Sentimental Analysis and Topic Modeling End to End Application

A PROJECT REPORT

submitted by

CB.EN.U4ELC20011	**Avanish Jha**
CB.EN.U4ELC20056	**Ratneshwar Kumar Bharti**
CB.EN.U4ELC20057	**Ravi Gupta**
CB.EN.U4ELC20060	**Rudraksh Singh**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

IN

ELECTRICAL AND COMPUTER ENGINEERING

AMRITA SCHOOL OF ENGINEERING, COIMBATORE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

APRIL 2024

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING,COIMBATORE -641 112

BONAFIDE CERTIFICATE

This is to certify that this project entitled **"Sentimental Analysis and Topic Modeling End to End Application"** submitted by

CB.EN.U4ELC20011	**Avanish Jha**
CB.EN.U4ELC20056	**Ratneshwar Kumar Bharti**
CB.EN.U4ELC20057	**Ravi Gupta**
CB.EN.U4ELC20060	**Rudraksh Singh**

in partial fulfillment of the requirements for the award of the **Degree of Bachelor of Technology** in **ELECTRICAL & COMPUTER ENGINEERING** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Engineering.

R. Ranjith
Supervisor
Assistant Professor (Sr. Gr.)
Department of Electrical and Electronics Engineering
Amrita School of Engineering
Coimbatore- 641112

Balamurugan S.
Chairperson
Professor
Department of Electrical and Electronics Engineering
Amrita School of Engineering
Coimbatore- 641112

This project report was evaluated by us on……

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

In our rapidly digitizing world, the exponential surge in user-generated content across various platforms demands robust and agile processing tools for extracting actionable insights. Addressing this imperative, our project showcases an avant-garde mobile application that seamlessly integrates advanced Natural Language Processing (NLP) techniques, with a primary focus on Sentiment Analysis (SA) and Topic Modeling.

For Sentiment Analysis, we harness the power of Logistic Regression, a renowned statistical and machine learning method tailored for binary outcomes. By modeling the log odds of the probability of a particular sentiment based on textual features, Logistic Regression offers both efficiency and interpretability. Its strength lies in its ability to handle non-linear relationships, its clarity in presenting the impact of individual words or phrases on sentiment, and its adaptability to various text classification challenges. In a mobile environment, these attributes translate to real-time, accurate sentiment categorizations, empowering businesses with immediate feedback on user sentiments and facilitating data-driven decision-making.

Concurrently, the application's Topic Modeling facet is powered by the Latent Dirichlet Allocation (LDA) method. LDA, a generative probabilistic model, adeptly uncovers concealed thematic structures in extensive text collections. By deciphering the blend of topics within documents and the assortment of words within topics, LDA delivers a nuanced understanding of dominant discussions and narratives.

Preliminary assessments highlight the application's prowess in rendering precise sentiment determinations and identifying cogent topic categorizations. By synthesizing methodologies and insights from leading-edge research, this mobile application stands as a beacon of the confluence of academic rigor and practical utility, setting a benchmark for future endeavors in mobile-centric NLP solutions.

CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP Natural Language Processing

SVM Support Vector Machine

LR Logistic Regression

NB Naive Bayes

Max-Ent Maximum Entropy

LDA Latent Dirichlet Allocation

NMF Non Negative Matrix Factorization

LSA Latent Semantic Analysis

BERT Bidirectional Encoder Representations from Transformers

Chapter 1

INTRODUCTION

1.1 Introduction to Sentiment Analysis (SA)

1.1.1 Background and Context

Sentiment Analysis (SA), colloquially known as opinion mining, emerges from the confluence of Natural Language Processing (NLP) and text analytics. Its primary goal is to discern the sentiment or emotion encapsulated within textual data, whether it's positive, negative, or neutral, and in some advanced systems, even more nuanced emotions like joy or disappointment[1]. As the digital landscape burgeoned with user-generated content from various platforms, ranging from social media to e-commerce websites, the significance of SA has been accentuated.

1.1.2 Significance of SA in Mobile Applications

The proliferation of mobile apps has created new channels for users to express their opinions and feelings. User reviews, ratings, social media mentions, and other unstructured text data contain a wealth of sentiment information. Performing sentiment analysis on this data can uncover trends and patterns that would otherwise be difficult to detect manually.For example, an app developer could analyze user reviews over time to identify major peaks in negative sentiment. This could signal issues with a recent app update that have frustrated users. By drilling down on the specific features or bugs mentioned in those negative reviews, developers can pinpoint areas to fix.

Sentiment analysis can also reveal differences in how users perceive new features or design changes. A common application is comparing sentiment before and after an app redesign. A significant increase in negative sentiment could indicate the update has been received poorly by users. Analyzing the sentiment shift can provide guidance on where the update missed the mark for users.[2]

Overall, sentiment analysis provides mobile app developers an invaluable listening tool. By continuously monitoring user feedback and sentiment patterns, developers can identify problems early, understand user pain points, and gain insights to guide the app optimization process. This ultimately leads to higher user retention, engagement, and satisfaction.

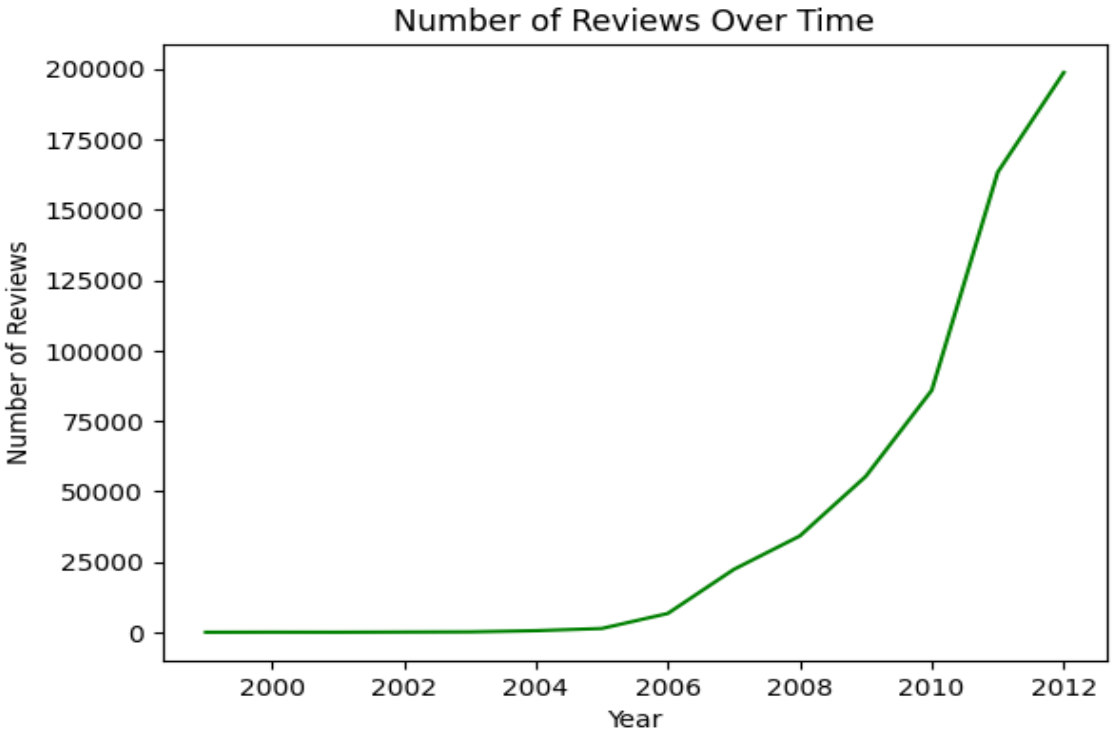

Figure 1.1: Number of Reviews Over Time for Amazon Food Reviews (1999-2012)

1.2 Introduction to Topic Modeling

1.2.1 Background and Context

Topic Modeling is an unsupervised machine learning technique that identifies topics in a large volume of text. By analyzing the co-occurrence patterns of words, it seeks to extract clusters or 'topics' that represent semantic themes present within the documents[5]. Originating from the realms of text mining and Natural Language Processing (NLP), its foundations are rooted in the need to understand and categorize vast textual datasets, which, in the age of digital information, are growing exponentially.

1.2.2 Significance of Topic Modeling in Mobile Applications

Mobile applications, which are a nexus of user interactions and feedback, generate a plethora of textual data. This data, though rich in insights, is often unstructured and vast. Topic Modeling stands as a beacon in this context, helping categorize user feedback into coherent themes or topics, aiding developers in understanding common concerns, praises, or areas of improvement[5]. Beyond feedback, Topic Modeling can also be instrumental in content recommendation within

apps, ensuring users receive information most relevant to their interests.

