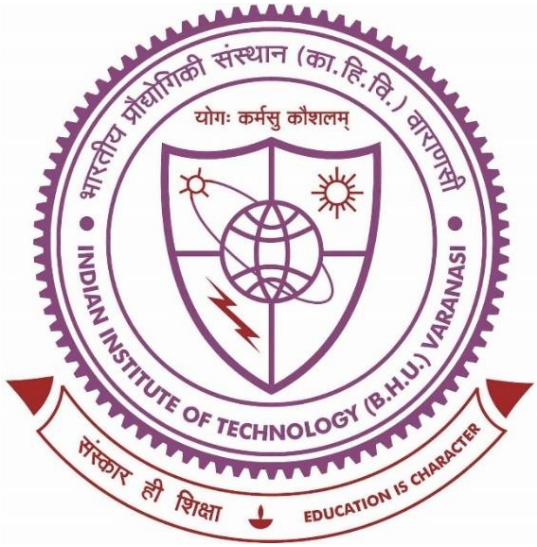
ENRICHING CUSTOMER EXPERIENCE THROUGH SENTIMENT ANALYSIS IN THE BANKING, FINANCIAL SERVICES AND INSURANCE SECTOR: A CASE OF NATIONAL BANK



Thesis submitted in partial fulfillment

for the Award of

MASTER OF TECHNOLOGY

in

DECISION SCIENCES AND ENGINEERING

by

MAHESH PURBIA

DEPARTMENT OF MECHANICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY
(BANARAS HINDU UNIVERSITY)
VARANASI – 221005

Roll No: 21222004 ***Session: 2022-23***

Indian Institute of Technology (BHU)

Department of Mechanical Engineering

Varanasi

CERTIFICATE

It is certified that the work contained in the thesis entitled as **"Enriching Customer Experience through Sentiment Analysis in the Banking, Financial Services and Insurance Sector: A Case of National Bank " By MAHESH PURBIA (Roll No: 21222004**) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree. Candidate has owned the onus of the originality of his research work and plagiarism**.**

Prof. P. Bhardwaj

Supervisor

DECLARATION BY THE CANDIDATE

I, **MAHESH PURBIA**, certify that the work embodied in this thesis is my own bonafide work and carried out by me under the supervision of **Prof. P. Bhardwaj** from **January 2022** to **June 2023**, at the Mechanical Engineering Department, Indian Institute of Technology (BHU), Varanasi. The matter embodied in this thesis has not been submitted for the award of any other degree/diploma. I declare that I have faithfully acknowledged and given credits to the research workers wherever their works have been cited in my work in this thesis. I further declare that I have not willfully copied any other's work, paragraphs, text, data, results, etc., reported in journals, books, magazines, reports dissertations, theses, etc., or available at websites and have not included them in this thesis and have not cited as my own work. I own whole responsibility of originality of this work and plagiarism.

Date:

Place: Varanasi MAHESH PURBIA

CERTIFICATE BY THE SUPERVISOR

It is certified that the above statement made by the student is correct to the best of my knowledge.

Prof. P. Bhardwaj
Supervisor
Mechanical Engineering Department
Indian Institute of Technology (BHU)
Varanasi

Prof. Santosh Kumar

Head of Department of Mechanical Engineering
Indian Institute of Technology (BHU)
Varanasi - 221005

COPYRIGHT TRANSFER CERTIFICATE

Title of the Thesis: *Enriching Customer Experience through Sentiment Analysis in the Banking, Financial Services and Insurance Sector: A Case of National Bank*

Name of the Student: MAHESH PURBIA

COPYRIGHT TRANSFER

The undersigned hereby assigns to the Indian Institute of Technology (Banaras Hindu University) Varanasi all rights under copyright that may exist in and for the above thesis submitted for the award of the *Master of Technology in Decision Sciences and Engineering.*

Date:

Place: Varanasi MAHESH PURBIA

Note: However, the author may reproduce or authorize others to reproduce material extracted verbatim from the thesis or derivative of the thesis for author's personal use provided that the source and the Institute's copyright notice are indicated.

ACKNOWLEDGEMENT

It is indeed my proud privilege to express my deep sense of gratitude, respect, indebtedness and sincere regard to my supervisor **Prof. P. Bhardwaj**, Department of Mechanical Engineering for his excellent supervision, valuable guidance, motivation, encouragement, and helpful advice throughout the course of this study. Without his valued suggestions and support, the research reported in the thesis could not have been completed. It has been an enriching scientific experience and I would like to thank him for providing an opportunity to work under his guidance.

I am thankful to **Prof. Santosh Kumar**, Head of the Department of Mechanical Engineering, for his encouragement in this work.

I want to express my sincere gratitude and appreciation to **Mr. Pawan Kumar Agarwal** (**AVP Analytics Client Insight**) for dedicating valuable time, offering insightful guidance, and providing immense help throughout the entire process. Their expertise and support were instrumental in the successful completion of my research.

I would like to thank **Prof. Anil Kumar Agrawal**, **Dr. Ajinkya N. Tanksale, Dr. Cherian Samuel, Dr. Lakshay and Dr. Saurabh Pratap** for their continued support and fruitful suggestions throughout my thesis work.

I sincerely thank the technical staff, **Mr. Anil Kumar Singh and Mr. Rajendra Prasad, and office staff Mr. Anil**, who helped me in various ways during this research work.

I am highly obliged to thank my seniors, especially Mr. Ankit Chouksey and Mr. Meghavatu Krishna Prasanna Naik, for their affection and support during my research. I have been blessed with a friendly and cheerful group of fellow mates. I sincerely thank Mr. Himanshu, Mr. Shubham, Mr. Jasvindar, Mr. Eshwar, Mr. Sourav, and Mr. Manideep. They directly or indirectly supported my research work. Their companionship and lively discussions inside and outside the simulation and optimization laboratory were great sources of inspiration.

And above all, I owe a special thanks to my entire family, especially my parents and brother for all their encouragement and support throughout and for providing me with the opportunity to be where I am today.

Space does not allow me to mention each person by name. I am extremely grateful and obliged for sparing their valuable time and enabling me with sufficient knowledge, motivation, and moral support to successfully complete the project.

Finally, I bow my head humbly before the almighty God, Shiva. Without whose blessings, this work would have been impossible.

Date:

Place: Varanasi MAHESH PURBIA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AUC	Area under the curve
BFSI	Banking, Financial Services and Insurance
BOW	Bag of Words
TF IDF	Term Frequency and Inverse Document Frequency
CRM	Customer Relationship Management
FN	False Negative
FP	False Positive
LDA	Latent Dirichlet Allocation
ML	Machine Learning
NLP	Natural Language Processing
POS	Part-of-speech
ROC	Receiver operating characteristic
SMOTE	Synthetic Minority Over-sampling Technique
TN	True Negative
TP	True Positive
VADER	Valence Aware Dictionary for Sentiment Reasoning

ABSTRACT

Sentiment analysis is crucial for gaining insights into customer feedback, opinions, regarding products and services. This research project aims to develop an effective sentiment analysis model for bank call center transcribed data. The objective is to automate the classification of customer sentiments into negative, positive or neutral categories based on the recorded conversations.

Research was carried out in three phases. In Phase 1, lexicon-based methods, including VADER, SentiWordNet, and TextBlob, were employed to generate sentiment labels for the call conversation data. A majority count approach was utilized to determine the final sentiment label, allowing insights to be gained into the performance and limitations of lexicon-based techniques.

In Phase 2, the focus was on developing a machine learning model for sentiment classification using the labeled transcribed data from Phase 1. Different text vectorization methods, such as BOW and TF IDF, were explored, and the class imbalance issue was addressed using the SMOTE algorithm. Several Machine Learning algorithms, including Logistic Regression, Linear SVC, and XG Boost Classifier, were evaluated to identify the best-performing model for the dataset. Hyperparameter tuning was performed to optimize the model's performance and generalization capabilities. Furthermore, an ensemble voting classifier was built to leverage the strengths of individual models and improve the overall performance.

In Phase 3, the emphasis is on model deployment. The saved text vectorized model and trained classifier model are loaded for practical usage. New input data undergoes data pre-processing, including cleaning and formatting, followed by text vectorization to transform the

text into numerical representations. The ML model is then utilized to predict sentiment labels for the processed input data. Finally, the numerical labels are mapped back to sentiment tags or categories for a more straightforward interpretation and analysis of the predicted sentiments.

The results of ensemble were found with accuracy as 0.89, precision as 0.88, recall as 0.89, and f1 score as 0.88. Also, the ROC-AUC scores demonstrate the classifier's ability to distinguish between sentiment classes was 0.82 for negative sentiment while 0.87 for neutral and 0.86 for positive sentiments. The micro-average ROC AUC of 0.9699 and a macro-average ROC AUC of 0.8539 was achieved for ensemble classifier indicating its effectiveness in classifying sentiments over all classes.

The developed sentiment analysis model offers a practical solution for banks and financial institutions to gain actionable insights from customer interactions. By automating sentiment classification, banks can improve customer experience, identifying opportunities for improvement and making choices based on data to maximize product offerings and customer service initiatives.

By exploring several methods, including both lexicon-based and machine-learning techniques, this study adds to the area of sentiment analysis. The performance of the developed model is assessed using real-world call center data pertaining to a nationalized bank, highlighting its capability to accurately classify customer sentiments. By examining various procedures and leveraging both lexicon-based and machine learning methods, this research expands the understanding and effectiveness of sentiment analysis in practical contexts, particularly in customer sentiment classification.

Keywords: sentiment analysis, lexicon-based methods, machine learning, text vectorization, model deployment.

CHAPTER 1

INTRODUCTION

With intensifying competition and growing customer expectations, organizations are striving to deliver exceptional customer experiences. The success of the customer-business connection is contingent on a number of elements, including the product or service quality, the reputation of the brand name or the firm itself, and most importantly, customer trust. Building and maintaining customer trust is an essential part in developing customer loyalty toward the firm(Too et al., 2001; Zboja & Voorhees, 2006). In order to expand their customer base, companies are increasingly focusing on gathering customer feedback and leveraging it to make data-driven decisions that align with customer preferences. So is the Banking, Financial Services and Insurance (BFSI) sector experiencing a significant shift towards customer-centricity in recent years. Understanding customer sentiment and feedback has become crucial for organizations to improve their services i.e., identify areas for enhancement, and cultivate long-term customer relationships.

1.1 Customer Experience in the BFSI Sector

In today's rapidly evolving business landscape, organizations are facing intensified competition and an array of complex challenges. The demand for globalization and the ever-changing business environment requires companies to adapt and continually enhance their products and systems. This adaptation is crucial to improve service quality and maintain a competitive edge in the market(Yasin et al., 2004).

A customer can be described as an individual who receives, consumes, or purchases products and services from a company. The opinions and feedback provided by customers after

experiencing these offerings are invaluable in effectively managing and aligning business operations to meet their needs. Such insights enable businesses to make informed decisions and tailor their products and services to serve their customers better. Customer experience is crucial to the success of a business, and the Banking, Financial Services, and Insurance (BFSI) sector is no exception.

Understanding the impact of service quality on an organization's financial outcomes is a top priority today. It directly influences profitability and other financial indicators, making it essential for business success. By improving service quality, organizations can enhance customer satisfaction, retention, and overall financial performance (Zeithaml et al., 1996). By focusing on improving customer experience and service quality, organizations in the BFSI sector aim to attract and retain customers, increase customer satisfaction, and ultimately drive financial success. The ability to effectively analyze customer sentiment and feedback become essential in achieving these objectives and making informed decisions to enhance offerings and customer satisfaction.

1.2 Importance of Sentiment Analysis in the BFSI Sector

Customer feedback is a valuable resource for organizations in the banking, financial services, and insurance (BFSI) sector to improve their offerings and enhance customer experiences. In recent years, businesses have recognized the significance of customer feedback as a valuable source of insights to improve products, services, and customer experiences. By actively taking feedback through methods like surveys, online reviews, and monitoring social media, businesses can learn valuable information about what customers like, what worries them, and what they expect. This helps businesses make better decisions and enhance their products and services to cater the customer needs effectively. (Tara Ramroop, 2023). Analyzing this feedback enables enterprises to identify areas for improvement, make informed decisions, and implement changes to enhance offerings and customer satisfaction. With

customers increasingly expecting personalized and seamless interactions, providing exceptional customer service has become a key differentiator for banks and financial institutions.

The advancement of NLP approaches has led to the emergence of sentiment analysis as a powerful tool in the BFSI sector. Sentiment analysis automatically determines the text's polarity and categorizes it as positive, negative, or neutral. Organizations can gain a deeper understanding of customer sentiment by applying sentiment analysis to unstructured textual data from various sources like call transcriptions, online reviews, and social media.

Call centers serve as a significant touchpoint in the customer journey, where customers seek assistance, raise concerns, and express their opinions. Analyzing the sentiment of these interactions can offer valuable insights into customer satisfaction and enable the BFSI sector to enhance its customer experience strategies.

Customer service departments become the go-to point for customers when they encounter problems beyond their ability to resolve, leading to businesses investing a significant $1.3 trillion annually to handle a massive volume of 265 billion customer service calls (Dylan Azulay, 2019; Trips Reddy, 2017).

Traditionally, customer feedback in the BFSI sector has been collected through surveys and questionnaires, which provide valuable but limited insights. However, Sentiment analysis, a subfield of NLP, has emerged as a powerful tool for analyzing customer sentiment from unstructured textual data, automatically determining the text's polarity and categorizing it as positive, negative, or neutral.

NLP can evaluate call transcriptions, categorize conversation topics, and detect consumer sentiment. It is becoming more popular as a valuable tool for businesses to enhance decision-making processes and deliver better customer service. By leveraging sentiment

analysis in customer call centers, banks can gain a deeper understanding of customer sentiment and proactively address customer needs and concerns (Sophia Lam et al., 2019).

Furthermore, in the BFSI sector, sentiment analysis faces unique challenges because of the complex nature of financial language, the diversity of customer interactions, and the need for real-time analysis. Developing a practical sentiment analysis framework explicitly tailored to the BFSI sector requires sophisticated NLP techniques and an understanding of the domain-specific nuances.

1.3 Challenges in Implementing Sentiment Analysis in the BFSI Sector

Implementing sentiment analysis in the banking, financial services, and insurance (BFSI) sector presents unique challenges. These challenges arise from the complex nature of financial language, the diversity of customer interactions, and the need for real-time analysis. It is important to address these challenges to effectively leverage sentiment analysis and extract meaningful insights.

1.3.1 Restriction to Confidential Data

The BFSI sector handles sensitive customer data, making data privacy and security essential considerations when implementing sentiment analysis. Ensuring compliance with data protection regulations and maintaining high-security standards throughout the sentiment analysis process is of utmost importance to protect customer information and maintain trust.

Given the highly sensitive nature of customer data in the BFSI sector, access to this confidential information is typically restricted to mitigate privacy and security risks. Implementing Sentiment analysis within this context requires diligent handling and robust security measures to ensure the confidentiality of customer data. Implementing sentiment

analysis while adhering to data protection regulations and ensuring the confidentiality of customer data requires careful handling and robust security measures.

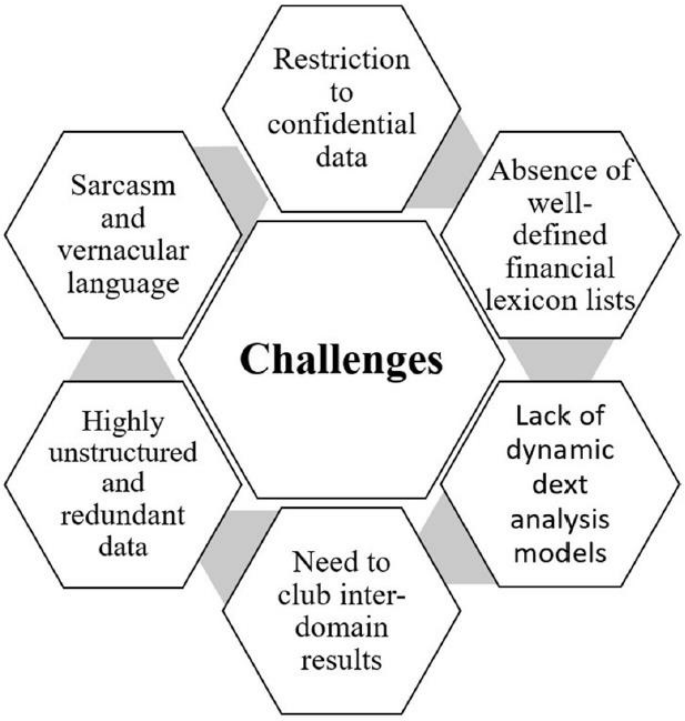


Figure 1.1 Major Challenges to Text Mining in Finance (Gupta et al., 2020)

1.3.2 Highly Unstructured and Redundant Data

In the BFSI sector, customer feedback is often unstructured and encompasses various sources such as emails, call transcripts, social media posts, and online reviews. Extracting valuable sentiment insights from this data is challenging due to redundancy and irrelevance. To address this, efficient data preprocessing techniques and filtering mechanisms are crucial to handle the volume and unstructured nature of the data, ensuring that relevant sentiment insights can be extracted effectively.

1.3.3 Absence of Well-Defined Financial Lexicon Lists

Unlike general sentiment analysis, sentiment analysis in the BFSI sector requires domain-specific knowledge and understanding of financial terms and concepts. Building

comprehensive financial lexicon lists that encompass the unique vocabulary and terminology of the BFSI sector can be a challenge. Without well-defined lexicons, sentiment analysis models may struggle to accurately interpret sentiment related to financial products, services, or events.

1.3.4 Lack of Dynamic Text Analysis Models

Sentiment analysis models need to continuously evolve and adapt to changing customer sentiment patterns and emerging linguistic trends. The BFSI sector experiences evolving customer expectations, market dynamics, and regulatory changes that can impact customer sentiment. Developing dynamic text analysis models that can adapt to these changes is crucial to maintain the accuracy and relevance of sentiment analysis results.

1.3.5 Sarcasm and Vernacular Language

Customers may express their sentiment using sarcasm, idiomatic expressions, or vernacular language, which can be challenging for sentiment analysis models. These models need to be trained to recognize and interpret such linguistic nuances accurately to avoid misclassification of sentiment.

1.3.6 Need to Club Inter-Domain Results

In the BFSI sector, sentiment analysis may involve analyzing customer feedback from various domains such as banking, insurance, and investment services. Combining and interpreting sentiment results across these different domains can be challenging due to variations in terminology, context, and customer expectations. To acquire a thorough picture of client sentiment in the BFSI industry, it is essential to develop approaches for combining cross-domain sentiment data and deriving holistic insights. The BFSI sector involves intricate financial terminology, jargon, and concepts. Sentiment analysis algorithms need to be trained and fine-tuned to accurately interpret and analyze customer sentiment in this specialized

domain. Ensuring the sentiment analysis model understands the context and meaning of financial terms is essential to avoid misinterpretations.

