Analyzing Apple, Samsung, and Xiaomi Consumer Reviews (2021–2024): Aspect Classification and Sentiment Analysis using BERT, RoBERTa, and DistilBERT



Conduct by:

Name: Abdullah Yahya Moqbel Al-Sabahi

NIM: 21523270

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA

FAKULTAS TEKNOLOGI INDUSTRI

UNIVERSITAS ISLAM INDONESIA

2025

SUPERVISOR ENDORSEMENT PAGE

Analyzing Apple, Samsung, and Xiaomi Consumer Reviews (2021–2024): Aspect Classification and Sentiment Analysis using BERT, RoBERTa, and DistilBERT

THESIS

Written By:

Student Name : Abdullah Yahya Moqbel Al-Sabahi

Student ID : 21523270

Yogyakarta, May 23, 2025

Advisor,

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

EXAMINER ENDORSEMENT PAGE

Analyzing Apple, Samsung, and Xiaomi Consumer Reviews (2021–2024): Aspect Classification and Sentiment Analysis using BERT, RoBERTa, and DistilBERT

THESIS

has been defended in front of the examiner as one of the requirements to obtain a Bachelor of Informatics degree from the Undergraduate Program in Informatics at the Faculty of Industrial Technology, Universitas Islam Indonesia

Yogyakarta, May 23, 2025

Chair

Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.

Examiner 1

(Dr. Raden Teduh Dirgahayu, S.T., M.Sc)

Examiner 2

(Sheila Nurul Huda, S.Kom., M.Cs)

Acknowledged by,

Head of Undergraduate Program in Informatics

Faculty of Industrial Technology

Universitas Islam Indonesia

(Ir. Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

AUTHENTICITY STATEMENT

The undersigned:

Name	:	Abdullah Yahya Moqbel Al-Sabahi
NIM	:	21523270

Final project with title:

Enhancing Aspect-Based Sentiment Analysis with Transformer Models: A Modular and Temporal Evaluation of Product Reviews for Apple, Samsung, and Xiaomi Devices

Stating that all components and contents in this final project are my own work. If in the future it is proven that some parts of this work are not my own work, the final project submitted as my own work is ready to be withdrawn and ready to bear any risks and consequences.

Thus this statement letter is made, hopefully it can be used properly.

Yogyakarta, May 14, 2025



(Abdullah Yahya Moqbel Al-Sabahi)

DEDICATION

This final project is presented to Allah SWT as a form of worship of the author as His servant, to both parents of the author who continue to provide moral and material support to the author, as well as all parties who have accompanied the author's learning process during his undergraduate education.

MOTO

FOREWORD

In the name of Allah, the Most Gracious, the Most Merciful. With deep gratitude and intellectual commitment, I present this final project, which explores Aspect-Based Sentiment Analysis (ABSA) through the lens of transformer-based models and their application to consumer electronics product reviews. This research represents the culmination of a comprehensive inquiry into sentiment trends, aspect-level insights, and temporal dynamics extracted from over 20,000 reviews of smartphones and smartwatches from Apple, Samsung, and Xiaomi, spanning the years 2021 to 2024.

The successful completion of this research would not have been possible without the exceptional support and wisdom of my supervisor, Associate Professor Dhomas Hatta Fudholi. His unwavering guidance, insightful feedback, and deep expertise were instrumental not only in shaping the technical and methodological direction of this work but also in inspiring intellectual growth throughout my academic journey. His patience, encouragement, and commitment to academic excellence have left a lasting impact on my approach to research. I am profoundly grateful for the opportunity to learn under his supervision. I also extend sincere appreciation to the Informatics Study Program at Universitas Islam Indonesia (UII) for cultivating a learning environment that continuously supports innovation in Natural Language Processing (NLP) and applied machine learning.

I extend heartfelt appreciation to my family, whose unwavering support and encouragement have sustained me throughout the journey from data collection and annotation to model fine-tuning and critical analysis. Their presence was instrumental in helping me overcome the challenges associated with conducting multi-model comparisons and longitudinal sentiment analysis. I would also like to express my sincere thanks to Ahmad Fathan Hidayatullah for his valuable guidance and advice.

This research aims to make a meaningful contribution to the field of NLP, particularly in the domains of product analytics, consumer behavior modeling, and aspect-level opinion mining. It is my sincere hope that this work serves as both a reference and inspiration for future research in Aspect-Based Sentiment Analysis, especially in real-world, multilingual, and time-sensitive applications.

Yogyakarta, May 14, 2025

(Abdullah Yahya Moqbel Al-Sabahi)

ABSTRACT

The exponential growth of online consumer reviews has amplified the need for effective analytical techniques to accurately capture user sentiment toward specific product features. This research provides a detailed analysis of Apple, Samsung, and Xiaomi consumer reviews collected from major e-commerce platforms from 2021 to 2024, employing transformer-based Natural Language Processing (NLP) models BERT, RoBERTa, and DistilBERT. Specifically, two critical tasks were conducted separately: Aspect Classification, identifying eight pre-defined product aspects (Battery, Performance, Camera, Display, Price, Multimedia, Fitness & Health Tracking, and Customization), and Overall Sentiment Classification, assigning each review a general polarity (positive, neutral, negative).

Results indicate that RoBERTa achieved the highest accuracy (F1 score of 0.791) in aspect classification tasks, effectively handling nuanced linguistic variations within reviews. In overall sentiment classification, BERT provided the most balanced performance across sentiment classes, whereas DistilBERT, despite lower accuracy, significantly reduced computational overhead with the fastest training times (approximately 6.3 minutes), highlighting an important trade-off between speed and accuracy.

Temporal analysis of over 20,000 reviews from 2021 to 2024 highlights distinct sentiment trends across brands. Apple consistently achieved high user satisfaction, particularly for Performance and Camera aspects, reflecting stable positive sentiment. Samsung demonstrated significant improvements in Fitness & Health Tracking, aligning with evolving consumer preferences for smartwatch features. Xiaomi, while competitive on Price, showed variable sentiment for Battery and Performance, indicating areas for targeted improvement. These findings, derived from a modular approach separating aspect classification and overall sentiment analysis, provide granular insights into consumer preferences over time.

This research contributes methodologically by decoupling aspect classification from sentiment analysis, thereby improving transparency and reducing model complexity. While product aspects were extracted using transformer-based models, sentiment analysis was conducted only at the overall review level. This design choice improves scalability and annotation consistency. Practically, the findings offer actionable insights for product improvement, customer satisfaction management, and competitive analysis within the consumer electronics market.

Keywords: Aspect Classification; Sentiment Analysis; Consumer Reviews; BERT; RoBERTa; DistilBERT; Transformer Models; Temporal Analysis; Consumer Electronics.

GLOSSARY

Term	Definition
ABSA	Aspect-Based Sentiment Analysis – A sentiment analysis technique that identifies specific aspects within a text and determines the sentiment expressed toward each.
ATE	Aspect Term Extraction – The process of identifying and extracting aspect terms or product features mentioned in a review or text.
Aspect Term Polarity	A classification task that assigns a sentiment polarity (positive, negative, or neutral) to each extracted aspect term.
BERT	Bidirectional Encoder Representations from Transformers – A transformer-based deep learning model trained bidirectionally to understand word context.
DistilBERT	A compressed version of BERT that is faster and smaller while retaining most of its performance capabilities.
EDA	Exploratory Data Analysis – A statistical approach used to explore patterns, trends, and anomalies in a dataset.
Embedding	A method for converting words or phrases into numerical vectors for computational processing in machine learning models.
Fine-Tuning	The process of adapting a pre-trained model to a specific task or domain by further training it on targeted data.
NLP	Natural Language Processing – A field of artificial intelligence that focuses on enabling machines to understand, interpret, and generate human language.
RoBERTa	Robustly Optimized BERT Pretraining Approach – An improved version of BERT that removes next sentence prediction and uses dynamic masking.
SA	Sentiment Analysis – A computational method for detecting and classifying sentiment polarity (positive, negative, neutral) in text.
Tokenization	The process of breaking text into smaller units (tokens), such as words or subwords, for analysis by NLP models.
Transformer	A deep learning model architecture based on self-attention mechanisms, used for processing sequential data like text.

