to maximising the margin between the classes. The slack variables ζ_i allow for some degree of misclassification, which is controlled by the regularisation parameter C. The kernel function $\phi(x_i)$ is used to map the input data into a higher-dimensional space, enabling linear separation in cases where the data is not linearly separable in its original space (Scikit-learn Documentation, 2024).

For this study, selecting an appropriate kernel function for the Support Vector Classification model was crucial. The kernel function plays a key role in transforming the input data into a higher-dimensional space where it becomes easier to find a separating hyperplane between different sentiment classes (e.g., positive, neutral, negative) (Sasidharan,2024)

Several kernel functions were considered for this task:

- **Linear Kernel**: The linear kernel assumes a linear relationship between the features, which can be effective when the sentiment data is linearly separable in the original feature space. However, given the complexity and the nuanced nature of language in customer reviews, the linear kernel might not capture the subtleties of sentiment expression, such as sarcasm or nuanced opinions (Müller, et al., 2001).

- **Polynomial Kernel**: This kernel maps the original features into a polynomial feature space, allowing the model to capture more complex relationships between words in the reviews. While the polynomial kernel can handle non-linear separations, it may introduce higher complexity and computational cost, which can be a concern given the size of the dataset (Shawe-Taylor & Cristianini, 2004).
- **Radial Basis Function (RBF) Kernel**: The RBF kernel was eventually chosen as it is particularly suited for this sentiment analysis task due to its ability to handle non-linear classifications by mapping the data into an infinite-dimensional space (Cao , et al., 2008). This is ideal for text data, where relationships between words and phrases are often non-linear. The RBF kernel can effectively capture these complex relationships, making it a strong candidate for accurately classifying sentiments in customer reviews (Hsu, et al., 2016).

However, the effectiveness of the kernel is highly dependent on the correct tuning of key hyperparameters (Padierna, et al., 2016). Hyperparameter tuning is a vital process that enhances the performance of the SVC model by optimizing key parameters. In this study, two critical hyperparameters were carefully tuned:

- **C Parameter**: The C parameter was adjusted to balance the trade-off between correctly classifying as many sentiment labels as possible (i.e., positive, neutral, negative) while maintaining a strong decision boundary that generalizes well to new, unseen data. A lower C value might allow some misclassifications in the training data but would result in a broader margin and potentially better generalization (Cunha, et al., 2022).Conversely, a higher C value would aim for higher accuracy in the training data, potentially at the risk of overfitting (Ying, 2019). Given the varied nature of customer sentiments, tuning the C parameter was crucial to ensuring that the model effectively captures both strong and subtle sentiment cues (Cortes & Vapnik, 1995).
- **Gamma Parameter**: The Gamma parameter in the RBF kernel controls how far the influence of a single training example reaches (Ben-Hur and Weston, 2010) For the sentiment analysis task, tuning Gamma allowed the model to

balance between a smooth decision boundary and a more complex boundary that can capture subtle shifts in sentiment (Hsu et al., 2003). A lower Gamma value was tested to allow each training example to have a broader influence, which is beneficial in cases where sentiment expressions are more general and dispersed. On the other hand, a higher Gamma value was tested to allow the model to focus more closely on specific sentiment expressions, which is useful for capturing strong sentiment signals (Bishop, 2006).

Mathematically, the RBF kernel is defined as:

$$k(x, x') = exp\left(-\gamma \left\|x - x'\right\|^2\right) \qquad (2)$$

In the RBF kernel, x and x' represent two data points in the feature space. The term $\|x - x'\|$ refers to the Euclidean distance between these two points, which measures how far apart they are in the feature space. The parameter γ plays a critical role in defining the extent of influence that a single training example has on the decision boundary (Schölkopf & Smola, 2002).

A large γ value results in a narrow influence range, meaning that each data point only affects its immediate surroundings. This can lead to a very complex decision boundary that closely fits the training data, which may capture the specific nuances of the data but also risks overfitting(Ying, 2019). Overfitting occurs when the model becomes too tailored to the training data, losing its ability to generalise well to new, unseen data (Goodfellow, Bengio, & Courville, 2016).

On the other hand, a small γ value results in a broader influence range, where each data point affects a wider area of the feature space. This leads to a smoother decision boundary that is less sensitive to the specific details of individual data points, potentially improving the model's ability to generalise. By avoiding overfitting, a small γ value can help the model perform better on new data by not overly conforming to the training set's peculiarities (Hastie, Tibshirani, & Friedman, 2009).

To determine the optimal values for the hyperparameters C and γ in the SVC model, a method known as grid search with cross-validation was employed. This process involves two key components: grid search and cross-validation (Jain, 2024).

Grid search is an exhaustive search method used to find the optimal hyperparameters for a given model. It systematically evaluates a predefined set of hyperparameter values by training the model on the training data using each combination of parameters (Bergstra & Bengio, 2012). For instance, in the case of SVC, the grid search would involve testing various combinations of C and γ values. The grid search optimises the objective function of the SVC model formula (Tyagi, 2023).

Within this framework, the grid search evaluates the model's performance over a grid of hyperparameter values:

$$C \in \{C_1, C_2, \ldots, C_m\}, \quad \gamma \in \{\gamma_1, \gamma_2 \ldots \gamma_n\} \qquad (3)$$

For each combination (C_i, γ_j) the SVC model is trained and evaluated, allowing the selection of the combination that yields the best performance based on the chosen evaluation metric (Kuhn and Johnson, 2013).

Random Forest

Random Forest is an ensemble learning method primarily used for classification and regression tasks (Sruthi, 2024). It operates by constructing a multitude of decision trees during training and outputting the mode of the classes (classification) or mean prediction (regression) of the individual trees. The core idea behind Random Forest is to improve the predictive accuracy and control overfitting by combining the predictions of multiple decision trees, each trained on different subsets of the data (Breiman, 2001).

Random Forest is built upon the concept of bagging (Bootstrap Aggregating), which involves generating multiple datasets by sampling the original data with replacement (Breiman, 1996). Each of these datasets is then used to train a separate decision tree. During prediction, each tree in the forest gives a classification, and the class with the most votes becomes the model's prediction data. This process makes Random Forest resilient to overfitting, as the model relies on multiple trees rather than a single tree, thus smoothing out predictions (Liaw & Wiener, 2002).

Since this is a classification task, we used the Gini index which is commonly used to guide the branching of nodes in decision trees.

$$Gini = 1 = \sum_{i=1}^{c} (p_i)^2 \quad (4)$$

It evaluates each branch of a node by considering the class and its associated probability, helping to identify which branch is more likely to occur. In this context, p_i denotes the relative frequency of the observed class in the dataset, c while represents the total number of classes (Schott, 2019).

Random Forest is particularly well-suited for this study due to its capacity to handle high-dimensional data and its inherent ability to withstand overfitting—critical factors when working with text data similar to our online reviews. For our analysis, the features derived from word embeddings result in a high-dimensional feature space therefore, the ability of Random Forest to incorporate random feature selection at each split in the decision tree proves advantageous in managing this complexity (Breiman, 2001).

Moreover, Random Forest's capacity to provide feature importance scores enhances the interpretability of the model's decisions. This allows for a deeper understanding of which words or phrases in the reviews most strongly influence the sentiment classification. This interpretability is particularly valuable in this study, where understanding the underlying factors driving customer sentiment is as essential as achieving accurate sentiment predictions (Cutler, et al., 2012).

In Random Forest models, several key parameters influence the behaviour and performance of each decision tree. Understanding these parameters is essential for tuning the model to achieve optimal results (Srivastava, 2024). Below is an explanation of the key parameters and how they were used in the context of this research:

- **n_estimators**: This parameter defines the number of trees in the forest. A higher number of trees generally improves the model's accuracy but comes at the cost of increased computational resources (Data Camp, 2023).
- **max_depth**: This parameter specifies the maximum depth of the trees. By controlling the depth, we can limit the complexity of the model, helping to prevent overfitting (Data Camp, 2023).
- **min_samples_split**: This parameter controls the minimum number of samples required to split an internal node. By setting this parameter, we ensure that the trees do not become overly complex by splitting on nodes with very few samples (Fraj, 2017).

Understanding the key parameters that influence the behaviour of each decision tree within the Random Forest is crucial for maximizing the model's accuracy. However, to truly unlock its potential, it's essential to fine-tune these parameters through hyperparameter tuning, ensuring that the model is both accurate and generalizes well to new data (Svetnik, et al., 2003). Hyperparameter tuning for the Random Forest model will be conducted using Grid Search with cross-validation. This involves searching over a grid of possible values for the parameters; n_estimators, max_depth, and min_samples_split (Jain, 2024). Cross-validation ensures that the model is evaluated on multiple splits of the data, thus providing a more reliable estimate of its performance. The combination of parameters that yielded the best cross-validation score was then selected for the final model (Seong, 2024).

The cross-validation process seeks to minimize the following loss function:

$$CVerror = \frac{1}{K}\sum_{k=1}^{K} \quad error_k \tag{5}$$

K represents the number of folds used in cross-validation and $error_k$ is the error on the k-th fold.

This approach ensures that the Random Forest model is optimally configured for the sentiment classification task, balancing the trade-off between bias and variance (Pandian, 2024)

Naive Bayes Classifier

Naive Bayes is a probabilistic classifier that is particularly effective for text classification tasks, making it a strong candidate for sentiment analysis in this study (Manning, et al., 2008). The core of the Naive Bayes classifier is based on Bayes' theorem, which provides a way to calculate the posterior probability of a class given the features of the input data (Bahtiar, et al., 2023).Despite the "naive" assumption that features are independent of each other, which rarely holds in real-world scenarios, Naive Bayes has shown remarkable performance in various applications, especially in text classification (Joachims, 1998).

Naive Bayes is well-suited due to its ability to efficiently handle the high-dimensional data typically generated by text features such as word frequencies or TF-IDF scores (Robertson, 2004). The model operates under the assumption that the presence of a particular word in a review is independent of the presence of any other word, given the sentiment class (Rennie, et al., 2003). This simplification allows the Naive Bayes classifier to be computationally efficient and quick to train, which is beneficial when working with large datasets of customer reviews (McCallum & Nigam, 1998).

It is grounded in Bayes' theorem, which can be expressed as (Glenn, 2023):

$$P(\, A|B) = P(B|A) * P(A)/P(B) \tag{6}$$

Where:

- P(A\B) represents the probability of event A occurring given that event B has occurred (posterior probability).
- $P(B|A)$ denotes the probability of event B occurring given that event A has occurred (likelihood).
- $P(A)$ is the overall probability of event A happening (prior probability).

- $P(B)$ is the total probability of event B occurring (evidence).

In this study, the Multinomial Naive Bayes variant is used, which is particularly appropriate for discrete features like word counts (Jurafsky & Martin, 2020). The model calculates the likelihood of a given review belonging to a particular sentiment class (e.g., positive, neutral, or negative) based on the frequency of words in the review (Chen & Goodman, 1996).

Furthermore, Naive Bayes offers an interpretable model where the contribution of each word to the final classification can be understood in probabilistic terms. This transparency is valuable in this study, as it allows for analysis not only of overall sentiment but also of the specific words and phrases that most strongly influence customer sentiment (Yang & Liu, 1999).

While Naive Bayes assumes feature independence, which is a simplification, this assumption does not significantly detract from the model's performance in text classification tasks like the one undertaken in this study (Zhang, 2004). Its efficiency, combined with its effectiveness in high-dimensional spaces, makes it a compelling choice for sentiment analysis in the context of restaurant reviews.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks have become a staple in deep learning due to their efficacy in processing grid-like data structures, such as images and text sequences (LeCun, et al., 2015). For our study, CNNs will be employed to analyse our customer reviews, as they are adept at identifying patterns, such as phrases or specific words, that indicate sentiment. The application of CNNs allows for the extraction of local features within text data, which is crucial for understanding the nuances in customer feedback (Zhang, et al., 2015).

CNNs operate by applying convolutional filters to the input data, which, in this case, are the word embeddings representing customer reviews. The primary components of a CNN include:

- **Convolutional Layer**: This layer applies a set of convolutional filters to the input data to extract features (Sumbati, 2024).

- **Pooling Layer**: After the convolution, a pooling layer, often MaxPooling, is used to reduce the spatial dimensions of the feature maps while retaining the most important features (Sumbati, 2024).
- **Flattening and Dense Layers**: Once the features have been extracted and pooled, they are flattened into a single vector, which is then passed through fully connected (dense) layers. The final layer typically uses a softmax activation function to output a probability distribution over the sentiment classes (Goyal, 2024).