While Topic Modeling offers promising results, its application in the mobile domain is fraught with challenges. The concise nature of feedback, the dynamism of user interactions, and the evolving nature of app features make the extraction of stable topics challenging. Additionally, understanding the optimal number of topics or ensuring they remain interpretable and distinct can be complex[6]

1.3 Objectives

For this project, the following tasks have to be considered as objectives:

1. Real-time feedback and insights for users.
2. Enhancing business decisions through user feedback analysis.
3. Creation of a mobile application integrating sentiment analysis and topic modeling.

1.4 Report Outline

The report consists of the introductory chapter and the other chapters as follows:

Chapter 2 describes the Algorithms used for SA

Chapter 3 gives a detailed description about the dataset.

Chapter 4 gives Comparison between different algorithms. Results and Analysis is shown in this chapter.

Chapter 2

LITERATURE REVIEW

Sentiment Analysis (SA) and Topic Modeling are pivotal techniques in the realm of Natural Language Processing (NLP), and their importance in extracting valuable insights from user-generated content in mobile applications cannot be overstated. As mobile platforms burgeon with user reviews, feedback, and discussions, effective techniques like Logistic Regression for SA and LDA for Topic Modeling are paramount. This literature survey provides a deep dive into these methodologies based on several key works.

Sentiment Analysis of Product Reviews[1]:

This comprehensive review underscores the significance of understanding consumer sentiments from product reviews. The myriad of methodologies explored in this paper, from traditional machine learning to deep learning techniques, underscores the versatility and challenges of SA. Notably, the effectiveness of logistic regression, a relatively simple yet powerful method for SA, is highlighted.

NLP in Customer Service[2]:

NLP's transformative role in enhancing customer service interactions is explored in this work. The paper elaborates on various methodologies, with a notable mention of logistic regression's efficacy in real-time SA, making it invaluable for immediate feedback systems in customer service platforms.

Comparison of Different Machine Learning Algorithms for Sentiment Analysis[3]:

In an era where a multitude of algorithms exists for SA, this paper's comparative analysis stands out. The strength of logistic regression, particularly its interpretability and efficiency, is discussed. Its performance metrics, in comparison to other algorithms, provide valuable insights for its selection in SA projects.

NLP: Current Trends[4]:

This paper offers a panoramic view of the current trends in NLP. From sentiment analysis to topic modeling and other NLP applications, the authors discuss the challenges and opportunities inherent in the field. The study highlights the rapid advancements in deep learning and their implications in NLP.

Different Topic Modeling Models[5]:

Topic modeling's essence lies in extracting structured topics from vast unstructured data. This research dives deep into various topic modeling techniques, with LDA standing out due to its effectiveness and widespread usage. The paper provides a thorough understanding of LDA's methodology, its applications, and the insights it can offer.

Sentiment Analysis Using VADER and Logistic Regression Techniques[6]:

This study's focus on logistic regression offers a detailed perspective on its application in SA. The comparison with VADER, another sentiment analysis tool, provides a comprehensive understanding of logistic regression's nuances, strengths, and potential areas of improvement.

A Survey on Sentiment Analysis Methods, Applications, and Challenges[7]:

This overarching survey provides a holistic view of SA, discussing from rule-based methods to machine learning techniques. The emphasis on logistic regression's applicability across diverse platforms, from social media to e-commerce, underscores its importance in the SA domain.

In conclusion, the combination of Logistic Regression for Sentiment Analysis and LDA for Topic Modeling promises robust and effective insights, especially in mobile applications. The literature provides a foundational understanding of these techniques, their strengths, and their challenges, guiding the successful implementation of a "Sentiment Analysis & Topic Modeling Mobile Application" project.

Chapter 3

THEORETICAL FRAMEWORK

The theoretical underpinnings of any research endeavor provide the bedrock upon which practical implementations are built. In the realm of text analytics, particularly Sentiment Analysis and Topic Modeling, there exist intricate mathematical and algorithmic foundations that guide the extraction of meaningful insights from textual data.

3.1 Theoretical Foundations of Sentiment Analysis

Sentiment Analysis, at its core, is a classification problem where pieces of text are categorized into predefined sentiment classes[1]. Depending on the granularity required, this can range from binary classification (positive/negative) to multi-class (e.g., positive/neutral/negative) or even multi-label scenarios.

3.1.1 Logistic Regression

One of the most widely-used algorithms for such classification tasks is Logistic Regression. Unlike its name suggests, Logistic Regression is used for binary classification problems, and its extension, Multinomial Logistic Regression, tackles multi-class problems.

Nature of the Algorithm:

- Logistic Regression operates by estimating probabilities using a logistic (sigmoid) function. This ensures that the estimated probabilities are between 0 and 1, making them interpretable as class probabilities.
- The decision boundary in Logistic Regression is linear, which means it works best when the data points of different classes can be separated by a straight line (in 2D), a plane (in 3D), or a hyperplane (in higher dimensions).

Mathematical Formulation:

Given a feature vector x, the probability $P(Y = 1|x)$

$$P(Y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{3.1}$$

Here e is the base of natural logarithms, and B_0 and B_1 are the parameters of the model.

Applications and Insights:

- Beyond its vast applications, Logistic Regression's strength lies in its interpretability. Each feature's weight provides insights into its impact on the sentiment classification[4].

3.1.2 Gradient Boosting

Gradient Boosting is an ensemble learning technique that combines the predictions of multiple weak models, usually decision trees, to create a strong predictive model. It operates in an iterative manner, with each new model correcting the errors of the previous ones, gradually improving overall predictive performance.

Nature of the Algorithm:

- Gradient Boosting is an ensemble learning method that builds a strong predictive model by combining the predictions of multiple weak models, usually decision trees.
- It works in an iterative manner, where each new tree corrects the errors of the previous ones, gradually improving the model's performance.

Mathematical Formulation:

1. **Initialization**:

   Start with an initial estimate, which can be the average of the target values (for regression problems) or the log odds ratio (for classification problems).

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma) \tag{3.2}$$

2. **Iterative Updates**:

   For each stage $m = 1, 2, \ldots, M$, where M is the number of boosting iterations:

(a) **Compute Pseudo-Residuals**:

Calculate the negative gradient (pseudo-residuals) of the loss function $L(y, F)$ with respect to the model predictions F at the previous step.

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{3.3}$$

(b) **Fit a Weak Learner**:

Fit a weak learner (usually a decision tree) to the pseudo-residuals.

$$h_m(x) = fit(x, r_{im}) \tag{3.4}$$

(c) **Compute Multiplier**:

Compute a multiplier γ_m that minimizes the loss when added to the current model.

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)) \tag{3.5}$$

(d) **Update the Model**:

Update the model with the weak learner scaled by the multiplier.

$$F_m(x) = F_{m-1}(x) + \nu\gamma_m h_m(x) \tag{3.6}$$

Here, ν is the learning rate, a parameter that scales the contribution of each tree.

3. **Final Model**:

The final model is a sum of the initial estimate and the contributions from all the weak learners.

$$F_M(x) = F_0(x) + \nu \sum_{m=1}^{M} \gamma_m h_m(x) \tag{3.7}$$

Applications and Insights:

- Gradient Boosting is widely used for both regression and classification problems. It can be applied to detect anomalies in data. It is robust to outliers and can handle complex relationships in data[3].

3.1.3 Support Vector Machines

A supervised learning algorithm that is used for classification and regression. SVM finds the optimal hyperplane that separates data points of different classes in feature space. It aims to maximize the margin between the classes, and it can handle both linear and non-linear decision boundaries through the use of kernel functions.

Nature of the Algorithm:

- SVM is a supervised learning algorithm used for classification and regression.
- It works by finding the hyperplane that best separates the data into different classes while maximizing the margin between the classes.

Mathematical Formulation:

1. **Data Representation**:
   Consider a set of training examples $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where each x_i is a feature vector representing a data point (e.g., a text document) and y_i is the class label (e.g., +1 for positive and -1 for negative sentiment).

2. **Objective Function**:
   The goal is to find a hyperplane defined by the weight vector w and bias b that separates the classes with the maximum margin. The hyperplane can be represented by the equation $w \cdot x + b = 0$.

3. **Optimization Problem**:
   To find the optimal w and b, solve the following optimization problem:

   $$\min_{w,b} \frac{1}{2}\|w\|^2$$

   subject to the constraints:

   $$y_i(w \cdot x_i + b) \geq 1, \quad foralli = 1, \ldots, n$$

4. **Support Vectors**:
   Data points that lie on the margins (defined by $w \cdot x + b = \pm 1$) are called support vectors.

These are the critical elements of the training set as they are the closest to the hyperplane and determine its position and orientation.