Table of Contents

EXAMINER ENDORSEMENT PAGE ii
DEDICATION iv
MOTO v
FOREWORD vi
ABSTRACT vii
GLOSSARY viii
TABLE OF TABLES xii
TABLE OF FIGURES xiii
CHAPTER I 1
1.1 Background of the Study 1
1.2 Problem Statement and Research Questions 1
1.3 Scope and Delimitation 2
Scope 2
Delimitations: 2
1.4 Research Objectives 2
1.5 General Research Methodology 3
1.6 Report Structure 4
CHAPTER II 5
2.1 Aspect Classification and Sentiment Analysis 5
2.2 Transformer Models in ABSA 6
2.2.1 BERT 6
2.2.2 RoBERTa 7
2.2.3 DistilBERT 8
2.3 Modular ABSA Architecture 9
2.4 Challenges in ABSA for Product Reviews 11
2.5 Critical Model Analysis and Research Contribution 12
2.6 Key Contributions of This Research 15
CHAPTER 3 17
3.1 Research Understanding 19
3.2 Data Collection 20
3.2.1 Data Types 20
3.2.2 Estimated Data Volume 21

3.2.3 Web Scraping Approach....21
3.2.4 Data Quality Assurance....22
3.3 Data Preparation....23
3.3.1 Text Preprocessing....24
3.4 Exploratory Data Analysis....33
3.4.1 Review Distribution by Brand....33
3.4.2 Review Distribution by Device Type....34
3.4.3 Combined Brand and Device Distribution....34
3.4.4 Word Frequency and Aspect Validation....35
3.5 Human Annotation Process....36
3.5.1 Aspect Definition and Thematic Analysis....36
3.5.2 Annotation Procedure....38
3.5.3 Inter-Annotator Agreement....39
3.5.5 Annotation Impact....39
3.6 Modeling....42
3.6.1 Model Fine-Tuning....42
3.6.2 Experiments Overview....43
3.6.3 Hybrid Prediction Pipeline....45
3.7 Model Evaluation....47
3.7.1 Evaluation Metrics....47
3.7.2 Evaluation Strategy by Task....48
3.7.3 Overfitting Prevention and Validation Protocol....48
CHAPTER IV....50
4.1 Overview....50
4.2 Results....50
4.2.1 Aspect Term Extraction Task (ATE)....50
4.2.2 Sentiment Analysis Task (SA)....59
4.2.3 Results Analysis....63
4.3 Discussion....65
4.4 Model Performance Results....66
4.4.1 Experimental Evolution....66
4.4.2 Final Modular Model Performance....79
4.4.3 Sentiment Analysis Results (Pretrained Model)....92

4.4.4 Trade-off Summary 93
4.5 Conclusion 93
CHAPTER V 95
5.1 Conclusion 95
5.2 Theoretical Implications 96
5.3 Practical Recommendations 96
5.4 Limitations of the Study 97
5.5 Suggestions for Future Research 97
REFERENCES 98
APPENDIX 104

TABLE OF TABLES

Table 2.1 Comparison of BERT, RoBERTa, and DistilBERT (Novita AI., May 2024).....9
Table 2.2 Common ABSA Challenges and Their Impact 12
Table 2.3 Deep Learning Models for ABSA: A Literature Review 14
Table 3.1 Presents an example of dataset.....21
Table 3.2 Comparative Examples Before and After Normalization 25
Table 3.3 Example on Removing Special Characters 27
Table 3.4 Tokenizer characteristics for all models.....28
Table 3.5 Comparing Tokenization in BERT and RoBERTa 29
Table 3.6 Aspect Definitions Based on Annotated Review Categories 37
Table 3.7 Aspect Categories and Associated Keywords 38
Table 3.8 Sample of Manually Annotated Reviews with Aspects and Overall Sentiment 39
Table 3.9 Summary of Experimental Evolution 44
Table 4.1 Summarizes the frequency of each aspect 51
Table 4.2 Presents the raw counts of key aspects by brand 52
Table 4.3 Shows the aspect breakdown by device category 53
Table 4.4 Shows the top co-mentioned aspect pairs 54
Table 4.5 Summarizes the normalized temporal trends across key periods 55
Table 4.6 Summarizes this relationship 56
Table 4.7 Overall evaluation shows 58
Table 4.8 Aspect-level F1 scores 58
Table 4.9 Training Time 59
Table 4.10 Summarizes the sentiment output per model 59
Table 4.11 Summarizes sentiment distribution per brand 61
Table 4.12 Shows sentiment distribution by device type.....62
Table 4.13 Summarizes sentiment volume across selected aspects 63
Table 4.14 Aspect Term Extraction Performance (F1-Scores by Aspect).....67
Table 4.15 Sentiment Classification Performance (Per Class) 68
Table 4.16 Sentiment Classification Report (BERT) 70
Table 4.17 Sentiment Classification Report (RoBERTa).....71
Table 4.18 Sentiment Classification Report (DistilBERT) 72
Table 4.19 BERT + LoRA Classification Report.....74
Table 4.20 RoBERTa + LoRA Classification Report.....75
Table 4.21 DistilBERT + LoRA Classification Report 77
Table 4.22 Experiments Table.....78
Table 4.23 Trade-off Summary 93

TABLE OF FIGURES

Figure 1.1 Research Workflow 4
Figure 2.1 ABSA Pipeline Diagram 6
Figure 2.2 BERT architecture 7
Figure 2.3 RoBERTa architecture 8
Figure 2.4 Modular ABSA pipeline 10
Figure 2.5 Training Time vs. F1 Score Comparison 15
Figure 3.1 Research Methodology Workflow 19
Figure 3.2 Data Quality Assurance Process 23
Figure 3.3 Overall Workflow of Data Preparation 24
Figure 3.4 Workflow of Text Preprocessing 24
Figure 3.5 Workflow for Removing Punctuation 26
Figure 3.6 Workflow Special Characters 27
Figure 3.7 Tokenization Process Example 30
Figure 3.8 Embedding Process in Transformer Models 32
Figure 3.9 Number of Reviews per Brand 33
Figure 3.10 Review Distribution by Device Type 34
Figure 3.11 Stacked Bar Chart of Reviews by Brand and Device Type 35
Figure 3.12 Word Cloud of Most Frequent Terms in Review Texts 36
Figure 3.13 Human Annotation Workflow 41
Figure 3.14 Aspect Frequency Distribution in Annotated Dataset 41
Figure 3.15 Fine-Tuning Process 46
Figure 3.16 Final Modular ABSA Architecture 46
Figure 3.17 Prediction Framework 49
Figure 4.1 Aspect Frequency Distribution 51
Figure 4.2 Aspect Mentions by Brand 52
Figure 4.3 Aspect Mentions by Device Type 53
Figure 4.4 Aspect Co-occurrence Heatmap 54
Figure 4.5 Monthly Trends of Top Aspects (2021–2024) 55
Figure 4.6 Average Aspect Mentions by Review Length 57
Figure 4.7 Sentiment Distribution per Model 60
Figure 4.8 Sentiment Distribution by Brand 61
Figure 4.9 Sentiment Distribution by Device Type 62
Figure 4.10 Sentiment Distribution by Aspect 63
Figure 4.11 Aspect Train vs Val loss 68
Figure 4.12 Sentiments Train vs Val loss 69
Figure 4.13 Sentiment Confusion Matrix 69
Figure 4.14 BERT Confusion Matrix 70
Figure 4.15 BERT & RoBERTa – Training and Validation Loss and Accuracy 71
Figure 4.16 RoBERTa Confusion Matrix 71
Figure 4.17 DistilBERT Training and Validation Loss 72
Figure 4.18 DistilBERT Confusion Matrix 73
Figure 4.19 BERT + LoRA Train vs Validation Loss 74