Grid search and cross-validation will be used to find the optimal configuration, which was crucial for improving the model's accuracy in classifying customer sentiments (Bergstra & Bengio, 2012).

Recurrent Neural Networks (RNN)

Recurrent Neural Networks are particularly well-suited for sequence-based data, such as text, where the order of words is important for understanding context. In this study, RNNs were used to capture the temporal dependencies within the customer reviews, which is essential for accurately predicting sentiment based on the sequential nature of the text (Hochreiter & Schmidhuber, 1997).

RNNs function by maintaining a hidden state that is updated at each time step based on the current input and the previous hidden state. This allows the network to retain information about previous words in the sequence, making it effective for tasks where context is crucial (Pascanu, Mikolov, & Bengio, 2013).

The RNN's operation can be described by the following equations (Nabi, 2019):

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b_h) \quad (7)$$

Where:

- h_t is the hidden state at time step t.
- W_h and W_x are weight matrices.
- x_t is the input at time step t.

- b_h is the bias vector.
- σ is the activation function, such as `$tanh$`

Hyperparameter tuning for the RNN was also carried out using $keras\_tuner$, focusing on parameters such as:

- Number of Units in the RNN Layer: The size of the hidden state, varied between 50 and 200.
- Learning Rate: The learning rate was tested in the range of 0.001 to 0.01.
- Dropout Rate: Regularization parameter to prevent overfitting, typically tested between 0.2 and 0.5.

This tuning process ensured that the RNN was well-optimized for handling the sentiment analysis task, particularly given the sequential nature of the text data (Hochreiter and Schmidhuber, 1997).

3.4.3 Qualitative Analysis

The qualitative analysis in this study was focused on uncovering the underlying themes within the customer reviews of the five fast-food restaurants. This aspect of the research was critical for gaining deeper insights into customer sentiments beyond what quantitative measures could capture. The qualitative approach allowed us to explore the nuances of customer feedback, including their opinions on the restaurant's ambiance, food quality, and service experience. We applied a combination of text preprocessing and topic modeling techniques to uncover underlying themes in the data

Text Preprocessing

The first step involved cleaning and preparing the textual data for analysis. We used the Spacy library to remove stop words, punctuation, and other irrelevant information (Honnibal and Montani, 2017). Tokenization was performed to break down the text into individual words or tokens (Webster and Kit, 1992). We also applied lemmatization to reduce words to their base forms, making the text more uniform and easier to analyze.

Vectorization

Vectorization is the process of converting textual data into numerical data that can be used by machine learning models (Kozhevnikov and Pankratova, 2020). For this study, we utilized the CountVectorizer from the Scikit-learn library. The CountVectorizer works by transforming the text into a matrix of token counts. This approach essentially builds a vocabulary from the text data and then constructs a sparse matrix where each row corresponds to a document (e.g., a review), and each column corresponds to a token (e.g., a word) from the vocabulary (Verma, 2022). It works by employing the following procedures (Sharma, 2020):

1. **Tokenization**: The text is first split into tokens (usually words). For example, the sentence "I love burgers" would be tokenized into ["I", "love", "burgers"].
2. **Vocabulary Creation**: The vectorizer then creates a vocabulary, which is a set of all unique tokens (words) present in the entire text corpus. For example, if our corpus is ["I love burgers", "I hate fries"], the vocabulary would be ["I", "love", "burgers", "hate", "fries"].
3. **Count Matrix Creation**: The text is then transformed into a count matrix. Each document is represented as a vector where each element represents the count of a token from the vocabulary in that document.

In this study, $CountVectorizer$ was used to convert the processed text reviews into this count matrix, which was then fed into subsequent models like Latent Dirichlet Allocation (LDA) for topic modeling. This transformation was crucial as it allowed the models to analyze the frequency and distribution of words across the reviews, providing a basis for identifying key topics and patterns in customer feedback (Kozhevnikov and Pankratova, 2020).

Topic Modeling

Topic modeling is a crucial technique used in natural language processing to uncover hidden structures or themes within a corpus of documents (Peddireddi, 2023). In this study, two different topic modeling techniques were employed: Latent Dirichlet Allocation (LDA) and Non-Negative Matrix Factorization (NMF).

1) **Latent Dirichlet Allocation**: LDA is a probabilistic model that identifies underlying topics within a set of documents. The model assumes that each document is a mixture of several topics, and each topic is a mixture of words. The goal of LDA is to reverse-engineer this process to discover the original topics and word distributions. This is how it works (Blei, et al., 2003):
   a) **Topic Distribution**: LDA assumes that documents are generated by first choosing a distribution over topics. For each document d, a topic distribution $\theta_d$is drawn from a Dirichlet distribution Dir(α), where α is the parameter of the Dirichlet prior on the per-document topic distributions.
   b) **Word Distribution**: Then, for each topic k, a word distribution ϕk is drawn from another Dirichlet distribution Dir(β), where β is the parameter of the Dirichlet prior on the per-topic word distributions.
   c) **Document Generation**: For each word in the document, a topic z_n is chosen from the topic distribution θ_d, a word w_n is then generated from the word distribution corresponding $\phi_{z_n}$to the chosen topic.

The LDA model aims to estimate the distributions that most likely generated the observed documents.

In our study, LDA was used to identify the underlying topics within the fast-food restaurant reviews. We set the number of topics to five, based on an initial analysis of the data, which allowed the model to extract the most prominent words for each topic, thereby helping us to categorize the reviews into distinct themes.

2) **Non-Negative Matrix Factorization**: To complement the findings from LDA, we also applied Non-Negative Matrix Factorization (NMF). Unlike LDA, which is a probabilistic model, NMF is a linear algebra-based technique that factorizes the document-term matrix into two lower-dimensional matrices: one

representing topics and the other representing the topic distribution in the documents (Lee and Seung, 1999).

Generally, this is how NMF works (Goyal, 2021):

a) **Matrix Factorization**: Given a document-term matrix X , where X_{ij} represents the frequency of term j in document i, NMF factorizes X into two non-negative matrices W and H, such that:

$$X \approx W * H \tag{8}$$

Here:

W is an $m * k$ matrix, where m is the number of documents and k is the number of topics. W represents the topic distribution across documents. H is a $k * n$ matrix, where n is the number of terms in the vocabulary. H represents the word distribution across topics.

b) **Objective Function**: The factorization is achieved by minimizing the difference between X and $W * H,$ typically measured by the Frobenius norm:

$$\underset{W,H}{min} \|X - W\|^2 F \tag{9}$$

subject to $W \geq 0 \; and \; H \geq 0$

NMF was employed alongside LDA to ensure that we captured all significant topics within the dataset. NMF's linear algebra approach provided a different perspective on the data, helping to identify topics that might not have been as prominent in the LDA analysis but were still significant within the reviews (Casalino, et al., 2016). This complementary analysis ensured a comprehensive understanding of the underlying themes in customer feedback.

3.4.4 Evaluation Metrics

Evaluation metrics are essential tools in assessing the effectiveness of machine learning models, particularly in classification tasks. They provide a means to quantify how well a model is performing in predicting the correct categories, going beyond simple accuracy to consider aspects like precision, recall, and the balance between these factors with the F1 score (Dalianis, 2018). In sentiment analysis, where models must correctly identify and classify the sentiment expressed in text, these metrics become crucial in ensuring that the model not only performs well on average but also handles imbalanced datasets and various types of classification errors (John & Kartheeban, 2019). By employing a range of evaluation metrics, we can gain a more comprehensive understanding of a model's strengths and weaknesses, ultimately leading to better model selection and tuning for optimal performance. Here are the metrics we employed for our study:

Accuracy

This metric measures the proportion of correctly classified instances out of the total instances. While accuracy provides a general overview of the model's performance, it is not always sufficient, especially for imbalanced datasets (Sumeet, 2024).

Accuracy can be represented as:

$$\frac{TP+TN}{TP+TN+FP+FN} \qquad (10)$$

Where:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Precision

Precision measures the ratio of true positive predictions to the total number of positive predictions made by the model (Sumeet, 2024). This metric was crucial for determining the reliability of the model's positive predictions. It works using:

$$Precision = \frac{TP}{TP+FP} \tag{11}$$

Recall

Recall, or sensitivity, is the ratio of true positive predictions to the actual number of positive instances in the dataset. This metric was used to evaluate how well the model captured all relevant instances of the positive class (Sumeet, 2024).

$$Recall = \frac{TP}{TP+FN} \tag{12}$$

F1 Score

The F1 score is the harmonic mean of precision and recall, providing a balanced metric that considers both false positives and false negatives. It was particularly useful for evaluating models where class distributions were uneven (Python Programmer, 2023).

$$F1\,Score = 2 * \frac{Precison * Recall}{Precision + Recall} \tag{13}$$

Confusion Matrix

The confusion matrix provided a comprehensive breakdown of the model's performance by showing the number of true positives, true negatives, false positives, and false negatives. This helped in visualizing the strengths and weaknesses of the models (Sumeet, 2024).

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Table 2: Here is a table illustrating the structure of a Confusion Matrix.

3.4.5 Experimental Setup

The experimental setup was straightforward, with no specific configurations or setups deemed critical for this study. The experiments were run on standard computing hardware, and no additional resources or environments were required.

3.4.6 Ethical Considerations

Although the study did not involve sensitive data or human subjects directly, ethical considerations were taken into account. An ethical review form was submitted via the LEAS system to ensure that the research adhered to institutional guidelines and standards.

Chapter 4

Implementation

The implementation phase of this research involved the practical application of various data analysis techniques and machine learning models to analyse customer sentiments expressed in online reviews. This section details the step-by-step process followed to translate the theoretical methodologies into actionable insights.

4.1 Research studies *not* involving human participants

This research did not involve any human participants. All data used in the analysis were publicly available online reviews, this ensures that the study adhered to ethical standards without infringing on privacy or consent issues. We strictly focused on secondary data analysis, making it both ethically sound and aligned with best practices in data-driven research.

The implementation section primarily focuses on applying the methods discussed in Chapter 3 to uncover key trends in customer sentiment and provide actionable recommendations for improving customer satisfaction and retention. This involved a multi-step process that included data loading and cleaning, text preprocessing, sentiment analysis using machine learning and deep learning models, topic modeling, and finally, comparative analysis across the different restaurants.

4.2 Software and Tools

The implementation was carried out in Python, using a range of libraries and tools specifically suited for natural language processing and machine learning tasks. The first step in our implementation was to set up the Python environment.

We used Jupyter Notebook as our development environment due to its interactivity and ease of use for iterative development and data visualization. By using the command, “*!pip install pandas numpy matplotlib seaborn scikit-learn spacy tensorflow wordcloud imblearn*”, these libraries were installed and imported to ensure we had the necessary resources for processing English text:

- **Pandas**: This was used for data manipulation and analysis, particularly to load and explore the dataset (Pandas Documentation, 2024).
- **Numpy**: This provided support for numerical computations (Harris, et al., 2020).
- **Matplotlib and Seaborn**: These were used to create visualizations that helped us understand the data distribution (Waskom, 2021).
- **Scikit-learn**: This is a machine learning library in Python that provides a wide range of supervised and unsupervised learning algorithms. It is designed to be accessible and efficient, making it an ideal tool for data mining and data analysis (Pedregosa, et al., 2011).
- **Spacy**: This was used for advanced NLP preprocessing tasks, such as tokenization, lemmatization, and word embeddings. For NLP tasks, the Spacy library required a specific language model, which was downloaded using the command: `$!python\ -m\ spacy\ download\ en\_core\_web\_md$` (Explosion, 2024)
- **TensorFlow**: This is an end-to-end open-source platform for machine learning that enabled us to easily build and deploy deep learning models (Abadi, 2025).
- **Keras**: This is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, Theano, or CNTK (Chollet, 2015).
- **Word cloud**: This was utilized to generate visual word clouds from the text data to see the word distribution of reviews in relation to the actual customer sentiments (Mueller, 2020).
- **Imbalearn**: This is an extension to scikit-learn that deals specifically with imbalanced datasets. It provides various methods to under-sample and over-sample data, which helps in improving model performance when dealing with

datasets where one class is significantly underrepresented. (Lemaître, et al., 2017).

4.3 Loading the Data

The dataset, stored in an Excel file named “Reviews.xlsx”, was loaded into a pandas DataFrame we named “data”. This was done by using the function “*data = pd.read_excel(‘Reviews.xlsx’)*”. This was an essential step to make the data accessible for further analysis(Pandas, 2024).