1.4 Sentiment analysis approaches

Sentiment analysis is essential in analyzing text to understand people's sentiments. It involves extracting and studying emotions and opinions expressed in written content. With the increasing use of social media, online reviews, and other text sources, sentiment analysis has become valuable for understanding public opinion, market research, and decision-making. This section explores different approaches used in sentiment analysis, including methods based on sentiment word dictionaries, machine learning and deep learning techniques.

There are several kinds of sentiment analysis methodologies. These approaches encompass a wide range of techniques and methodologies to analyze text and determine the sentiment expressed within it. Figure 1.2 provides an overview of these approaches, and each of them will be discussed in the following subsections.

1.4.1 Lexicon based approaches

Sentiment Analysis with Lexicon-based methods involves the utilization of pre-built sentiment word dictionaries. These dictionaries contain a collection of sentiment words along with their associated polarity, which can be neutral, negative, or positive. The underlying assumption is that the sentiment of a text can be inferred by aggregating the sentiment scores of the individual words within the text.

Lexicon-based approaches are utilized to categorize text into several sentiment categories, including negative, neutral, and positive. Alternatively, if desired, the classification can be limited to negative and positive sentiments. These approaches are considered

unsupervised learning techniques, as they don’t need labeled training data. Instead, they rely on the polarity of words in a given text to determine its overall sentiment.

The process involves using a predetermined dictionary or lexical resources to classify words into categorical labels (positive, negative, or neutral) or assign them numeric scores. The sentiment lexicons used in these approaches have predefined sentiment values assigned to words before they are used for sentiment analysis.

To analyze the sentiment of a document using a lexicon-based approach, an initial score, usually set to zero, is assigned. Each positive word encountered in the document increments the score, while each negative word encountered decrements the score. After considering all the words in the document, the overall score is assessed by comparing it to a threshold value. Based on this comparison, the document is classified as having negative, positive, or neutral sentiments, often referred to as the polarity of the document.

Lexicon-based approaches are simple and computationally efficient, as they rely on pre-existing sentiment word dictionaries. However, they may have limitations in capturing nuanced sentiments and handling context-specific or domain-specific expressions.

Lexicon-based approaches in sentiment analysis can be categorized into two main methods (i) Dictionary based and (ii) Corpus based, as described in the following subsection.

1.4.1.1 Dictionary based techniques

The dictionary-based approach relies on sentiment word dictionaries or lexicons that contain predefined words and their associated sentiment polarities, such as “*positive, negative, or neutral.”* The text is analyzed by matching the document's words against the sentiment lexicon entries. A sentiment score is then computed by considering the occurence of positive & negative words encountered in the text. This method is well known for its simplicity and computing efficiency, which makes it suitable for analyzing massive text volumes.

However, it may struggle with context-dependent sentiments and handling negations or sarcasm effectively.

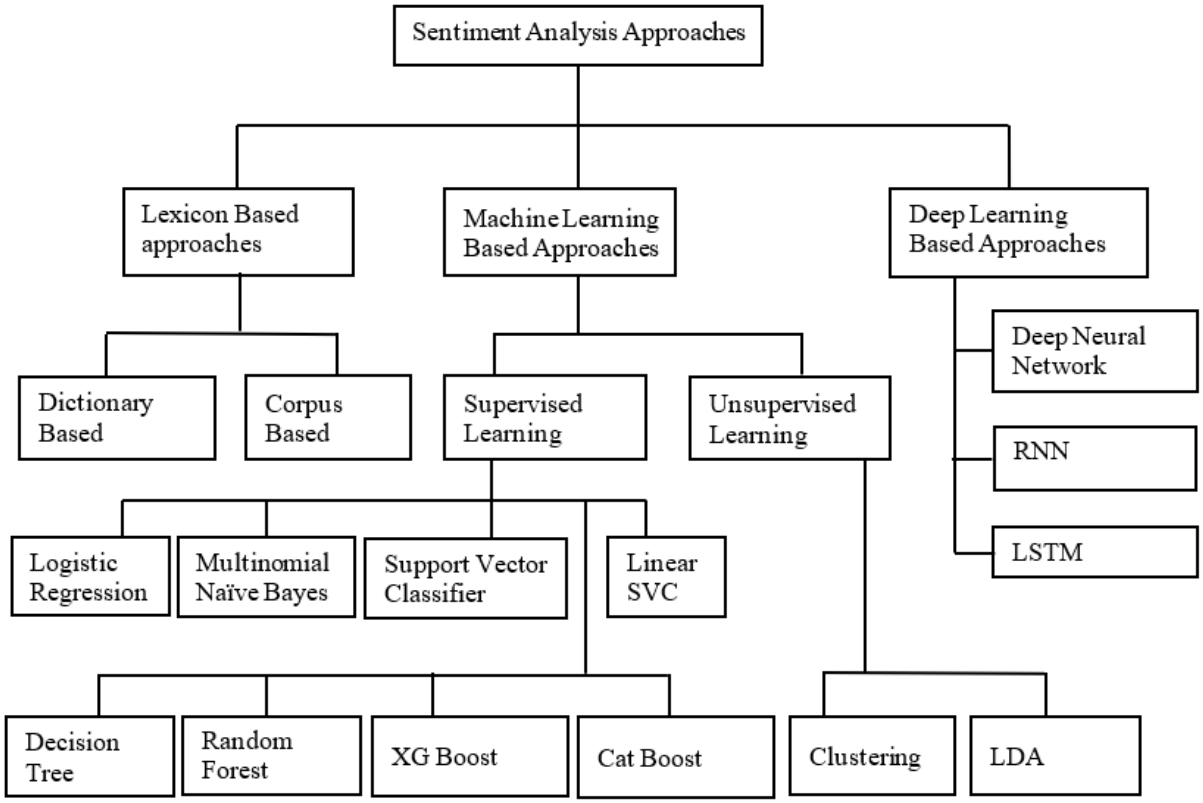


Figure 1.2 Sentiment Analysis Approaches

Sentiment Analysis with Valence Aware Dictionary and sEntiment Reasoner.

Hutto & Gilbert (2014) developed VADER, a rule-based sentiment analysis framework designed exclusively for social media sentiment analysis. The lexicon used in VADER incorporates not only individual words but also idioms and commonly used phrases. The lexicon used in VADER consists of over 9,000 lexical features. Through meticulous selection and validation, over 7,500 lexical characteristics with verified valence values were chosen. Each aspect in the lexicon is given a valence value ranging from -4 (very negative) to 4 (highly

positive), with 0 indicating neutrality. VADER takes into account the overall sentiment expressed by the combination of these lexical features.

Sentiment Analysis with TextBlob

TextBlob, due to Hazarika et al. (2020), is a Python library that offers a user-friendly API for basic NLP tasks TextBlob's sentiment function provides two properties: polarity and subjectivity. The polarity score represents the sentiment of the text and goes from -1 as negative to 1 as positive, with 0 indicating neutral sentiment. TextBlob's sentiment analysis function utilizes a pre-trained sentiment classifier to assign polarity scores to text.

Subjective sentences typically express personal opinion, emotion, or judgment, whereas objective sentences convey factual information without personal bias. The subjectivity score, which goes from 0 to 1, indicates how subjective the text is, with 0 and 1 describing objectivity and subjectivity, respectively.

Sentiment Analysis with SentiWordNet.

SentiWordNet due to (Sebastiani et al., 2010) is a sentiment lexicon that provides sentiment ratings to words based on their synsets (sets of synonymous words) in WordNet. SentiWordNet is a resource that provides sentiment ratings to WordNet synsets, indicating whether they are positive, negative, or neutral. SentiWordNet is a resource that provides sentiment ratings to WordNet synsets, indicating whether they are positive, neutral, or negative. The latest version, SentiWordNet 3.0, is built on WordNet 3.0 and provides automatic annotations for sentiment analysis purposes.

Every synset in SentiWordNet is assigned three numerical scores, (i) Pos(s), (ii) Neg(s), and (iii) Obj(s), which represents positive, negative, and objective aspects of the synset, respectively. The scores range from 0 to 1, and their sum is always 1.

1.4.1.2 Corpus based techniques

Corpus-based techniques are used in sentiment analysis to discover sentiment in a specific topic by examining a sizable corpus of text. In this method, an initial seed list of sentiment words is used as the basis for the analysis, which then searches a vast corpus for more sentiment words. By examining the context in which these words appear, the approach helps identify sentiment words with specific orientations or sentiments that are relevant to the given context.

Corpus-based techniques typically require a large labeled training dataset to analyze sentiment effectively. These methods rely on the statistical patterns observed in the corpus to determine sentiment. By analyzing the co-occurrence and frequency of words in the corpus, sentiment can be inferred(Medhat et al., 2014).

One common application of corpus-based sentiment analysis is using machine learning algorithms to train sentiment classifiers on labeled datasets. These classifiers learn patterns from the corpus and can then be utilized to categorize sentiments in new, unseen text.

The Corpus based approach can be performed using a statistical or semantic approach.

Statistical approach

The statistical approach in sentiment analysis uses statistical techniques to identify co-occurrence patterns and extract sentiment information from large annotated corpora. By analyzing word frequencies in positive and negative contexts, words can be categorized as positive, negative, or neutral. Additionally, it leverages the observation that similar opinion words tend to appear together, allowing for the determination of unknown word polarity through relative co-occurrence frequency, often using measures like Pointwise Mutual Information (PMI) by (Read, 2004). This approach finds applications in detecting review

manipulation and enhances sentiment analysis by capturing semantic orientation characteristics.

Latent Semantic Analysis (LSA) is a statistical method applied in sentiment analysis to analyze relationships between documents and their mentioned terms, revealing meaningful patterns(Landauer et al., 1998). LSA has been used to understand factors influencing helpfulness votes in reviews and to model semantic orientation characteristics. Combined with methods like semantic orientation inference from PMI (SO-PMI), it produces weighted features and advances sentiment analysis by providing a deeper understanding of sentiment expressions and semantic associations.

Semantic approach

In sentiment analysis, the Semantic method assigns sentiment values to words based on their semantic similarity. It employs principles that take into account word similarity, assigning similar sentiment scores to semantically related terms. WordNet, a lexical database, for example, contains semantic associations between words that can be used to compute sentiment polarity. Sentiment values for unknown words can be acquired by expanding an initial list of sentiment terms with WordNet synonyms and antonyms and assessing the sentiment polarity based on the relative count of positive and negative synonyms. The Semantic approach is often combined with statistical methods to improve sentiment analysis tasks(Brooke, 2009).

1.4.2 Machine-learning based approaches

ML approaches have been frequently used in sentiment analysis applications, allowing for automatic analysis and categorization of text input based on patterns and links discovered in training data. Two main types of ML approaches are used in sentiment analysis (i) supervised and (ii) unsupervised, and are discussed in the following subsection.

1.4.2.1 Supervised Learning:

Supervised ML based techniques are widely used in sentiment analysis to categorize text into predefined sentiment categories based on labeled training data. Here are some commonly used supervised ML algorithms for sentiment analysis:

Logistic Regression

Due to Cox & David R (1958), Logistic Regression uses a logistic function to estimate the relationship between the input features and one or more output variables based on the probability of belonging to a particular class. It is commonly used in text classification tasks due to its simplicity and interpretability.

Multinomial Naïve Bayes

Due to W. Zhang & Gao (2011), This is a probabilistic classifier based on Baye's theorem with the naïve assumption of a mutually independent pair of features, e.g., words. It works well with text classification tasks and is particularly suitable for cases where the features represent word counts or frequencies. The model calculates the probabilities of individual class given the input numerical text features and then categorizes the class with the highest probability as predicted sentiment.

Support Vector Classifier (SVC)

Due to Cortes & Vapnik (1995), The algorithm aims to find the optimal hyperplane that separates the classes in the higher-dimensional space that maximizes the margin between classes. Using different kernels, it can handle both linearly separable and non-linearly separable data.

Linear Support Vector Classifier

Due to Ladicky & Torr (2011), Linear SVC is a variant of SVC that uses a linear kernel. It creates a linear decision boundary in the original feature space. It is helpful with text categorization problems where there may be a lot of features (words).

Decision Tree Classifier

Due to Quinlan (1986), Decision Tree Classifier is a non-parametric classification algorithm that creates a tree-like structure by repeatedly splitting data depending on feature values. By following the path from the root node to the leaf nodes, decisions are made based on the feature values. Decision trees are intuitive and can capture complex relationships between text features.

Random Forest Classifier (Bagging)

Due to Breiman (2001), This is an ensemble of decision trees built using bootstrap aggregation, which involves sampling the training data with replacement and selecting random feature subsets. The final assessment is made by combining the outputs of the individual trees.

Extreme Gradient Boosting Algorithm

Due to T. Chen & Guestrin (2016), XGBoost Algorithm is a boosting algorithm that enhances the performance of weak learners, specifically decision trees, to form a strong learner. It combines the strengths of boosting and gradient descent techniques to create an ensemble of weak learners that iteratively improves the overall predictive performance. The algorithm builds decision trees sequentially, with each subsequent tree aiming to rectify the errors made by the previous trees.

Categorical Boosting Algorithm

Categorical Boosting Algorithm (CatBoost) due to Prokhorenkova et al., (2018) is another boosting algorithm explicitly designed for categorical features. It incorporates a

symmetric tree structure and utilizes gradient-based optimization algorithms. It handles categorical variables naturally without the need for pre-processing, such as one-hot encoding. CatBoost utilizes gradient boosting and provides fast and accurate predictions for text classification tasks.

1.4.2.2 Unsupervised ML approaches

Unsupervised ML approaches in sentiment analysis aim to discover patterns and structures within the data without relying on labeled examples. These techniques are handy when labeled data is unavailable. Some standard unsupervised machine learning techniques used in sentiment analysis include

Clustering

Due to Macqueen (1967), the Clustering algorithm is used to group similar data points based on their features or characteristics. In the context of sentiment analysis, clustering techniques can be employed to identify groups of texts that exhibit similar sentiment patterns. By analyzing the features or representations of the text data, clustering algorithms can automatically group texts into clusters without any prior knowledge of the sentiment categories.

Clustering techniques provide a way to discover inherent sentiment patterns in an unsupervised manner. They can reveal clusters of texts with similar sentiment, enabling further analysis of the sentiments expressed within each cluster.

Latent Dirichlet Allocation (LDA)

Due to David M. Blei et al. (2003), LDA is a probabilistic generative model commonly used in NLP and topic modeling. It is an unsupervised ML technique that aims to discover hidden topics in a group of texts. LDA assumes that documents are composed of a mixture of topics, and each topic is characterized by a distribution of words. By analyzing the word distributions across documents, LDA infers the underlying topic distribution and identifies the

latent topics present in the data. In the context of sentiment-analysis, LDA can be applied to uncover hidden topics related to sentiment and gain insights into the themes or concepts driving the expressed sentiment within a text corpus.

LDA is particularly useful when labeled examples are unavailable, as it does not require prior knowledge or annotations of sentiment categories. It enables researchers to explore the sentiment structure of the text data by revealing the latent topics that contribute to different sentiments. By understanding the underlying topics associated with the sentiment, LDA offers a valuable tool for sentiment analysis tasks, allowing for deeper insights and potentially discovering new dimensions or categories of sentiment in an unsupervised manner.

1.4.3 Deep Learning Approaches

Deep learning approaches have grown substantially in various fields, including sentiment analysis. The use of deep learning algorithms for sentiment analysis is examined in this section.

1.4.3.1 Deep Neural Network (DNN)

DNNs are highly effective machine learning models with multiple artificial neural network (ANN) layers between the input and output layers (Bengio, 2009; Schmidhuber, 2015). These hidden layers enable the network to learn hierarchical representations of the input data, making it capable of capturing complex patterns and relationships. DNNs have been widely used to extract informative features from text data and perform sentiment classification.

The deep layers of a DNN allow it to learn abstract representations of the input text, enabling the network to understand complex relationships between words and phrases. By leveraging these learned representations, DNNs can effectively capture the nuanced aspects of sentiment expressed in text, leading to more accurate sentiment analysis.

1.4.3.2 RNN

Recurrent neural networks (RNNs), a specialized type of neural network architecture, are created especially for processing sequential input. RNNs keep internal memory to accommodate sequential dependencies, unlike standard feedforward neural networks that process data in a single run. This memory allows the network to retain information about the previous inputs it has processed, enabling it to capture temporal dynamics in the data (Rumelhart et al., 1986).

In sentiment analysis, RNNs can appropriately capture the sentiment represented in a sentence by considering the order of the words. By maintaining a hidden state that summarizes the information from previous words, RNNs can capture contextual dependencies and grasp how the sentiment evolves throughout the sentence. This sequential modeling capability makes RNNs particularly suitable for sentiment analysis tasks, where the sentiment expressed in a sentence can be influenced by preceding words.

1.4.3.3 LSTM

RNNs' inability to capture long-term dependencies is a problem that is addressed by the Long Short Term Memory (LSTM) variation. Regular RNNs can suffer from the vanishing gradient problem, where gradients gradually becomes smaller over time and are difficult to detect in long-range dependencies. LSTMs overcome this issue by introducing memory cells and gating mechanisms (Hochreiter & Schmidhuber, 1997).

LSTMs have a more complex structure compared to regular RNNs. They incorporate memory cells with the capacity to store information over long sequences, enabling them to capture long-term dependencies in the data. Additionally, LSTMs have input, forget, and output gates that regulate the movement of information into, out of, and within the memory cells. This gating mechanism enables LSTMs to selectively retain and update information,

making them effective in capturing the sentiment-bearing words and their relationships in text data.

1.5 Evaluation Metrics for Multi Class Classification

In machine learning, problems that involve classifying data into more than two classes are referred to as "multi-class classification" problems. Here, the objective is to assign each instance to a single preset class. The success of a multi-class classification model's ability to correctly classify cases across different classes must be measured using the suitable assessment measures. Due to Grandini et al. (2020), some frequently used assessment metrics for multi-class classification are briefly explained in the following subsections. These metrics provide valuable insights into the performance of the model.

1.5.1 Confusion Matrix

In the evaluation of classification algorithms, many metrics are based on the Confusion Matrix. A square matrix called the confusion matrix is used to assess how well a classification model is working. The rows serve as the actual class names, while the columns serve as the predicted class labels, providing a visual depiction. The count or proportion of instances assigned to each class is shown in figure 3.1 in each matrix cell.

The confusion matrix is divided into four distinct components, namely (i) True Positive (TP); (ii) True Negative (TN); (iii) False Positive (FP), and (iv) False Negative (FN). TP is the number of positive cases predicted correctly. TN are all those negative cases that are predicted negatively. FP is the number of negative cases that were predicted positive. FN shows all those positive cases that are predicted as negative. Based on these cases, various metrics were proposed like accuracy, recall, precision, and F1-score. The following sub-sections explain each of these individually.