5. **Sentiment Classification**:
   Once the model is trained, a new text document x can be classified by evaluating the sign of $w \cdot x + b$. A positive sign indicates a positive sentiment, and a negative sign indicates a negative sentiment.

Applications and Insights:

- SVM is used in image classification tasks. It is employed in text classification problems. It performs well even in high-dimensional spaces. The regularization parameter in SVM helps prevent overfitting.[1].

3.1.4 Random Forest

An ensemble learning method that constructs a collection of decision trees during training. Each tree is trained on a random subset of the data, and a random subset of features is considered at each split. The final prediction is made by aggregating the predictions of all individual trees, providing a robust and accurate model.

Nature of the Algorithm:

- Random Forest is an ensemble learning method that constructs a multitude of decision trees during training.
- It outputs the mode of the classes for classification problems or the mean prediction for regression problems.

Mathematical Formulation:

Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes (classification) of the individual trees.

1. **Training Phase**:

   - Given a training set $X = \{x_1, x_2, ..., x_n\}$ with corresponding target values $Y = \{y_1, y_2, ..., y_n\}$, where each x_i is a feature vector representing a document and y_i is its sentiment label.
   - A number of decision trees are constructed. For each tree:
     (a) A random sample of the training set is selected with replacement (bootstrap sample).
     (b) At each node of the tree, a random subset of features is chosen, and the best split on these features is used to split the node. The process is repeated recursively.

2. **Prediction Phase**:

   - For a new document represented by a feature vector x, each tree in the forest makes a prediction about the sentiment.
   - The final prediction is made based on the majority vote of all the trees in the forest.

$$\hat{y} = mode\{tree_1(x), tree_2(x), ..., tree_k(x)\}$$

   where $tree_i(x)$ is the prediction of the i-th tree.

Note: Random Forest is particularly effective for sentiment analysis because it can handle high-dimensional feature spaces and complex data structures often found in text data.

Applications and Insights:

- It is used in image recognition tasks. It is applied in credit scoring models. It is used for tasks like gene expression analysis. It is less prone to overfitting compared to individual decision trees. It provides a measure of the importance of each feature in the prediction.[2].

Chapter 4

DATA DESCRIPTION

Amazon Fine Food Reviews:

Context

This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all approximating 500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

Data Includes:

- Reviews from Oct 1999 - Oct 2012
- 568,454 reviews
- 256,059 users
- 74,258 products
- 260 users with > 50 reviews

Figure 4.1: Word cloud of the reviews.

Number of Attributes/Columns in data: 10

Attribute Information:

- Id
- ProductId - unique identifier for the product
- UserId - unqiue identifier for the user
- ProfileName
- HelpfulnessNumerator - number of users who found the review helpful
- HelpfulnessDenominator - number of users who indicated whether they found the review helpful or not.
- Score - rating between 1 and 5
- Time - timestamp for the review
- Summary - brief summary of the review
- Text - text of the review

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	5	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	1	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	4	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	2	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	5	1350777600	Great taffy	Great taffy at a great price. There was a wid...

Figure 4.2: First five rows of the Amazon Fine Food Reviews Dataset.

Chapter 5

RESULTS AND ANALYSIS FOR SENTIMENT ANALYSIS MODELS

5.1 Logistic Regression

Accuracy: 86.27%

Classification Report:

Table 5.1: Classification Report for Logistic Regression

	Precision	Recall	F1-Score	Support
Class 0	0.73	0.65	0.69	16181
Class 1	0.51	0.16	0.25	8485
Class 2	0.89	0.97	0.93	89025

Confusion Matrix:

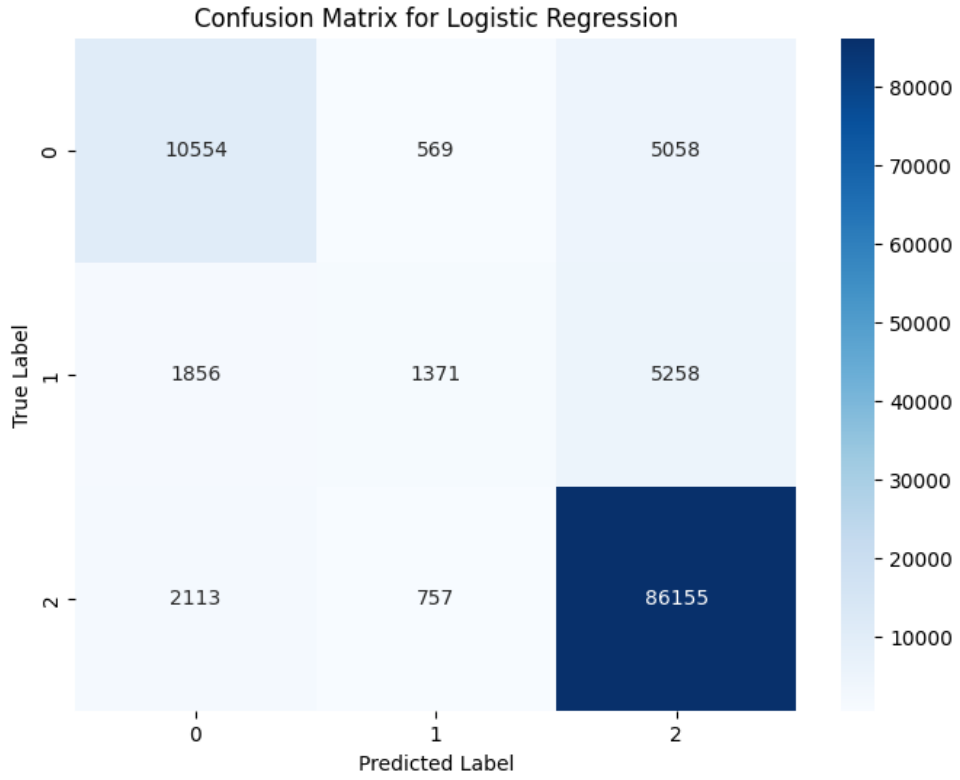

Figure 5.1: Confusion Matrix for Logistic Regression

5.2 Support Vector Machines (SVM)

Accuracy: 86.15%

Classification Report:

Table 5.2: Classification Report for SVM

	Precision	Recall	F1-Score	Support
Class 0	0.72	0.65	0.68	16181
Class 1	0.58	0.09	0.16	8485
Class 2	0.89	0.97	0.93	89025

Confusion Matrix:

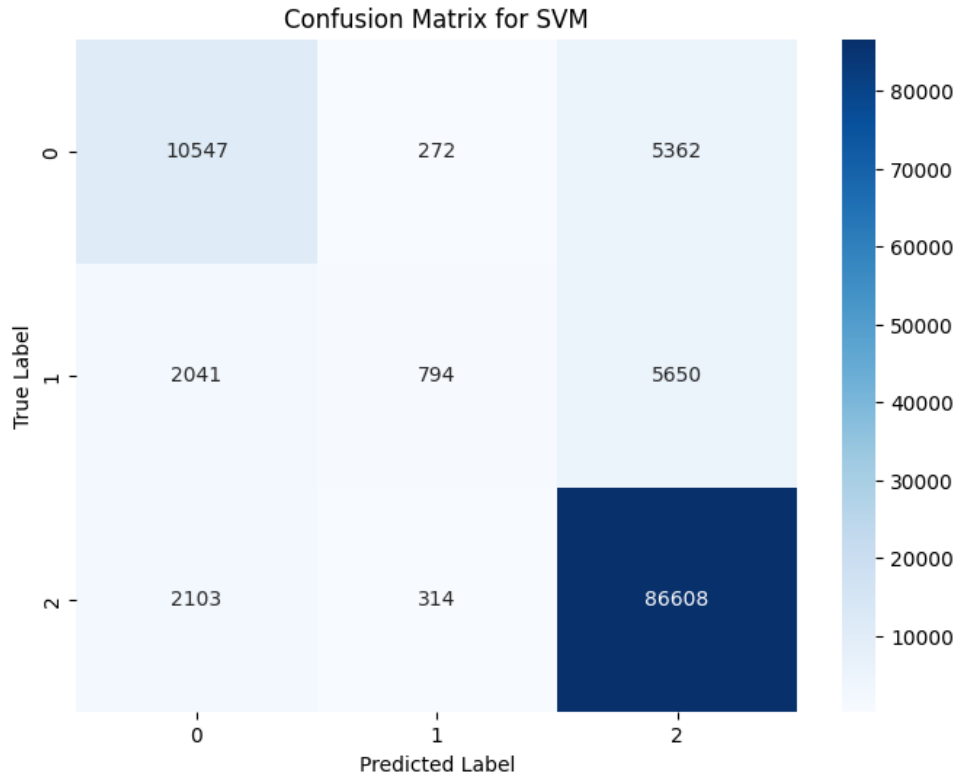

Figure 5.2: Confusion Matrix for SVM

5.3 Gradient Boosting Classifier

Accuracy: 81.91%

Classification Report:

Table 5.3: Classification Report for Gradient Boosting

	Precision	Recall	F1-Score	Support
Class 0	0.81	0.27	0.40	161811
Class 1	0.58	0.04	0.08	8485
Class 2	0.82	0.99	0.90	89025

Confusion Matrix:

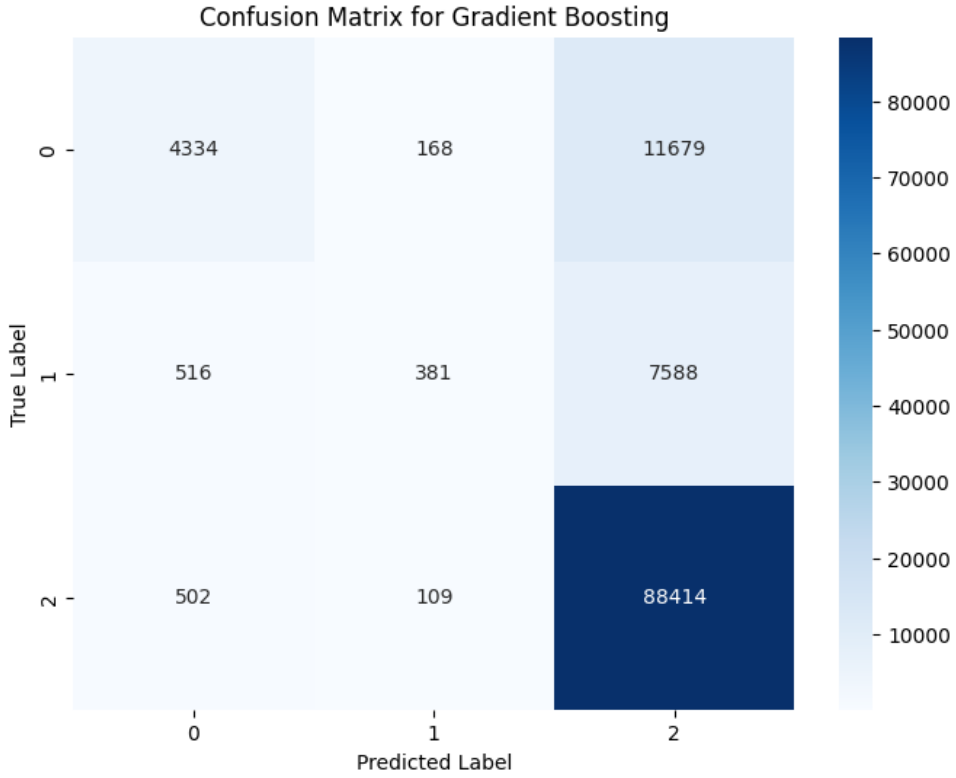

Figure 5.3: Confusion Matrix for Gradient Boosting

5.4 Random Forest Classifier

Accuracy: 89.50%

Classification Report:

Table 5.4: Classification Report for Random Forest

	Precision	Recall	F1-Score	Support
Class 0	0.92	0.61	0.73	16181
Class 1	0.98	0.39	0.55	8485
Class 2	0.89	0.99	0.94	89025

Confusion Matrix:

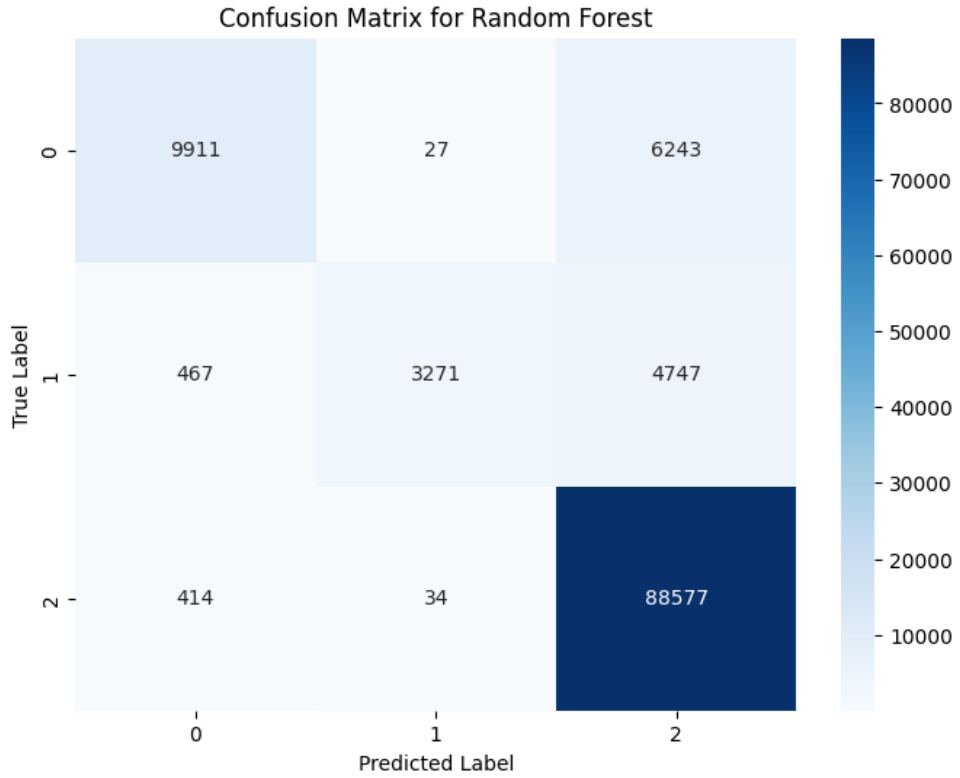

Figure 5.4: Confusion Matrix for Random Forest

5.5 Model Comparison

5.5.1 Model Metrics:

Table 5.5: Model Metrics Comparison

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	86.27%	84.07%	86.27%	84.35%
SVM	86.15%	83.98%	86.15%	83.60%
Gradient Boosting	81.91%	80.14%	81.91%	76.74%
Random Forest	89.50%	90.06%	89.50%	88.15%