4.4 Data Exploration and Initial Analysis

The next task was to inspect the structure of the data, check for missing values, and identify potential issues. To gain an initial understanding, we displayed the first few rows of the dataset by using “*data.head()”*. This command provided a quick look at the dataset, revealing columns such as “Restaurant Name”, “Restaurant Type”, “Star Rating”, “Sentiment”, and “Reviews”. Following that we went ahead to analyse the following:

4.4.1 Checking for Null Values

The dataset was examined for missing or null values, which can significantly impact the performance of machine learning models. The $isnull()$ function in Python was used to detect any null entries across the dataset (Pandas, 2024). This revealed that there were 1911 missing rows in the Reviews column.

4.4.2 Statistical Summary of Data Columns

A statistical summary was generated to understand the central tendencies and distributions of the numerical columns, such as star ratings. Using the $describe()$ function, key metrics like mean, median, mode, standard deviation, and range were computed (Pandas, 2024). This summary provided insights into the distribution of ratings, highlighting whether they were skewed towards positive or negative reviews and identifying any potential outliers that might skew the analysis.

4.4.3 Data Distribution and Visualisation

Data visualisation techniques were employed to gain a clearer understanding of the distribution of various features. Histograms were used to visualize the distribution of star ratings, while bar plots were used to show the frequency distribution of sentiments. These visualisations were generated using Python's matplotlib and seaborn libraries, providing a more intuitive grasp of the data's structure and any underlying patterns (Matplotlib Documentation, 2024; Seaborn Documentation, 2024).

To visualize the distribution of star ratings, the following steps were taken:

Importing Visualization Libraries

'*import seaborn as sns'* and *'import matplotlib.pyplot as plt*' were used to import the packages we had previously installed.

Creating a Count Plot

'*sns.countplot(x='Star Rating', data=data)*' was used to visualize the distribution of the various ratings across the restaurants.

Setting the Figure Size and Displaying the Plot

We used 'plt.figure(figsize=(8, 6))' and '*plt.show()*' were used to set the figure size and display the plot. These visualizations were vital in understanding how customers rated their experiences and identifying any skewness or bias in the ratings.

4.5 Data Cleaning and Preprocessing

Data cleaning is a critical step in any data analysis project. It ensures that the data used for modeling is accurate, consistent, and free from errors. In our case, data cleaning involved handling missing values, removing duplicate entries, and cleaning the text data by removing non-informative characters. Here is how they were applied:

4.5.1 Handling Missing Values

Missing values in the 'Reviews' column could potentially lead to biased results. To address this, we removed the missing using: $dropna()$ function (McKinney, 2010).

4.5.2 Handling Duplicated Values

Duplicate entries can distort the analysis by giving undue weight to certain reviews. To address this, we removed the duplicated entries by using the '*data.drop_duplicates(inplace=True)*' function in python. The original dataset contained 5,000 reviews; however, following data cleaning and deduplication processes, the number of reviews was reduced to 3,089. This ensured that each review was unique, preventing any biases from duplicated or missing data (McKinney, 2010).

4.5.3 Normalization

The text data required additional cleaning to remove non-informative characters such as URLs, emojis, and special symbols. This was essential for preparing the text for further analysis, including sentiment analysis and topic modeling. The *str.lower()* function was used to convert all text to lowercase. The *str.replace()* function was, used to remove special characters, and the *strip()* function was applied to remove any leading or trailing whitespace (NLTK Documentation, 2024).

4.5.4 Tokenization and Lemmatization

Tokenization involves breaking down text into individual words and lemmatization reduces words to their base or root form(Webster & Kit, 1992). These processes were implemented using the Spacy library. We loaded the Spacy English model: '*import spacy*', '*nlp = spacy.load('en_core_web_md')*' .This model includes pre-trained word vectors, making it efficient for NLP tasks. We then created a preprocessing to process each review:

```
'def preprocess_text(text):
doc = nlp(text)
tokens = [token.lemma_ for token in doc if not token.is_stop and token.is_alpha]
```

return ' '.join(tokens)'

This function takes a review as input, tokenizes and lemmatizes it, and removes stopwords and punctuation. The output is a cleaned version of the text, ready for vectorization.

4.5.5 Word Embeddings

The *en_core_web_md* model from SpaCy was used again to generate 300-dimensional embeddings, which served as the input features for the subsequent machine learning models (Honnibal & Montani, 2017). This approach was chosen due to its proven effectiveness in handling text data, particularly in tasks requiring an understanding of the context in which words are used.

4.5.6 Vectorization

Vectorization is the process of converting text data into numerical vectors that machine learning models can process(Goyal, 2021). We used TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, which assigns a weight to each word based on its frequency in the document and across all documents(Huilgol, 2024). We imported it by using the function '*from sklearn.feature_extraction.text import TfidfVectorizer'*, then proceeded with '*vectorizer = TfidfVectorizer(max_features=5000)*' and '*X = vectorizer.fit_transform(data['Processed Reviews'])*'. This process converted the cleaned text into a matrix of numerical features, with each feature representing the importance of a word in the review. The resulting matrix `X` was used as input for machine learning models.

4.6 Training Machine Learning Models

With the pre-processed text data ready, the next step was to train machine learning models to predict sentiment. The data was split into training and testing sets to evaluate the models' performance.

4.6.1 Splitting the data

The data was split into 80% training and 20% testing using: '*from sklearn.model_selection import train_test_split', 'X_train, X_test, y_train, y_test = train_test_split(X, data['Sentiment'], test_size=0.2, random_state=42)*'. This ensured that the models could be evaluated on unseen data, providing a measure of their generalization performance.

4.6.2 Random Forest Classifier

The Random Forest classifier is a powerful ensemble method that builds multiple decision trees and merges them to obtain a more accurate and stable prediction(Saini, 2024).

Training the model

To import the model package we used '*from sklearn.ensemble import RandomForestClassifier*' to import the model. Following that, '*rf_model=RandomForestClassifier(n_estimators=100, random_state=42)*' was implemented. It creates an instance of the *RandomForestClassifier* named *rf_model* with *100* decision trees (*n_estimators=100*) and sets the random seed to 42 (*random_state=42*) to ensure reproducibility of the results. The function '*rf_model.fit(X_train, y_train)*' was used to train the *rf_model* using the training data i.e. X_train and the corresponding labels (y_train).

Evaluating the Model

The '*y_pred = rf_model.predict(X_test)'* function uses the trained *rf_model* to make predictions on the test data (X_test). The predict method generates the predicted class labels for each sample in the test dataset, which are then stored in the variable y_pred. We imported the classification report using '*from sklearn.metrics import classification_report*'. This provided detailed metrics, including precision, recall, and F1-score, all of which are crucial for assessing the model's performance across different sentiment classes.

4.6.3 Support Vector Classifier

SVCs are effective in high-dimensional spaces which makes them well-suited for text data(Premanand, 2023).

Training the SVC Model

'*from sklearn.svm import SVC'* imported the Support Vector Classification class from the SVM module of the scikit-learn library. We then used '*svc_model = SVC(kernel='linear', random_state=42)*' to create an instance of the SVC class named *svc_model* with a specified kernel function and a random seed. The *kernel='linear'* parameter specifies that a linear kernel will be used. The *random_state=42* parameter sets the seed for random number generation, ensuring that the results can be duplicated. We proceeded to train the model '*svc_model.fit(X_train, y_train)*'. This uses the training data in *X_train* and the corresponding labels in *y_train*. The fit method optimizes the hyperplane that best separates the classes in the feature space. For a linear kernel, the algorithm will attempt to find a straight line (in two dimensions) or a hyperplane (in higher dimensions) that maximally separates the data points of different classes.

Evaluating the SVC Model

We applied '*y_pred_svc = svc_model.predict(X_test)*' line uses to make predictions on the test data , *X_test* by using the trained *svc_model* . The predict method generates the predicted class labels for each sample in the test dataset. These predicted labels are stored in the variable y_pred_svc.

4.6.4 Naïve Bayes Classifier

This model had a slightly different approach because we used TD-IDF vectorizer instead of word embeddings. We split the TF-IDF-transformed data '*X_tfidf*' and the corresponding *y* labels into training and testing sets using "*X_train_tfidf, X_test_tfidf, y_train, y_test = train_test_split (X_tfidf, y, test_size=0.2, random_state=42)".*The *test_size=0.2* parameter specifies that 20% of the data was used for testing, and *random_state=42* ensures the split can be used again by setting a seed for the random number generator(Ray, 2024).

Training the model

To train the model we used the python functions: "*nb = MultinomialNB()*" and "*nb.fit(X_train_tfidf, y_train)*". These functions initialised a Multinomial Naive Bayes classifier and trained it using our training data '*X_train_tfidf*' and training labels '*y_train*'.

Evaluating the model

"*y_pred_nb = nb.predict(X_test_tfidf)*" With this function, the trained Naive Bayes model was used to predict the class labels for the test data i.e. *X_test_tfidf*. The predicted labels are stored in the variable *y_pred_nb*.

4.7 Training Deep Learning Models

To capture more complex patterns in the data, deep learning models were employed. We explored Convolutional Neural Networks and Recurrent Neural both of which are commonly used in text classification tasks.

4.7.1 Convolutional Neural Network

CNNs are particularly effective at recognizing spatial hierarchies in data. In text analysis, this translates to understanding the relationship between words within a sentence. The approach to training the deep learning models is also quite different from the one we took for the machine learning models (Harshi, 2021).

Convert String Labels to Integer Labels

For this step, we used *LabelEncoder*. This class from *sklearn.preprocessing* is used to convert categorical string labels into integer labels. In Python it was implemented by using (Sharma, 2024):

label_encoder =LabelEncoder(),

y_train_encoded = label_encoder.fit_transform(y_train)

y_test_encoded = label_encoder.transform(y_test)

The *fit_transform* method is applied to the training labels "*y_train*" to encode them as integers. The same encoding is applied to the test labels "*y_test*" using the transform method, ensuring consistent mapping between the label strings and integers.

One-Hot Encode the Integer Labels

For this step, we implemented the following code:

```
y_train_one_hot = to_categorical(y_train_encoded, num_classes=3)
y_test_one_hot = to_categorical(y_test_encoded, num_classes=3)
```

The *to_categorical* function from *keras.utils* was used to convert the integer-encoded labels into one-hot encoded vectors. One-hot encoding is a process that converts categorical integer labels into binary matrix representations, where each class is represented by a unique binary vector. This is crucial for multi-class classification problems where the output layer of the neural network uses a softmax activation function. The parameter *num_classes=3* specifies the number of unique classes in the dataset (Sharma, 2024).

Building and Compiling the CNN Model

Here is the function we used for this step:

```
cnn = Sequential([
    Embedding(input_dim=X_train.shape[1], output_dim=128),
    Conv1D(128, 5, activation='relu'),
    GlobalMaxPooling1D(),
    Dense(10, activation='relu'),
    Dense(3, activation='softmax')  # 3 output neurons for 3 classes
])
```

Where*:*

Sequential is class from *keras.models* that allowed us to build a linear stack of neural network layers. The *Embedding* layer converted the input data into dense vectors of fixed size. It's primarily used to transform text data into word embeddings. Here, *input_dim=X_train.shape[1]* indicates the size of the input dimension and *output_dim=128* specifies the dimension of the dense embedding vectors. *Conv1D* applies a 1D convolutional filter along our text data. It has 128 filters, a kernel size of 5, and uses the *relu* activation function to introduce non-linearity. *GlobalMaxPooling1D* performed max pooling across the entire sequence length, reducing the output from the convolutional layers to a fixed size vector. It helped to reduce the dimensionality and computational complexity of the model while retaining important features. *Dense* is a fully connected layer that has 10 neurons with *relu* activation, providing additional non-linearity and learning capacity to the network. The final Dense layer has 3 neurons, one for each class, with a softmax activation function, which is used for multi-class classification problems.

Compiling the Model

To assemble everything so we can finally use the model, we applied the function: *cnn.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy']).* The *optimizer='adam'* is a popular optimization algorithm used in training neural networks due to its efficiency and low memory requirements. The *categorical cross-entropy* loss function is used for multi-class classification problems where the output labels are one-hot encoded. It calculates the difference between the actual and predicted probability distributions of the classes. Then the model's performance will be evaluated using *accuracy*, which is a metric that measures the proportion of correctly classified samples (Sharma, 2021).