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 1.3 Confusion Matrix Representation

1.5.2 Accuracy

By computing the ratio of instances that were correctly classified to all of the model's predictions in the dataset, it determines how accurate the model's predictions are overall.

$$\text{Accuracy} = \frac{\mathbf{TP+TN}}{\mathbf{TP+TN+FP+FN}}$$

1.5.3 Precision

The Ratio of TP values to all the Predicted Positive Values (column total of predicted positives). It provides information on the percentage of predictions in the positive class that were actually positive.

In other words, Precision indicates how reliable the model's positive predictions are. A higher precision value signifies a lower rate of false positives.

$$\text{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FP}}$$

1.5.4 Recall (Sensitivity or True Positive Rate)

This expression is the fraction of TP cases to the overall positive cases in the data. It provides information on the percentage of all positive samples identified as positive by the classifier. A higher recall value indicates a lower rate of false negatives.

$$\text{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}}$$

1.5.5 Specificity (True Negative Rate)

This expression is the proportion of all negative cases that the classifier identified as negative.

$$\text{Specificity} = \frac{\mathbf{TN}}{\mathbf{TN+FP}}$$

$$\text{FPR} = 1\text{- Specificity}$$

1.5.6 F-1 Score

A balanced statistic that combines recall and precision into a single number is the F1-score. The F1 score is actually computed by taking the harmonic mean of precision and recall. It is typically more valuable than accuracy if we have an Imbalance class distribution.

$$\text{F1 Score} = \mathbf{2}\left(\frac{\boldsymbol{Precision \times Recall}}{\boldsymbol{Precision + Recall}}\right)$$

$$\text{F1 Score} = \frac{\mathbf{2TP}}{\mathbf{2TP+FP+FN}} = \frac{\mathbf{TP}}{\mathbf{TP}+\frac{1}{2}(\mathbf{FP+FN})}$$

1.5.7 Micro, Macro Weighted Averaging

```
Linear SVC (BOW) - Accuracy: 0.8729508196721312
Linear SVC (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.42      0.43        19
           1       0.30      0.33      0.32         9
           2       0.94      0.94      0.94       216

    accuracy                           0.87       244
   macro avg       0.56      0.56      0.56       244
weighted avg       0.87      0.87      0.87       244
```

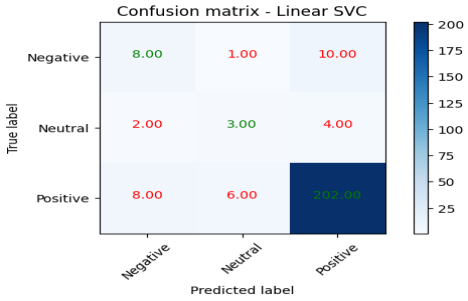


Figure 1.4 Confusion Matrix Example for Multiclass Classification

1.5.7.1 Micro Average Score

Micro-averaging calculates the metric by considering the Total TPs, Total FPs, and Total FNs across all classes. It gives more weight to classes with a larger number of instances. All samples equally contribute to the final averaged metric

For Negative Sentiment:

- TP = 8
- FP = 2+8 = 10
- FN = 1+10 = 11
- TN = 3+4+6+202 = 215

So, the Performance measure for Negative.

- Precision = 8 / (8+10) = 0.44
- Recall = 8 / (8+11) = 0.42
- F1 Score = (2*0.44*0.42) / (0.44+0.42)

  = 0.369/0.86

  = 0.43

Similarly, we can compute performance measures for other class as well.

For Positive Sentiment:

- TP = 202
- FP = 10 + 4 = 14
- FN = 8 + 6 = 14
- TN = 3+2+8+1 = 14

So, the Performance measures for Positive

- Precision = 202/ (202+14) = 0.94
- Recall = 202 / (202+14) = 0.94
- F1 Score = (2*0.94*0.94) / (0.94+0.94)
  = 1.76/1.88
  = 0.94

For Neutral Sentiment:

- TP = 3
- FP = 6+1 = 7
- FN = 2+4 = 6
- TN = 8+8+10+202 = 215

So, the Performance measures for Neutral

- Precision = 3/(3+7) = 0.30
- Recall = 3/(3+6) = 0.33
- F1 Score = (2*0.30*0.33) / (0.30+0.33)
  = 0.198 / 0.63
  = 0.32

- Total TP = 8+202+3 = 213
- Total FP = 10+14+7 = 31
- Total FN = 11+14+6 = 31

So, Micro Average Scores for Neutral

- Precision = 213/(213+31) = 0.872
- Recall = 213/(213+31) = 0.872
- Micro F1 Score = (2*0.87*0.87) / (0.87+0.87)
  = 1.513/ 1.74
  = 0.87

Precision = Recall = Micro F1 = Accuracy

1.5.7.2 Macro Average Score

It computes the performance measure separately for individual classes and then calculate the average value. It gives equal weight to each class, regardless of class imbalance. All classes equally contribute to the final averaged metric

Macro F1 Score = (0.43 + 0.94 + 0.32) / 3 = 0.56

1.5.7.3 Weighted Average Score

Weighted average calculates the metric for individual classes, weighted by the number of cases in individual class. It considers class imbalance by assigning higher weights to classes with fewer instances. Each classes contribution to the average is weighted by its size

Weighted F1 Score = (0.43*19 + 0.94*216 + 0.32*9) / (19+216+9)

= (8.17+203.4+2.88) / 244 = 0.878

1.5.8 ROC – AUC Curve

The ROC (Receiver Operating Characteristics) curve is used to measure classification models performance. It draws a probability curve for TPR (Sensitivity) against the FPR (1 - Specificity) at different threshold values. By using ROC plot, ROC AUC score is calculated which is the Area Under the ROC plot and its value scale from 0 to 1. An ROC AUC of 0.5 indicates random guessing, while 1 indicates a perfect classifier.

1.6 Objective of the Thesis

This thesis aims to develop a multiclass sentiment classification model and its application to bank call center transcribed data.

To develop a robust multiclass sentiment classification model for transcribed data of a nationalized bank call center using NLP and ML techniques. This involves exploring various NLP methods, feature extraction methods, and ML algorithms to accurately categorize customer sentiments as negative, positive or neutral.

1.7 Outline of the Thesis

The thesis has been structured in the following manner. First chapter, Introduction, contains a general summary of the research topic and explores the background of customer experience in the BFSI sector. It highlights the importance of sentiment analysis in this industry

and discusses the challenges faced when implementing sentiment evaluation techniques. It also includes a section on the Sentiment analysis methodologies, which discusses various approaches to sentiment analysis. It also explores the evaluation metrics commonly utilized for multi-class classification tasks.

The second chapter contains a Literature Survey, which is a detailed evaluation of the existing literature on sentiment analysis. It identifies and analyzes research gaps while highlighting the relevant approaches and techniques employed in the field.

In the third chapter, Problem, and Dataset Description, the specific problem addressed in the thesis is defined, and a full explanation of the dataset used in the study is provided. This chapter sets the context for the subsequent analysis.

Continuing to the fourth chapter, Solution Approach: Methodology, the proposed framework for sentiment analysis is outlined. It encompasses various elements, including lexicon-based sentiment generation, text pre-processing techniques, supervised machine learning algorithms, strategies for handling imbalanced datasets, exploration of different machine learning models, hyperparameter tuning through Grid Search CV, ensemble modeling using Voting Ensemble, and the deployment of the developed model.

Lastly, the fifth and final chapters, Conclusion and Future Scope, summarize the thesis findings and discuss their implications. It also provides insights into potential future research directions within the sentiment analysis domain. A list of the sources cited throughout the thesis is provided at the end.

CHAPTER 2

LITERATURE SURVEY

This Chapter Covers literature survey on Sentiment Analysis using NLP techniques and related Machine Learning and Deep Learning Approaches. Sentiment analysis, is an NLP technique that analyzes and determines the sentiment polarity of textual data. It has gained significant attention in the field of NLP and has become a prominent research area (Liu, 2012).

Text classification is a process of categorizing texts into groups. It finds applications in diverse domains like news organization, product review analysis, spam filtering, and document organization in digital libraries. Texts can be classified based on their topic, whether they meet specific criteria, or even their sentiment (Aggarwal & Zhai, 2012). Approaches for text sentiment classification commonly include traditional machine learning algorithms or deep learning methods (Kowsari et al., 2019).

In the literature survey entitled "*A Survey of the Applications of Text Mining in Financial Domain*" conducted by Kumar and Ravi (2016), the authors reviewed 89 research papers published between 2000 and 2016. The survey emphasizes on how text mining is used in many financial fields, such as stock market prediction, FOREX rate prediction, customer relationship management (CRM), and cyber security. The paper highlights key issues, identifies gaps, and proposes future research directions in this field (Kumar & Ravi, 2016).

The literature review conducted by Gupta et al. (2020), titled "*Comprehensive Review of Text-Mining Applications in Finance*" provides a comprehensive analysis of the impact and uses of text mining in the field of finance. The authors examine its relevance in banking, financial projections, and corporate finance, reviewing existing literature, highlighting recent

studies, discussing text-mining methods used in finance, addressing challenges, and exploring future prospects (Gupta et al., 2020)

In their study, Tien Thanh Vu et al. (2012) introduced a model that utilized Twitter messages to predict the price movements of stocks. They categorized the sentiment of the messages as either positive or negative and used this information to predict the stock prices of four enterprises "*Amazon, Apple, Microsoft, and Google*". They employed a Decision Tree (DT) algorithm for the classification task, considering historical data spanning 41 days.

In their research work, Yu et al. (2013) conducted a study investigating the influence of social media on the stock market performance of companies across different industries. They employed a Naive Bayes classifier to analyze the postings collected from numerous sources such as forums, blogs, and Twitter. Using these postings collected from multiple sources for 824 firms from six distinct sectors (pharmacy, software, health sector, hotel, and savings institutions) were analyzed. For performance evaluation, they considered stock return value and risk. They calculated sentiment scores for each document based on these values to assess the impact of social media on stock market performance.

Dey et al. (2009) proposed a stock market analysis system based on financial news. They employed "*Latent Dirichlet Allocation (LDA)*" to extract topic and kernel k-means algorithm for clustering topic-document data. They discovered important events and their market effects by analyzing the clusters with Sensex raw data. The method was built utilizing capital market news from the Indian stock exchange. The authors concluded that their text clustering-based approach provides insights into how events impact the stock market, enabling the design of better predictors. They also mentioned ongoing work on developing a market prediction system incorporating major real-time events through text mining.

Pang et al. (2002) investigated sentiment classification of movie reviews using unigram, bigram, and n-gram features. To classify the reviews, they used standard ML algorithms such as Naive Bayes, Maximum Entropy, and Support Vector Machines (SVM). The results of their experiments showed that these techniques outperformed human-generated values, indicating the effectiveness of the ML approaches in sentiment classification.

Chen et al. (2020) have done a comprehensive literature survey on applying NLP in the field of Financial Technology (FinTech). The paper explores NLP's role in Know Your Customer (KYC), Know Your Product (KYP), and Satisfy Your Customer (SYC) scenarios, analyzing both formal and informal textual data. The authors discuss dynamic product feature updates and customer satisfaction in B2C and C2C business models. The study identifies past challenges and proposes future research directions in FinTech and the open finance trend.

Deng et al. (2011) proposed a stock price prediction model incorporating features from time series data and social networks. The model utilizes numerical dynamics, sentiment analysis, and technical analysis of historical price and volume as input features. The stock price movements are modeled as a regression problem and solved using a Multiple Kernel Learning (MKL) regression framework. The experimental results show that the suggested strategy performs better than baseline methods in magnitude prediction metrics. The study concludes that incorporating features beyond stock prices themselves improves prediction performance. Future research directions include exploring different data sources and considering contextual dynamics in sentiment analysis.

Wu et al. (2014) developed a sentiment analysis framework for stock markets. They integrated popular sentiment analysis techniques with machine learning approaches like SVM and GARCH modeling. Using data from Sina Finance, they found a correlation between stock price volatility and sentiment in stock forums. In terms of classification accuracy, statistical

machine learning surpassed semantic approaches. The study also revealed that investor sentiment had a stronger impact on value stocks than growth stocks. Overall, their approach provided decision support for sentiment analysis in online stock forums.

The paper by Nopp & Hanbury (2015) examines the application of sentiment analysis in measuring a bank's attitude towards risk. The study analyses text data from CEO letters and outlook sections of annual bank reports to explore the effectiveness of sentiment analysis in banking supervision. The findings highlight both opportunities and limitations in using sentiment analysis for risk assessments. While individual bank predictions based on sentiment analysis are relatively inaccurate, aggregated analysis reveals significant correlations between textual uncertainty/negativity and future changes in quantitative risk indicators. The study suggests that sentiment analysis can be a valuable tool for macroprudential analyses in assessing risks in the banking system.

2.1 RESEARCH GAP

Despite the increasing interest in sentiment analysis, there is a gap in research focusing specifically on sentiment analysis for customer call centers in the BFSI sector. Existing studies often generalize sentiment analysis approaches without considering the unique characteristics and challenges of this sector. Consequently, there is a need for a comprehensive and tailored sentiment analysis framework that leverages NLP techniques to enrich the customer experience in the BFSI sector.

In this research, the objective is to develop such a framework and explore its effectiveness in improving customer experience through sentiment analysis of customer call center interactions. By addressing this research gap, it is the objective to contribute to the advancement of sentiment analysis techniques in a nationalized bank of the BFSI sector and provide actionable insights for banks to enhance their customer service strategies.

CHAPTER 3

PROBLEM AND DATA SET DESCRIPTION

This chapter explores the problem statement and the dataset used in the study, specifically in the context of the National Bank. In the following subsections, we will discuss the problem, define the objectives to overcome it, and describe the dataset employed for the study.

3.1 PROBLEM DESCRIPTION

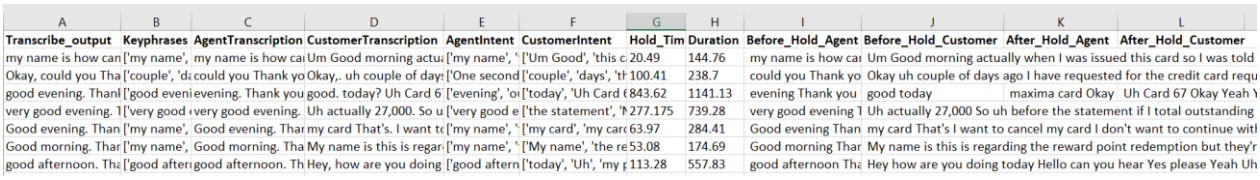
In today's competitive BFSI sector, understanding customer sentiment is crucial for improving customer satisfaction and loyalty. As part of this effort, bank call centers collect a vast amount of customer interaction data. Analyzing this data can provide valuable insights into customer sentiment towards various bank products and services. However, manually reviewing and categorizing these interactions can be time-consuming and inefficient. As a result, an automated sentiment analysis tool that can accurately categorize customer sentiments as neutral, negative, or positive based on transcribed call center conversations.

3.2 OBJECTIVE

With a focus on a nationalized bank, this study aims to develop a sentiment analysis model for bank call center transcribed data, focusing on classifying customer sentiments towards bank products as positive, negative, or neutral. The model will leverage NLP techniques and machine learning algorithms to automate the sentiment classification process. By accurately classifying customer sentiments, the objective is to provide the bank with actionable insights to enhance customer experience, discover areas for improvement, and make data-driven decisions to optimize their product offerings and customer service strategies.

3.3 DATASET DESCRIPTION

Due to a non-disclosure agreement, specific details about the dataset of this particular bank cannot be disclosed. However, it is essential to note that the dataset utilized in this study consists of a collection of 812 transcribed call center conversations from a bank's customer service interactions.



A	B	C	D	E	F	G	H	I	J	K	L
Transcribe_output	Keyphrases	AgentTranscription	CustomerTranscription	AgentIntent	CustomerIntent	Hold_Tim	Duration	Before_Hold_Agent	Before_Hold_Customer	After_Hold_Agent	After_Hold_Customer
my name is how can	['my name',	my name is how ca	Um Good morning actu	['my name', '	['Um Good', 'this c	20.49	144.76	my name is how ca	Um Good morning actually	when I was issued this card so I was told	
Okay, could you Tha	['couple', 'd	could you Thank yo	Okay,. uh couple of day	['One second	['couple', 'days', 'th	100.41	238.7	could you Thank yo	Okay uh couple of days ago	I have requested for the credit card requ	
good evening. Thanl	['good eveni	evening. Thank you	good. today? Uh Card 6	['evening', 'o	['today', 'Uh Card 6	843.62	1141.13	evening Thank you	good today	maxima card Okay	Uh Card 67 Okay Yeah Y
very good evening. T	['very good	very good evening.	Uh actually 27,000. So u	['very good e	['the statement', 'I	277.175	739.28	very good evening T	Uh actually 27,000 So uh	before the statement if I total outstanding	
Good evening. Than	['my name',	Good evening. Thar	my card That's. I want t	['my name', '	['my card', 'my car	63.97	284.41	Good evening Than	my card That's I want to cancel my card I don't want to continue with		
Good morning. Thar	['my name',	Good morning. Tha	My name is this is regar	['my name', '	['My name', 'the re	53.08	174.69	Good morning Thar	My name is this is regarding the reward point redemption but they'r		
good afternoon. Tha	['good after	good afternoon. Th	Hey, how are you doing	['good aftern	['today', 'Uh', 'my p	113.28	557.83	good afternoon Tha	Hey how are you doing today Hello can you hear Yes please Yeah Uh		

Figure 3.1 Dataset used in the study

The snapshot of the obtained data is shown in Figure 3.1. the Dataset Comprises of the following information.

1. **'Transcribe_output'**: This column represents the transcribed text output of the overall conversation. It includes the text generated through speech-to-text conversion or any other transcription process.
2. **'Keyphrases':** This column contains key phrases or important terms extracted from the transcribed text in the overall conversation.
3. **'AgentTranscription':** This column contains the transcriptions or text spoken by the agent during the conversation. It represents the dialogue or messages conveyed by the customer service representative or agent.
4. **'CustomerTranscription':** This column contains the transcriptions or text spoken by the customer during the conversation. It represents the dialogue or messages conveyed by the customer or user.

5. **'AgentIntent':** This column contains important terms extracted from the Agent Conversation transcribed text.
6. **'CustomerIntent':** This column contains important terms extracted from the Customer Conversation transcribed text.
7. **'Hold_Time':** This column represents the duration of time the conversation was put on hold in seconds. It provides information about how long the customer or agent had to wait during the conversation.
8. **'Duration':** This column represents the duration of the entire conversation, including both active conversation time and hold time in seconds.
9. **'Before_Hold_Agent':** This column contains the text or actions of the agent before the conversation was put on hold. It captures the agent's interactions or messages just before the hold occurred.
10. **'Before_Hold_Customer':** This column contains the text or actions of the customer before the conversation was put on hold. It captures the customer's interactions or messages just before the hold occurred.
11. **'After_Hold_Agent':** This column contains the text or actions of the agent after the conversation resumed from hold. It captures the agent's interactions or messages once the hold was over.
12. **'After_Hold_Customer':** This column contains the text or actions of the customer after the conversation resumed from hold. It captures the customer's interactions or messages once the hold was over.