Figure 4.20 BERT + LoRA Confusion Matrix....75
Figure 4.21 RoBERTa + LoRA Train vs Validation Loss....76
Figure 4.22 RoBERTa + LoRA Confusion Matrix....76
Figure 4.23 DistilBERT + LoRA Train vs Validation Loss....77
Figure 4.24 DistilBERT + LoRA Confusion Matrix....77
Figure 4.25 Training vs Validation (accuracy and loss) BERT....80
Figure 4.26 Training vs Validation (accuracy and loss) RoBERTa....80
Figure 4.27 Training vs Validation (accuracy and loss) DistilBERT....80
Figure 4.28 Confusion Matrixes for Battery - BERT....81
Figure 4.29 Confusion Matrixes for Camera - BERT....81
Figure 4.30 Confusion Matrixes for Customization - BERT....82
Figure 4.31 Confusion Matrixes for Display - BERT....82
Figure 4.32 Confusion Matrixes for Fitness & Health Tracking - BERT....83
Figure 4 33 Confusion Matrixes for Multimedia - BERT....83
Figure 4.34 Confusion Matrixes for Performance - BERT....84
Figure 4.35 Confusion Matrixes for Price - BERT....84
Figure 4.36 Confusion Matrixes for Battery - RoBERTa....85
Figure 4.37 Confusion Matrixes for Camera - RoBERTa....85
Figure 4 38 Confusion Matrixes for Customization - RoBERTa....86
Figure 4.39 Confusion Matrixes for Display - RoBERTa....86
Figure 4.40 Confusion Matrixes for Fitness & Health Tracking - RoBERTa....87
Figure 4.41Confusion Matrixes for Multimedia - RoBERTa....87
Figure 4.42 Confusion Matrixes for Performance - RoBERTa....88
Figure 4 43 Confusion Matrixes for Price - RoBERTa....88
Figure 4.44 Confusion Matrixes for Battery - DistilBERT....89
Figure 4.45 Confusion Matrixes for Camera - DistilBERT....89
Figure 4.46 Confusion Matrixes for Customization - DistilBERT....90
Figure 4.47 Confusion Matrixes for Display - DistilBERT....90
Figure 4.48 Confusion Matrixes for Fitness & Health Tracking - DistilBERT....91
Figure 4.49 Confusion Matrixes for Multimedia - DistilBERT....91
Figure 4.50 Confusion Matrixes for Performance - DistilBERT....92
Figure 4.51 Confusion Matrixes for Price - DistilBERT....92

CHAPTER I
INTRODUCTION

1.1 Background of the Study

The proliferation of digital platforms has dramatically reshaped consumer behavior, making online reviews a pivotal source of information influencing purchase decisions. It is estimated that over 80% of consumers rely heavily on online reviews to assess product quality, usability, and reliability (Zhang et al., 2021). For consumer electronics, such as smartphones and smartwatches, online reviews offer detailed insights into product strengths and weaknesses, creating a valuable but complex dataset for businesses and researchers.

Traditional sentiment analysis methods typically provide a simplistic, overall sentiment label (positive, neutral, negative), lacking granularity and potentially obscuring valuable insights. To address this limitation, more advanced methods specifically Aspect Classification and Sentiment Analysis are increasingly employed. Aspect classification identifies distinct product features mentioned in reviews, while sentiment analysis assesses the general sentiment expressed, providing clearer insights into consumer opinions (Hu & Liu, 2022).

The rise of transformer-based NLP models, notably BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019), has significantly improved performance in natural language understanding tasks. These models, with their deep contextual representations and computational efficiencies, present significant opportunities to improve accuracy in aspect identification and sentiment classification.

This study focuses explicitly on consumer reviews for smartphones and smartwatches from Apple, Samsung, and Xiaomi, collected between 2021 and 2024 from major e-commerce platforms including Amazon, Walmart, Flipkart, and Best Buy.

1.2 Problem Statement and Research Questions

Despite significant advancements in NLP, extracting precise aspect-level details and accurately classifying sentiment at scale remain challenging. Existing methods often merge these two tasks, particularly in multi-classification scenarios, leading to reduced interpretability and potential inaccuracies in understanding consumer sentiment dynamics (Sun et al., 2019; Rustam et al., 2021).

Thus, this research aims to answer the following key questions:

- RQ1: How effectively can transformer-based models separately perform aspect classification and overall sentiment classification in consumer electronics reviews?

- RQ2: Which transformer model BERT, RoBERTa, or DistilBERT provides the most accurate aspect classification and overall sentiment classification?
- RQ3: What are the most frequently mentioned product aspects, and how does overall consumer sentiment vary across reviews mentioning these aspects for Apple, Samsung, and Xiaomi?
- RQ4: What temporal trends in consumer sentiment can be observed in reviews collected from 2021 to 2024?

1.3 Scope and Delimitation

Scope

This research focuses on consumer reviews for smartphones and smartwatches from Apple, Samsung, and Xiaomi, collected from Amazon, Walmart, Flipkart, and Best Buy between 2021 and 2024. Reviews initially presented in languages other than English were translated into English for consistency and accuracy. Aspect classification covers eight predefined product aspects: Battery, Performance, Camera, Display, Price, Multimedia, Fitness & Health Tracking, and Customization. Sentiment classification is performed at the overall review level rather than at the individual aspect level.

Delimitations:

- Sentiments are assigned at the review level rather than at the sentence or phrase level, which may generalize mixed sentiments within a single review.
- The aspect classification used eight predefined categories, potentially overlooking other emerging or niche aspects.
- Only reviews from four prominent e-commerce platforms (Amazon, Walmart, Flipkart, Best Buy) were analyzed, potentially excluding diverse demographics.

1.4 Research Objectives

The primary objective is to independently assess product aspects and overall consumer sentiment using transformer-based NLP models. Specific objectives include:

- Developing separate models for aspect classification and overall sentiment analysis.
- Comparing performance between BERT, RoBERTa, and DistilBERT.
- Analyzing sentiment distribution and frequency across brands and product types.
- Identifying temporal trends in consumer sentiment from 2021 to 2024.
- Providing practical insights for enhancing product quality and customer satisfaction strategies

1.5 General Research Methodology

The research adhered to a structured and sequential methodology, which is summarized in the following steps and visually represented in Figure 1.1. This figure provides an overview of the research workflow, encompassing the entire process from initial problem identification to final evaluation and interpretation of results:

a. Problem Identification: The study began by recognizing the need for a more nuanced, aspect-specific analysis of consumer sentiment in electronics reviews an area underserved by traditional sentiment models.
b. Literature Review: A comprehensive survey of existing literature was conducted to understand ABSA techniques and the application of transformer models such as BERT, RoBERTa, and DistilBERT in sentiment classification.
c. Data Collection: Over 20,000 reviews of smartphones and smartwatches were collected from Amazon, Flipkart, Walmart, and Best Buy, focusing on Apple, Samsung, and Xiaomi devices released between 2021 and 2024.
d. Preprocessing and Annotation: The dataset was cleaned and normalized. A stratified sample was manually annotated to identify eight predefined aspects and assign sentiment labels (positive, neutral, or negative) to each aspect and the review overall.
e. Model Development and Fine-Tuning: A modular pipeline was implemented. Aspect classification and sentiment classification were handled in two distinct modules using transformer models. Fine-tuning was performed on annotated data to optimize performance.
f. Evaluation and Comparative Analysis: Each model's performance was assessed using accuracy, precision, recall, F1-score, and training time. Results were benchmarked to determine the most suitable model for each task.
g. Insights and Interpretation: Aspect-sentiment patterns were aggregated to identify brand-specific and time-based trends. Visualizations such as heatmaps and frequency charts supported interpretation.

This structured approach ensured methodological rigor and facilitated both model evaluation and meaningful insights from consumer feedback.

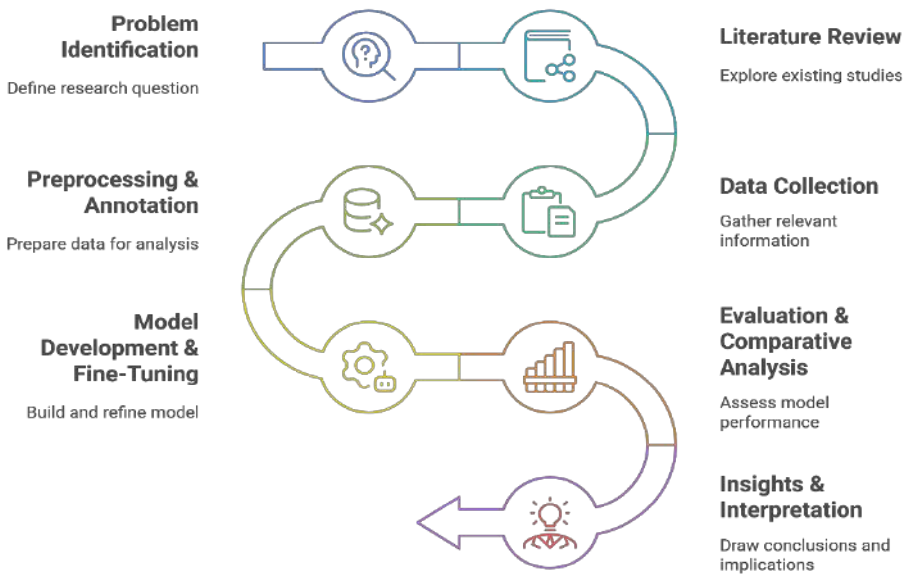


Figure 1.1 Research Workflow: illustrates the overall research workflow, summarizing each methodological phase from problem identification through to model evaluation and analysis.