Training the CNN Model

Finally we trained our model using: '*cnn.fit(X_train, y_train_one_hot,epochs=5,batch_size=64,validation_data=(X_test,y_test_one_hot))'*

In this function, *fit* is used to train the CNN model on the training data *X_train* and *y_train_one_hot*. The training process involves adjusting the weights of the neural network based on the loss calculated on the training set. *epochs=5* specifies the number of times the entire training dataset will pass through the neural network. In this case, the model will train for 5 epochs. The *batch_size=64* means the number of training samples that will be used in each iteration of training. And then the *validation_data* represents the validation data to be used for evaluating the model after each epoch. It helps monitor the model's performance on unseen data, allowing for early stopping if overfitting is detected (Harshi, 2021).

4.7.2 Recurrent Neural Network

Recurrent Neural Network (RNN) is a type of neural network architecture designed for processing sequential data, such as text, speech, or time series data. In sentiment analysis, RNNs can capture the dependencies between words in a sentence for a more nuanced understanding of customer reviews. The approach we took to implement RNN in Python is very similar to what we did with CNN, except for a few things.

Building the RNN Model

We began by implementing the foundational structure of the RNN model using the Keras Sequential API. The model architecture is designed to process sequential data, which is crucial for tasks such as sentiment analysis where the order of words can significantly influence the meaning of a sentence(Hochreiter & Schmidhuber, 1997).

```
rnn = Sequential([
Embedding(input_dim=X_train.shape[1], output_dim=128),
SimpleRNN(128, activation='relu'),
    Dense(3, activation='softmax')

])
```

In this architecture, the *Embedding layer* transforms the input words into 128-dimensional vectors, effectively capturing semantic relationships between words. The *SimpleRNN* layer, which contains 128 units, processes these word sequences, maintaining a hidden state that evolves over time. The activation function used here is *ReLU* (activation='relu'). Finally, the *Dense* layer with 3 units and a softmax activation function outputs probabilities for the three sentiment classes (Negative, Neutral, Positive), allowing the model to make predictions based on the highest probability (Kalita, 2024).

Compiling the Model

To compile the model we used the function, "*rnn.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])*" . The features of the code serve the same purpose as they did with the CNN model.

Training the Model

We trained our RNN model by implementing, "*rnn.fit(X_train,y_train_one_hot,epochs=5,batch_size=64,validation_data=(X_test,y_test_one_hot))".*

4.8 Challenges and Solutions

One of the most significant challenges encountered was the underperformance of the machine learning models. Despite careful data preparation and model selection, many models consistently achieved accuracy rates of less than 85%, a threshold generally considered suboptimal for predictive tasks in sentiment analysis (Zhang, et al., 2018). This underperformance prompted a deeper investigation into the underlying causes and the exploration of strategies to enhance model efficacy.

4.8.1 Data Imbalance and Model Sensitivity

A critical issue contributing to the underperformance was the imbalance in the sentiment classes within the dataset. Like many real-world datasets, the distribution of sentiments in the online reviews was skewed, with certain classes (e.g., positive sentiments) being more prevalent than others (e.g., neutral sentiments). This imbalance often leads to biased model predictions, where the model favours the majority class at the expense of the minority classes (Johnson and Khoshgoftaar, 2019). Consequently, models such as Naive Bayes and even more sophisticated algorithms like CNNs and RNNs struggled to accurately classify the less represented classes and this resulted in lower overall accuracy and poor recall for minority classes.

4.8.2 Model Complexity and Overfitting

Another challenge was managing the complexity of the models, particularly for deep learning architectures such as CNNs and RNNs. While these models are powerful in capturing intricate patterns in data, they are also prone to overfitting, especially when the training data is not sufficiently large or representative (Goodfellow, Bengio, and Courville, 2016). Overfitting occurs when the model becomes too tailored to the training data, capturing noise rather than the underlying signal, leading to diminished performance on unseen data (Ying, 2019). This issue was observed during the evaluation phase, where high training accuracy did not translate into equivalent test accuracy, indicating that the models were not generalising well to new data.

4.8.3 Hyperparameter Tuning for Machine Learning Models

Given these challenges, it became clear that enhancing model performance required a more nuanced approach. We implemented SMOTE to address class imbalance and hyperparameter tuning to optimise model configurations (Chawla et al., 2002). By integrating these techniques, we aimed to mitigate the limitations observed and achieve higher accuracy and generalisation across all sentiment classes, ultimately leading to more reliable and actionable insights from the sentiment analysis.

SMOTE for Random Forest

It was particularly important given the skewed distribution of sentiment classes in our data, where positive sentiments were more prevalent than neutral or negative sentiments. The SMOTE process involved resampling the feature set X and the target labels y to create a more balanced dataset for model training (Chawla et al., 2002).

Following the application of SMOTE, we used *GridSearchCV* to perform hyperparameter tuning on the Random Forest model. We defined a parameter grid for Random Forest, including variations in the number of estimators *n_estimators*, the maximum depth of trees *max_depth*, and the minimum number of samples required to split an internal node *min_samples_split.* GridSearchCV systematically evaluated different combinations of these parameters through cross-validation, ultimately selecting the configuration that maximised accuracy on the training data(Bergstra and Bengio, 2012). This refined model was then used to make predictions on the test set, resulting in improved performance metrics, as indicated by the classification report (Table 3).

SMOTE for SVM

Applying SMOTE to our SVM model involved resampling the feature set *X* and the target labels *y*, leading to a dataset where all sentiment classes were adequately represented.

After applying SMOTE, the resampled data was split into training and testing sets using an 80-20 split. To achieve the best model performance, we employed *GridSearchCV* for hyperparameter tuning (Bergstra & Bengio, 2012). A parameter grid was defined for the SVC model, including variations in the regularisation *parameter C*, the kernel coefficient *gamma*, and the kernel type *rbf* and *linear* (Müller, et al., 2001). GridSearchCV systematically explored these combinations through cross-validation, ultimately selecting the optimal set of hyperparameters that maximised the accuracy on the training data (Bergstra and Bengio, 2012).

The best-performing model was then used to predict sentiment classes on the test set, and its performance was evaluated using key metrics such as accuracy, precision, recall, and F1-score (Dalianis, 2018). These metrics provided a comprehensive assessment of how well the model performed across all sentiment classes, reflecting the impact of SMOTE and hyperparameter tuning on improving the model's ability to accurately classify sentiments (Table 5).

4.8.4 Hyperparameter Tuning for Deep Learning Models

We utilised the *Keras Tuner library* to perform hyperparameter tuning for the RNN and CNN models.

Tuning RNN

The process involved defining a search space for key hyperparameters, including the output dimension of the embedding layer *embedding_output_dim*, the number of units in the RNN layer *rnn_units*, and the activation function used in the RNN layer *rnn_activation*. These hyperparameters significantly influence the model's capacity to learn patterns in sequential data, which is crucial for tasks such as sentiment analysis (LeCun, et al., 2015).

To explore this search space effectively, we employed the *Hyperband* algorithm via Keras Tuner, which is designed to efficiently allocate resources among hyperparameter configurations by balancing the exploration of a wide range of parameters with the fine-tuning of promising configurations (Li et al., 2018). The tuner was set up to optimise for validation accuracy *val_accuracy*, running a series of training epochs with different hyperparameter settings and evaluating performance on a validation set.

Once the tuning process was complete, the best-performing model was selected based on its validation accuracy. This model was then evaluated on the test set, where it achieved a higher accuracy score, demonstrating the effectiveness of the tuning process (Figure 7).

Tuning CNN

Similar to the RNN model, we utilised the *Keras Tuner* library here. The tuning process began by defining a search space for key hyperparameters, including the output dimension of the embedding layer *embedding_output_dim*, the number of filters in the convolutional layer *filters*, and the number of units in the dense layer *units*. These hyperparameters were systematically varied to identify the configuration that would yield the best validation accuracy (Li et al., 2018).

The *Hyperband* algorithm efficiently allocated computational resources by testing a wide range of hyperparameter configurations in the initial rounds and then focusing on the most promising configurations in subsequent rounds. This approach ensured a thorough exploration of the hyperparameter space while maintaining computational efficiency (Falkner, et al., 2018).

Once the tuning process was completed, the best-performing model was selected based on its validation accuracy. This optimised model was then evaluated on the test set, where its performance was measured using accuracy as well as a detailed classification report. The classification report provided insights into the model's precision, recall, and F1-score across the different sentiment classes (Table 6).

4.9 Topic Labelling

In the process of extracting key themes from the customer reviews, we utilised two topic modelling techniques: Latent Dirichlet Allocation and Non-Negative Matrix Factorization. These models are particularly effective for uncovering the underlying structure in large collections of textual data, allowing us to identify distinct topics that reflect the primary themes discussed by customers (Blei, et al., 2003).

To begin, we used *CountVectorizer* to transform the pre-processed reviews into a matrix of token counts. This method converts the text data into a structured format that can be fed into the LDA and NMF models. The *CountVectorizer* was configured with parameters such as *max_df=0.95* to ignore words that appear in more than 95% of the documents (likely common stopwords or overly frequent terms) and *min_df=2* to include only those words that appear in at least two documents, ensuring that the topics generated are meaningful and relevant to the broader dataset (Rajaraman & Ullman, 2011).

4.9.1 Latent Dirichlet Allocation

LDA is a generative probabilistic model that assumes documents are mixtures of topics and that topics are mixtures of words (Kibe, 2024). For this study, we configured the LDA model to extract five distinct topics (n_components=5). The LDA model was trained on the matrix of token counts generated by the *CountVectorizer*, and the resulting topics were analysed by examining the top words associated with each topic. The top words provide a concise summary of the main themes captured by each topic, offering insights into what customers frequently discussed in their reviews (Blei, Ng, and Jordan, 2003).

4.9.2 Non-Negative Matrix Factorization

In addition to LDA, we applied NMF, a linear algebra-based technique that factorises the document-term matrix into two lower-dimensional matrices, capturing the latent topics within the text (Bhangale, 2023). Like LDA, we configured NMF to extract five topics *n_components=5*. NMF is particularly useful for text analysis as it allows for more interpretable results, with topics represented as additive combinations of words (Lee and Seung, 1999). After training the NMF model on the same token count matrix, we analysed the top words in each topic, similar to the approach taken with LDA.

4.9.3 Topic Interpretation and Labelling

After extracting the topics, we manually reviewed the top words associated with each topic to assign meaningful labels. For instance, a topic with words like "service," "staff," "friendly," and "time" was labelled as "Customer Service." This step ensured that the topics identified by the models were relevant to our research questions and could be interpreted in the context of the restaurant industry.

Integration with Sentiment Analysis

The identified topics were then integrated with sentiment analysis results to understand how sentiment varied across different aspects of the restaurant experience. For example, we examined whether the sentiment associated with "Service Quality" was generally positive, negative, or neutral, and how this sentiment correlated with overall customer satisfaction. We implemented a series of steps using Python's data manipulation and visualization libraries. The process involved filtering the dataset to focus on specific topics, grouping the data by sentiment and restaurant, and then visualising the results:

topic_data = df[df['NMF_Topic_Label'] == label]

Here, label represents the specific topic label being analysed. The filtered data *topic_data* contains only the reviews related to that topic (Pedregosa et al., 2011). Next, the filtered data was grouped by restaurant name and sentiment, and the number of reviews for each sentiment category (Positive, Neutral, Negative) was counted. This grouping was essential for understanding how each restaurant was perceived in relation to the given topic. The grouping and counting were performed as follows:

sentiment_count=topic_data.groupby(['RestaurantName','Sentiment']).size().unstack(fill_value=0)

The *groupby* function was used to aggregate data by 'Restaurant Name' and 'Sentiment', and unstack(*fill_value=0*) was used to transform the data into a more readable format, with missing values filled with zeros (McKinney, 2010).

This approach provided a comprehensive view of customer sentiment distribution across various topics, offering valuable insights into the performance of each restaurant in specific areas of customer experience.

Chapter 5

Results & Discussion

Our overall motivation for this study was to conduct a detailed comparative analysis on the performance of the five fast-food restaurants by using sentiment analysis. In this scenario, each of these establishments— Village Burger, AZN Sandwich Bar, Farm Burger Dunwoody, Cheeseburger Bobby's, and Lucky's Burger & Brew Marietta— is a worthy competitor looking to breakthrough within the fast-food sector. We mainly concentrated on three key factors in the online reviews assessed, these were the dining experience, customer service, and food quality, with the goal of influencing strategic decisions, projecting economic viability, and ultimately boosting customer happiness and retention.

5.1 Overview of Analysis Methods and Results

The study utilised five machine learning models, including Random Forest, Naive Bayes, Support Vector Machine, Convolutional Neural Networks and Recurrent Neural Networks, to conduct sentiment analysis on online reviews. The sentiments were categorised into negative, neutral, and positive to provide a comprehensive view of customer perceptions across various topics.