The data, pertaining to a nationalized bank, was accordingly preprocessed and then analyzed with the approaches already discussed and implementation is discussed in next chapter.

CHAPTER 4

SOLUTION APPROACH: METHODOLOGY

This chapter presents the solution approach and methodology employed in the study to address the sentiment analysis problem in the context of the National Bank. The proposed framework and various techniques used for sentiment generation and sentiment classification will be discussed in detail.

4.1 PROPOSED FRAMEWORK

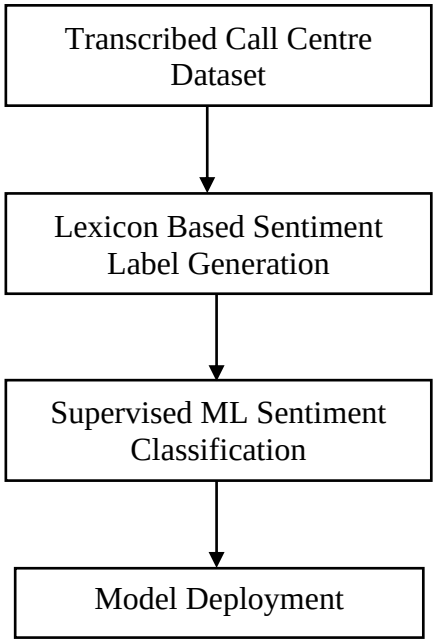


Figure 4.1 Proposed Framework

4.2 LEXICON BASED SENTIMENT GENERATION

As discussed in Introduction (Chapter 1), we used three different techniques to Calculate Sentiment Score such as Vader Sentiment, Textblob, SentiWordNet. All these techniques give polarity scores in different ranges. Vader Sentiment Polarity scores varies in

the range between -4 to 4, Textblob Sentiment Polarity Scores varies in the range of -1 to 1 and SentiWordNet Polarity Score varies in the range between 0 to 1.

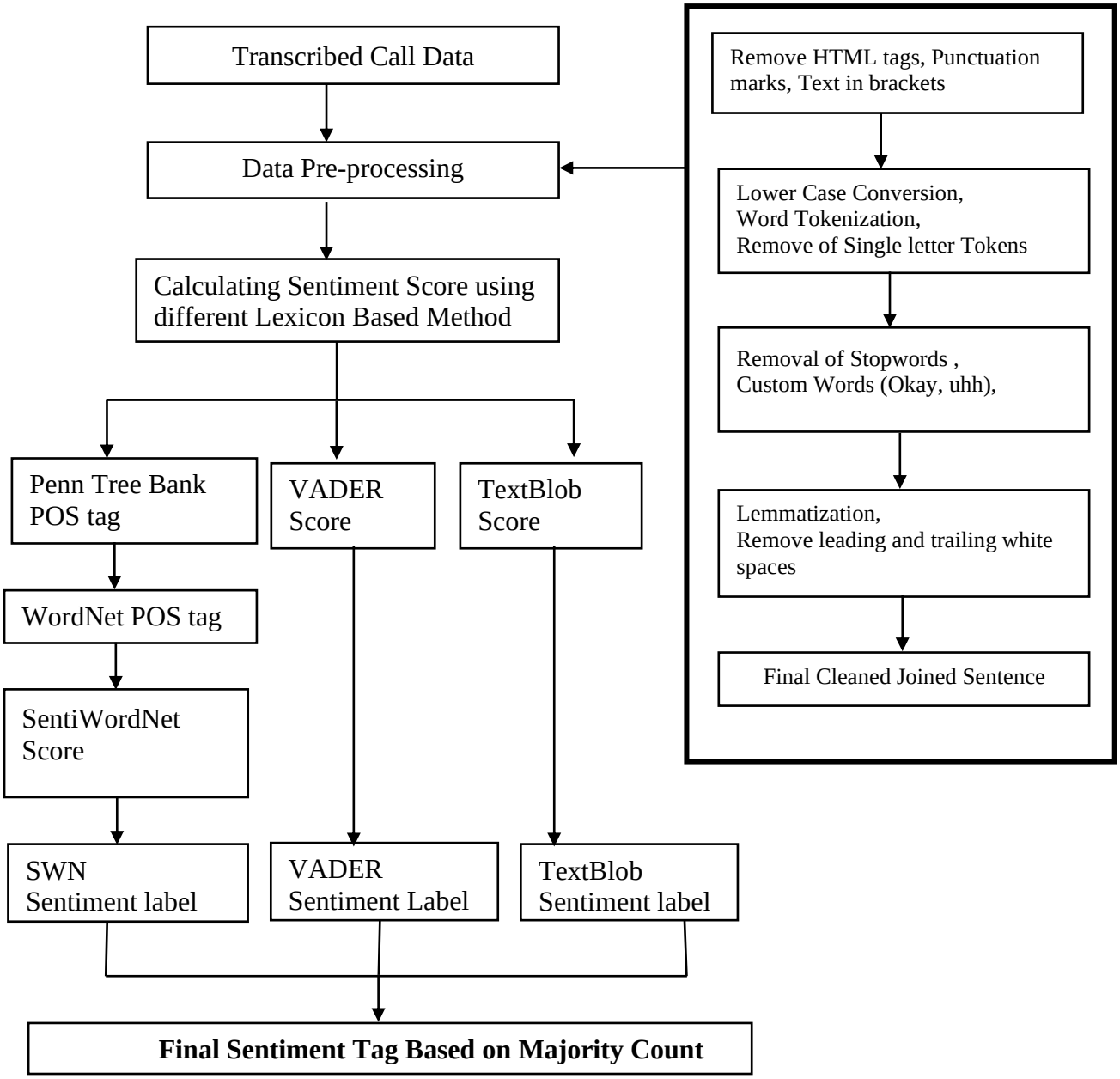


Figure 4.2 Lexicon Based Sentiment Analysis Framework

4.2.1 DATA PRE-PROCESSING

Preprocessing data or text is essential in NLP applications such as sentiment analysis (Gurusamy & Kannan S, 2014). It involves converting unprocessed text data into a format appropriate for machine learning algorithms and analysis. The following subsection describes the steps used for preprocessing data used in our study.

4.2.1.1 Tokenization

Due to Webster & Kit (1992), Tokenization is a method of splitting a text or document into smaller components known as tokens. Tokens can be distinct words, phrases, or even characters depending on the level of granularity required.

4.2.1.2 Stop Words Removal

Stop word removal, adopted from Kaur & Kaur Buttar (2018), are commonly occurring words in a language that generally don't have significant meaning and do not contribute much to the overall understanding of the text. Articles (e.g., "a," "an," "the"), pronouns (e.g., "he," "she," "it"), prepositions (e.g., "in," "on," "at"), and conjunctions (e.g., "and," "but," "or") are example of stop words.

4.2.1.3 Lemmatization

Lemmatization, adopted from Manning et al., (1946), is the process of reducing words to their base or root form, known as the lemma. The lemma represents the dictionary form of a word and aids in the reduction of several inflected versions of a word to a single representation. For example, the lemma of the words "running," "runs," & "ran" is "run.".

It considers the context and meaning of the word in order to determine the appropriate lemma. The purpose of lemmatization is to normalize words and reduce their variability, which can benefit tasks like text classification, information retrieval, and other natural language processing applications.

4.2.2 Text Pre-Processing for SentiWordNet

SentiWordNet is a lexical resource that provides scores to synsets (sets of synonymous words) in WordNet, a widely used lexical database. Text preprocessing is necessary before using SentiWordNet for sentiment analysis to ensure the accuracy and effectiveness of the analysis.

4.2.2.1 Parts of Speech (POS) Tagging

It involves assigning grammatical tags to each word in a sentence, indicating its part of speech (e.g., noun, verb, adjective). POS tagging is essential for SentiWordNet because sentiment scores are assigned at the synset level, and knowing the correct POS tag helps in matching words with their corresponding synsets.

In NLTK, we can perform POS tagging using the pos_tag() function. It accepts a list of tokens (words) and returns a list of tuples, each containing a word and its matching POS tag.

POS Tags Required by SentiWordNet:

SentiWordNet uses WordNet's POS tags to assign sentiment scores to words. WordNet uses different POS tags compared to the standard POS tags used in NLTK. The mapping between the two sets of POS tags is as follows:

NLTK POS Tags (Santorini, 1990)**:**

NLTK Tags	POS
NN	Noun
VB	Verb
JJ	Adjective
RB	Adverb

Table 4.1 NLTK POS Tags

WordNet POS Tags: (Miller & Fellbaum, 1998)

WordNet Tags	POS
n	Noun
v	Verb
a	Adjective
r	Adverb

Table 4.2 WordNet POS Tags

4.2.2.2 Synset Mapping

SentiWordNet assigns sentiment scores to synsets, not individual words. Therefore, after tokenization, POS tagging, and lemmatization, the next step is to map each word to its corresponding synset. This involves accessing WordNet's database and finding the appropriate synset for each word based on its lemma and POS tag.

4.2.2.3 Sentiment Score Calculation

Once the words are mapped to their respective synsets, SentiWordNet provides sentiment scores for each synset. These scores typically represent the positivity, negativity, and neutrality of the synset. The sentiment scores can be combined and aggregated to calculate the overall sentiment score for a given text.

4.3 SUPERVISED MACHINE LEARNING BASED SENTIMENT CLASSIFICATION

As discussed in Chapter 1, the sentiment classification using the machine learning approach is illustrated in Figure 4.3.

4.3.1 DATA PRE-PROCESSING

The exact steps for text cleaning were followed as those used during sentiment label generation using lexicon-based models.

- Removes HTML tags using the BeautifulSoup library.
- Removes content within square brackets using regular expressions.
- Removes characters except letters using regular expressions.
- Removes punctuation using regular expressions.

- Tokenizes the text into individual words.
- Removes stopwords and custom stopwords.
- Filters out tokens with a length of 1 (single characters).
- Lemmatizes the tokens using the NLTK lemmatizer.
- Joins the tokens back into a single string.

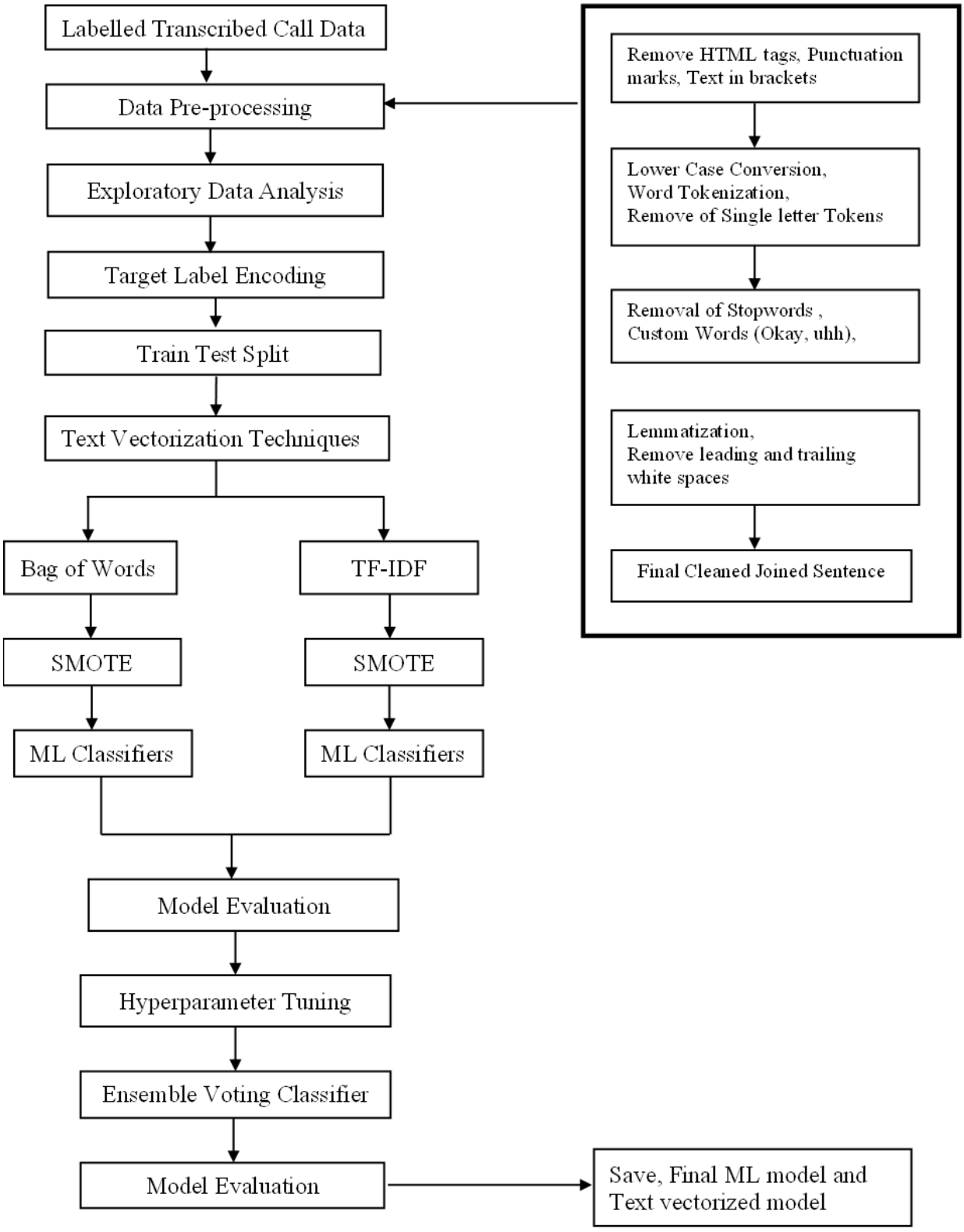


Figure 4.3 Supervised ML Approach for Sentiment Classification

	Transcribe_output	Overall_Sentiment	Cleaned_Text
0	my name is how can I delight you today? Um Good morning actually when I was issued this card so I was told that uh uh no uh will be, it will be but again in this month I have received a message that imposed, Yeah the card holder of this card. Okay, let me check your statement. Yes. Charges are included in this month. Okay. Now would you like to raise the request for? Alrighty. Thank you so much. Could you please send me your mother name? Uh 14,700 one. Okay, thanks for the verification so I will request for reversal this amount will be not refundable amount triple nine. It's only credit on your credit card. Okay, so are you okay with that now? Okay ma'am. Thank you. And firstly you have to uh pay this statement. Refundable amount will be adjusted in your next month statement. You have to pay the bill and the payment next month? Yes sir, definitely. And uh for a minute. Yeah, yes sir, I can number, it will be taken, it will be credit on your credit. Okay. Welcome. Is there anything else that I can assist you with? Thank you. Thank you so much. Okay thank you for calling us requested. You have a nice day. Okay.	Neutral	name delight today um good morning actually issued card told month received message imposed yeah card holder card let check statement yes charge included month would like raise request alrighty thank much could please send mother name one thanks verification request reversal amount refundable amount triple nine credit credit card thank firstly pay statement refundable amount adjusted next month statement pay bill payment next month yes sir definitely minute yeah yes sir number taken credit credit welcome anything else assist thank thank much thank calling u requested nice day
1	Okay, could you Thank you? So uh couple of days ago I have requested for the credit card. Okay, request Uh I got a lot of, I wanted to know the reason Sure. One second. Okay, so if you want to the rest of the annual membership take very quick please. Okay, Yeah, I think no worries. So he goes uh some issues uh sub sub category selected the requested decline, but I have taken the request for verification purposes? Yes. Sorry. Okay, tell me your name, party, I know that I just yeah. Sorry uh tell me, repeat again. Uh so which month annual membership fee will be charged? December. December. Okay, very quick Turnaround time, did we take of I remember confirmation will be refunded if you want then I will be mentioned on there, please mention that. Okay now if you want that then I will be Mhm. request Okay. policies is there anything else for now? This will be okay. Thank you so much. Thank you so much. Have a wonderful day. Okay.	Neutral	could thank couple day ago requested credit card request got lot wanted know reason sure one second want rest annual membership take quick please yeah think worry go issue sub sub category selected requested decline taken request verification purpose yes sorry tell name party know yeah sorry tell repeat month annual membership fee charged december december quick turnaround time take remember confirmation refunded want mentioned please mention want mhm request policy anything else thank much thank much wonderful day
2	good evening. Thank you for choosing our bank. My name is, how may I delight you today? Uh Card 67 maxima card Okay. Yeah. Yeah well Okay. Okay. Okay. October. October okay. October 600%. resident 80 double 10 uh 001 eight double zero double 01. resident verification. Mhm. double Okay 80 double 114 last time. resident yeah. bank I do request number registered mobile number dot com K Y C. Dot dot R D S. bank dot com. Welcome to our banks. Please enter the things which we have sent on your registered number followed by the last night. Congratulations. Yeah I would like to receive dot dot dot dot bank dot com finish. Yeah the sorry sorry many address address A D. H. D. For bangalore. A for S for haryana, P. For U. U. U. For umbrella U. U. In uh Police partner we have 6793. Okay 80 double 103 zero double 103 number one number one number one number one thank you for thank you for writing customers and look forward document, make sure that okay. Uh huh. Dot com. Okay what capital dot okay. capital Y T dot dot C. A R. D S. bank dot com. Okay. Mhm. Okay yeah Okay yeah busy lane Okay yeah 25 quick information. Okay. Okay. Uh it's mutler request number mobile number and email mobile number R. D. For Banksy for no no no no no R P for character Okay I C B Okay five 6070 double one 5670. No No. 5 6 zero 560 70 70 double one double one R. P. C. B. R. P. T. B. pay Okay. Number of reward points and the multiple reward points. Okay. Credit card. Okay. Yeah thank you. Okay. Okay www dot dot com application limit increase. Yeah. Personal loan, personal loan. Yeah. Check 7 600 â‚¹100 up to 1000 thousands Many up to 1000 reward points. Transaction transaction maximum. Okay take care, utility bill payment. Okay. Alright. Grocery growth Uh huh, correct somebody reward point Okay. Sorry 900 reward points. cash Okay.	Neutral	good evening thank choosing bank name may delight today card maximum card yeah yeah well october october october resident double eight double zero double resident verification mhm double double last time resident yeah bank request number registered mobile number dot com dot dot dot com bank dot com welcome bank please enter thing sent registered number followed last night congratulation yeah would like receive dot dot dot dot bank dot com finish yeah sorry sorry many address address bangalore haryana umbrella police partner double zero double number one number one number one number one thank thank thank writing customer look forward document make sure huh dot com capital dot capital dot dot bank dot com mhm yeah yeah busy lane yeah quick information mutler request number mobile number email mobile number banksy character five double one zero double one double one pay number reward point multiple reward point credit card yeah thank www dot dot com application limit increase yeah personal loan personal loan yeah check thousand many reward point transaction transaction maximum take care utility bill payment alright grocery growth huh correct somebody reward point sorry reward point cash

Figure 4.4 **Pre-processed Text**

4.3.2 EXPLORATORY DATA ANALYSIS

EDA is an essential step in understanding the characteristics and patterns of the dataset. In the following subsection, we examine the data set to gain insights into the frequency of label data, word frequency for each sentiment label, and generate word clouds to visualize the prominent terms in each sentiment class.