1.6 Report Structure

This thesis consists of five structured chapters:

- Chapter I: Introduction (background, objectives, scope).
- Chapter II: Literature Review (theoretical foundations, previous research).
- Chapter III: Methodology (detailed methods, data handling).
- Chapter IV: Results and Discussion (model results, insights, comparisons).
- Chapter V: Conclusion (key findings, limitations, future directions).

CHAPTER II
LITERATURE REVIEW

2.1 Aspect Classification and Sentiment Analysis

Aspect Classification and Sentiment Analysis extends traditional sentiment analysis by identifying specific product features within a review and evaluating the overall sentiment expressed. Unlike traditional Aspect-Based Sentiment Analysis (ABSA), which assigns polarity to each aspect (e.g., battery: positive, display: negative), this study separates Aspect Term Extraction (ATE) and Sentiment Classification (SC) into distinct tasks (Zhang et al., 2021). ATE identifies features like battery or display, while SC determines the overall sentiment of the review (positive, neutral, or negative), providing a clearer understanding of consumer opinions in domains like consumer electronics, where reviews often contain mixed sentiments (Hu & Liu, 2022).

This modular approach improves model transparency and reduces complexity, especially in multi-aspect reviews with ambiguous or contrasting sentiments. ATE focuses on extracting product features, while SC assesses the broader emotional tone, allowing for more interpretable results when analyzing user preferences, pain points, and areas of satisfaction or dissatisfaction.

Recent advancements in Natural Language Processing (NLP), particularly transformer-based models, have enhanced the effectiveness of Aspect Classification and Sentiment Analysis. Models like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and DistilBERT (Sanh et al., 2019) provide contextual embeddings that improve both aspect recognition and overall sentiment classification.

To illustrate the framework applied in this study, Figure 2.1 presents the Modular Aspect Classification and Sentiment Analysis Pipeline. As shown, Aspect Term Extraction is handled by fine-tuned transformer models, while Sentiment Classification is conducted using pretrained models. During inference, the outputs of both modules are combined to provide

insights into aspect mentions and their correlation with overall review sentiment, offering a granular and interpretable analysis process.

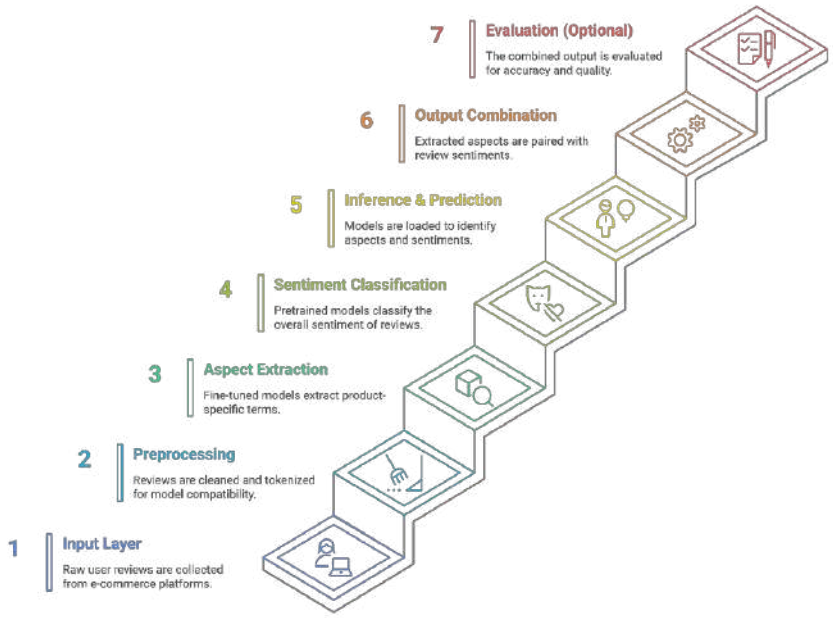


Figure 2.1 ABSA Pipeline Diagram

2.2 Transformer Models in ABSA

To fully understand the capabilities of transformer models within ABSA, we examine individual architectures that contribute to model performance.

2.2.1 BERT

BERT (Bidirectional Encoder Representations from Transformers) was pre-trained on massive corpora using masked language modeling and next sentence prediction. It processes input in both directions, enabling nuanced contextual understanding. BERT uses WordPiece tokenization and a 12-layer architecture in its base form. Its strength lies in capturing complex dependencies between aspect terms and sentiment indicators within reviews (Devlin et al., 2019).

To illustrate BERT's internal structure and tokenization mechanism, Figure 2.2 presents the model's architecture along with the WordPiece tokenization flow. This diagram demonstrates how BERT processes input tokens into embeddings and passes them through

multiple attention-based encoder layers to generate rich, context-aware representations used in downstream tasks such as aspect recognition and sentiment classification.

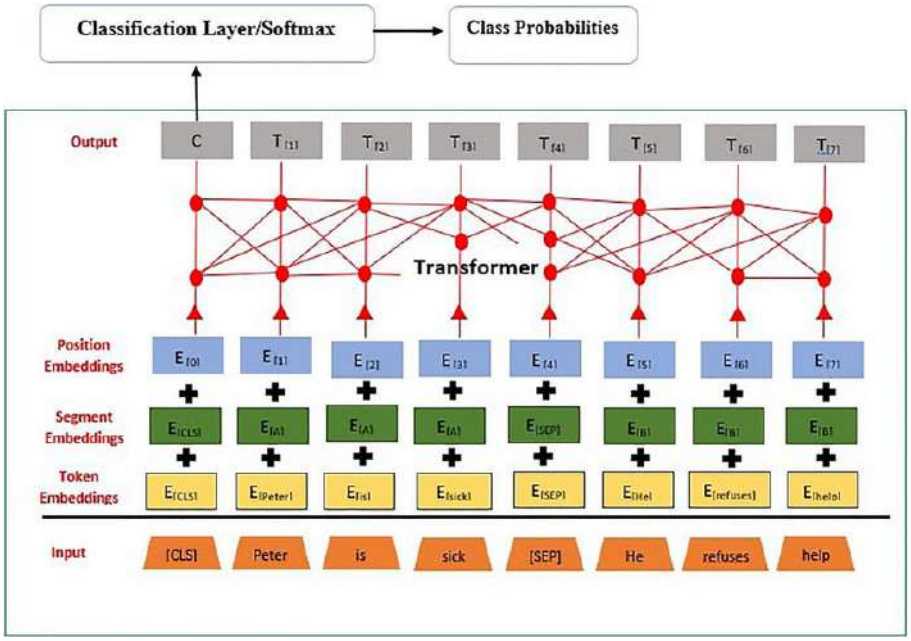


Figure 2.2 BERT architecture with WordPiece tokenization flow (Dinithi Vithanage)

2.2.2 RoBERTa

RoBERTa builds on BERT by removing the next sentence prediction objective, increasing batch size, and training on a more extensive dataset with dynamic masking. It uses Byte-Pair Encoding (BPE) for tokenization and is known for capturing fine-grained semantics in noisy textual data like product reviews (Liu et al., 2019).

Figure 2.3 provides a visual overview of RoBERTa's architecture, highlighting its key structural differences from BERT and emphasizing its capability to better generalize over diverse review contexts encountered in this study.