5.1.1 Random Forest Results

Here are the key metrics and results we used in evaluating our model:

- **Accuracy**: 0.93
- **Precision**: Average precision across all classes was 0.94, indicating the model’s high ability to correctly identify the positive, neutral, and negative sentiments without much false positives.

- **Recall**: With a recall of 0.94, the RF model was effective in identifying the correct sentiment, showing minimal false negatives.
- **F1-Score**: The F1-score was consistently around 0.94, suggesting a balance between precision and recall.

Metric	Negative	Neutral	Positive	Macro Avg	Weighted Avg
Precision	0.93	0.95	0.93	0.94	0.94
Recall	0.92	0.98	0.92	0.94	0.94
F1-Score	0.93	0.97	0.93	0.94	0.94
Support	412	433	445		

Table 3: Here is a table displaying the results of the Random Forest model

The high accuracy and balanced precision, recall, and F1-scores indicate that the Random Forest model performs well in categorising the sentiments, making it reliable for this task. The model's performance across all sentiment classes was nearly identical, demonstrating its efficiency and generalisation capability.

5.1.2 Support Vector Machine Results

The results of the Support Vector Machine support the analysis and assessment previously made by Samal and Panda (2017). Here are the key metrics and results we used in evaluating our model:

- **Accuracy:** 0.95
- **Precision:** The SVM model showed exceptional precision, with an average precision of 0.96, particularly excelling in identifying neutral sentiments (0.98).
- **Recall:** With an average recall of 0.95, the model efficiently identified the correct sentiment without missing relevant instances.

- **F1-Score:** The F1-score was high across all classes, averaging 0.96, indicating an excellent balance between precision and recall.

Metric	Negative	Neutral	Positive	Macro Avg	Weighted Avg
Precision	0.97	0.98	0.92	0.96	0.96
Recall	0.92	0.98	0.96	0.95	0.96
F1-Score	0.94	0.98	0.94	0.95	0.96
Support	412	433	445		

Table 4: Here is a table displaying the results of the SVM model

The SVM model outperformed the others, showing remarkable accuracy and balanced precision, recall, and F1-scores across all sentiment categories. This makes SVM the most effective model for sentiment analysis in this study, capable of distinguishing subtle differences in customer sentiment with high reliability.

5.1.3 Naïve Bayes Results

Naive Bayes is a simple probabilistic model that applies the Bayes' theorem. It often performs well in text classification tasks although not as much in this study. Here are the key metrics and results we used in evaluating our model:

- **Accuracy**: 0.60
- **Precision**: The model had an overall precision of 0.60, with the highest precision observed for positive sentiments (0.69). However, it struggled with neutral sentiments, with precision dropping to 0.52.
- **Recall**: The model's recall was uneven, with a significantly higher recall for negative sentiments (0.72) and much lower for neutral (0.47).
- **F1-Score**: The F1-score, which balances precision and recall, was moderate, with the lowest being for neutral sentiments (0.49).

Metric	Negative	Neutral	Positive	Macro Avg	Weighted Avg
Precision	0.6	0.52	0.69	0.6	0.6
Recall	0.72	0.47	0.61	0.6	0.6
F1-Score	0.65	0.49	0.65	0.6	0.6
Support	412	433	445		

Table 5: Here is a table displaying the results of the Naïve Bayes model

The Naive Bayes model struggled, particularly with neutral sentiment classification, resulting in lower overall performance. This can be attributed to the model's simplistic assumptions about feature independence, which do not hold well in complex sentiment analysis tasks.

5.1.4 Convolutional Neural Network Results

In sentiment analysis, CNNs can capture hierarchical patterns in text. Here are the key metrics and results we used in evaluating our model:

- **Accuracy**: 0.8445
- **Precision**: The CNN model demonstrated strong precision in identifying positive sentiments (0.86) but struggled with neutral sentiments, where precision was 0.00.
- **Recall**: The recall was high for positive sentiments (0.96) but low for neutral (0.00) and negative sentiments (0.70), indicating some difficulty in correctly identifying negative and neutral sentiments.
- **F1-Score**: The F1-score was highest for positive sentiments (0.91) but low for neutral sentiments (0.00). This suggests that while the model is effective in identifying positive sentiments, it struggles with other categories.

Metric	Negative	Neutral	Positive	Macro Avg	Weighted Avg
Precision	0.79	0	0.86	0.55	0.79
Recall	0.7	0	0.96	0.56	0.84
F1-Score	0.74	0	0.91	0.55	0.81
Support	141	37	433		

Table 6: Here is a table displaying the results of the CNN model

The CNN model performed well in identifying positive sentiments, with high accuracy and F1-scores for this class. However, the poor performance in identifying neutral sentiments, as indicated by the zero precision and recall, highlights a limitation of the model. The model might require further hyperparameters tuning or it might have performed better if we had a larger and more diverse dataset.

5.1.5 Recurrent Neural Network Results

The RNN model was evaluated using the same dataset. We focused on its ability to classify sentiments into positive, neutral, and negative categories. The model achieved a test accuracy of 0.8134 which reflects its performance in capturing the temporal dependencies in the text data. However, the classification report indicates some areas where the model struggled, particularly with neutral sentiments.

- **Accuracy**: 0.8134
- **Precision**: The overall precision was moderate, with notable performance in the positive sentiment class (0.83). However, the model showed poor precision for neutral sentiments (0.00), indicating difficulty in correctly identifying these instances.
- **Recall**: The recall was relatively low, especially for neutral sentiments (0.03), reflecting the model's struggle to detect neutral sentiments correctly.

- **F1-Score**: The F1-score was highest for positive sentiments (0.88) but very low for neutral sentiments (0.05), suggesting that the model had challenges in maintaining a balance between precision and recall for certain classes.

Metric	Negative	Neutral	Positive	Macro Avg	Weighted Avg
Precision	0.74	1	0.83	0.86	0.82
Recall	0.62	0.03	0.94	0.53	0.81
F1-Score	0.68	0.05	0.88	0.53	0.78
Support	141	37	433		

Table 7: Here is a table displaying the results of the RNN model

Our RNN model demonstrated moderate performance, particularly excelling in classifying positive sentiments. However, its inability to accurately classify neutral sentiments indicates that the model may require further tuning or additional data to improve its generalisation across all sentiment categories. The low precision and recall for neutral sentiments are areas of concern that need addressing for the model to be considered reliable in a broader sentiment analysis context.

The SVM model remains the standout in this analysis, demonstrating the highest accuracy, precision, recall, and F1-scores, especially when compared to other models. The CNN and RNN models performed moderately well but had challenges in accurately identifying neutral sentiments, as evidenced by their lower precision and recall for this class. The Random Forest model offered a balanced performance, while the Naive Bayes model struggled significantly, particularly with neutral sentiment classification.

These results highlight the importance of model selection and tuning in sentiment analysis tasks. SVM and Random Forest models appear to be more effective for this particular dataset, while CNN and RNN, despite their powerful capabilities in other domains, may require further optimisation to improve their performance in sentiment analysis.

5.2 Sentiment Distribution and Key Insights

The sentiment distribution across the restaurants was analysed by breaking down the reviews into positive, neutral, and negative categories. This analysis provided key insights into how each restaurant is perceived in the context of the three main topics: dining experience, customer service, and food quality.

5.2.1 Village Burger

Village Burger stands out as the top performer in the sentiment analysis, particularly excelling in the area of food quality. It received the highest number of positive reviews (535) and the lowest negative reviews (60). This indicates strong customer satisfaction, likely driven by consistently high food quality and a favourable dining experience.

Opportunities

While Village Burger’s performance is commendable, maintaining this standard requires continuous innovation in menu offerings and customer engagement. Furthermore, the restaurant can capitalise on its strengths by exploring new customer segments or expanding its footprint.

5.2.2 Cheeseburger Bobby’s

Cheeseburger Bobby’s faced significant challenges, with the highest number of negative reviews (182), particularly related to customer service. This suggests that customers have had suboptimal experiences, which could be detrimental to long-term customer retention and business viability.

Recommendations

To address these challenges, Cheeseburger Bobby's should prioritise improvements in customer service. Training staff to be more responsive and courteous, streamlining service processes, and perhaps revisiting the service model could turn around customer perceptions. Focusing on resolving these issues could shift neutral and negative sentiments toward the positive, thereby improving overall customer satisfaction (Zeithaml, Bitner and Gremler, 2013).

5.2.3 Farm Burger Dunwoody

Farm Burger Dunwoody showed a balanced sentiment distribution with a strong leaning towards positive reviews. The restaurant is performing well but has room for improvement, especially in areas like service efficiency and dining experience (Figure 8).

Recommendations

By enhancing the efficiency of service and optimising the dining experience, Farm Burger Dunwoody can transform these strengths into a competitive advantage. This might involve optimising kitchen operations, revising the menu to reduce complexity, or investing in staff training to ensure consistency in service delivery (Parasuraman, Zeithaml and Berry, 1988).

5.2.4 Lucky’s Burger & Brew Marietta

Lucky's Burger & Brew Marietta received a substantial number of positive reviews, particularly in food quality and the overall dining experience (Figure 6). However, like Farm Burger, there is potential to improve, particularly in service efficiency and staff interactions.

Recommendations

To leverage its strong points, Lucky's should focus on fine-tuning its service processes and enhancing the customer interaction experience. This could involve implementing a customer feedback loop that allows the restaurant to address service issues promptly (Kotler and Keller, 2016).

5.2.5 AZN Sandwich Bar

AZN Sandwich Bar had a relatively high number of negative reviews (161), indicating areas that need improvement, especially in the dining experience (Figure 6). However, the balance with positive reviews (346) shows potential for growth if these challenges are addressed.

Recommendations

AZN should consider revising its approach to the dining experience. This could involve rethinking the restaurant's ambiance, improving food presentation, or diversifying the menu to better meet customer expectations. Implementing changes based on customer feedback could help AZN shift more neutral and negative reviews towards positive (Oliver, 1999).

5.3 Key Themes

The qualitative analysis aimed to identify underlying topics within the reviews using methods like Latent Dirichlet Allocation and Non-Negative Matrix Factorization. The topics identified included Dining Experience, Customer Service, Service Efficiency, Food Quality, and Service and Food. These topics are critical as they directly influence customer satisfaction and retention (Blei, Ng and Jordan, 2003; Lee and Seung, 1999).

5.3.1 Dining Experience

The dining experience appeared to be a strong point for Lucky's Burger & Brew Marietta and Farm Burger Dunwoody. Cheeseburger Bobby's, despite having a majority of positive reviews, might benefit from addressing the issues raised in the negative reviews. Village Burger, although having fewer comments, seems to have a well-received dining experience among its customers.

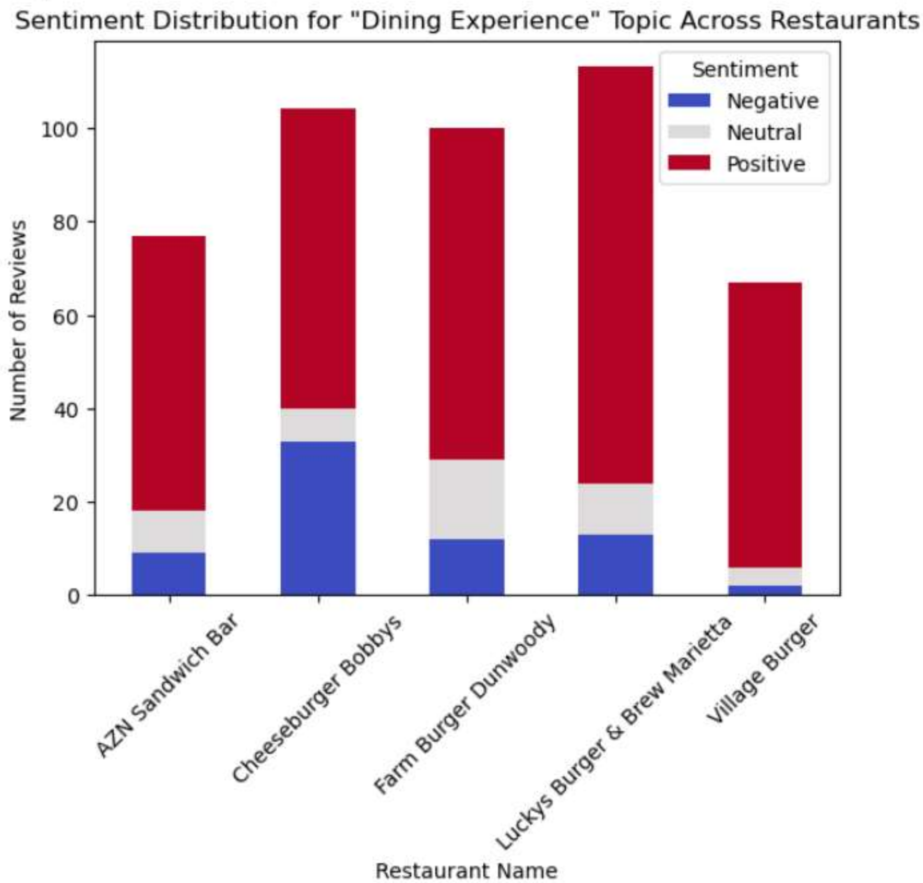

Figure 6: A graph displaying the Sentiment Distribution for "Dining Experience" across the five restaurants.