4.3.2.1 Frequency of Label Data

To understand the distribution of sentiment labels in the dataset, we analyze the frequency of each label. This analysis helps us gain an understanding of the data imbalance and the prevalence of different sentiment categories.

In figure 4.5 shows the distribution of sentiment labels for our dataset used.

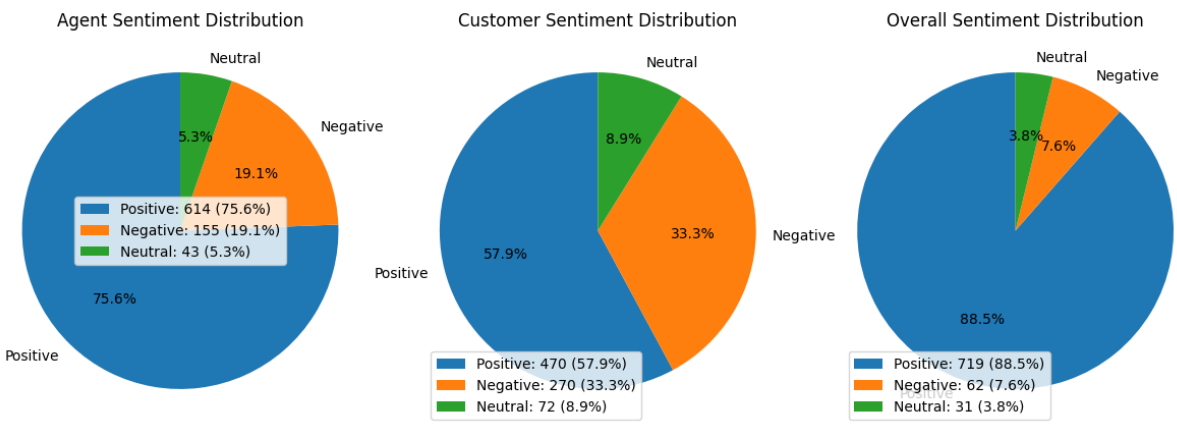


Figure 4.5 **Frequency of Sentiment Labels**

4.3.2.2 Word Frequency for each Sentiment Labels

Analyzing the word frequency for each sentiment label provides insights into the most frequently occurring words in different sentiment categories.

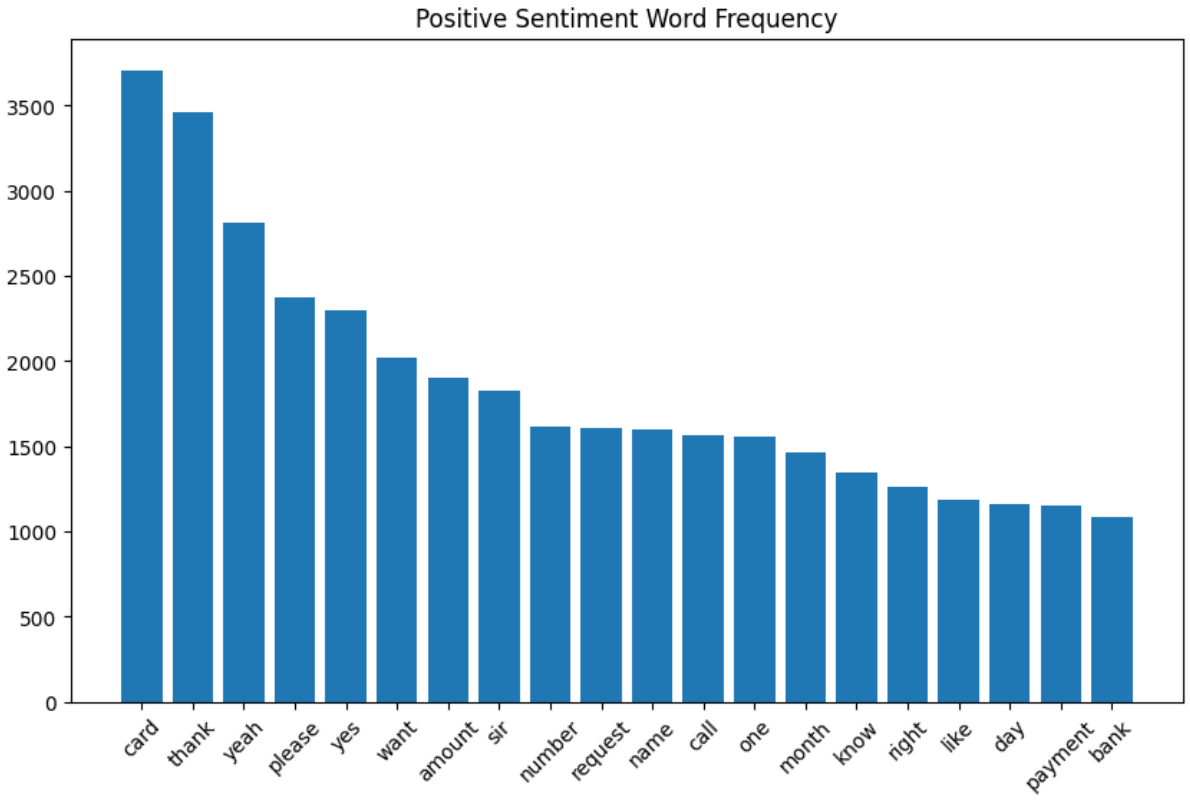


Figure 4.6 **Word Frequency Count for Positive Sentiments**

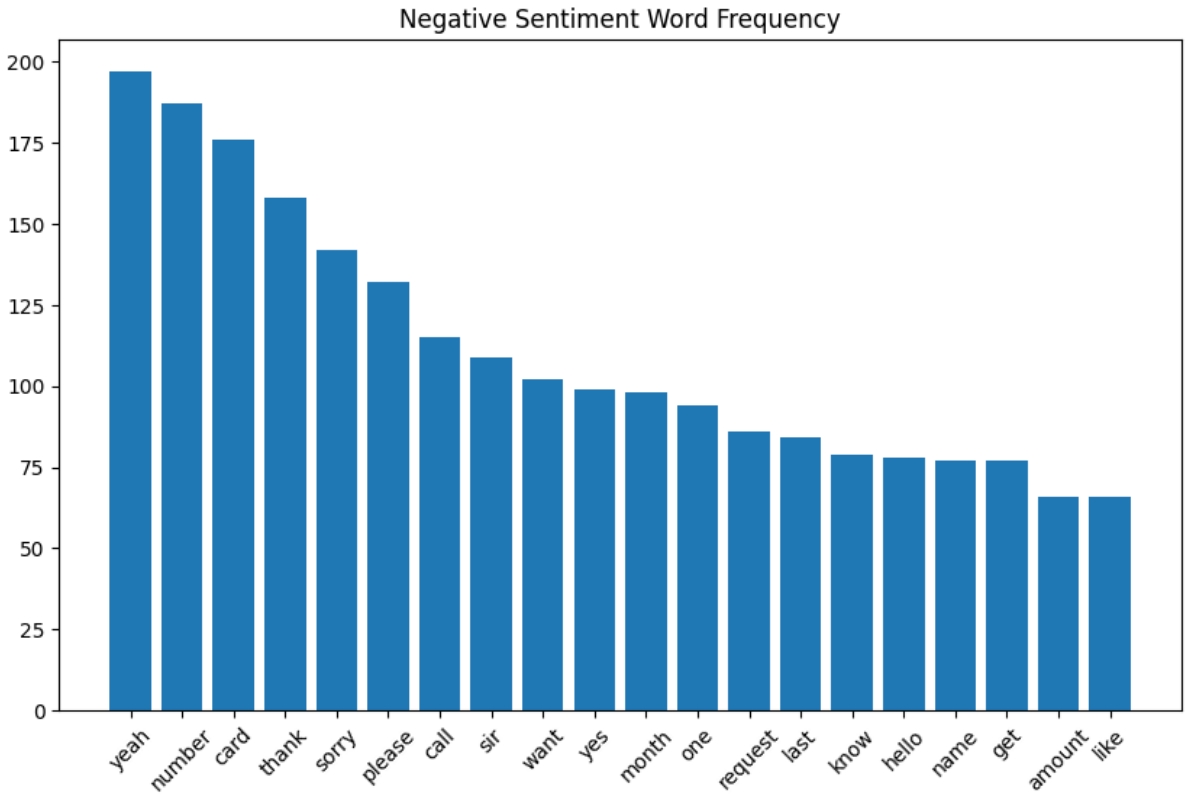


Figure 4.7 Word Frequency Count for Negative Sentiment

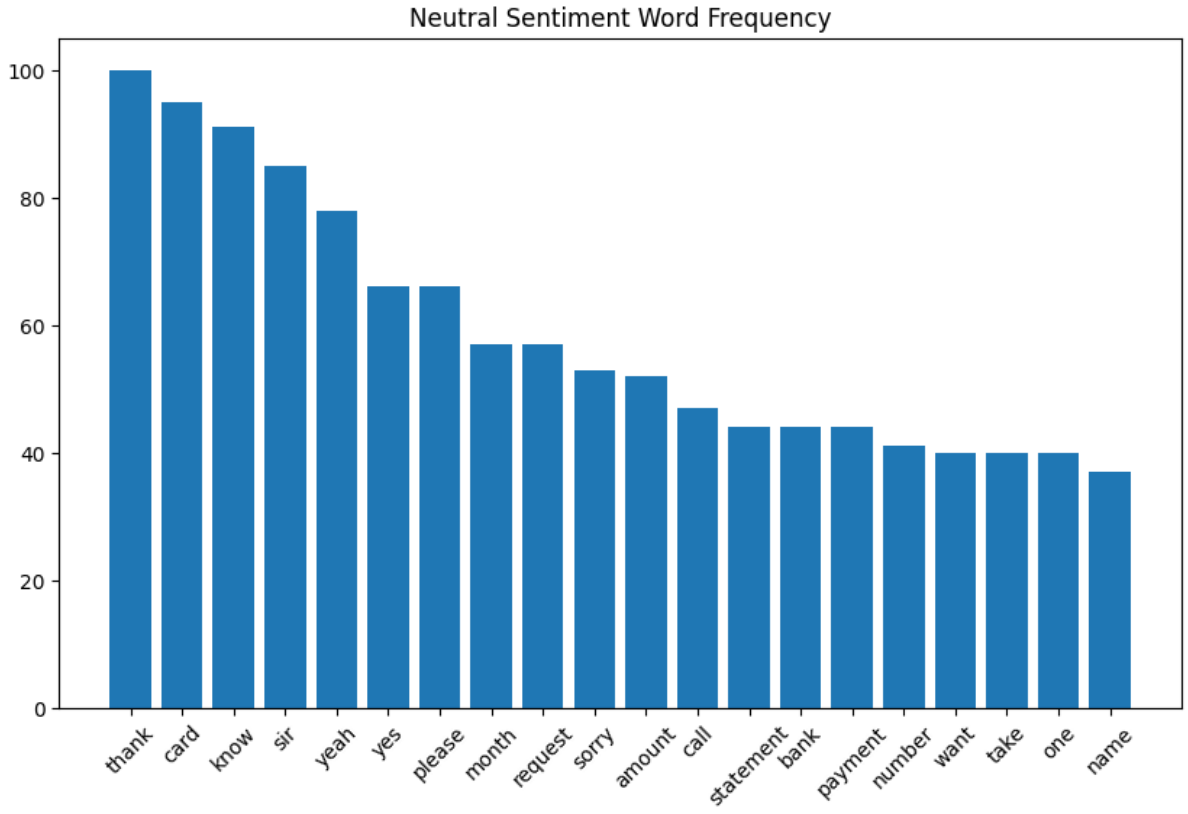


Figure 4.8 Word Frequency Count for Neutral Sentiment

4.3.2.3 Word Cloud Formation for each Sentiment Class

Word clouds are visual representations of the words that appear most frequently in a corpus of text. They offer an intuitive and visually appealing way to understand the importance and prominence of different words in a given dataset. The size and visual prominence of a word in a word cloud are determined by both its frequency and importance within the document.

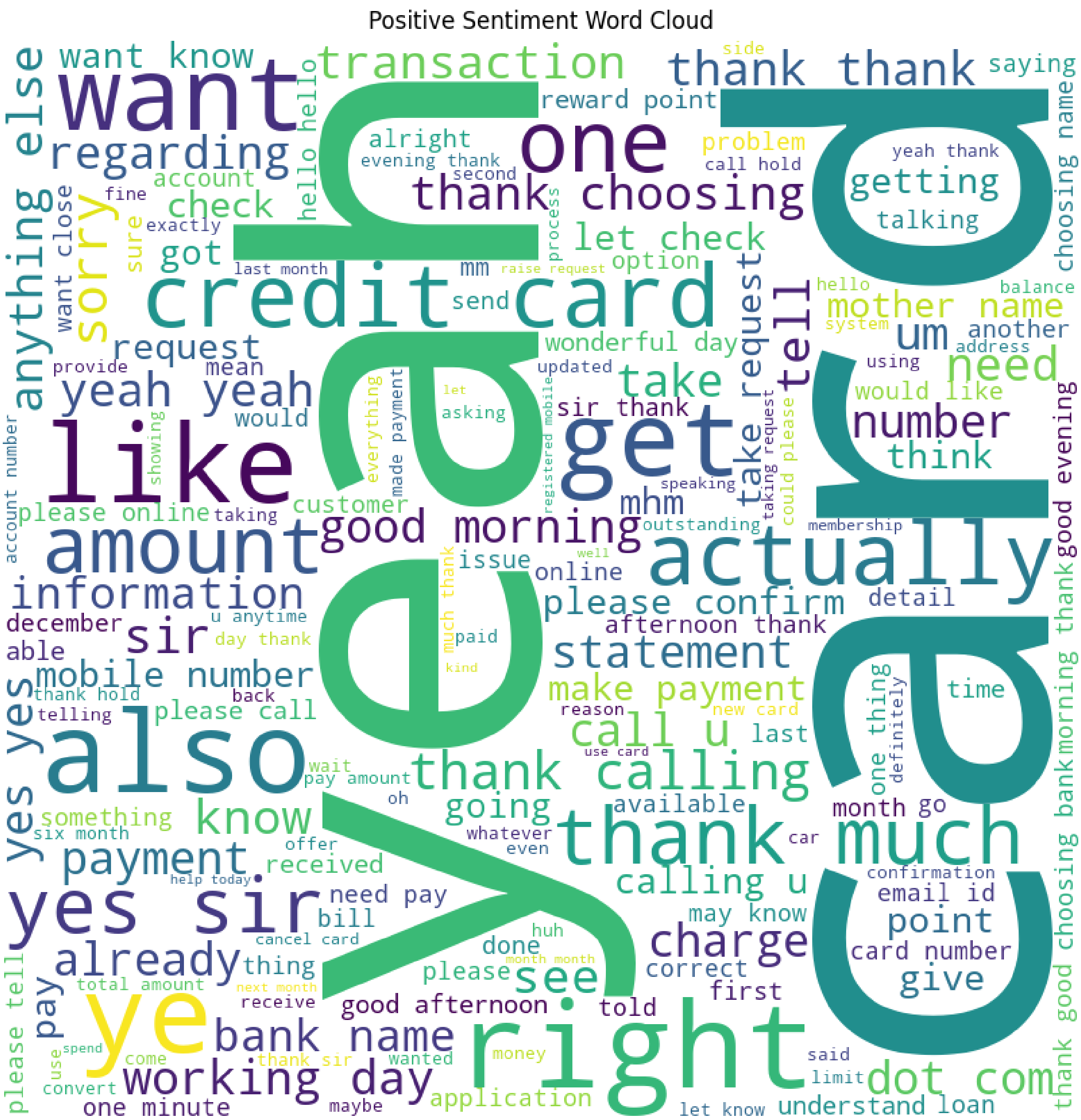


Figure 4.9 Word Cloud for Positive Sentiment

Figure 4.10 Word Cloud for Negative Sentiment

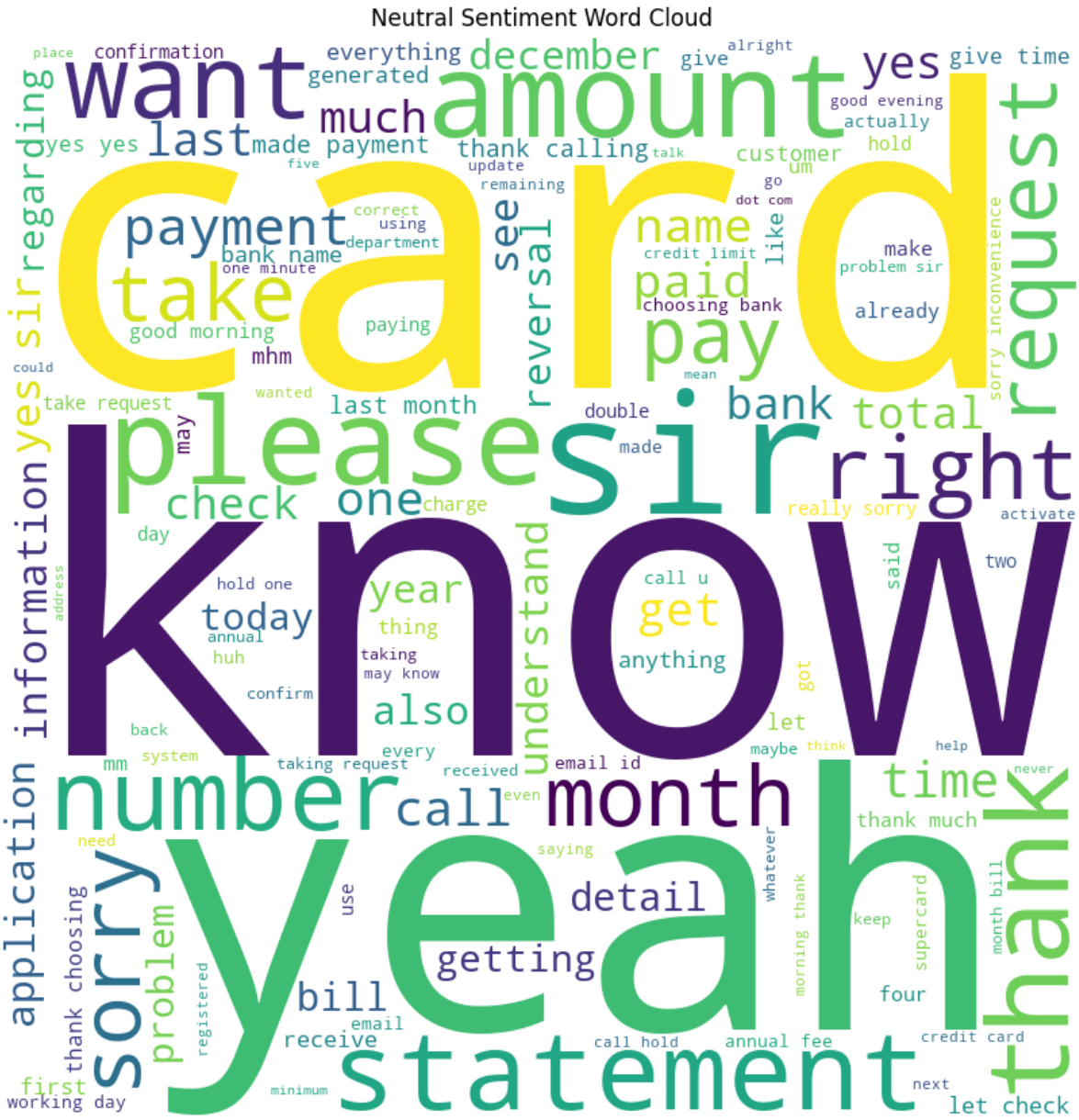


Figure 4.11 Word Cloud for Neutral Sentiments

4.3.3 Label Encoding

We apply label encoding to prepare the sentiment labels for machine learning models. Label encoding is a technique that converts categorical labels into numerical representations. This allows the machine learning algorithms to understand and process the sentiment labels effectively.