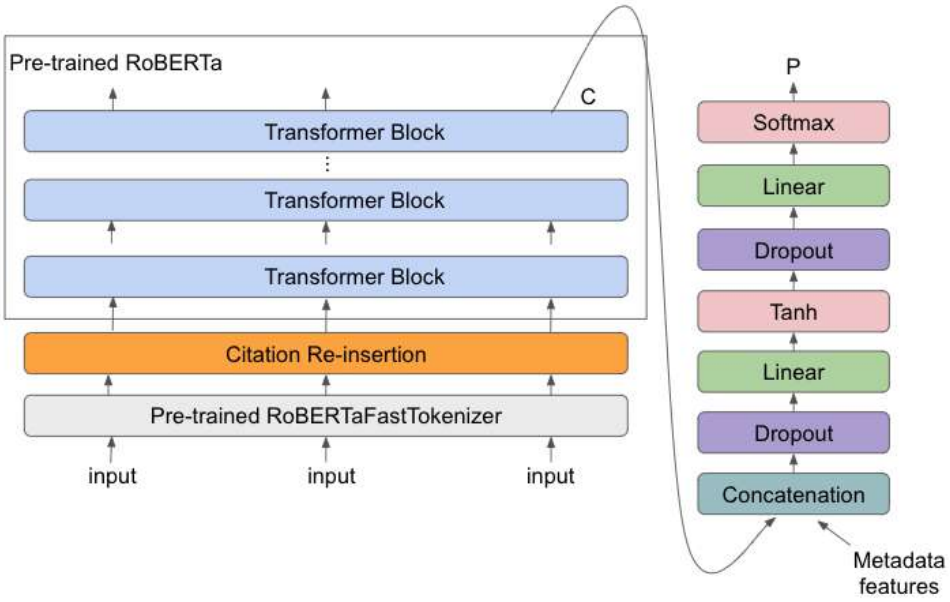


Figure 2.3 RoBERTa architecture (Zihan Huang., Jun 2021)

2.2.3 DistilBERT

DistilBERT is a smaller, faster version of BERT, designed to maintain 97% of BERT's performance while cutting down on computational cost (Sanh et al., 2019). However, due to its reduced complexity, DistilBERT has a tendency to overfit on dominant sentiment classes, particularly in imbalanced datasets.

To compare the performance and efficiency trade-offs among the three models used in this research, Table 2.1 presents a side-by-side comparison of BERT, RoBERTa, and DistilBERT. This comparative visualization underscores RoBERTa's superior accuracy, BERT's robust alignment with sentiment nuances, and DistilBERT's lightweight architecture suitable for faster inference.

Table 2.1 Comparison of BERT, RoBERTa, and DistilBERT (Novita AI., May 2024)

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	Base: 110 Large: 340	Base: 110 Large: 340	Base: 66	Base: ~110 Large: ~340
Training Time	Base: 8 × V100 × 12 days* Large: 64 TPU Chips × 4 days (or 280 × V100 × 1 days*)	Large: 1024 × V100 × 1 day; 4-5 times more than BERT.	Base: 8 × V100 × 3.5 days; 4 times less than BERT.	Large: 512 TPU Chips 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	5% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	Base: 16 GB BERT data Large: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation-based modeling

2.3 Modular ABSA Architecture

While Section 2.2 focused on the individual capabilities and training characteristics of transformer models, this section builds on the discussion of individual models by describing how they are modularly integrated within the ABSA pipeline. In this study, Aspect Term Extraction (ATE) and Sentiment Classification (SC) are treated as separate tasks, enabling better optimization, interpretability, and generalizability (Cambria et al., 2020).

Transformer models are fine-tuned specifically for ATE using annotated datasets, enabling them to identify relevant product features within reviews. Meanwhile, sentiment classification is handled by robust, pretrained models that assign an overall polarity to the review. By decoupling these tasks, the modular approach mitigates task interference and enhances the overall performance and stability of the system.

This modular design not only allows task specialization but also reflects real-world data limitations. Due to the lack of fine-grained sentiment labels per aspect, the sentiment classifier operates at the review level, not aspect level. This design choice was also influenced by scalability needs and annotation efficiency.

The architecture of this modular pipeline is illustrated in Figure 2.4. It shows how aspect terms are extracted independently from sentiment classification, with the latter producing a single polarity label that is applied uniformly to all identified aspects within a review. While this approach simplifies integration, it can result in generalized aspect–sentiment pairings, especially when mixed sentiments are present in the same review.

In the context of this study, the term modules refer specifically to the three transformer-based language models BERT, RoBERTa, and DistilBERT that are independently employed within the modular ABSA pipeline. Each model functions as a self-contained module responsible for either Aspect Term Extraction (ATE) or Sentiment Analysis (SA), allowing for flexible experimentation and task-specific optimization across the sentiment analysis process.

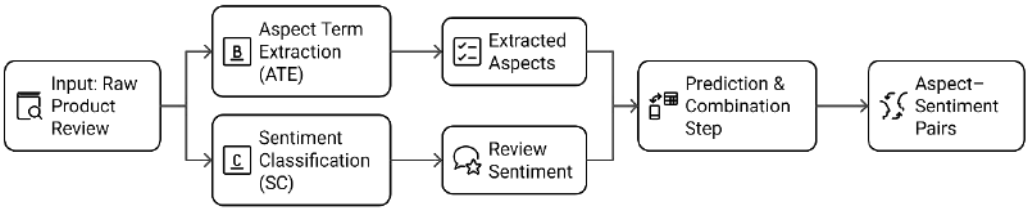


Figure 2.4 Modular ABSA pipeline

Although Figure 1.1, Figure 2.1, Figure 2.4, and Figure 3.1 use different visual formats, they represent a consistent and unified methodology. Figure 1.1 introduces the overall research workflow at a high level, outlining each stage from data collection to evaluation. Figure 2.1 elaborates on this by presenting the ABSA architecture with a focus on modularity specifically the separation of Aspect Term Extraction (ATE) and Sentiment Analysis (SA). Figure 2.4 provides a more technical perspective, detailing the decoupled inference process, where fine-tuned transformer models (BERT, RoBERTa, and DistilBERT) are applied independently for aspect and sentiment tasks before merging results. Lastly, Figure 3.1 revisits the process from a data preparation and experimentation standpoint, tying together implementation stages within the modular ABSA pipeline. Thus, the figures are complementary representations of the same methodological structure and should be interpreted as layered views of the same system.

2.4 Challenges in ABSA for Product Reviews

While the modular ABSA framework enhances flexibility and interpretability, its application to real-world product reviews introduces several persistent challenges. One of the foremost issues is the informal and noisy nature of user-generated content, which frequently contains slang, emojis, abbreviations, typos, and inconsistent grammar. These elements can undermine the effectiveness of tokenization and aspect extraction even after preprocessing (Hu & Liu, 2022).

Another complication arises from mixed sentiments within a single review where users often praise one feature (e.g., “great display”) while criticizing another (e.g., “poor battery life”). When sentiment classification is applied globally rather than per aspect, such contradictions lead to semantic mismatches between aspect terms and assigned sentiments.

Furthermore, aspect ambiguity poses difficulties in both model training and evaluation. Synonymous expressions like “screen,” “display,” or “panel” may refer to the same concept, yet without proper normalization or synonym resolution, these variations can fragment aspect identification and reduce learning effectiveness.

Aspect sparsity also limits performance: certain features such as “Customization” or “Fitness & Health Tracking” appear too infrequently in the data, making it difficult for models to generalize and learn meaningful representations.

Compounding these issues is the lack of aspect-level sentiment annotations in most publicly available datasets. Without granular labels, models are constrained to predicting overall sentiment, reducing the precision and utility of the ABSA pipeline.

Reviews in non-English languages were translated into English before analysis. This language normalization step introduces variability in tone and phrasing, which can affect aspect extraction and sentiment classification.

Review length introduces additional complexity. Short reviews often lack sufficient context and tend to be vague, while longer reviews contain intricate sentence structures and multiple aspects, which increase the likelihood of aspect boundary errors and misclassified sentiment.

Lastly, model-specific challenges were observed in this study. For instance, DistilBERT was prone to overfitting on dominant sentiment classes, leading to biased predictions particularly a tendency to overpredict positive sentiments in multi-class classification tasks where class imbalance is present (Sun et al., 2019; Rustam et al., 2021). Table 2.2 provides a concise overview of these challenges, describing their causes and their specific impact on the ABSA pipeline.