5.3.2 Customer Service

Overall, customer service was a critical factor across all restaurants. The high number of negative reviews for Farm Burger Dunwoody and Cheeseburger Bobby's suggests that these establishments may need to focus more on improving customer service to enhance customer satisfaction. AZN Sandwich Bar, while receiving more positive reviews, still has significant room for improvement.

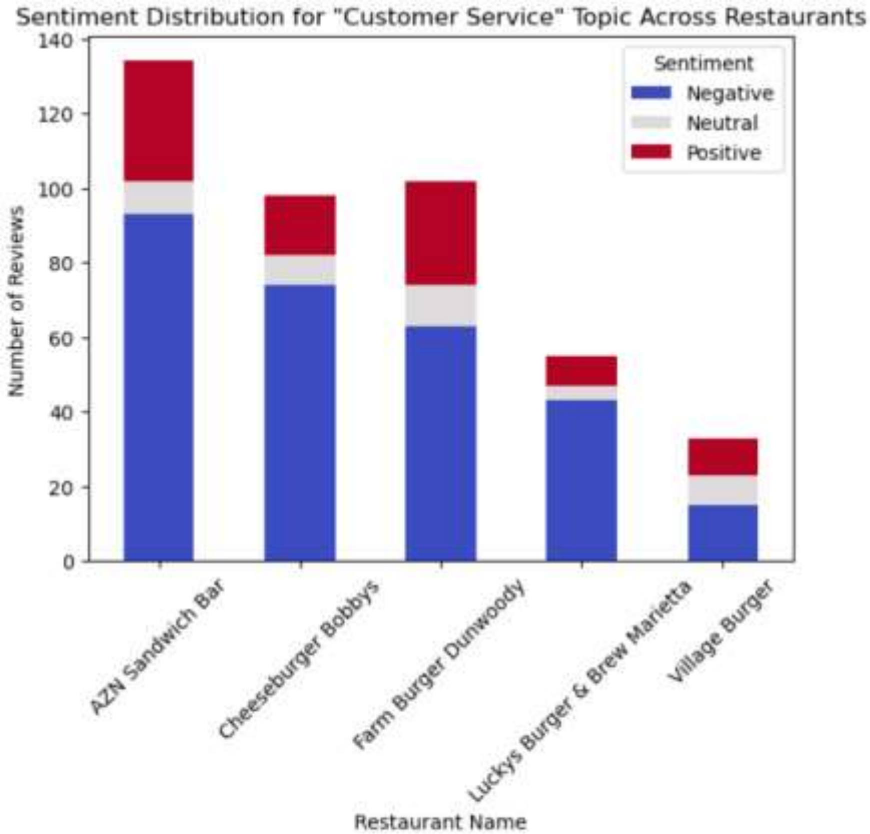

Figure 7: A graph displaying the Sentiment Distribution for "Customer Service" across the five restaurants.

5.3.3 Service Efficiency

This is another aspect we found related to the Customer Service theme. Service efficiency seemed to be a key strength for Village Burger, which outperforms the other establishments in this category. AZN Sandwich Bar and Cheeseburger Bobby's show mixed results, indicating some inconsistency in service efficiency that may need to be addressed. Farm Burger Dunwoody and Lucky's Burger & Brew Marietta, while receiving fewer reviews, appear to have effective service efficiency as well.

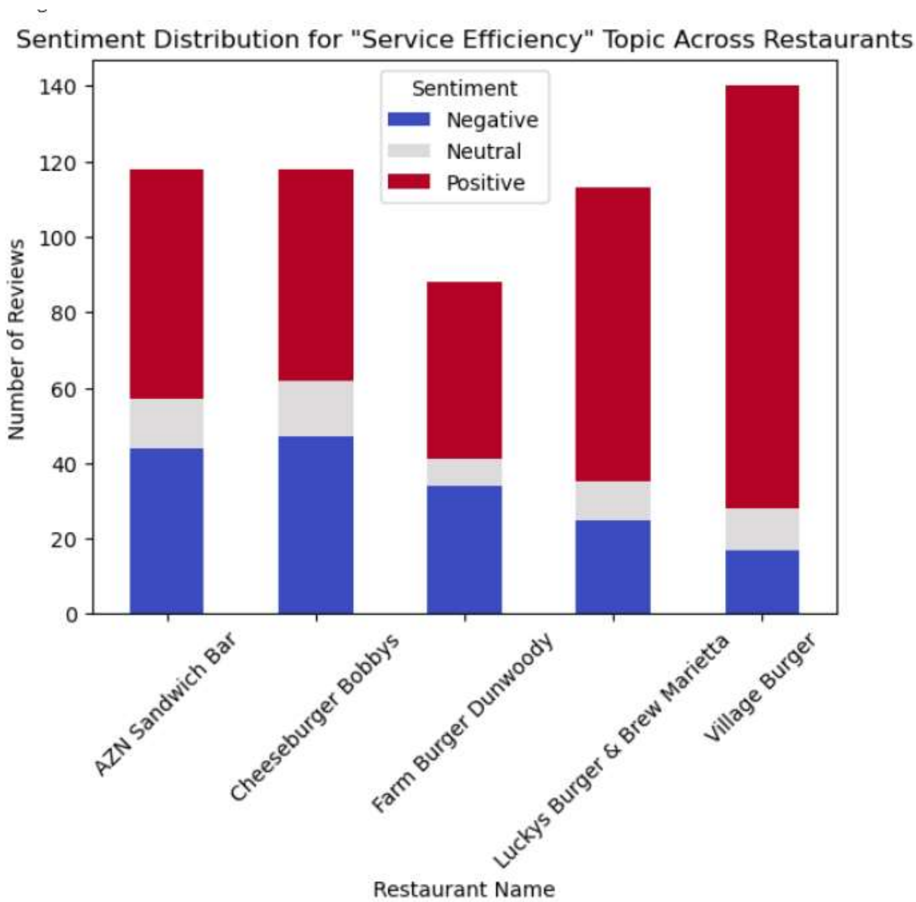

Figure 8: A graph displaying the Sentiment Distribution for "Service Efficiency" across the five restaurants.

5.3.4 Food Quality

Food quality is a clear differentiator for Village Burger, which leads in both the number of reviews and the positivity of those reviews. Farm Burger Dunwoody and Cheeseburger Bobby's also perform well in this category, indicating that food quality is a core strength across these establishments. AZN Sandwich Bar, while having fewer reviews, is positively perceived in terms of food quality.

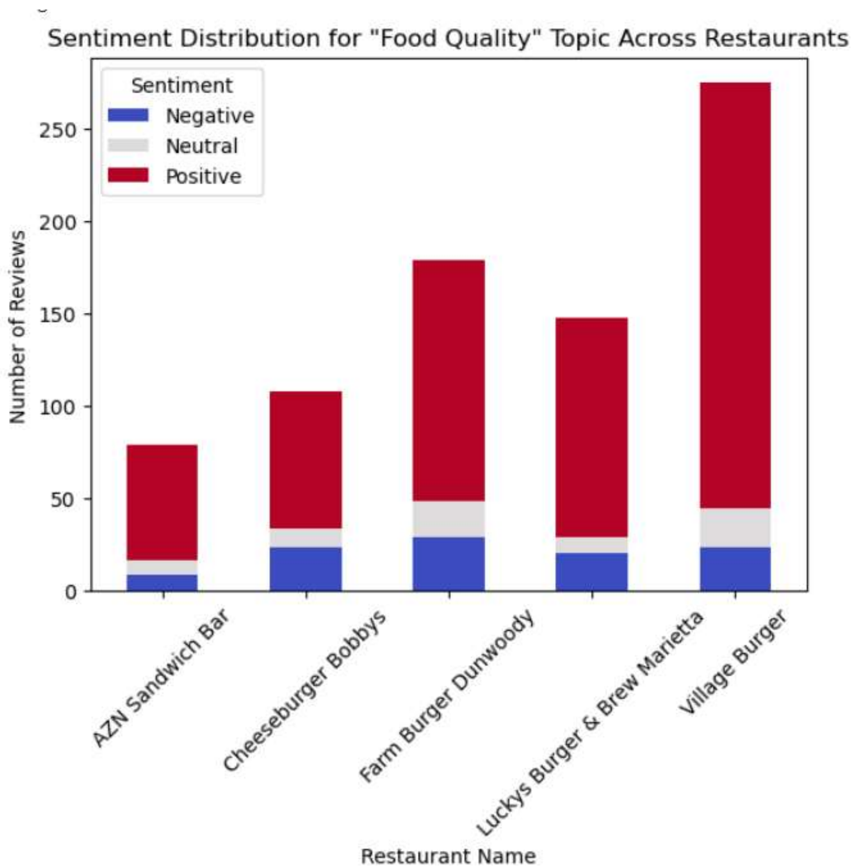

Figure 9: A graph displaying the Sentiment Distribution for "Food Quality" across the five restaurants.

5.3.5 Service and Food

The combination of service and food quality is a powerful driver of customer satisfaction. The analysis showed that restaurants that excelled in both areas, like Village Burger, received the most positive reviews. Cheeseburger Bobby's and AZN Sandwich Bar also perform well, indicating that customers appreciate the overall experience at these establishments. Farm Burger Dunwoody, although receiving fewer reviews, is also positively regarded in this area.

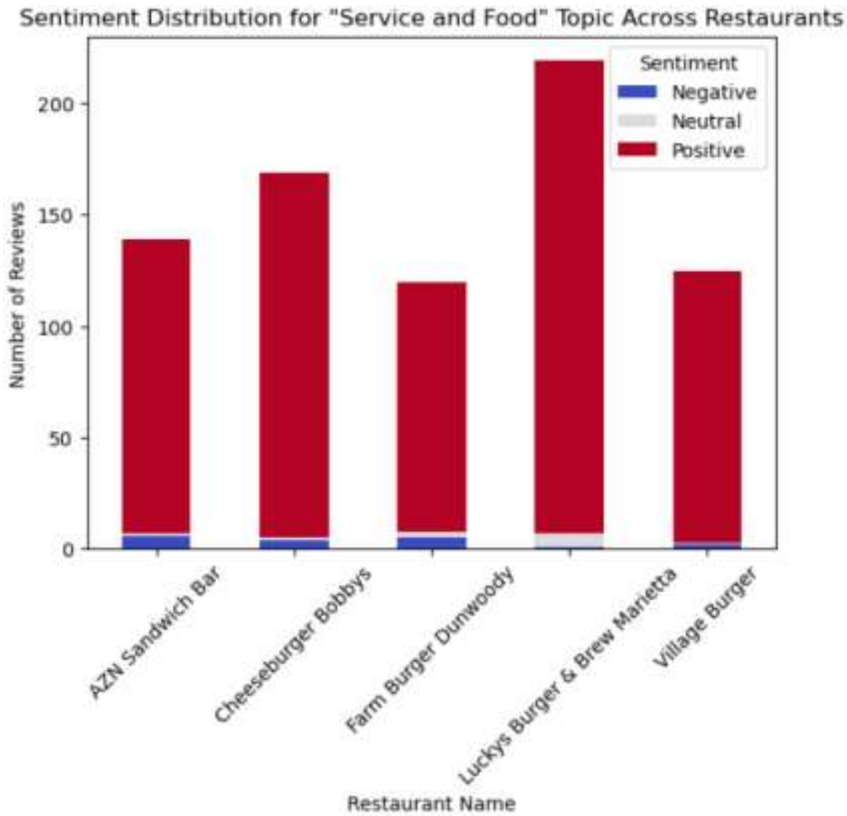

Figure 10: A graph displaying the Sentiment Distribution for "Service and Food" across the five restaurants.