5.5.2 Model Comparison Plot:

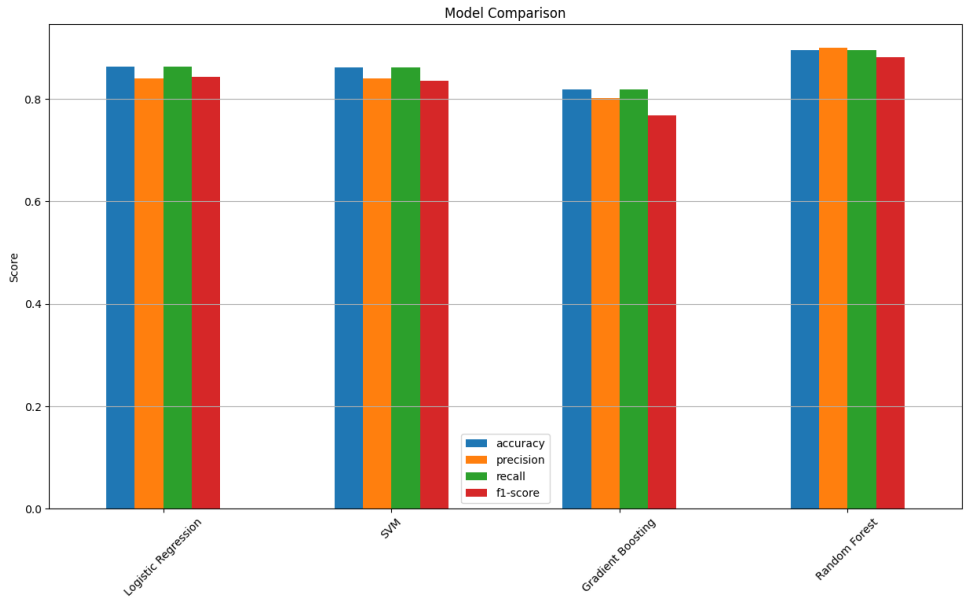

Figure 5.5: Model Comparison Plot

5.6 Feature Importance (Random Forest)

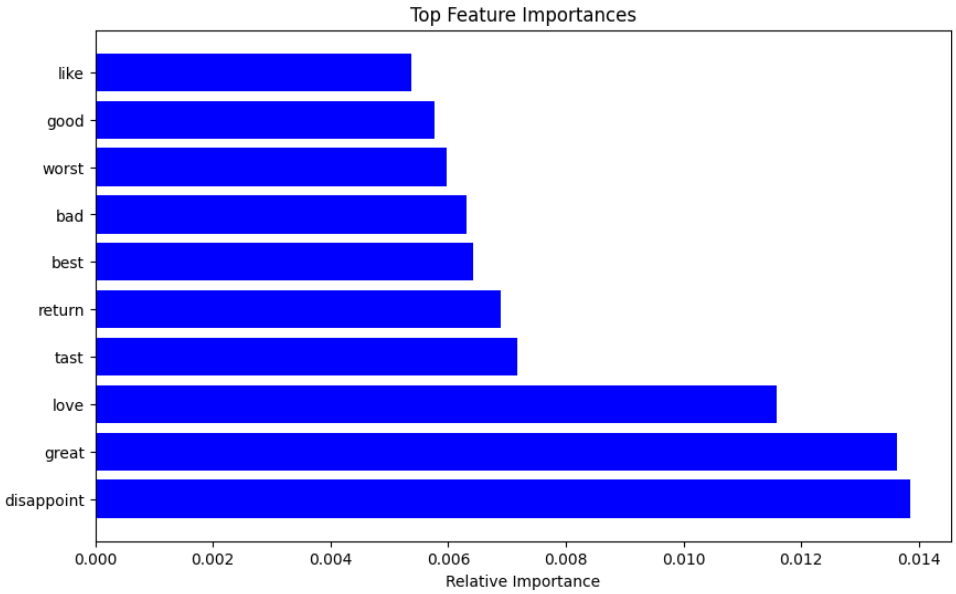

Figure 5.6: Feature Importance for Random Forest

5.7 Conclusion

The presented ensemble learning algorithms—Gradient Boosting, Support Vector Machines (SVM), and Random Forest—stand as formidable tools in the realm of machine learning. Each algorithm possesses unique characteristics that cater to diverse problem domains. Gradient Boosting excels in sequential error correction, SVM provides geometrically motivated hyperplane solutions, and Random Forest showcases robustness against overfitting.

For our specific application, the choice of Random Forest proves particularly compelling. Random Forest's resilience to overfitting, capacity to handle noisy data, and ability to provide insightful feature importance rankings align seamlessly with the challenges of our dataset. In scenarios where interpretability and robust performance are paramount, Random Forest emerges as a pragmatic choice over the other algorithms. As machine learning evolves, Random Forest continues to demonstrate its versatility and reliability, making it a valuable asset in our pursuit of accurate and interpretable predictive modeling.

Chapter 6

FINE-TUNING BERT FOR SENTIMENT CLASSIFICATION

6.1 Introduction

This chapter introduces the process of fine-tuning the BERT (Bidirectional Encoder Representations from Transformers) model for sentiment classification, utilizing a subset of 5000 rows from the Amazon Fine Food Reviews dataset. BERT, developed by Google, has revolutionized the field of natural language processing (NLP) by enabling models to understand the context of words in a sentence more effectively than previous methods. Our objective is to leverage the powerful pre-trained BERT model, adapting it with a smaller, specific dataset to accurately classify reviews into positive or negative sentiments.

The application of BERT for sentiment classification showcases the model's capability to understand complex language nuances, making it an ideal choice for analyzing consumer reviews. By fine-tuning BERT on a targeted dataset, we aim to achieve high accuracy in sentiment analysis, demonstrating the model's versatility and efficiency in processing and understanding human language. This chapter will cover the necessary steps for dataset preparation, model fine-tuning, training, and evaluation, outlining the approach taken to adapt BERT for our specific sentiment classification task.

6.2 BERT Model Overview

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking model in the field of natural language processing (NLP), introduced by researchers at Google in 2018. It represents a significant leap forward in the ability of machines to understand human language due to its unique architecture and training approach.

6.2.1 Architecture

BERT's architecture is based on the transformer model, which relies on attention mechanisms to understand the context of words in a sentence. Unlike traditional models that read text input sequentially (either left-to-right or right-to-left), BERT processes text in both directions simulta-

neously. This bidirectional approach allows BERT to capture a deeper understanding of context and nuance in language.

The model is composed of multiple layers of transformer blocks, with each block containing two main components: a multi-head self-attention mechanism and a fully connected feed-forward network. This design enables BERT to process and relate words to all other words in a sentence, regardless of their positional distance.

6.2.2 Pre-training and Fine-tuning

One of the key features of BERT is its pre-training on a large corpus of text before being fine-tuned for specific tasks. During pre-training, BERT learns language representations by performing two unsupervised tasks:

1. **Masked Language Model (MLM):** Randomly masking words in a sentence and predicting the masked words based on their context. This task helps BERT learn a broad understanding of language structure and vocabulary.

2. **Next Sentence Prediction (NSP):** Given two sentences, BERT predicts whether the second sentence is the logical continuation of the first. This task helps the model understand relationships between sentences.

After pre-training, BERT can be fine-tuned with additional output layers for a wide range of NLP tasks, such as sentiment analysis, question answering, and language inference. Fine-tuning requires relatively less data and computation time compared to training a model from scratch, making BERT highly effective and efficient for specific applications.

6.2.3 Impact

The introduction of BERT has led to significant improvements in the performance of NLP systems on a variety of tasks. Its ability to understand the context and meaning of words in text has enabled more accurate and nuanced language processing, setting new standards for machine understanding of human language.

In summary, BERT's innovative architecture, combined with its pre-training and fine-tuning capabilities, offers a versatile and powerful tool for a wide range of NLP applications. Its impact extends beyond academic research, providing practical benefits in industries where under-

standing human language is crucial, such as customer service, content analysis, and automated recommendations.

6.3 Dataset Preparation and Preprocessing

The Amazon Fine Food Reviews dataset comprises approximately 500,000 food product reviews from Amazon. For this project, we extracted a subset of 5000 reviews to fine-tune the BERT model. The dataset was preprocessed as follows:

1. Sentiment Labeling: Reviews with a score greater than 3 were labeled as positive (1), and the rest as negative (0).
2. Tokenization: The BERT tokenizer was used to tokenize the text, adding special tokens, padding, and creating attention masks necessary for the model.

6.4 Model Fine-Tuning and Training

6.4.1 Loading the Pre-trained BERT Model

The process begins with loading the pre-trained 'bert-base-uncased' model. This model has been pre-trained on a large corpus of English text and understands the language to a significant extent. However, to adapt it to the specific task of sentiment classification, it needs to be fine-tuned. For this task, the model is loaded with a binary classification head on top (since the task is to classify sentiment as positive or negative).

6.4.2 Preparing Data for Training

This demonstrates the preparation of input data for the model, which involves tokenizing the text reviews into a format BERT understands. This includes:

- **Tokenization:** Converting raw text into tokens (words or subwords) that are present in BERT's vocabulary.
- **Attention Masks:** Creating masks to differentiate between the tokens and padding (since each input sequence needs to be of the same length for batch processing).
- **Input IDs:** Mapping tokens to their IDs in BERT's vocabulary.

6.4.3 Data Loaders

To efficiently handle the data during training and validation, data loaders are used. They ensure that the data is fed into the model in batches, making it manageable and speeding up the training process. The notebook sets up separate data loaders for both training and validation datasets, using appropriate batching strategies to optimize memory usage and computational efficiency.

6.5 Model Training

In our approach to fine-tuning the BERT model for sentiment classification, we have employed a detailed and methodical training process. Below outlines the steps we have taken to ensure effective model training:

1. **Initialization of the Pre-trained BERT Model:** We have started by loading the 'bert-base-uncased' model, pre-trained by Google. This model is adapted for binary classification by adding a classification layer on top. Our choice of the 'bert-base-uncased' model is due to its efficiency in processing English text and its proven effectiveness in various NLP tasks.

2. **Data Preparation and Tokenization:** We have tokenized our dataset using BERT's tokenizer, converting the text reviews into a format suitable for the model. This process involves generating input IDs and attention masks, essential for the model to differentiate between real data and padding. Our data preparation step ensures that the model receives input in a consistent and understandable format.

3. **Setting Up Data Loaders:** We have set up data loaders for both the training and validation datasets. These data loaders handle batching and shuffling of the data, making the training process more efficient and manageable. By utilizing data loaders, we have streamlined the process of feeding data into the model for both training and evaluation phases.

4. **Training Loop:** We have conducted the training over multiple epochs, utilizing the AdamW optimizer and a linear learning rate scheduler with warm-up steps. This approach allows for dynamic adjustment of the learning rate, optimizing the training process. Throughout the training, we have calculated the loss and adjusted the model's parameters accordingly to minimize this loss. Our training loop is designed to ensure that the model learns effectively from the training data.

5. **Validation and Performance Evaluation:** Alongside training, we have evaluated the model's performance on a validation set. This step is crucial for assessing the generalization of the model to new data. We have measured the model's accuracy, precision, recall, and F1 score on the validation dataset, providing a comprehensive understanding of its performance.