4.3.4 Train Test Split

We divided the dataset into training and testing sets to assess the performance of the machine learning models. The train test split is typically done to train the models on a subset of the data (training set) and then assess their performance on unseen data (testing set). In our study, we split the original dataset of 812 call center conversations into train data and test data by applying the train test split function with a test size of 0.3, resulting in a training dataset with 568 samples and a testing dataset with 244 samples.

We can gain insights into their accuracy and effectiveness in sentiment classification by evaluating the models' performance on the testing set.

4.3.5 Text Vectorization Techniques

Whenever we apply any algorithm in NLP, it works on numbers. Because ML algorithms cannot operate directly with raw text, it must be transformed into well-defined vectors of real numbers. Text vectorization transforms raw text input data into numerical representation, which the ML model supports.

Text vectorization is a crucial step in machine learning feature extraction, converting text into numerical vectors to extract distinct features for model training.

4.3.5.1 Bag of Words

It is a statistical method for converting text into a numerical representation so that it can be utilized for ML-based modeling and is adapted from (Y. Zhang et al., 2010). In this method, a text which can be a sentence or document, is represented as the Bag of its words, which means it only cares about the occurrence of words within a document and their count without considering the order of words in the sentence or document, hence called Bag of words. It involves two steps, i.e., (i) Creating a vocabulary of known words and (ii) Measuring the occurrence of these words within the text.

For example:

Consider below three sentences as described in Table 4.3

Step 1: Vocabulary would consist of – the, cat, sat, in, hat, with

Step 2: Count Vectorizer

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Table 4.3 Bag of Words Example (Zhou, 2019)

	ability	able	abrasion	abroad	absolute	absolutely	accept	acceptable	accepted	accepting	access	accident	according	accordingly	account	ac
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	

Table 4.4 Vectorized Bag of words Features for the input data

card	2702
thank	2630
yeah	2106
yes	1665
want	1536
sir	1398
number	1199
request	1196
know	1107
month	1097

Table 4.5 Top 10 Frequent Words with BOW Features

4.3.5.2 TF-IDF

It is a numerical statistic consisting of two terms, term frequency, and inverse document frequency, used in text mining to assess the significance of a word within a document or a collection of documents (Jones, 1972). To comprehend it, three fundamental terms are related to it: (i) Term frequency (TF); (ii) Document frequency (DF); and (iii) Inverse Document frequency (IDF), which will be discussed in detail in the following subsection.

- **TF:** is computed as the ratio of word occurrences to the total number of words in a document, assuming that higher term frequency indicates greater relevance to the document.

$$\mathbf{TF} = \frac{\textbf{Total number of times a word is present in document}}{\textbf{Total number of words in that document}}$$

- **DF:** It refers to the number of times a word appears in the collection of documents. It is used in calculating IDF. Words that appear in a smaller number of documents are generally more discriminative and contribute more to the overall TF-IDF score.

$$\mathbf{DF} = \frac{\textbf{Document containing word W}}{\textbf{Total number of document}}$$

- **IDF:** It quantifies the significance of a term within a document collection by taking the logarithm of the ratio between the total number of documents and the number of documents containing the term.

$$\mathbf{IDF} = \log\left(\frac{\textbf{Total number of document}}{\textbf{Document containing word W}}\right)$$

The Final TF-IDF is the product of TF and IDF. The higher score indicates the greater importance of the word in the corpus.

$$\textbf{TF-IDF} = \textbf{TF} * \textbf{IDF}$$

	ability	able	abrasion	abroad	absolute	absolutely	accept	acceptable	accepted	accepting	access	accident	according	accordingly	account
0	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.069136	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 4.6 Vectorized TF-IDF Features for the input data

	feature_names	tfidf_sum
303	card	59.586110
2437	yeah	50.151157
2441	yes	40.517871
2036	sir	39.070164
2376	want	36.941490
1847	request	33.512094
1458	number	33.059815
1392	month	32.178284
1539	payment	29.407347
1158	know	27.976978

Table 4.7 Top 10 Frequent words with TF-IDF Features

4.3.6 Handling Imbalanced Dataset

Imbalanced data refers to a situation in which the distribution of instances across different classes is significantly skewed. In our case, the number of instances representing negative and neutral sentiments is considerably lower than those expressing positive sentiments(Kotsiantis et al., 2006).

This class imbalance poses challenges for machine learning models, as they tend to be biased towards the majority class, which in our case consists of positive sentiments. Class imbalance poses challenges in training machine learning models as it can lead to biased predictions and poor performance on the minority class. As a result, the predictive performance of the model may be compromised, as it may struggle to accurately capture and classify the minority class (negative sentiments).

This section discusses the potential impact of class imbalance on model performance (Hensman & Masko, 2015). The class imbalance issue can have the following effects:

Reduced Accuracy: Due to the unequal distribution of classes, a model trained on imbalanced data may achieve high accuracy by simply predicting the majority class most of the time. However, such high accuracy can be misleading as the model fails to predict instances from the minority class accurately.

Biased Decision Boundary: Imbalanced datasets can result in decision boundaries that are biased towards the majority class. As a result, the model may struggle to correctly classify instances from the minority class, leading to poor recall and precision for that class.

Lack of Generalization: Models trained on imbalanced datasets may lack the ability to generalize well to new and unseen data. This is because the model is not exposed to enough

examples from the minority class during training, resulting in a biased understanding of the data distribution.

In this study, SMOTE has been employed to mitigate the issue of class imbalance, which is discussed in the following subsection.

4.3.6.1 SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) is a widely adopted resampling technique designed specifically for handling class imbalance. It generates synthetic examples of the minority class to increase its representation in the dataset, thus achieving a more balanced class distribution. The SMOTE algorithm selects a minority class instance and identifies its nearest neighbors. It then generates synthetic instances along the line segments connecting the instance and its neighbors in the feature space. These synthetic examples capture the underlying patterns and characteristics of the minority class, enabling the machine learning model to learn and generalize more effectively (Chawla et al., 2002).

By incorporating SMOTE into this study, the goal is to mitigate the impact of class imbalance and improve the predictive performance of the machine learning model. With a more balanced representation of negative sentiments, the model can better capture the nuances and intricacies of the minority class, leading to more accurate and reliable predictions.

The process of SMOTE can be summarized as follows:

1. **Identifying Minority Class Instances**: SMOTE starts by identifying instances from the minority class that requires oversampling.
2. **Identifying Nearest Neighbors**: For each minority class instance, SMOTE identifies its k nearest neighbors based on a distance metric (e.g., Euclidean distance). The value of k is typically set to 5.

3. **Creating Synthetic Samples**: Synthetic samples are generated by randomly selecting one or more nearest neighbors and creating synthetic instances along the line segments joining the selected neighbors. The synthetic samples are created by randomly selecting a fraction between 0 and 1 and multiplying it with the difference between the feature values of the instance and its selected neighbor(s).

4. **Balancing the Dataset**: The synthetic samples are added to the original dataset, effectively increasing the number of instances in the minority class and achieving a more balanced dataset.

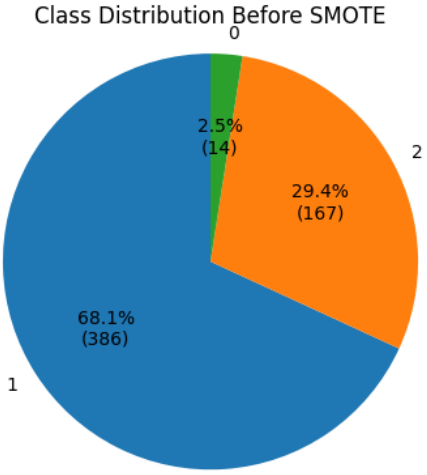


Figure 4.12 Class Distribution Before SMOTE

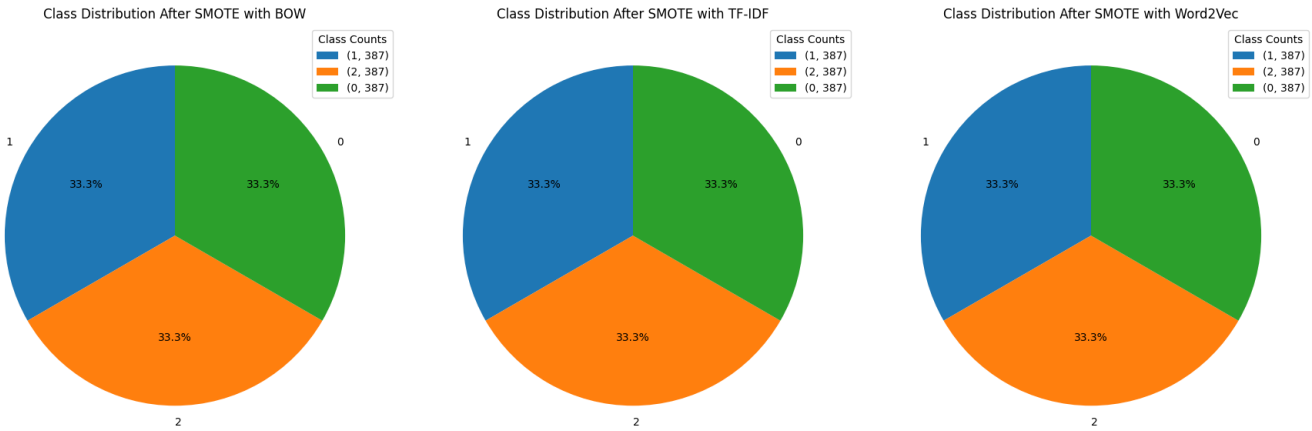


Figure 4.13 Class Distribution after SMOTE

4.3.7 Machine Learning Models Result

This section presents the results obtained from applying various machine learning models to the dataset. We specifically focus on the performance of these models when using SMOTE with Bag-of-Words (BOW) representation and SMOTE with TF-IDF representation.

4.3.7.1 ML Models on SMOTE with BOW

After applying SMOTE to address the class imbalance and using the Bag-of-Words (BOW) representation, we evaluated several ML models on the dataset. The models were trained on the balanced dataset and tested on unseen data. The evaluation metrics like accuracy, precision, recall, F1 score, and Confusion Matrix, were calculated to assess the models' performance in sentiment classification.

```
Logistic Regression (BOW) - Accuracy: 0.8811475409836066
Logistic Regression (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.37      0.42        19
           1       0.33      0.33      0.33         9
           2       0.93      0.95      0.94       216

    accuracy                           0.88       244
   macro avg       0.59      0.55      0.57       244
weighted avg       0.87      0.88      0.88       244
```

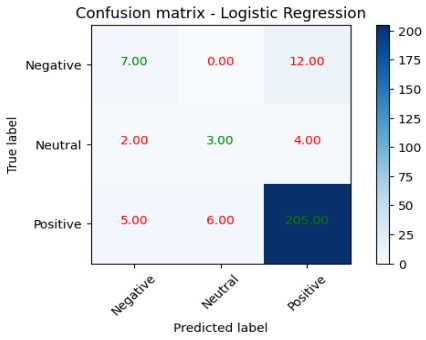


Figure 4.14 Logistic Regression Model Evaluation on BOW Features

```
Multinomial Naive Bayes (BOW) - Accuracy: 0.8852459016393442
Multinomial Naive Bayes (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.11      0.17        19
           1       0.00      0.00      0.00         9
           2       0.90      0.99      0.94       216

    accuracy                           0.89       244
   macro avg       0.47      0.37      0.37       244
weighted avg       0.83      0.89      0.85       244
```

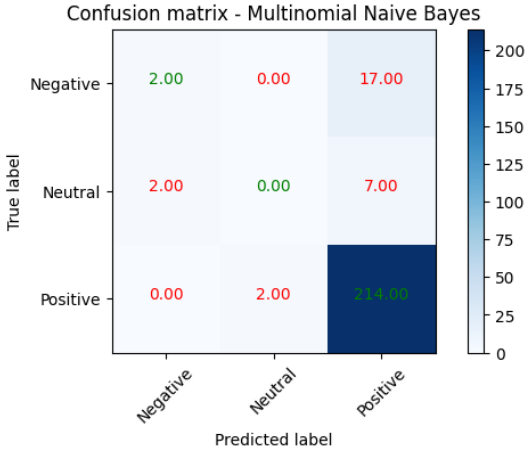


Figure 4.15 Multinomial Naïve Bayes Model Evaluation on BOW Features

```
Decision Tree Classifier (BOW) - Accuracy: 0.7622950819672131
Decision Tree Classifier (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.12      0.21      0.16        19
           1       0.12      0.22      0.15         9
           2       0.92      0.83      0.88       216

    accuracy                           0.76       244
   macro avg       0.39      0.42      0.40       244
weighted avg       0.83      0.76      0.79       244
```

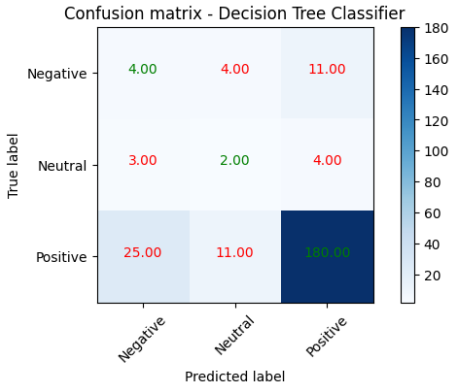


Figure 4.16 Decision Tree Model Evaluation on BOW Features

```
Support Vector Classifier (BOW) - Accuracy: 0.8524590163934426
Support Vector Classifier (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.24      0.21      0.22        19
           1       0.25      0.33      0.29         9
           2       0.93      0.93      0.93       216

    accuracy                           0.85       244
   macro avg       0.47      0.49      0.48       244
weighted avg       0.86      0.85      0.85       244
```

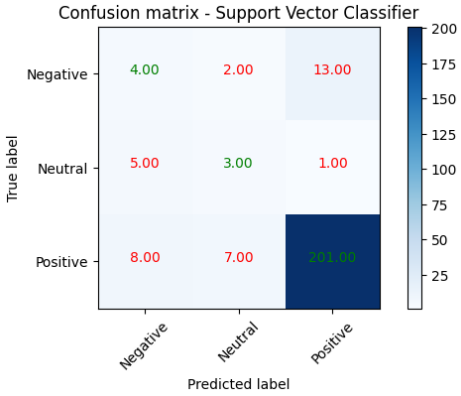


Figure 4.17 Support Vector Model Evaluation on BOW Features

```
Linear SVC (BOW) - Accuracy: 0.8729508196721312
Linear SVC (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.42      0.43        19
           1       0.30      0.33      0.32         9
           2       0.94      0.94      0.94       216

    accuracy                           0.87       244
   macro avg       0.56      0.56      0.56       244
weighted avg       0.87      0.87      0.87       244
```

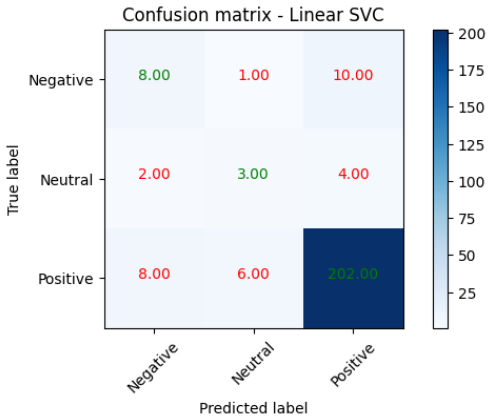


Figure 4.18 Linear SVC Model Evaluation on BOW Features

```
Random Forest Classifier (BOW) - Accuracy: 0.8975409836065574
Random Forest Classifier (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.55      0.32      0.40        19
           1       0.33      0.11      0.17         9
           2       0.92      0.98      0.95       216

    accuracy                           0.90       244
   macro avg       0.60      0.47      0.51       244
weighted avg       0.87      0.90      0.88       244
```

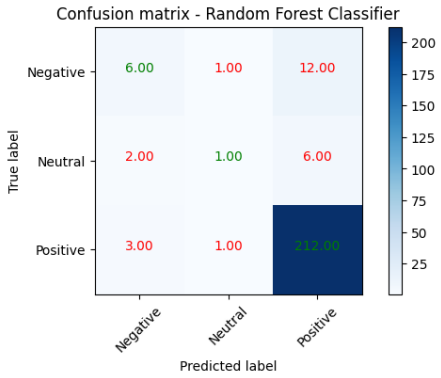


Figure 4.19 Random Forest Classifier Model Evaluation on BOW Features

```
XG Boost (BOW) - Accuracy: 0.9016393442622951
XG Boost (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.62      0.26      0.37        19
           1       1.00      0.11      0.20         9
           2       0.91      0.99      0.95       216

    accuracy                           0.90       244
   macro avg       0.85      0.46      0.51       244
weighted avg       0.89      0.90      0.88       244
```

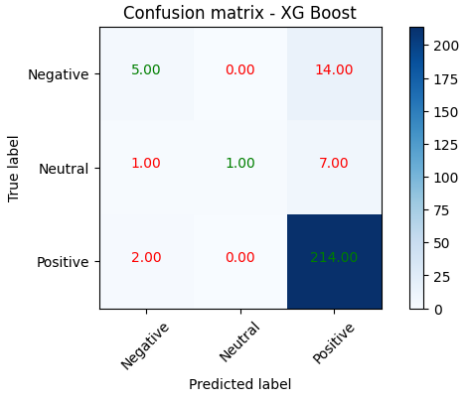


Figure 4.20 XG Boost model Evaluation on BOW Feature

```
Cat Boost (BOW) - Accuracy: 0.8647540983606558
Cat Boost (BOW) - Classification Report:
              precision    recall  f1-score   support

           0       0.36      0.21      0.27        19
           1       0.10      0.11      0.11         9
           2       0.92      0.95      0.94       216

    accuracy                           0.86       244
   macro avg       0.46      0.43      0.44       244
weighted avg       0.85      0.86      0.86       244
```