Table 2.2 Common ABSA Challenges and Their Impact

Challenge	Description	Impact on ABSA Pipeline
Informal and Noisy Text	Includes slang, emojis, typos, inconsistent grammar	Decreases model accuracy during tokenization and aspect extraction
Mixed Sentiment in Reviews	Reviews often contain both praise and criticism	Single overall sentiment may not align with each extracted aspect (semantic mismatch)
Aspect Ambiguity	Synonyms or vague terms used for the same concept (e.g., “screen” vs. “display”)	Reduces consistency in aspect identification and frequency tracking
Aspect Sparsity	Rare aspects like "Customization" have limited training examples	Results in poor performance on low-frequency classes in F1 evaluations
Lack of Aspect-Level Sentiment Labels	Most datasets only offer overall sentiment, not per aspect	Forces sentiment classification to remain global, limiting sentiment granularity
Review Length Variation	Short reviews are vague; long reviews are complex and hard to parse	Affects both ATE precision and sentiment pairing accuracy
Model Overfitting and Class Bias	Smaller models tend to overfit or favor dominant classes	Leads to imbalanced predictions skewed toward dominant sentiments

2.5 Critical Model Analysis and Research Contribution

Among the tested models, RoBERTa demonstrated superior performance in extracting aspects due to its pretraining techniques. It effectively handled fine distinctions in feedback on features like battery life or camera performance (Liu et al., 2019).

While BERT offered strong sentiment alignment and DistilBERT was efficient, both had limitations. DistilBERT’s architecture, while lightweight, favored dominant classes in sentiment prediction. Despite these constraints, its speed makes it suitable for real-time applications (Sanh et al., 2019).

The modular ABSA design, separating ATE and SC tasks, reduces task interference and allows better scalability. This method supports adaptability to future extensions such as cross-lingual or multimodal ABSA (Cambria et al., 2020).

A unique contribution of this study is the integration of temporal sentiment dynamics across over 20,000 reviews (2021–2024). This innovation enables longitudinal insights into how sentiment evolves by brand and product type a gap rarely addressed in prior ABSA literature (Xu et al., 2020).

Among the models compared, RoBERTa achieved the highest accuracy in aspect classification, while BERT showed more balanced performance in overall sentiment prediction, especially across neutral and negative labels. These findings help guide future model selection based on task specificity and computational constraints.

To contextualize this study's methodological approach and contributions, Table 2.3 presents a comparative overview of recent deep learning studies in ABSA. These include variations in model architecture, dataset type, and identified limitations, helping to frame the novelty and scalability of the current research.

Table 2.3 Deep Learning Models for ABSA: A Literature Review

Title	Methods Used	Contributions	Limitations	Research Gap
Aspect-Based Sentiment Analysis Using BERT (Zhang et al., 2021)	BERT-based ABSA classifier	Demonstrated effectiveness of BERT for ABSA	Sensitive to noisy, unstructured review data	Limited work on dynamic product sentiment over time
Utilizing Transformer Models for Analysing Fake Reviews (Chen et al., 2022)	DeBERTa for sentiment + fake review detection	Achieved high accuracy in fake review classification	Dataset imbalance affects model generalization	Focused only on deception detection, not fine-grained aspects
Sentiment Classification System Using BiGRU (Ali et al., 2020)	BiGRU-based classification	High accuracy in emotion-rich airline reviews	Low scalability and high computational cost	Did not explore transformer models or modular ABSA design
Improving Performance of ABSA using RoBERTa (Kim et al., 2021)	RoBERTa-based ABSA with CRISP-DM pipeline	RoBERTa outperformed other baselines	Large model size limited deployment	Lacked real-time or temporal sentiment tracking
DistilRoBERTa2GNN Hybrid for ABSA (Singh et al., 2023)	DistilRoBERTa + GNN	Novel hybrid architecture for mixed-domain ABSA	Semantic loss in informal language	Did not address sentiment drift or time-series modeling
Hierarchical LSTM with Attention for Product ABSA (Wang et al., 2021)	H-LSTM with attention mechanism	Improved context sensitivity in ABSA	Complex architecture hard to scale	No modular separation of ATE and SC
Deep CNN for Domain-specific Sentiment Mining (Rahman et al., 2020)	Convolutional networks for domain-adapted ABSA	Achieved good results on niche datasets	Poor generalization beyond training domain	No adaptation to diverse product review formats

To further support comparative performance analysis, Figure 2.5 visually presents each model's trade-off between training time and classification accuracy (F1-score). RoBERTa achieved the highest F1 score (0.791) in aspect classification but required the longest training duration (~11.8 minutes). In contrast, DistilBERT completed training fastest (~6.3 minutes) but with a lower F1 score (0.753). BERT offered a balanced trade-off between performance and efficiency.

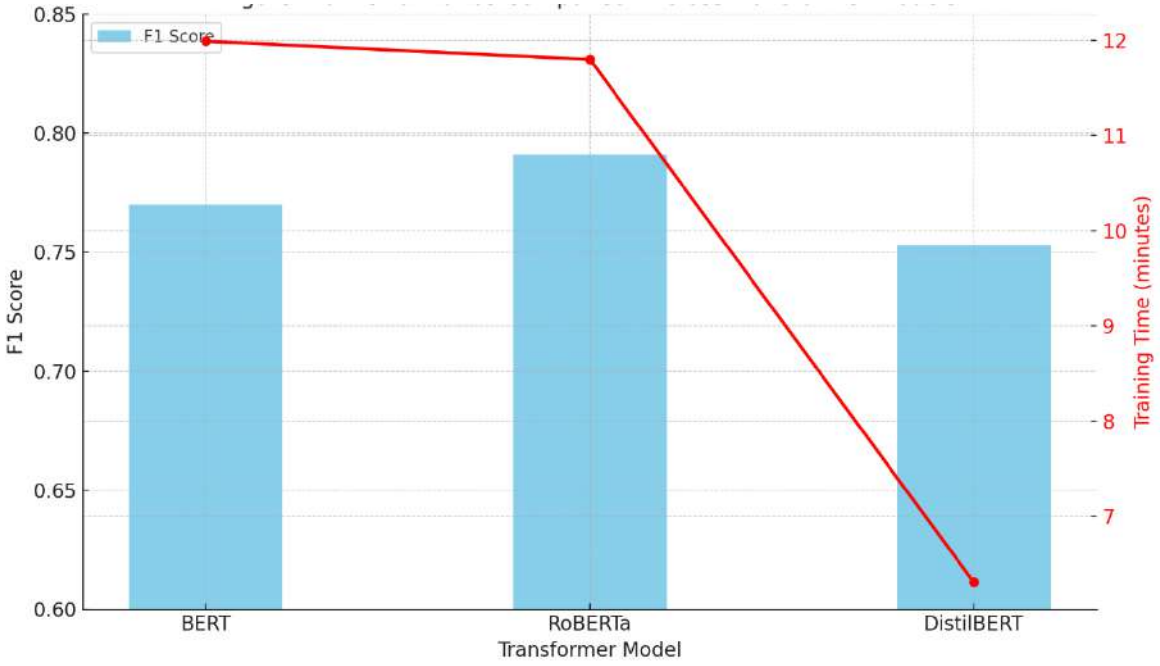


Figure 2.5 Training Time vs. F1 Score Comparison for BERT, RoBERTa, and DistilBERT

2.6 Key Contributions of This Research

- Modular Architecture: Unlike many joint-task ABSA models, this research separates aspect classification and sentiment classification into distinct modules. This approach reduces task interference and enhances system interpretability and maintainability.
- Temporal Sentiment Dynamics: The analysis spans consumer reviews from 2021 to 2024, uncovering longitudinal trends in user sentiment across brands and product types a rarely addressed area in prior ABSA literature.
- Broad Dataset Coverage: The study uses over 20,000 reviews sourced from multiple platforms (Amazon, Flipkart, Walmart, Best Buy), covering diverse writing styles, devices (phones and watches), and brands (Apple, Samsung, Xiaomi).

- Real-World Scalability: The modular structure and evaluation of lightweight models (e.g., DistilBERT) ensure that the proposed system is not only accurate but also deployable in commercial or resource-constrained environments.