5.4 Strategic Recommendations

Lower-performing restaurants can learn valuable lessons from the success of better-performing competitors like Village Burger. Here are some strategic recommendations:

5.4.1 Focus on Service Excellence

Cheeseburger Bobby's, in particular, should focus on transforming its customer service approach by adopting a customer-first mentality, training staff extensively, and creating a culture of service excellence (Schneider and Bowen, 1995).

5.4.2 Enhance the Dining Experience

A positive dining experience is not just about food but also about the ambiance, cleanliness, and overall customer experience. Cheeseburger Bobby's should consider re-evaluating their dining environments, possibly redesigning their spaces to create a more welcoming and enjoyable atmosphere (Bitner, 1992).

5.4.3 Consistency in Food Quality

Consistency in delivering high-quality food is essential for building customer loyalty. Restaurants struggling in this area should focus on maintaining high standards, conducting regular quality checks, and listening to customer feedback to make necessary adjustments (Sloan, Legrand and Chen, 2013).

5.4.4 Improve Service Efficiency

Efficient service leads to a better customer experience and higher satisfaction. Restaurants like AZN Sandwich Bar and Cheeseburger Bobby's should streamline their operations, perhaps by adopting new technologies or revising their service processes to reduce waiting times and improve overall efficiency (Chase and Apte, 2007).

5.4.5 Leverage Technology for Continuous Improvement

The use of sentiment analysis models like SVM and Random Forest can provide ongoing insights into customer perceptions. By continuously monitoring online reviews and customer feedback, these restaurants can identify emerging issues and address them proactively (Liu, 2012).

It is quite understandable that not all these recommendations are easily achievable. There might be a lot of complexities and constraints to deal with in the process. However continuous innovation and listening to what customers are saying about the restaurant in general is an essential driving force to continuous success.

Chapter 6

Conclusion

The research sought to evaluate customer sentiment, predict business viability, and inform strategic decisions for the five fast-food restaurants by analysing online reviews.

6.1 Successes of the study

The study successfully employed a combination of quantitative and qualitative methods to gain a comprehensive understanding of customer perceptions across various dimensions such as food quality, service efficiency, dining experience, and customer service. By using machine learning models like SVM, CNN, RNN, Random Forest, and Naive Bayes, we were able to accurately classify customer sentiments into positive, neutral, and negative categories. The SVM model emerged as the most effective in this context, providing high accuracy and balanced performance across all sentiment classes.

The qualitative analysis of key themes within the reviews revealed valuable insights into customer priorities and concerns. Village Burger, for example, stood out for its superior food quality and efficient service, while Lucky's Burger & Brew Marietta was noted for its exceptional dining experience. These insights offer actionable information for each restaurant, guiding them towards areas of improvement and reinforcing their strengths.

Furthermore, the integration of topic modelling provided a nuanced understanding of the factors driving customer satisfaction or dissatisfaction. This dual approach not only validated the sentiment classification but also provided a deeper layer of context that pure sentiment analysis might overlook.

6.2 Limitations of the Study

Despite its successes, the study had several limitations that should be acknowledged:

1. **Data Limitations**: The study relied on the availability and quality of online reviews, which may not represent the entire customer base. Some restaurants had fewer reviews, which could have influenced the robustness of the results. Additionally, the reviews are inherently subjective and may be influenced by individual biases, which the models might not fully account for.
2. **Model Limitations**: While SVM performed exceptionally well, the CNN and RNN models struggled with certain sentiment categories, particularly neutral sentiments. This could be due to the complexity of sentiment analysis, where neutral sentiments often lack clear markers compared to positive or negative sentiments. The Naive Bayes model's performance also highlighted its limitations in handling complex text data, as it underperformed in comparison to the other models (Rennie, et al., 2003).
3. **Contextual Understanding**: Although topic modelling provided insights into key themes, it does not capture the full context of customer reviews. For example, the same word or phrase might have different implications depending on the review's context, which the models might not fully grasp. This limitation suggests that a more context-aware model could enhance future analysis (Hui, 2019).
4. **Generalisation**: The results of this study are specific to the selected five fast-food restaurants and may not be generalisable to other restaurants or food industries. Different regions, customer demographics, and competitive landscapes could yield different results, limiting the study’s applicability to other contexts.

6.3 Opportunities for Future Work

The limitations of this study open up several avenues for future research:

1. **Enhanced Data Collection**: Future studies could aim to collect a more extensive and diverse dataset, including reviews from different regions or platforms, to provide a more comprehensive analysis. Incorporating other data sources such as social media, customer surveys, and direct feedback could also enrich the dataset.
2. **Improving Model Performance**: There is potential to enhance the performance of the CNN and RNN models by experimenting with more sophisticated architectures or combining them with attention mechanisms that better capture the context and nuances of sentiment in text data. Additionally, fine-tuning models with domain-specific knowledge could improve their ability to differentiate between subtle sentiment classes (Feng, 2023).
3. **Context-Aware Analysis**: Future research could explore context-aware natural language processing models, such as transformers, to better understand the nuances of customer reviews. These models could provide deeper insights into how certain words or phrases contribute to the overall sentiment, considering the full context of each review (Hui, 2019)..

6.4 Final Thoughts

This study provides a meaningful contribution to understanding customer sentiments in the fast-food industry by combining advanced machine learning techniques with qualitative analysis. While there are limitations, the findings offer valuable insights that can guide strategic decisions for the participating restaurants. By addressing the study's limitations and exploring the identified opportunities for future work, researchers and practitioners can further enhance the effectiveness of sentiment analysis in driving business success.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y. & Zheng, X. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org. Available at: https://www.tensorflow.org/

Aureliano-Silva, L., Spers, E. E. & Leung, X., 2021. The Effect of Online Reviews on Restaurant Visit Intentions: Applying Signaling and Involvement Theories. *Journal of Hospitality and Tourism Technology*.

AZN Sandwich Bar, 2024. AZN Sandwich Bar. [Online] Available at: https://atlanta.aznsandwichbar.com/ [Accessed 23 July 2024].

Bahtiar, S. A. H., Dewa, C. K. & Ahmad, L., 2023. Comparison of Naïve Bayes and Logistic Regression in Sentiment Analysis on Marketplace Reviews Using Rating-Based Labeling. *Journal of Information Systems and Informatics*, 5(3).

Beheshti, N., 2022. Cross Validation and Grid Search: Using sklearn's GridSearchCV on random forest model. *Medium*, 5 February.

Beuscart, J. S., Mellet, K. & Trespeuch, M., 2016. Reactivity without Legitimacy? Online Consumer Reviews in the Restaurant Industry. *Journal of Cultural Economy*, 9(5), pp. 458-475.

Bhangale, A., 2023. Introduction to Topic Modelling with LDA, NMF, Top2Vec and BERTopic. *Medium*. Available at: https://medium.com/blend360/introduction-to-topic-modelling-with-lda-nmf-top2vec-and-bertopic-ffc362d4d44e [Accessed 12 August 2024].

Bishop, C.M., 2006. *Pattern recognition and machine learning.* New York: Springer.

Blei, D. M., Ng, A. Y. & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research,* 3, pp. 993-1022.

Boell, S.K. & Cecez-Kecmanovic, D., 2015. On being 'systematic' in literature reviews in IS. *Journal of Information Technology,* 30(2), pp.161-173.

Breiman, L., 1996. Bagging predictors. *Machine Learning,* 24(2), pp.123-140.

Breiman, L., 2001. Random forests. *Machine Learning,* 45(1), pp.5-32.

Broken Rice Media, 2024. Broken Rice Media. [Online] Available at: https://www.brokenricemedia.com/commercial [Accessed 23 July 2024].

Cambria, E., Schuller, B., Xia, Y. & Havasi, C., 2017. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems,* 28(2), pp.15-21.

Casalino, G., Del Buono, N. & Mencar, C., 2016. Non-negative matrix factorizations for intelligent data analysis. *In Non-Negative Matrix Factorizations for Intelligent Data Analysis.* Springer, pp. 1-22.

Cao, H., Naito, T. & Ninomiya, Y., 2008. Approximate RBF Kernel SVM and Its. *The 1st International Workshop on Machine Learning for Vision-based Motion Analysis.*

Çano, E. & Morisio, M., 2018. A Deep Learning Architecture for Sentiment Analysis. *In ICGDA '18 Proceedings of the International Conference on Geoinformatics and Data Analysis,* pp. 122-126. ACM.

Cheeseburger Bobby's, 2024. Cheeseburger Bobby's. [Online] Available at: https://order.myguestaccount.com/menu/5f497b4795b701a1168b45c5?from_locations=1&appid=5f495e75505ee9eb097b2404#burgers [Accessed 23 July 2024].

Cheeseburger Bobby's, 2024. TripAdvisor. [Online] Available at: https://www.tripadvisor.com/Restaurant_Review-g9743413-d4606849-Reviews-Cheeseburger_Bobby_s-Peachtree_Corners_Georgia.html [Accessed 23 July 2024].

Chawla, N.V., Bowyer, K.W., Hall, L.O. & Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, pp.321-357.

Chen, S.F. & Goodman, J., 1996. An Empirical Study of Smoothing Techniques for Language Modeling. *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL),* pp. 310-318.

Chollet, F. (2015). Keras: The Python Deep Learning library. Available at: https://keras.io

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine Learning,* 20(3), pp.273-297.

Cutler, D.R., Edwards Jr., T.C., Beard, K.H., Hess, K.L., Gibson, J. & Lawler, J.J., 2012. Random forests for classification in ecology. *Ecology,* 93(2), pp.332-340.

Cunha, V. C., Magoni, D., Inácio, P. R. M. & Freire, M. M., 2022. Advanced Information Networking and Applications. *In: Impact of Self C Parameter on SVM-based Classification of Encrypted Multimedia Peer-to-Peer Traffic,* pp. 180-193.

Dalianis, H. (2018). *Evaluation Metrics and Evaluation.* In: *Clinical Text Mining*. Springer, Cham.

Data Camp, 2023. Random Forest Classification with Scikit-Learn. [Online] Available at: https://www.datacamp.com/tutorial/random-forests-classifier-python#:~:text=we%20are%20using%3A-,n_estimators,-%3A%20the%20number%20of [Accessed 1 August 2024].

Doan, T. & Kalita, J., 2016. Sentiment Analysis of Restaurant Reviews on Yelp with Incremental Learning. *Proceedings of the 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA,* pp. 697-700.

Explosion. (2024). *spaCy 3.0 Documentation.* [Online]. Available: https://spacy.io/

Falkner, S., Klein, A. & Hutter, F., 2018. BOHB: Robust and efficient hyperparameter optimization at scale. *Proceedings of the 35th International Conference on Machine Learning (ICML 2018),* pp.1437-1446.

Farm Burger, 2024. Farm Burger. [Online] Available at: https://order.myguestaccount.com/menu/5f497b4795b701a1168b45c5?from_locations=1&appid=5f495e75505ee9eb097b2404#burgers [Accessed 23 July 2024].

Farhadloo, M. & Rolland, E., 2016. Fundamentals of Sentiment Analysis and Its Applications. *In: C. C. Aggarwal, ed. Machine Learning for Text.* Cham: Springer, pp. 1-36.

Feng, B., 2023. Deep learning-based sentiment analysis for social media: A focus on multimodal and aspect-based approaches. *Chongqing, International Conference on Machine Learning and Automation.*

Fraj, M. B., 2017. In Depth: Parameter tuning for Random Forest. *Medium,* 21 December.

Goyal, C., 2021. Part 5: Step by Step Guide to Master NLP – Word Embedding and Text Vectorization. *Analytics Vidhya.* Available at:

https://www.analyticsvidhya.com/blog/2021/06/part-5-step-by-step-guide-to-master-nlp-text-vectorization-approaches/ [Accessed 18 August 2024].

Goyal, C., 2021. Part 15: Step by Step Guide to Master NLP – Topic Modelling using NMF. *Analytics Vidhya.* Available at: https://www.analyticsvidhya.com/blog/2021/06/part-15-step-by-step-guide-to-master-nlp-topic-modelling-using-nmf/ [Accessed 2 August 2024].

Goyal, M., 2024. Dense In Deep Learning. *Code 360*, 27 March.

Gwinnett Daily Post, 2024. Gwinnett Daily Post. [Online] Available at: https://www.gwinnettdailypost.com [Accessed 23 July 2024].

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), pp.357-362.

Harshi, H., 2021. Convolutional Neural Network. *Analytics Vidhya*, 28 June. Available at: https://medium.com/analytics-vidhya/cnn-32fbc1706a89 [Accessed 18 August 2024].