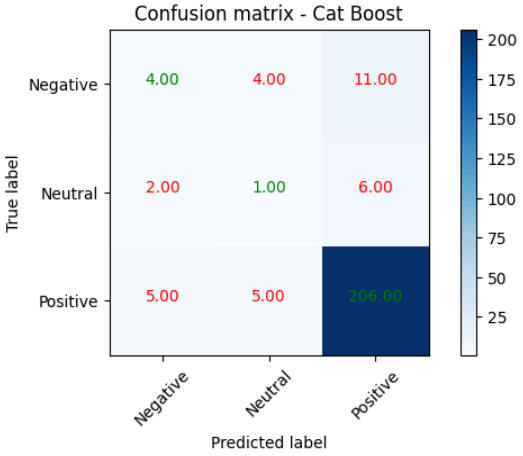


Figure 4.21 Cat Boost Model Evaluation on BOW Features

Summary Table : SMOTE with BOW

	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.881148	0.872385	0.881148	0.875881
Multinomial Naive Bayes	0.885246	0.834912	0.885246	0.848091
Decision Tree Classifier	0.762295	0.831223	0.762295	0.793287
Support Vector Classifier	0.852459	0.855145	0.852459	0.853525
Linear SVC	0.872951	0.873543	0.872951	0.873190
Random Forest Classifier	0.897541	0.870735	0.897541	0.878874
XG Boost Classifier	0.901639	0.891692	0.901639	0.876318
Cat Boost	0.864754	0.849765	0.864754	0.855448

Table 4.8 Summary Table for Model Evaluation on BOW Features

4.3.7.2 ML Models on SMOTE with TF-IDF

Similarly, we applied SMOTE to address the class imbalance and used the TF-IDF representation for training and testing various machine learning models. The models were evaluated based on accuracy, precision, recall, and F1 score to measure their performance in sentiment classification.

```
Logistic Regression (TFIDF) - Accuracy: 0.8811475409836066
Logistic Regression (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.26      0.33        19
           1       0.14      0.11      0.12         9
           2       0.92      0.97      0.95       216

    accuracy                           0.88       244
   macro avg       0.51      0.45      0.47       244
weighted avg       0.86      0.88      0.87       244
```

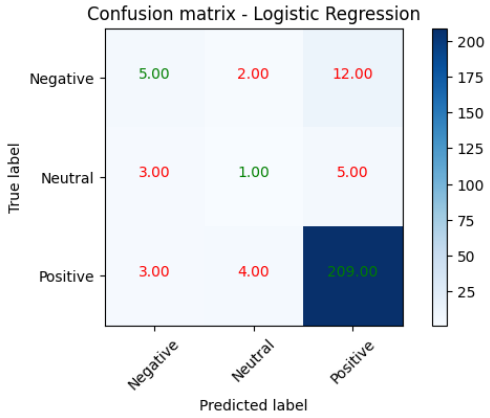


Figure 4.22 Logistic Regression Model Evaluation on TF-IDF Features

```
Multinomial Naive Bayes (TFIDF) - Accuracy: 0.8319672131147541
Multinomial Naive Bayes (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.21      0.21      0.21        19
           1       0.08      0.11      0.10         9
           2       0.93      0.92      0.92       216

    accuracy                           0.83       244
   macro avg       0.41      0.41      0.41       244
weighted avg       0.84      0.83      0.84       244
```

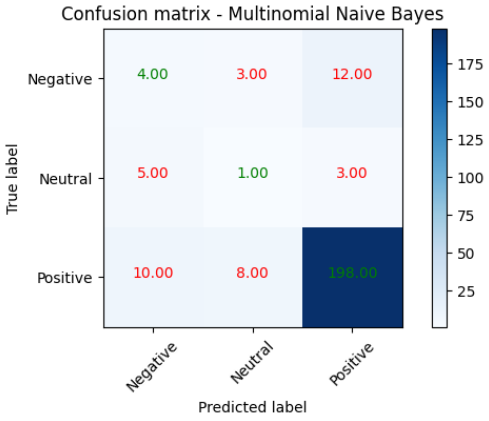


Figure 4.23 Multinomial Naïve Bayes Model Evaluation on TF-IDF Features

```
Decision Tree Classifier (TFIDF) - Accuracy: 0.774590163934426
Decision Tree Classifier (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.21      0.32      0.26        19
           1       0.06      0.11      0.07         9
           2       0.92      0.84      0.88       216

    accuracy                           0.77       244
   macro avg       0.40      0.42      0.40       244
weighted avg       0.83      0.77      0.80       244
```

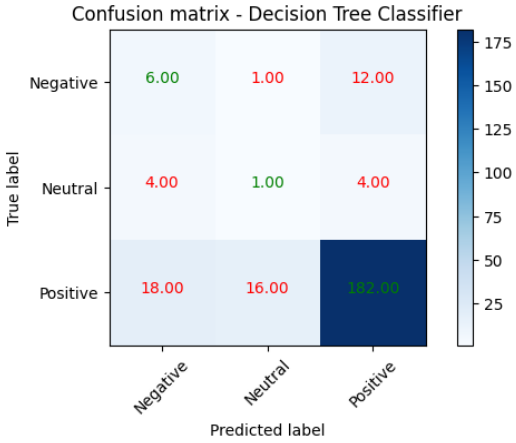


Figure 4.24 Decision Tree Model Evaluation on TF-IDF Features

```
Support Vector Classifier (TFIDF) - Accuracy: 0.8852459016393442
Support Vector Classifier (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        19
           1       0.00      0.00      0.00         9
           2       0.89      1.00      0.94       216

    accuracy                           0.89       244
   macro avg       0.30      0.33      0.31       244
weighted avg       0.78      0.89      0.83       244
```

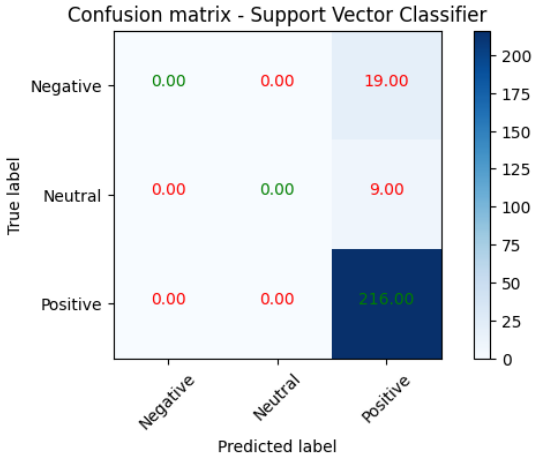


Figure 4.25 Support Vector Classifier Model Evaluation on TF-IDF Feature

```
Linear SVC (TFIDF) - Accuracy: 0.8811475409836066
Linear SVC (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.40      0.21      0.28        19
           1       0.00      0.00      0.00         9
           2       0.91      0.98      0.94       216

    accuracy                           0.88       244
   macro avg       0.44      0.40      0.41       244
weighted avg       0.84      0.88      0.86       244
```

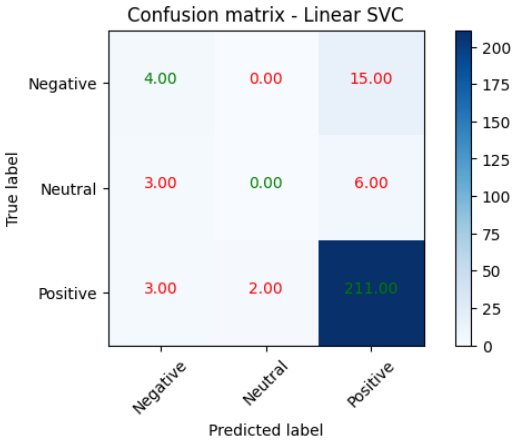


Figure 4.26 Linear SVC Model Evaluation on TF-IDF Feature

```
Random Forest Classifier (TFIDF) - Accuracy: 0.889344262295082
Random Forest Classifier (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.16      0.25        19
           1       0.00      0.00      0.00         9
           2       0.90      0.99      0.94       216

    accuracy                           0.89       244
   macro avg       0.50      0.38      0.40       244
weighted avg       0.84      0.89      0.85       244
```

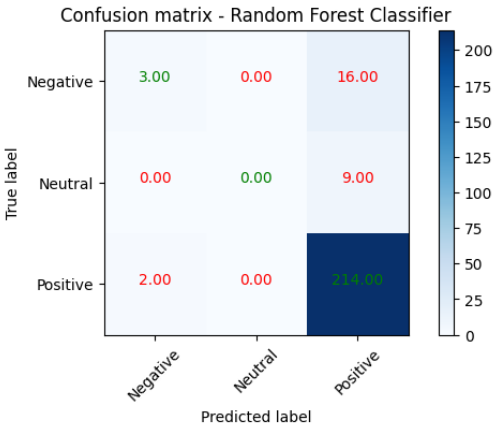


Figure 4.27 Random Forest Classifier Model Evaluation on TF-IDF Features

```
XG Boost Classifier (TFIDF) - Accuracy: 0.889344262295082
XG Boost Classifier (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.40      0.21      0.28        19
           1       1.00      0.11      0.20         9
           2       0.91      0.98      0.94       216

    accuracy                           0.89       244
   macro avg       0.77      0.43      0.47       244
weighted avg       0.87      0.89      0.86       244
```

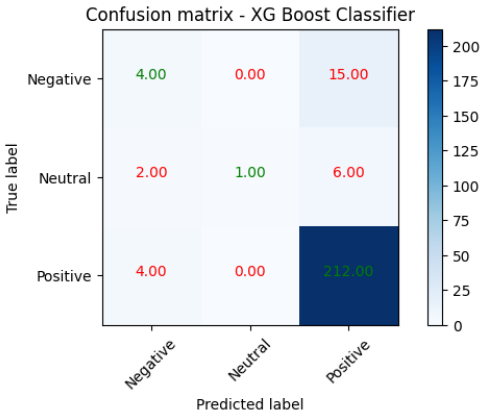


Figure 4.28 XG Boost Classifier Model Evaluation on TF-IDF Features

```
Cat Boost Classifier (TFIDF) - Accuracy: 0.8770491803278688
Cat Boost Classifier (TFIDF) - Classification Report:
              precision    recall  f1-score   support

           0       0.30      0.16      0.21        19
           1       0.00      0.00      0.00         9
           2       0.91      0.98      0.94       216

    accuracy                           0.88       244
   macro avg       0.40      0.38      0.38       244
weighted avg       0.83      0.88      0.85       244
```

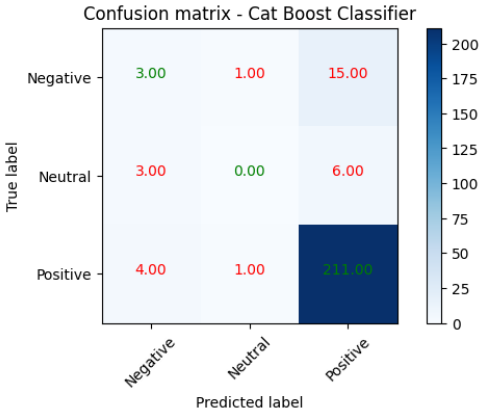


Figure 4.29 Cat Boost Classifier Model Evaluation on TF-IDF Features

Summary Table : SMOTE with TF-IDF

	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.881148	0.859321	0.881148	0.867745
Multinomial Naive Bayes	0.831967	0.842372	0.831967	0.837056
Decision Tree Classifier	0.774590	0.832446	0.774590	0.800946
Suport Vector Classifier	0.885246	0.783660	0.885246	0.831361
Linear SVC	0.881148	0.836263	0.881148	0.855351
Random Forest Classifier	0.889344	0.839368	0.889344	0.852182
XG Boost Classifier	0.889344	0.873493	0.889344	0.864814
Cat Boost Classifier	0.877049	0.828477	0.877049	0.849981

Table 4.9 Summary Table for Model Evaluation on TF-IDF Features

Based on the results obtained from the summary table, we have selected three machine learning models, namely Logistic Regression, Linear SVC, and XG Boost, for further analysis. These models have shown promising performance in terms of accuracy, precision, recall, and

F1 score. Considering the imbalanced nature of our dataset, the selection of these models was primarily based on their high F1 score, which considers both precision and recall. Additionally, we have thoroughly analyzed the classification report to assess the models' performance across various metrics.

Furthermore, upon comparing the performance of these models using both BOW and TF-IDF features, we observed that the models trained on BOW features consistently outperformed the models trained on TF-IDF features. This indicates that the BOW representation captures essential information for sentiment analysis in our dataset more effectively than TF-IDF. Therefore, we will focus our hyperparameter tuning and further analysis on the models trained with BOW features.

By selecting these three models, we aim to explore their capabilities further and optimize their performance through hyperparameter tuning. Each model brings unique strengths and features to the table, and by comparing their results and fine-tuning their parameters, we can identify the best-performing model for sentiment analysis on our dataset.

4.3.8 Hyperparameter Tuning – Grid Search CV

This section aims to optimize the performance of the selected ML models, namely Logistic Regression, Linear SVC, and XG Boost, through hyperparameter tuning. Unlike model parameters, hyperparameters are set prior to the learning process and are not learned from the data. They play a crucial role in controlling the behavior of the machine learning model during training and can significantly influence its performance and ability to generalize.

Examples of hyperparameters include the learning rate, regularization strength, maximum depth of a decision tree, and the number of neighbors in a k-nearest neighbors

algorithm. Unlike model parameters, which are learned from the training data, hyperparameters are set by the user or determined through a search process.

We will employ the Grid Search CV technique adopted from LaValle & Branicky, (2004) to identify the best combination of hyperparameters for each model. Grid Search CV is a systematic approach that exhaustively searches through a predefined set of hyperparameters and evaluates the model's performance using cross-validation. It systematically builds and evaluates multiple models with different hyperparameter combinations, allowing us to find the optimal set of hyperparameters that yield the best performance.

4.3.8.1 Logistic Regression

For hyperparameter tuning, we have considered the following parameters for logistic regression:

Penalty: This hyperparameter determines the type of regularization to be applied in logistic regression. We have tested two options: 'l1' (Lasso) and 'l2' (Ridge). L1 regularization adds a penalty term proportional to the absolute value of the coefficients, while L2 regularization adds a penalty term proportional to the squared magnitude of the coefficients.

Regularization parameter (C): The C parameter controls the inverse of the regularization strength. It balances the trade-off between fitting the training data and preventing overfitting. We have explored three values: 0.1, 1, and 10. Smaller values of C increase the regularization strength, while larger values allow the model to fit the training data more closely.

Max Iterations: This parameter defines the maximum number of iterations for the solver to converge to the optimal solution. We have tested three values: 100, 200, and 500. Increasing the max iterations can improve convergence but also increase the training time.

4.3.8.2 XG Boost Classifier

For hyperparameter tuning of XG Boost, we have considered the following parameters:

Learning Rate: The learning rate controls the step size at each boosting iteration. We have tested three values: 0.1, 0.01, and 0.001. A lower learning rate makes the model learn more slowly but can potentially improve generalization.

Max Depth: This parameter defines the maximum depth of each tree in the boosting process. We have explored three values: 3, 5, and 7. Increasing the max depth can make the model more expressive but also increase the risk of overfitting.

Number of Estimators: This parameter sets the number of boosting rounds or trees to be built. We have tested three values: 100, 300, and 500. It determines the overall complexity and the number of iterations the model will go through during training.

4.3.8.3 Linear SVC

For hyperparameter tuning of Linear SVC, we have focused on the following parameter:

Regularization parameter (C): The C parameter determines the trade-off between the misclassification of training examples and the simplicity of the decision boundary. We have explored three values: 0.1, 1, and 10. Smaller values of C increase the regularization strength and result in a wider margin, potentially sacrificing some training accuracy for improved generalization.

After performing grid search cross-validation with a cross-validation value of 3 on the models trained with BOW features as input, we obtained the following best hyperparameters:

- Logistic Regression Parameters: C: 1, max_iter: 100, penalty: l2
- XG Boost Classifier Parameters: learning_rate: 0.1, max_depth: 5, n_estimators: 300

- Linear SVC Parameters: C: 10

In the next section, we will implement the ensemble Voting classifier using the optimized models: Logistic Regression, XG Boost Classifier, and Linear SVC. This ensemble model will combine the predictions from these individual models using majority voting. By leveraging the strengths of these models and their unique approaches to sentiment analysis, we aim to improve the overall predictive performance.

After building the ensemble Voting classifier, we will evaluate its performance on our dataset. We will assess its accuracy, precision, recall, and F1 score to gauge its effectiveness in predicting sentiment labels. Additionally, we will analyze the confusion matrix and ROC AUC curve to gain insights into the model's performance across different classes.

The confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives for each sentiment class. This information helps us understand how well the ensemble model performs for different sentiment categories.

Furthermore, we will plot the Receiver Operating Characteristic (ROC) curve, which illustrates the trade-off between the true and false positive rates at various classification thresholds. The Area Under the Curve (AUC) score summarizes the performance of the ROC curve and provides a single metric to evaluate the ensemble model's ability to distinguish between positive and negative instances.

By evaluating the ensemble Voting classifier using these metrics and visualizations, we can assess its overall performance and determine its suitability for sentiment analysis on our dataset. Let's proceed with implementing the ensemble model and conducting the performance evaluation.

4.3.9 Ensemble Modelling – Voting Ensemble

In this section, we will explore ensemble modeling techniques adopted from Dietterichl, (2002), specifically the Voting Ensemble, which combines the predictions from multiple individual models to make final predictions. Ensemble modeling is a powerful approach that aims to improve the overall performance and robustness of machine learning models by leveraging the collective intelligence of multiple models.

They are based on the principle of "wisdom of the crowd," where combining the opinions of multiple individuals often leads to better decisions than relying on a single opinion. Similarly, in ensemble modeling, multiple models are trained independently on the same dataset, and their predictions are combined to make the final prediction. This helps mitigate individual models' weaknesses and capture a more comprehensive understanding of the underlying patterns in the data.

One popular ensemble modeling technique is the Voting Ensemble, which combines the predictions from multiple models using a majority voting strategy. The idea is to consider the predictions from each model and select the class label that receives the most votes. This approach is particularly effective when the individual models have diverse characteristics and make different types of errors, as combining their predictions can lead to improved accuracy and generalization.