This chapter thus establishes the theoretical foundation and positions this study's contributions within the broader ABSA and sentiment classification landscape.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter outlines the systematic methodological approach employed to analyze consumer feedback on recent products from leading consumer electronics brands: Apple's iPhone series (13–15) and Apple Watch series (7–8, Ultra 2), Samsung's Galaxy series (S21–S24) and Galaxy Watch series (4–6), and Xiaomi's Redmi Note series (11–13), Redmi Watch (3 Active, 4), and Redmi Smart Band 8 released between 2021 and 2024. As depicted in Figure 3.1, the structured and comprehensive methodology facilitates accurate extraction and classification of consumer sentiments related to specific product aspects, such as battery life, camera quality, user interface, and design.

Figure 3.1 aligns consistently and logically with the workflows presented previously in Figure 1.1 (Chapter 1) and Figure 2.4 (Chapter 2), reinforcing the structured, modular, and coherent nature of the methodology throughout the study.

Given the vast quantity and complexity of product reviews on platforms like Amazon, Best Buy, and social media, traditional sentiment analysis methods have struggled to capture nuanced aspect-level details effectively. To address this challenge, three state-of-the-art transformer-based models BERT, RoBERTa, and DistilBERT are utilized. Each model brings unique strengths to the ABSA task: BERT provides deep bidirectional context understanding, RoBERTa enhances training stability and predictive accuracy through optimized training objectives, and DistilBERT offers computational efficiency with minimal performance loss.

These models independently perform two core tasks: Aspect Extraction, which identifies and categorizes the key product aspects mentioned in reviews, and Aspect Sentiment Classification, which determines sentiment polarity (positive, neutral, negative) associated with each identified aspect. Leveraging these complementary models ensures robust handling of linguistic variations and informal textual structures characteristic of online consumer reviews.

The outlined methodological approach enables a detailed, in-depth analysis of consumer opinions, highlighting specific aspects that significantly influence customer satisfaction or dissatisfaction. The insights derived from this analysis provide actionable recommendations for businesses to enhance product quality, refine marketing strategies, and strengthen customer engagement.

Moreover, this research contributes to the academic field by demonstrating the effectiveness and comparative strengths of BERT, RoBERTa, and DistilBERT in performing nuanced aspect-level sentiment analysis. This comparative evaluation establishes benchmarks for future ABSA studies, guiding the selection and optimization of advanced NLP models in practical consumer feedback scenarios.

Figure 3.1 elaborates on the broader research flow introduced earlier in Figure 1.1. While Figure 1.1 offers a high-level conceptual overview from problem identification to results interpretation, Figure 3.1 breaks down the same process into detailed methodological stages including data handling, modeling, and evaluation. This consistent structure ensures that both diagrams serve complementary roles strategic orientation and operational detail.

Figure 3.1 visually summarizes the structured methodology adopted in this study. The diagram illustrates the logical flow across key phases: data collection, preprocessing, data analysis, model development, evaluation, and deployment. This clear visual representation reinforces the modular and sequential nature of the research process and enhances reader understanding of how each phase contributes to the overall study objectives.

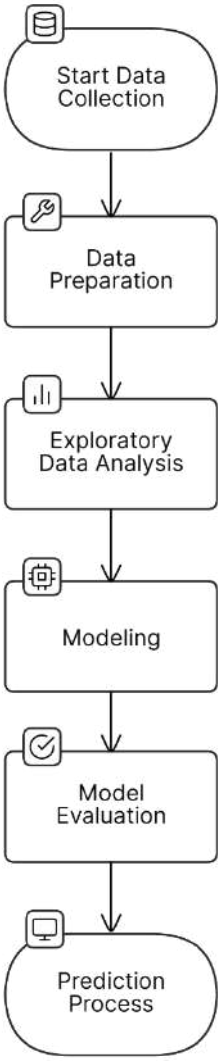


Figure 3.1 Research Methodology Workflow

3.1 Research Understanding

The Research Understanding phase is fundamental in establishing a clear comprehension of the research context and setting detailed, actionable objectives. This stage integrates creative analysis with structured investigation to effectively address the defined research challenges.

In this research, a modular approach inspired by Aspect-Based Sentiment Analysis (ABSA) is applied to evaluate consumer perceptions of Apple, Samsung, and Xiaomi products. The study separates two key tasks: Aspect Classification, which identifies significant product attributes mentioned in user-generated reviews (e.g., battery, camera), and Review-Level Sentiment Classification, which assigns an overall polarity (positive, neutral, negative) to each review.

While grounded in ABSA methodologies, this research does not perform sentiment classification at the aspect level, due to annotation constraints and the goal of maintaining interpretability and scalability. By leveraging advanced NLP models BERT, RoBERTa, and DistilBERT this design delivers precise and insightful outcomes that directly support the research goals and practical business decision-making.

3.2 Data Collection

Web scraping was selected as the primary data collection technique for this research due to its effectiveness in systematically capturing extensive and authentic user-generated reviews from popular e-commerce platforms. These platforms specifically Amazon, Best Buy, Flipkart, and Walmart are prominent sources of consumer feedback, particularly for electronic products such as smartphones and smartwatches from Apple, Samsung, and Xiaomi. This targeted scraping method was preferred for its precision and its capability to gather highly relevant and contextually rich data, essential for conducting an accurate Aspect-Based Sentiment Analysis (ABSA). Although the process was automated using Python libraries like BeautifulSoup, meticulous manual checks were integrated to ensure accuracy and contextual relevance. This combination of automated efficiency and manual curation significantly enhanced the reliability and specificity of the collected dataset, directly aligning with the defined research objectives.

This structured data collection methodology allowed the research to deeply investigate consumer sentiments and provided insights into the specific factors influencing product satisfaction and dissatisfaction. By carefully selecting relevant review data, this method ensured that the analysis directly supported strategic recommendations for businesses seeking detailed consumer insights into electronic products.

3.2.1 Data Types

The dataset compiled for this research primarily consists of textual consumer reviews sourced from Amazon, Best Buy, Flipkart, and Walmart. These reviews include comprehensive consumer-generated textual content, numerical ratings, and relevant metadata such as reviewer profiles and product specifics. Each review provides insights into the user's experience and opinions about distinct product features, including battery life, camera quality, design, and user interface. These detailed textual elements underpin the core analytical tasks of Aspect Extraction and Aspect Sentiment Classification, crucial for identifying nuanced sentiment trends and public perceptions regarding specific product attributes.

By concentrating on these textual reviews, the research effectively addresses its primary analytical goals, providing rich, contextually accurate sentiment data that allows for targeted and meaningful analysis of consumer feedback.

3.2.2 Estimated Data Volume

Considering the scope and precision of the scraping activities, approximately 20,000 product reviews have been successfully collected from the targeted platforms, covering the period between 2021 and 2024. This substantial dataset volume ensures robust analytical capability, offering statistically meaningful insights into customer sentiment patterns relevant to ABSA across various product types and brands. The diversity and volume of collected reviews enhance the validity and applicability of the findings, forming a comprehensive foundation for in-depth ABSA. Table 3.1 provides sample entries from the collected dataset, illustrating the structure of review text along with associated metadata such as date, brand, and product type.

Table 3.1 Presents an example of dataset

Text (Excerpt)	Date	Brand	Product Type
The camera is great, but the battery drains quickly.	2023-06-12	Apple	Phone
Love the sleek design. Worth the price.	2024-01-04	Samsung	Watch
Comfortable to wear but syncing takes too long.	2022-10-20	Xiaomi	Watch
iPhone 14 is smooth and fast. Face ID works even in the dark.	2023-11-25	Apple	Phone

3.2.3 Web Scraping Approach

To collect the required review data, a custom web scraping pipeline was developed using Python and the BeautifulSoup library. The scraper was designed to navigate product listing pages and extract relevant review sections, including review text, star ratings, review timestamps, and product metadata. Pagination controls and dynamic loading mechanisms were accounted for to ensure the scraper could collect reviews across multiple pages without duplication or data loss. To ensure compliance with platform policies and ethical standards, scraping frequency was regulated using timed requests and randomized user-agent headers. The scraper’s performance was iteratively validated by comparing outputs against visible site content and ensuring accurate parsing across all four target platforms.