Our training process has been meticulously designed to fine-tune the BERT model efficiently for the task of sentiment classification. By following these steps, we have leveraged the capabilities of BERT, achieving significant accuracy in classifying sentiments as either positive or negative based on the review text. Our approach demonstrates the practical application of pre-trained models in solving specific NLP tasks with high efficiency and accuracy.

Chapter 7

DEPLOYMENT AND APPLICATION OF SENTIMENT ANALYSIS MODELS

7.1 Introduction

In this chapter, we delve into the practical application of sentiment analysis models, highlighting the process from model deployment to the development of a user-centric mobile application. Emphasizing the transition from theoretical concepts to real-world applications, we explore the deployment of these models on a cloud platform, the design and development of a mobile application using cutting-edge tools, and the seamless integration of the application with the sentiment analysis models. This chapter aims to provide a comprehensive overview of how sentiment analysis can be utilized in mobile applications, offering insights into the technical challenges, solutions, and the impact of these technologies on user experience. Through this exploration, we aim to showcase the potential of sentiment analysis in enhancing mobile app functionalities and user engagement.

7.2 Deployment of Sentiment Analysis Model

The deployment of the Sentiment Analysis Model involves several key steps, starting with preprocessing text to remove special characters and stopwords, then leveraging a trained random forest model and TF-IDF vectorizer hosted on Google Cloud Storage. This process entails downloading these resources, loading them into the application, and applying them to user input to perform sentiment analysis. The code snippet provided outlines this process, including text preprocessing, model loading, prediction, and response formatting, to categorize input reviews into sentiment labels (Negative, Neutral, Positive) with associated probabilities. This deployment strategy highlights the integration of machine learning models with cloud technologies to enable scalable, real-time sentiment analysis in applications.

7.3 Mobile Application Development

The development of the mobile application aimed at providing a user-friendly platform for sentiment analysis involves two main components: UI/UX design using Figma and application development with Flutter.

7.3.1 UI/UX Design with Figma

Figma was employed to design the application's interface, focusing on simplicity and ease of use. The design process involved creating wireframes, user flow diagrams, and interactive prototypes. The goal was to ensure that users could navigate the application intuitively and perform sentiment analysis with minimal effort.

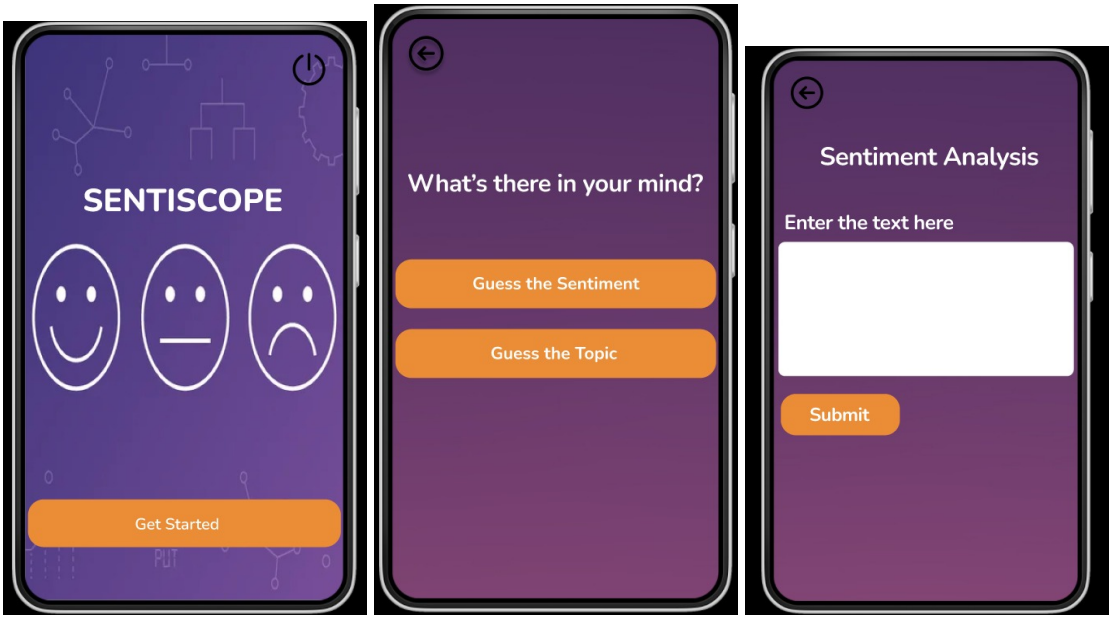

Figure 7.1: Screenshots of the UI/UX design in Figma.

7.3.2 Application Development with Flutter

Flutter, a popular open-source UI software development kit created by Google, was chosen for developing the mobile application. It enables the creation of natively compiled applications for mobile, web, and desktop from a single codebase. This section details the development process, including setting up the Flutter environment, coding the application, and testing on various devices.

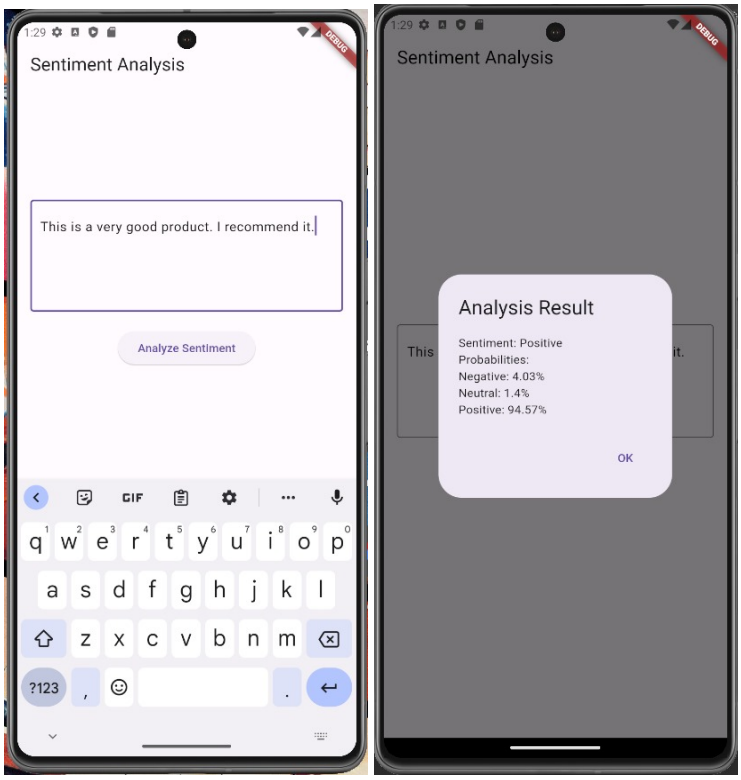

Figure 7.2: Screenshots of the application running on emulator.

The integration of the application with the sentiment analysis model is facilitated through a POST API, allowing users to submit text and receive analysis results directly in the app. This approach highlights the app's capacity to offer real-time sentiment analysis, leveraging cloud-hosted models for backend processing.

Chapter 8

TOPIC MODELING MODELS

8.1 LDA: Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative statistical model that is used to describe collections of discrete data such as text corpora. It's a type of topic model that allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. For example, in the context of text data, these unobserved groups are topics, and the model allows for documents to be represented as mixtures of these topics, which are themselves distributions over words.

8.1.1 Theoretical Framework

The theoretical framework of LDA is grounded in Bayesian statistics. The core idea is that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.

Generative Process

The model assumes the following generative process for each document in a corpus:

1. Choose N: Decide on the number of words N the document will have (according to a Poisson distribution).

2. Choose θ: Choose a topic mixture for the document (over a fixed set of K topics). This is done according to a Dirichlet distribution parameterized by α, resulting in a topic distribution θ for the document.

3. For each of the N words w in the document:

   (a) Choose a topic z: Select a topic z according to the multinomial distribution parameterized by θ.

   (b) Choose a word: From the topic z, select a word w according to the topic's multinomial distribution over the vocabulary, which is parameterized by β.

8.1.2 How It Works

To infer the hidden structure, LDA works backwards from the observed documents to infer the parameters of the generative model (θ for documents, β for topics).

Iterative Update

1. Initialization: Randomly assign each word in each document to one of the K topics.
2. Iterative Update (for Gibbs sampling):
   (a) For each document, and for each word within that document, compute the distribution over topics for that word (conditioned on all other words and their current topic assignments).
   (b) Update the topic assignment for the word, based on this distribution.

8.1.3 Key Components and Parameters

- α: Dirichlet prior on the per-document topic distributions.
- β: Dirichlet prior on the per-topic word distribution.
- θ: Topic distribution for a document.
- β: Word distribution for a topic.
- z: Topic assignments for each word.
- w: Observed words.