The Voting Ensemble can be implemented in two main ways: hard voting and soft voting. In hard voting, each model in the ensemble predicts a class label, and the majority class label is selected as the final prediction. In soft voting, each model provides a probability distribution over the class labels, and the class label with the highest average probability across all models is chosen as the final prediction. Soft voting considers the confidence level of each model's prediction, which can lead to better results. By leveraging the collective intelligence

of multiple models, ensemble modeling can help us achieve better results in sentiment analysis and other predictive tasks.

4.3.9.1 Ensemble Modelling – BOW

In this section, we will implement the ensemble modeling technique known as the Voting Ensemble using the BOW representation as input. The BOW approach represents text data by counting the occurrence of words in a document and creating a vector representation based on these word frequencies.

The ensemble classifier using the Bag-of-Words (BOW) representation has achieved the following performance metrics:

```
Ensemble Classifier Metrics:
Accuracy: 0.8934426229508197
Classification Report Ensemble Classifier - BOW :
              precision    recall  f1-score   support

           0       0.55      0.32      0.40        19
           1       0.43      0.33      0.38         9
           2       0.92      0.97      0.95       216

    accuracy                           0.89       244
   macro avg       0.63      0.54      0.57       244
weighted avg       0.88      0.89      0.88       244
```

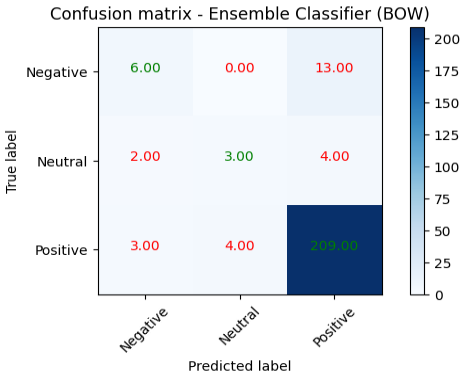


Figure 4.30 Ensemble Voting Classifier on BOW Features

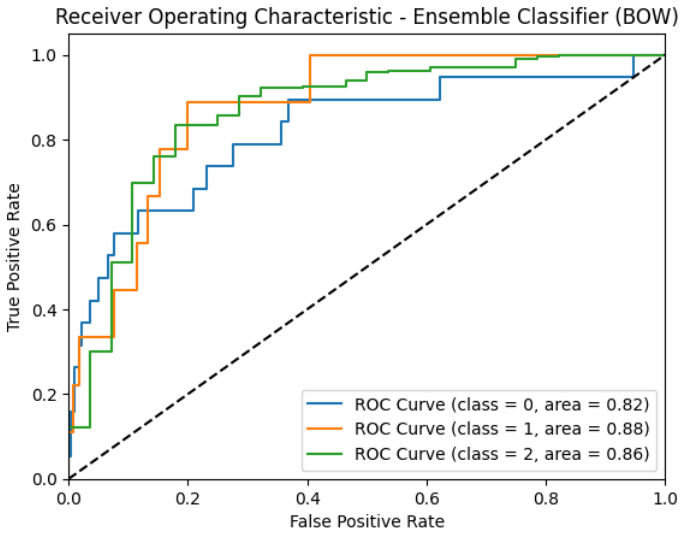


Figure 4.31 ROC Curve for Ensemble Voting classifier on BOW Features

The Voting Ensemble with BOW has been created by combining the predictions from the optimized models: Logistic Regression, XG Boost Classifier, and Linear SVC. Hyperparameter tuning has been performed on these models, and promising performance has been observed individually.

A majority voting strategy has been employed by the Voting Ensemble, where the final decision has been contributed to by each model's prediction. The ensemble prediction has been determined by selecting the class label that has received the most votes from the models

By combining the strengths of multiple models, the overall predictive performance can potentially be improved by the Voting Ensemble compared to the use of a single model. Different aspects of the data can be captured by the ensemble, diverse modeling techniques can be leveraged, and the weaknesses of individual models can be mitigated.

4.4 MODEL DEPLOYMENT

After developing and training our sentiment analysis model, the next step is to deploy it for practical use. Model deployment involves loading the saved text vectorization model and the trained classifier model using the joblib library (Varoquaux & Grisel, 2009), performing data pre-processing, applying text vectorization, making predictions using the machine learning model, and mapping the numerical labels to sentiment tags.

The deployment process can be outlined as follows:

Load the saved Text Vectorized Model: The text vectorization model, such as the Bag-of-Words (BOW) or TF-IDF model, is saved during the training phase. It needs to be loaded to transform the incoming text data into numerical feature vectors.

Load the Trained Classifier Model: The trained sentiment analysis classifier model, such as Logistic Regression, Linear SVC, or XG Boost Classifier, is saved as a serialized file. It needs to be loaded to make predictions on the input data.

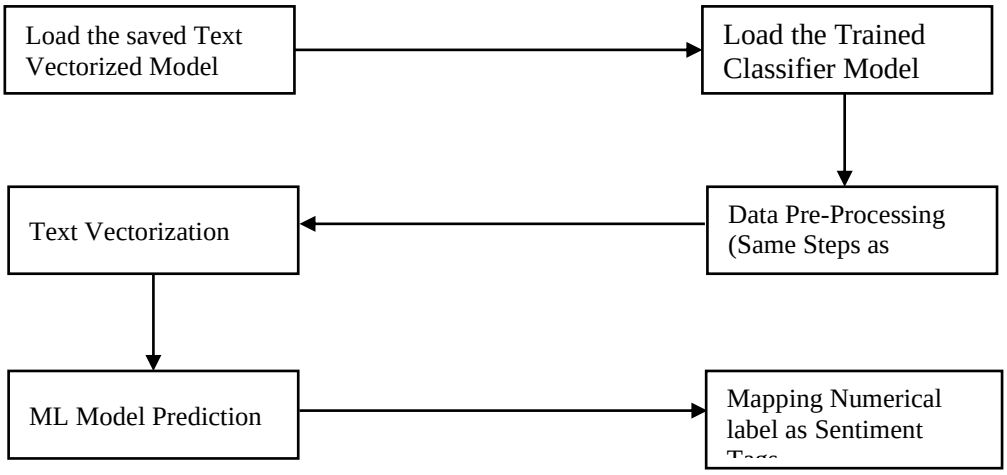


Figure 4.32 Model Deployment Framework

Data Pre-Processing: Perform the same data pre-processing steps as during the training phase. This includes removing noise, tokenization, removing stop words, stemming or lemmatization, and any other necessary pre-processing steps.

Text Vectorization: Apply the loaded text vectorization model to transform the pre-processed text data into numerical feature vectors. This step converts the text data into a format that can be understood by the machine learning model.

ML Model Prediction: Feed the transformed numerical feature vectors into the loaded classifier model to make predictions on the sentiment of the input data. The model will classify the sentiment as positive, negative, or neutral based on the learned patterns and features.

Mapping Numerical Label as Sentiment Tags: Once the predictions are obtained, map the numerical labels back to their corresponding sentiment tags. For example, assign "Positive" to label 1, "Negative" to label -1, and "Neutral" to label 0.

By following these deployment steps, we can apply the trained sentiment analysis model to new and unseen data, allowing us to analyze customer sentiments in real-time and gain valuable insights for decision-making and customer service improvements.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

In this chapter, we draw conclusions based on the findings and outcomes of our study on sentiment analysis. This chapter will provide a comprehensive summary of the key findings, implications, and research contributions. We will discuss the effectiveness of the proposed framework and methodologies in accurately capturing and classifying customer sentiment. Finally, we will outline the potential avenues for future research and development in the field of sentiment analysis in the BFSI sector, particularly concerning to the National Bank.

5.1 Conclusion

In this research project, conducted in three distinct phases, the primary objective was to develop a robust sentiment analysis model specifically tailored for bank call center transcribed data. The main aim of the study was to accurately classify customer sentiments into three categories: positive, negative, or neutral. By achieving this objective, the research sought to provide valuable insights that could be utilized to improve customer satisfaction and facilitate data-driven decision-making within the Banking, Financial Services, and Insurance (BFSI) sector. The ultimate goal was to leverage sentiment analysis techniques to enhance customer experiences and optimize operational processes within the BFSI industry.

In Phase 1, the lexicon-based methods, including VADER, SentiWordNet, and TextBlob, were utilized to generate sentiment labels for the transcribed call conversation input data. The final sentiment label was determined through the application of a majority count approach. Valuable insights into the performance and limitations of lexicon-based techniques were obtained during this phase.

In Phase 2 of the research project, the focus was on the development of a machine-learning model for sentiment classification, utilizing the labeled transcribed data generated in Phase 1. Different text vectorization techniques, including Bag-of-Words (BOW) and TF-IDF, were employed as the methods for text vectorization. These techniques were utilized to convert the textual data into numerical representations, facilitating the subsequent modeling and analysis stages of sentiment classification.

The class imbalance issue in our dataset was addressed by employing the SMOTE algorithm, which effectively balanced the dataset. By utilizing SMOTE, synthetic samples were generated for the minority class, resulting in a more balanced representation of the sentiment labels.

The performance of different machine learning algorithms was evaluated by experimenting with various models, including "*Logistic Regression, Multinomial Naïve Bayes, Support Vector Classifier, Linear Support Vector Classifier, Decision Tree Classifier, Random Forest Classifier, Extreme Gradient Boosting Algorithm, and Categorical Boosting Algorithm*." Upon evaluation, it was found that models such as Logistic Regression, Linear SVC, and XG Boost Classifier, when combined with the BOW representation, yielded the best results for the dataset.

To further optimize the performance of these models, hyperparameter tuning was conducted. This involved the fine-tuning of the parameters of the selected algorithms to enhance their accuracy and generalization capabilities. By adjusting the hyperparameters systematically, the aim was to find the optimal configuration for each model, thereby improving their overall performance on the sentiment classification task.

The aim was to build an ensemble voting classifier to leverage the strengths of individual models and improve the overall performance. The ensemble voting classifier, trained

on the BOW representation, resulted in a robust and accurate sentiment classification system. By combining the predictions of multiple optimized models using majority voting, the following results were achieved: Accuracy of 0.89, Precision of 0.88, Recall of 0.89, and F1 score of 0.88.

Furthermore, the ROC AUC scores demonstrate the classifier's ability to distinguish between sentiment classes. The individual class ROC AUC scores for negative, neutral, and positive sentiments were 0.82, 0.87, and 0.86, respectively. The ensemble classifier achieved a micro-average ROC AUC of 0.9699 and a macro-average ROC AUC of 0.8539, indicating its effectiveness in classifying sentiments across all classes.

In Phase 3, the sentiment analysis model was successfully deployed. The saved text vectorized model and trained classifier model were loaded. Data pre-processing, text vectorization, and sentiment label prediction were performed using the ML model. The numerical labels were then mapped back to sentiment tags to facilitate better interpretation of the results. This phase marked the final stage of the research project, where the developed sentiment analysis model was utilized to analyze new input data and provide meaningful sentiment classifications.

Overall, this research project has successfully developed an effective sentiment analysis model for bank call center transcribed data. The combination of lexicon-based techniques in Phase 1 and machine learning algorithms with optimized parameters in Phase 2 has enabled accurate sentiment classification. The model's insights can help improve customer satisfaction, identify areas for improvement, and make data-driven decisions to enhance bank products and customer service strategies.

5.2 Future Scope

While this research project has made significant progress in developing an effective sentiment analysis model for bank call center transcribed data, there are several avenues for future exploration and improvement. These future directions can enhance the model's performance, address existing limitations, and contribute to the advancement of sentiment analysis in the Banking Financial Services and Insurance sector.

Deep Learning Approaches: Using methods based on deep learning in sentiment analysis has yielded promising results in a variety of NLP tasks. Future research can explore the use of RNNs, LSTM networks (as outlined in Chapter 1), or transformers for sentiment classification in bank call center conversations. These architectures can capture complex patterns and contextual information, potentially enhancing the model's understanding of sentiment nuances.

Domain Adaptation: Adapting the sentiment analysis model to the specific domain of banking can improve its performance in analyzing customer sentiments accurately. Incorporating domain-specific knowledge, such as banking-related lexicons and financial sentiment dictionaries, can enhance the model's capacity to capture the unique characteristics and language used in the banking industry. Further research can explore techniques for domain adaptation and domain-specific sentiment analysis.

Multilingual Sentiment Analysis: In an increasingly globalized banking industry, analyzing customer sentiments in multiple languages becomes crucial. Future research can focus on developing multilingual sentiment analysis models that can effectively classify sentiments in different languages. Techniques such as transfer learning and cross-lingual models can be explored to leverage knowledge from high-resource languages to improve sentiment analysis in low-resource languages.

By pursuing these future research directions, we can further enhance the sentiment analysis model's accuracy, robustness, and applicability in the banking sector. These advancements can provide valuable insights to banks, leading to improved customer satisfaction, enhanced product offerings, and informed decision-making.

REFERENCES

Aggarwal, C. C., & Zhai, C. (2012). *A SURVEY OF TEXT CLASSIFICATION ALGORITHMS*. Springer, Boston, MA. https://doi.org/https://doi.org/10.1007/978-1-4614-3223-4_6

Breiman, L. (2001). RANDOM FORESTS. *Machine Learning*, *45*, 5–32. https://doi.org/10.1023/A:1010933404324

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. In *Journal of Artificial Intelligence Research* (Vol. 16).

Chen, C. C., Huang, H. H., & Chen, H. H. (2020). NLP in FinTech Applications: Past, Present and Future. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.2005.01320

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cox, & David R. (1958). The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, *20*(2), 215–232.

Deng, S., Mitsubuchi, T., Shioda, K., Shimada, T., & Sakurai, A. (2011). Combining technical analysis with sentiment analysis for stock price prediction. *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, 800–807. https://doi.org/10.1109/DASC.2011.138

Dey, L., Mahajan, A., & Haque, S. M. (2009). Document clustering for event identification and trend analysis in market news. *Proceedings of the 7th International Conference on*

Advances in Pattern Recognition, ICAPR 2009, 103–106. https://doi.org/10.1109/ICAPR.2009.84

Dylan Azulay. (2019, May 19). *Chatbots for Customer Service – 4 Current Applications*. Emerj.

Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for Multi-Class Classification: an Overview. *ArXiv*, *abs/2008.05756*. http://arxiv.org/abs/2008.05756

Gupta, A., Dengre, V., Kheruwala, H. A., & Shah, M. (2020). Comprehensive review of text-mining applications in finance. *Financial Innovation*, *6*(1). https://doi.org/10.1186/s40854-020-00205-1

Hazarika, D., Konwar, G., Deb, S., & Bora, D. J. (2020). Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing. *Proceedings of the International Conference on Research in Management & Technovation 2020*, *24*, 63–67. https://doi.org/10.15439/2020km20

Hutto, C. J., & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. http://sentic.net/

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. In *Journal of Documentation* (Vol. 28, Issue 1, pp. 11–21). https://doi.org/10.1108/eb026526

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information (Switzerland)*, *10*(4). https://doi.org/10.3390/info10040150

Kumar, B. S., & Ravi, V. (2016). A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, *114*, 128–147. https://doi.org/10.1016/j.knosys.2016.10.003

Ladicky, L., & Torr, P. H. S. (2011). Locally Linear Support Vector Machines. *Paper Presented at the Meeting of the ICML,Omnipress*, 985–992. http://dblp.uni-trier.de/db/conf/icml/icml2011.html#LadickyT11

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers.

Nopp, C., & Hanbury, A. (2015). Detecting Risks in the Banking System by Sentiment Analysis. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 591–600. https://doi.org/10.18653/v1/D15-1071

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics*, 79–86. https://doi.org/10.3115/1118693.1118704

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. *Paper Presented at the Meeting of the NeurIPS*, 6639–6649. http://arxiv.org/abs/1706.09516

Quinlan, J. R. (1986). Induction of Decision Trees. In *Machine Learning* (Vol. 1).

Sebastiani, F., Baccianella, S., & Esuli, A. (2010). SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *Proceedings of LREC*, *10*. http://sentiwordnet.

Sophia Lam, Charles Chen, Kristi Kim, George Wilson, J. Holt Crews, Matthew S. Gerber, University of Virginia, & Institute of Electrical and Electronics Engineers. (2019). Optimizing Customer-Agent Interactions with Natural Language Processing and Machine Learning. *2019 Systems and Information Engineering Design Symposium (SIEDS) : University of Virginia, Charlottesville, Virginia, USA, 26 April 2019.*, 1–6. https://doi.org/doi: 10.1109/SIEDS.2019.8735616

Tara Ramroop. (2023). *Customer feedback: 7 strategies to collect and leverage it - Zendesk.* Zendesk Blog. https://www.zendesk.com/in/blog/customer-feedback-hear-voice-customer/

Tien Thanh Vu, Quang Thuy Ha, Shu Chang, & Nigel Collier. (2012). An Experiment in Integrating Sentiment Features for Tech Stock Prediction in Twitter. *In Proceedings of the Workshop on Information Extraction and Entity Analytics on Social Media Data*, 23–38. https://www.researchgate.net/publication/270878444

Too, L. H. Y., Souchon, A. L., & Thirkell, P. C. (2001). Relationship Marketing and Customer Loyalty in a Retail Setting: A Dyadic Exploration. *Journal of Marketing Management*, *17*(3–4), 287–319. https://doi.org/10.1362/0267257012652140

Trips Reddy. (2017, October 17). *How Chatbots can Help Reduce Customer Service Cost by 30%*. . IBM Watson [Blog].

Wu, D. D., Zheng, L., & Olson, D. L. (2014). A decision support approach for online stock forum sentiment analysis. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *44*(8), 1077–1087. https://doi.org/10.1109/TSMC.2013.2295353

Yasin, M. M., Kunt, M., Alavi, J., & Zimmerer, T. W. (2004). TQM practices in service organizations: An exploratory study into the implementation, outcome and effectiveness.

Managing Service Quality: An International Journal, *14*(5), 377–389. https://doi.org/10.1108/09604520410557985

Yu, Y., Duan, W., & Cao, Q. (2013). The impact of social and conventional media on firm equity value: A sentiment analysis approach. *Decision Support Systems*, *55*(4), 919–926. https://doi.org/10.1016/j.dss.2012.12.028

Zboja, J. J., & Voorhees, C. M. (2006). The impact of brand trust and satisfaction on retailer repurchase intentions. *Journal of Services Marketing*, *20*(6), 381–390. https://doi.org/10.1108/08876040610691275

Zeithaml, V. A., Berry, L. L., & The, A. P. (1996). Conceptual Framework and Hypotheses Background. In *Journal of Marketing* (Vol. 60).

Zhang, W., & Gao, F. (2011). An improvement to naive bayes for text classification. *Procedia Engineering*, *15*, 2160–2164. https://doi.org/10.1016/j.proeng.2011.08.404

Zhang, Y., Jin, R., & Zhou, Z.-H. (2010). Understanding Bag-of-Words Model: A Statistical Framework. *International Journal of Machine Learning and Cybernetics*, *1*, 43–52. 10.1007/s13042-010-0001-0