Hochreiter, S. & Schmidhuber, J., 1997. Long Short-Term Memory. *Neural Computation*, 9(8), pp.1735-1780.

Hsu, C.-W., Chang, C.-C. & Lin, C.-J., 2016. A Practical Guide to Support Vector Classification. *ACM Digital Library*, 19 May.

Huda, A. S. M., Shoikot, M., Hossain, A. & Ila, J. I., 2019. An Effective Machine Learning Approach for Sentiment Analysis on Popular Restaurant Reviews in Bangladesh. *Newcastle, Newcastle University.*

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), pp.90-95.

Huilgol, P., 2024. Quick Introduction to Bag-of-Words (BoW) and TF-IDF for Creating Features from Text. *Analytics Vidhya.* Available at: https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/ [Accessed 18 August 2024].

IBIS World, 2024. Fast Food Restaurants in the US - Number of Businesses. *IBIS World,* 17 April.

Jain, A., 2024. All about GridSearch Cross validation. *Medium,* 5 February.

Joachims, T., 1998. Text Classification with Support Vector Machines: Learning with Many Relevant Features. *Proceedings of the European Conference on Machine Learning (ECML),* pp. 137-142.

John, S.M. & Kartheeban, K., 2019. Sentiment Scoring and Performance Metrics: Examination of Various Supervised Classifiers. *International Journal of Innovative Technology and Exploring Engineering (IJITEE),* 9(2S2), pp. 1120-1126. doi:10.35940/ijitee.B1111.1292S219.

Jones, A.B., 2018. Sentiment analysis of reviews: Text Pre-processing. *Medium,* 7 February. Available at: https://medium.com/@annabiancajones/sentiment-analysis-of-reviews-text-pre-processing-6359343784fb [Accessed 24 August 2024].

Jurafsky, D. & Martin, J.H., 2020. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 3rd ed. Prentice Hall.

Kalita, D., 2024. What is Recurrent Neural Networks (RNN)?. *Analytics Vidhya.* Available at: https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/ [Accessed 18 August 2024].

Kibe, K., 2024. Topic Modeling Using Latent Dirichlet Allocation (LDA). *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2023/02/topic-modeling-using-latent-dirichlet-allocation-lda/ [Accessed 12 August 2024].

Kowsari, K., Heidarysafa, M., Brown, D.E., Jafari Meimandi, K. & Barnes, L.E., 2018. RMDL: Random Multimodel Deep Learning for Classification. *Proceedings of the 2nd International Conference on Information System and Data Mining (ICISDM '18),* pp.19-28. Available at: https://doi.org/10.1145/3206098.3206111 [Accessed 24 August 2024].

Kuhn, M. & Johnson, K., 2013. *Applied predictive modeling.* Springer.

Kushniruk, H. & Rutynskyi, M., 2022. Post-COVID-19 perspectives – The case of Lviv, Ukraine. *In: Ana Pinto Borges, et al, ed. Gastronomy, Hospitality, and the Future of the Restaurant Industry: Post-COVID-19 Perspectives.* Hershey, PA: IGI Global, p. 24.

Lasek, A., Cercone, N. & Saunders, J., 2016. Restaurant Sales and Customer Demand Forecasting: Literature Survey and Categorization of Methods, s.l.: Springer.

Lemaître, G., Nogueira, F. & Aridas, C. K., 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research,* 18(17), pp.1–5. Available at: https://imbalanced-learn.org/stable/

LeCun, Y., Bengio, Y. & Hinton, G., 2015. Deep learning. *Nature,* 521(7553), pp.436-444.

Lee, D.D. & Seung, H.S., 1999. Learning the parts of objects by non-negative matrix factorization. *Nature,* 401(6755), pp.788-791.

Liu, B. (2012). *Sentiment analysis and opinion mining.* Morgan & Claypool Publishers.

Little, R.J.A. & Rubin, D.B., 2019. *Statistical Analysis with Missing Data.* 3rd ed. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley. DOI: 10.1002/9781119482260.

Lucky's Burger & Brew, 2024. Lucky's Burger & Brew. [Online] Available at: https://www.luckysburgerandbrew.com/marietta-menu [Accessed 23 July 2024].

Lucky's Burgers & Brew, 2024. TripAdvisor. [Online] Available at: https://www.tripadvisor.co.uk/Restaurant_Review-g35235-d2570068-Reviews-Lucky_s_Burgers_Brew-Roswell_Georgia.html [Accessed 23 July 2024].

Manning, C.D., Raghavan, P. & Schütze, H., 2008. *Introduction to Information Retrieval.* Cambridge University Press.

Matplotlib Documentation, 2024. Using Matplotlib. [online] Available at: https://matplotlib.org/stable/users/index.html [Accessed 12 Aug. 2024].

McCallum, A. & Nigam, K., 1998. A Comparison of Event Models for Naive Bayes Text Classification. *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*, pp. 41-48.

McKinney, W., 2010. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference (SciPy 2010)*, pp.51-56.

Medhat, W., Hassan, A. & Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), pp.1093-1113.

Mikolov, T., Chen, K., Corrado, G. & Dean, J., 2013. Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J., 2013. Distributed Representations of Words and Phrases and their Compositionality. *In Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS 2013)*, Lake Tahoe, NV, USA, 5-10 December 2013. Curran Associates Inc., pp. 3111–3119. Available at: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf [Accessed 10 August 2024].

Müller, K. R., Mika, S., Rätsch, G., Tsuda, K. & Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2), pp.181-201. doi:10.1109/72.914517.

Mueller, A.C., 2020. WordCloud. [online] Available at: https://github.com/amueller/word_cloud [Accessed 17 August 2024].

Nabi, J., 2019. Recurrent Neural Networks (RNNs). *Medium*, 11 July.

NLTK Documentation, 2024. *Natural Language Toolkit.* [online] Available at: https://nltk.org [Accessed 16 Aug. 2024].

NumPy Documentation, 2024. *NumPy fundamentals.* [online] Available at: https://numpy.org/devdocs/user/basics.html [Accessed 11 Aug. 2024].

Pandian, S., 2024. K-Fold Cross Validation Technique and its Essentials. *Analytics Vidhya*, 5 August.

Pandas Documentation, 2024. *pandas documentation.* [online] Available at: https://pandas.pydata.org/docs/index.html [Accessed 19 Aug. 2024].

Pascanu, R., Mikolov, T. & Bengio, Y., 2013. On the difficulty of training Recurrent Neural Networks. *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1310-1318.

Peddireddi, Y., 2023. Topic Modelling in Natural Language Processing. *Analytics Vidhya*. Available at: https://analyticsvidhya.com/blog/2021/05/topic-modelling-in-natural-language-processing/ [Accessed 16 August 2024].

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, É., 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, pp.2825–2830. Available at: https://scikit-learn.org/stable/

Pennington, J., Socher, R. & Manning, C. D., 2014. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532-1543.

Premanand, S., 2023. The A-Z guide to Support Vector Machine. *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/ [Accessed 18 August 2024].

Probst, P., Wright, M.N. & Boulesteix, A.L., 2019. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3), p.e1301.

Python Programmer, 2023. Evaluation Metrics for Classification. *Medium*. Available at: https://medium.com/@mlmind/evaluation-metrics-for-classification-fc770511052d [Accessed 25 Aug 2024].

Rajaraman, A. & Ullman, J.D., 2011. *Mining of Massive Datasets.* Cambridge: Cambridge University Press.

Ray, S., 2024. Naive Bayes Classifier Explained: Applications and Practice Problems of Naive Bayes Classifier. *Analytics Vidhya.* Available at: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/ [Accessed 18 August 2024].

Rennie, J.D.M., Shih, L., Teevan, J. & Karger, D.R., 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 616-623.

Robertson, S. E., 2004. Understanding Inverse Document Frequency: On Theoretical Arguments for IDF. *Journal of Documentation*, 60(5).

Ruder, S., Peters, M. E., Swayamdipta, S. & Wolf, T., 2019. Transfer learning in natural language processing. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pp. 15-18.

Saini, A., 2024. An Introduction to Random Forest Algorithm for beginners. *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/ [Accessed 18 August 2024].

Saumyab271, 2024. Stemming vs Lemmatization in NLP: Must-Know Differences. *Analytics Vidhya*. [online] Available at: https://analyticsvidhya.com/blog/2022/06/stemming-vs-lemmatization-in-nlp-must-know-differences/ [Accessed 11 Aug. 2024].

Schölkopf, B. & Smola, A.J., 2002. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT Press.

Schott, M., 2019. Random Forest Algorithm for Machine Learning. *Medium*, 25 April.

Scikit-learn Documentation, 2024. Support Vector Machines. [online] Available at: https://scikit-learn.org/stable/modules/svm.html [Accessed 13 Aug. 2024].

Seaborn Documentation, 2024. An introduction to seaborn. [online] Available at: https://seaborn.pydata.org/tutorial/introduction.html [Accessed 14 Aug. 2024].

Seong, S., 2024. Random Forest with Grid Search. *Medium,* 13 March.

Sennrich, R., Haddow, B. & Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* pp. 1715-1725.

Siddhartha, B.S., 2021. An Interpretation of Lemmatization and Stemming in Natural Language Processing. *Journal of University of Shanghai for Science and Technology*. Available at: https://www.researchgate.net/publication/348306833 [Accessed 24 Aug. 2024].

Srivastava, T., 2024. Tuning the parameters of your Random Forest model. *Analytics Vidya*, 25 June.

Sruthi, E. R., 2024. Understanding Random Forest Algorithm With Examples. *Analytics Vidhya*, 21 August.

Sumbati, L., 2024. Deep Learning (Part 3). CONVOLUTION NEURAL NETWORKS (CNNs). *Medium,* 14 April.

Sumeet, K. A. (2024). Metrics to Evaluate your Classification Model to take the Right Decisions. *Analytics Vidhya.* Retrieved from https://analyticsvidhya.com

Sumbati, L., 2024. Deep Learning (Part 3). CONVOLUTION NEURAL NETWORKS(CNNs). *Medium*, 14 April.

Suri, M., 2022. A Dummy's Guide to Word2Vec. *Medium.* Available at: https://medium.com/@manansuri/a-dummys-guide-to-word2vec-45644f3c673 [Accessed 21 Aug. 2024].

Svetnik, V., Liaw, A., Tong, C. & Culberson, J.C., 2003. Random forest: A classification and regression tool for compound classification and QSAR modeling. *Journal of Chemical Information and Computer Sciences*, 43(6), pp.1947-1958.

Tang, D., Qin, B. & Liu, T., 2015. Document modeling with gated recurrent neural network for sentiment classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp.1422-1432.

The National Restaurant Association, 2019. *Restaurant Industry 2030- Actionable Insights for the Future,* Washington: Restaurant.org.

Tukey, J.W., 1977. *Exploratory Data Analysis.* Reading, MA: Addison-Wesley.

Tyagi, K., 2023. SVM Hyperparameter Tuning using GridSearchCV | ML. *Geek4Geeks*, 11 January.

Uysal, A. K. & Gunal, S., 2014. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1), pp.104-112.

Verma, K., 2022. Using CountVectorizer to Extracting Features from Text. *GeeksforGeeks.* Available at: https://geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/ [Accessed 16 August 2024].

Village Burger, 2024. Village Buger. [Online] Available at: https://order.online/store/village-burger-cumming-625077/?hideModal=true&pickup=true&utm_source=sdk&visitorId=190ff072641936ade [Accessed 23 July 2024].

Warsi, K. A. & Syeedun, N., 2005. Food retailing: Fast Food Industry. *SSRN Electronic Journal.*

Waskom, M., 2021. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60), 3021.

Webster, J. & Watson, R.T., 2002. Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, 26(2), pp.xiii-xxiii.

Yadav, A. & Vishwakarma, D. K., 2021. Sentiment analysis using deep learning architectures: A review. *Artificial Intelligence Review,* 54(6), pp.4555-4625.

Ying, X., 2019. An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series,* 1168(2).

Young, T., Hazarika, D., Poria, S. & Cambria, E., 2018. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), pp.55-75.

Zhang, H., 2004. The Optimality of Naive Bayes. *Proceedings of the Seventeenth International Conference on Pattern Recognition (ICPR),* pp. 823-826.

Zhang, X., Zhao, J. & LeCun, Y., 2015. Character-level Convolutional Networks for Text Classification. *Proceedings of the 29th International Conference on Neural Information Processing Systems (NeurIPS),* pp. 649-657.

Zhou, C., Sun, C., Liu, Z. & Lau, F., 2015. A C-LSTM neural network for text classification. *arXiv preprint arXiv:1511.08630.*