8.2 LSI: Latent Semantic Indexing

Latent Semantic Indexing (LSI), also known as Latent Semantic Analysis, is a technique in natural language processing and information retrieval for analyzing relationships between a set of documents and the terms they contain by producing a set of concepts related to the documents and terms. LSI uses singular value decomposition (SVD) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text.

8.2.1 Theoretical Framework

LSI is based on the principle that words that are used in the same contexts tend to have similar meanings. The central idea behind LSI is to map both documents and terms into a latent semantic space where their semantic relationships can be studied.

Mathematical Foundation

The mathematical foundation of LSI lies in linear algebra, particularly in the use of SVD. SVD is used to decompose a term-document matrix into three matrices:

- T, the term matrix, where rows represent terms and columns represent concepts.
- S, the singular value matrix, which is diagonal and contains the singular values that indicate the importance of each concept.
- D^T, the document matrix, where columns represent documents and rows represent concepts.

8.2.2 How It Works

LSI begins by constructing a term-document matrix A where each entry a_{ij} represents the frequency of term i in document j. This matrix is then decomposed using SVD.

Singular Value Decomposition

SVD is applied to the term-document matrix to reduce its dimensionality, leading to a lower-dimensional approximation of the original matrix that captures the most important relationships between terms and documents.

8.2.3 Key Components and Parameters

- **Term-Document Matrix (A)**: A large, sparse matrix where rows represent terms and columns represent documents.
- **Singular Value Decomposition (SVD)**: A mathematical technique used to decompose A into T, S, and D^T.

- **Dimensionality Reduction**: The process of reducing the number of rows and/or columns of the term-document matrix to capture the most significant semantic structures.

8.3 Comparison of LDA and LSI

The performance of Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI) can be compared based on various metrics, with coherence score being one of the most informative. The coherence score measures the degree of semantic similarity between high scoring words in the topic. These scores typically range from 0 to 1, where a higher score means the topic is more coherent.

8.3.1 Coherence Scores

In our analysis, the coherence scores for the two models are as follows:

- **LDA Coherence Score:** The LDA model achieved a coherence score of approximately 0.59. This indicates a relatively high degree of topic coherence, suggesting that the model is able to find meaningful topics.
- **LSI Coherence Score:** The LSI model, on the other hand, obtained a coherence score of approximately 0.42. While this score indicates some level of topic coherence, it is significantly lower than that of the LDA model.

8.3.2 Discussion

The higher coherence score of the LDA model suggests that it may be more effective in capturing the underlying thematic structures in the corpus compared to the LSI model. This difference in performance could be attributed to the generative probabilistic approach of LDA, which allows for a more nuanced representation of document-topic and topic-word distributions. In contrast, LSI relies on linear algebra to decompose the term-document matrix, which may not capture the polysemous and homonymous nature of language as effectively.

Chapter 9

TOPIC MODELING WITH LLAMA 2 (BERTOPIC)

9.1 Introduction

This chapter investigates the utilization of Llama2 for Topic Modeling by adopting an innovative approach that obviates the necessity for individual document processing. We introduce BERTopic, a method distinguished by its modular architecture, which facilitates the incorporation of diverse large language models (LLMs) to refine topic representations. BERTopic's methodology unfolds across five well-defined steps, streamlining the process from document embedding to the extraction of key words that best represent each cluster.

9.2 BERTopic: An Overview

BERTopic leverages the capabilities of Llama2 in a novel topic modeling framework that simplifies the traditional process. Unlike conventional methods that require each document to be fed into the model separately, BERTopic introduces a streamlined, five-step sequence to enhance topic representation efficiency.

9.2.1 Process Overview

The methodology employed by BERTopic is sequential, encompassing the following stages:

1. **Document Embedding:** Initiate the process by embedding documents, transforming textual information into a high-dimensional space.

2. **Dimensionality Reduction:** Apply techniques to reduce the dimensionality of the embeddings, facilitating the identification of patterns within the data.

3. **Clustering Reduced Embeddings:** Cluster the embeddings post-reduction to group similar topics, enhancing the model's interpretability.

4. **Document Tokenization by Cluster:** Tokenize documents based on their assigned clusters, preparing the data for further analysis.

5. **Extraction of Cluster Keywords:** Conclude the process by identifying and extracting the words that best represent the essence of each cluster.

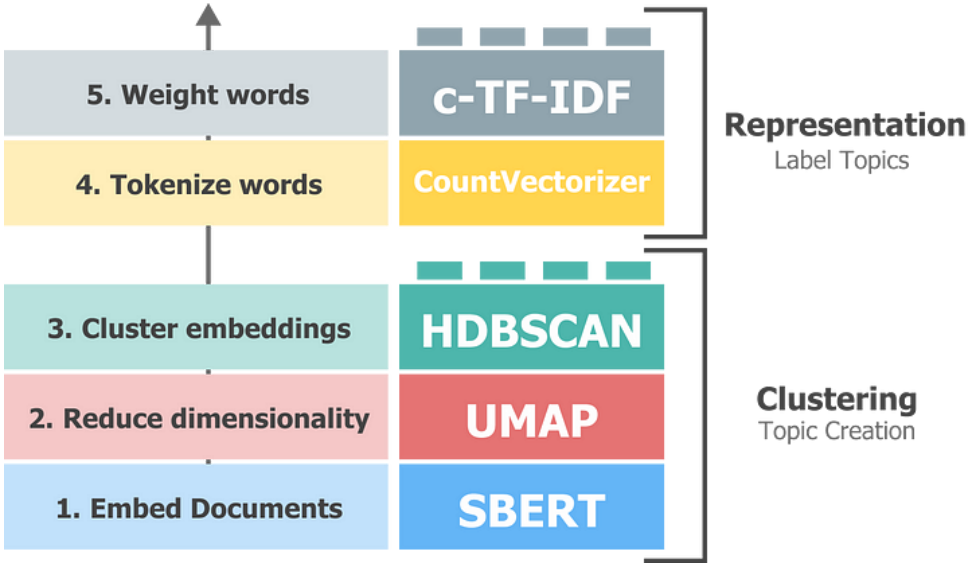

Figure 9.1: The 5 main steps of Bertopic.

9.3 Enhancing Topic Modeling with Llama 2

The emergence of large language models (LLMs) like Llama 2 heralds a new era in the field of topic modeling, providing us with tools to achieve much more nuanced topic representations than ever before. However, the sheer volume of data makes it impractical to process every document directly through Llama 2 due to computational limitations. While vector databases present a partial solution by facilitating document searches, they do not inherently guide us towards the specific topics of interest.

9.3.1 Synergizing BERTopic with Llama 2

To navigate these challenges, we adopt a two-pronged approach. Initially, we generate topics and clusters using BERTopic, a process that efficiently categorizes documents into coherent groups. Subsequently, we employ Llama 2 to refine these preliminary results, leveraging its advanced capabilities to fine-tune and distill the clustered information into more precise and accurate topic representations.

This method represents a fusion of strengths: BERTopic's ability to create distinct topics is complemented by Llama 2's sophisticated enhancement of these topic representations. The

result is a comprehensive topic modeling solution that maximizes the unique advantages of both tools, offering unparalleled insight into the thematic structures within large text corpora.

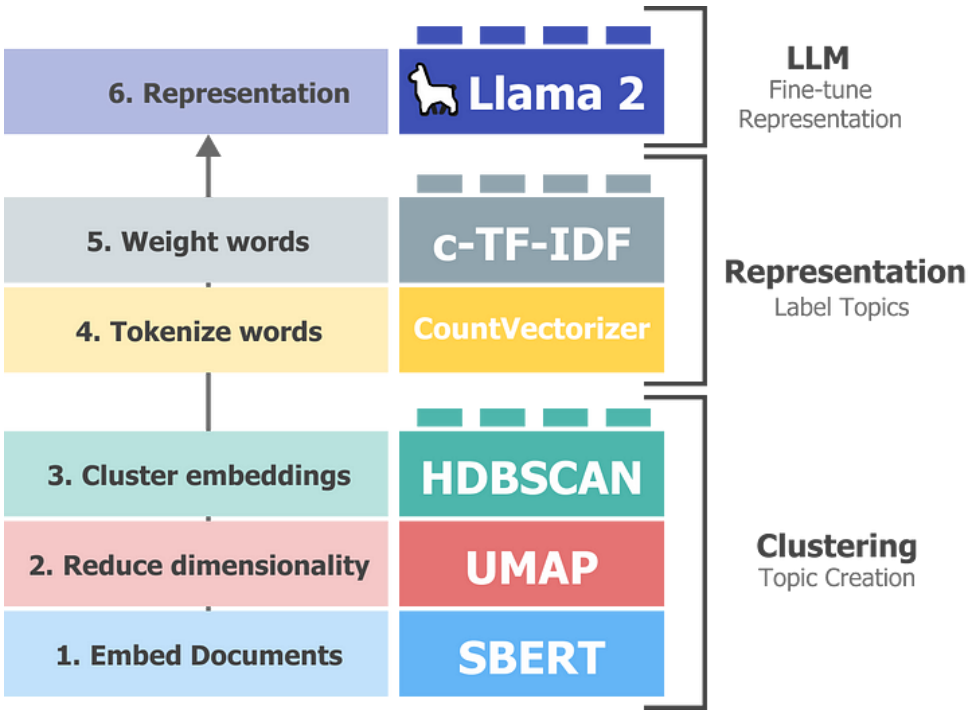

Figure 9.2: Llama 2 lets us fine-tune the topic representations generated by BERTopic.

9.4 Applying Llama 2 for Advanced Topic Modeling

Utilizing the cutting-edge capabilities of Llama 2, this section dives into the practical steps of implementing advanced topic modeling. The journey begins with setting up the environment by installing essential packages. We then prepare our dataset, consisting of ArXiv abstracts, for the modeling process. Authentication with HuggingFace is necessary to gain access to Llama 2, ensuring we comply with usage policies.

9.4.1 Optimization and Setup

To accommodate the large parameters of Llama 2 on available hardware, we employ optimization techniques like 4-bit quantization, significantly reducing memory requirements while maintaining performance. This step is critical for managing the model's computational demands efficiently.

9.4.2 BERTopic Integration

Following the optimization, we integrate BERTopic to generate and refine topics from our dataset. This involves pre-calculating embeddings for our documents and setting up sub-models within BERTopic for an enhanced clustering and labeling process. The integration showcases the synergy between BERTopic's clustering abilities and Llama 2's powerful language understanding, culminating in a sophisticated topic modeling solution.

9.4.3 Visualization and Interpretation

The culmination of our topic modeling effort is visualized through interactive and static methods, providing insightful representations of the topics discovered within the dataset. This visualization not only aids in the interpretation of the model's output but also highlights the nuanced relationships between different topics.

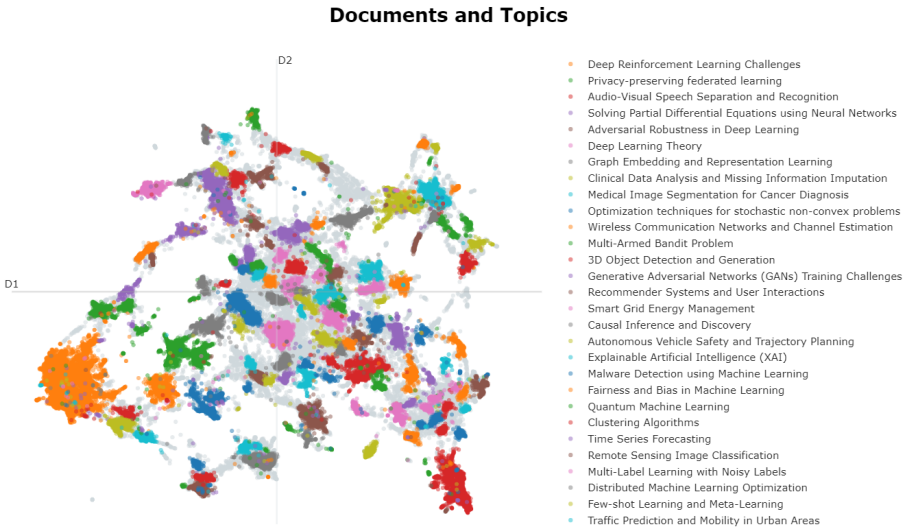

Figure 9.3: Topic Modeling Visualization.

9.5 Coherence Score Analysis of BERTopic

The effectiveness of topic modeling techniques can be quantitatively assessed by their coherence scores, which provide a measure of the semantic similarity between high-scoring words within each topic. BERTopic, in our analysis, has demonstrated a superior performance with a coherence score of 0.66. This score surpasses those obtained by traditional methods such as

Latent Dirichlet Allocation (LDA) and Latent Semantic Indexing (LSI), which scored 0.59 and 0.42, respectively.

9.5.1 Interpreting the Coherence Score

A coherence score of 0.66 for BERTopic implies that the topics generated by the model are significantly more coherent. Words within the same topic are more semantically related, which can lead to a deeper and more accurate understanding of the text corpus under study.

9.5.2 Comparison with Other Models

The improved coherence score suggests that BERTopic is capable of creating more meaningful and distinct topic separations when compared to LDA and LSI. This could be attributed to BERTopic's use of advanced embeddings and clustering techniques, which enhance the model's ability to capture and represent the nuances of large text corpora.

9.6 Conclusion

BERTopic's higher coherence score is indicative of its robustness in topic modeling, making it a preferable choice for researchers and practitioners seeking to extract insightful thematic patterns from complex datasets. The model's ability to generate coherent and thematically rich topics showcases its potential as a powerful tool in the realm of natural language processing.

By leveraging the combined strengths of BERTopic and Llama 2, we present a comprehensive approach to topic modeling that transcends traditional methods. This integration exemplifies how modern language models can be harnessed to uncover deep insights within extensive datasets, marking a significant advancement in the field of natural language processing.

Chapter 10

CONCLUSION

In this report, we've delved into sentiment analysis and ventured beyond, exploring the dynamic landscape of topic modeling. We've dissected the complexities of machine learning models from theoretical foundations to their integration into user-centric applications, culminating in a mobile application fortified by these models. Through this odyssey, the deployment on Google Cloud Platform epitomized the cloud's might in scalable machine learning solutions, while the marriage of Flutter, Figma, and sentiment analysis embodied the fusion of user experience with advanced technology.

Following this trajectory, we expanded our inquiry to encompass Topic Modeling, utilizing state-of-the-art techniques like LDA, LSI, and the sophisticated BERTopic with Llama 2. These methodologies elevated our understanding of textual data, offering nuanced insights through advanced algorithms and visualizations that transcend traditional analysis.

The synergy of BERTopic's clustering finesse with Llama 2's fine-tuning prowess demonstrated a hybrid vigor in extracting rich, coherent topics. The narrative of this report now weaves through the textured fabric of sentiment analysis and the intricate mosaic of topic modeling, each thread a testament to the power and potential of machine learning.

As we stand at the confluence of sentiment detection and topic discovery, we are reminded of the inexhaustible potential for future explorations. Prospects beckon to enhance sentiment analysis with deeper contextual understanding, while topic modeling beckons us towards untapped linguistic dimensions. The confluence of these technologies promises to redefine the bounds of language understanding and application development, charting new territories for artificial intelligence in our interconnected digital epoch.

REFERENCES

[1] S. T. K. Shivaprasad and J. Shetty, ”Sentiment Analysis of Product Reviews: A Review,” NMAM Institute of Technology Nitte, 2017. [Online]. Available: https://doi.org/10.1109/ICICCT.2017.7975207

[2] M. Mashaabi, A. Alotaibi, H. Qudaih, R. Alnashwan, and H. Al-Khalifa, ”Natural Language Processing in Customer Service: A Systematic Review,” King Saud University, Riyadh, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2212.09523

[3] G. Kaur and A. Sharma, ”Comparison of Different Machine Learning Algorithms for Sentiment Analysis,” Symbiosis International University, Pune, 2022. [Online]. Available: https://doi.org/10.1109/ICSCDS53736.2022.9760846

[4] D. Khurana, A. Koli, K. Khatter, and S. Singh, ”Natural language processing: state of the art, current trends, and challenges,” 2022. [Online]. Available: https://doi.org/10.1007/s11042-022-13428-4

[5] G. Papadia, M. Pacella, M. Perrone, and V. Giliberti, ”A Comparison of Different Topic Modeling Methods through a Real Case Study of Italian Customer Care,” Algorithms, vol. 16, no. 94, 2023. [Online]. Available: https://doi.org/10.3390/a16020094

[6] P. Dhanalakshmi, G. A. Kumar, B. S. Satwik, K. Sreeranga, A. T. Sai and G. Jashwanth, ”Sentiment Analysis Using VADER and Logistic Regression Techniques,” 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 139-144, doi: 10.1109/ICISCoIS56541.2023.10100565.

[7] Wankhade, M., Rao, A.C.S., Kulkarni, C. A survey on sentiment analysis methods, applications, and challenges. Artif Intell Rev 55, 5731–5780 (2022). https://doi.org/10.1007/s10462-022-10144-1

[8] M. Grootendorst, “Bertopic: Neural topic modeling with a class-based TF-IDF procedure,” arXiv.org, https://arxiv.org/abs/2203.05794 (accessed Apr. 10, 2024).