3.2.4 Data Quality Assurance

To ensure the accuracy, consistency, and analytical relevance of the collected data, a structured multi-step quality assurance process was implemented, as illustrated in Figure 3.2. For reviews originally in languages other than English, automated translation tools (Google Translate API integrated via Python) were utilized, followed by manual verification to ensure semantic consistency and correctness.

Next, the system detected and removed duplicate entries based on content and user metadata to avoid redundancy. Once uniqueness was confirmed, the reviews were examined for metadata completeness ensuring each entry included the brand, product name, device type, and numerical rating. Reviews missing this essential metadata were excluded.

Following structural validation, the content was assessed for its semantic relevance to Aspect-Based Sentiment Analysis (ABSA). Only reviews that referenced specific product aspects such as battery life, camera performance, screen quality, or design were retained. Irrelevant or overly generic reviews were discarded.

This pipeline, combining automation and logical branching, helped filter out noise and non-informative records while preserving meaningful data. The resulting dataset was thus optimized for ABSA tasks, enhancing the precision, credibility, and reliability of downstream sentiment classification.

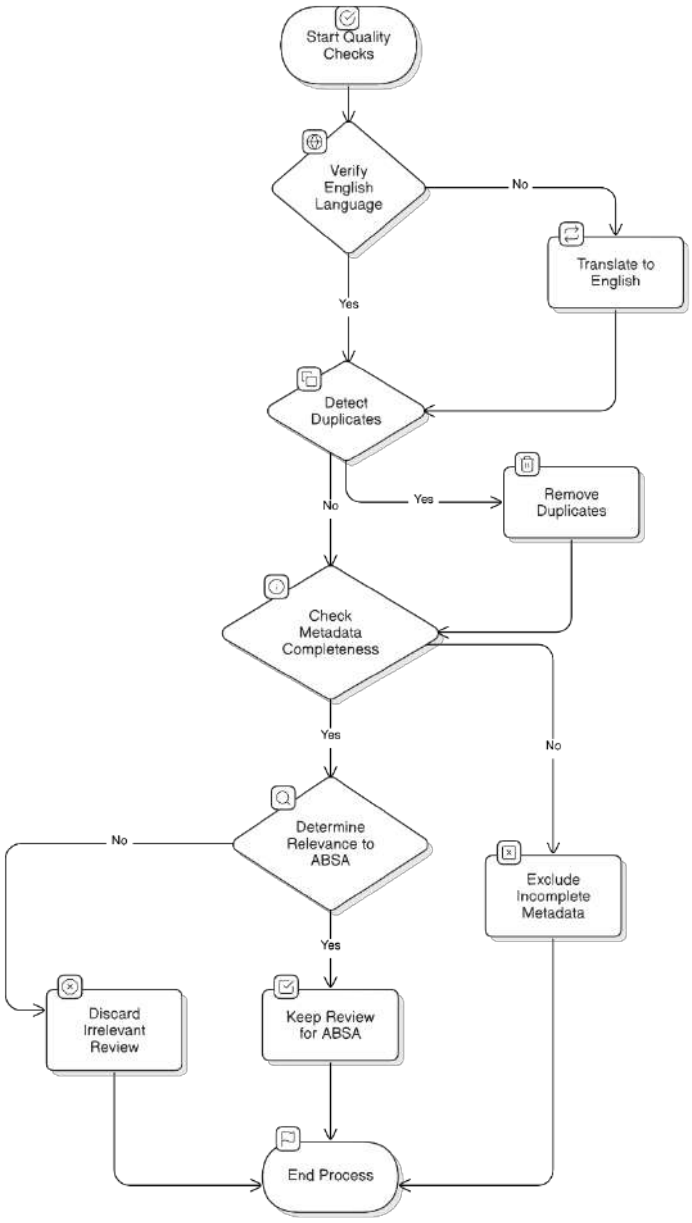


Figure 3.2 Data Quality Assurance Process

3.3 Data Preparation

The data preparation phase is essential to ensure the quality, structure, and consistency of textual data collected from electronic product reviews. Effective preparation significantly enhances model performance by reducing noise and facilitating meaningful insights during Aspect-Based Sentiment Analysis (ABSA). This phase comprises two critical stages: Text Preprocessing and Text Representation, illustrated comprehensively in Figure 3.3.

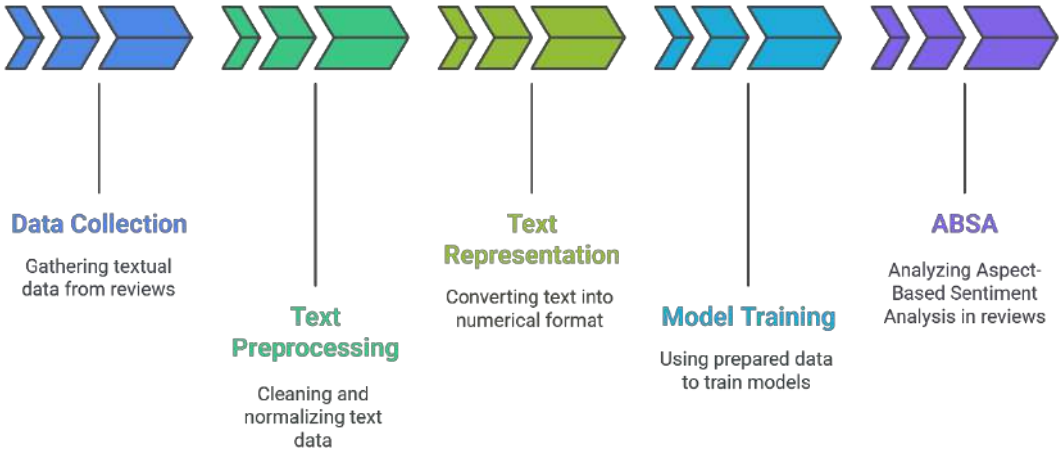


Figure 3.3 Overall Workflow of Data Preparation

3.3.1 Text Preprocessing

Text preprocessing is the initial step, aiming to refine the raw review data into an analyzable format by systematically addressing issues such as linguistic noise, inconsistent formatting, and incomplete entries (Jurafsky & Martin, 2019). This step includes normalization, tokenization, and handling incomplete data entries, These operations are outlined in detail in the subsections below and visually depicted in Figure 3.4, which presents the step-by-step workflow applied in this study to prepare the textual dataset for aspect-based sentiment analysis.

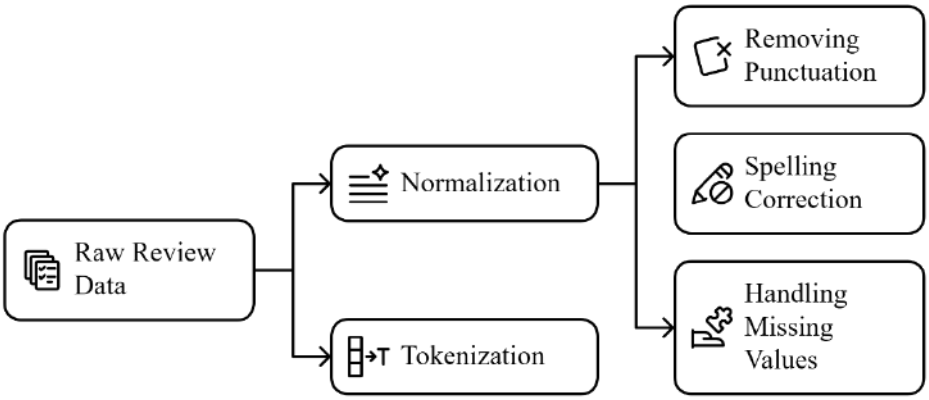


Figure 3.4 Workflow of Text Preprocessing

a. Normalization

Text normalization refers to the process of converting text into a uniform and standardized format, reducing variability while maintaining semantic integrity. In this research, stemming and stop-word removal were deliberately excluded, following recommendations by

Sun, Huang, & Qiu (2019), who caution that such steps may eliminate syntactically valuable cues needed for fine-grained sentiment classification. Instead, lightweight normalization steps were adopted to preserve sentiment-relevant tokens while reducing noise.

The normalization process included:

- Lowercasing all text to eliminate case-based variation.
- Removing punctuation and special characters using regular expressions.
- Spelling correction using automated libraries such as TextBlob.
- Whitespace normalization to remove repeated or trailing spaces.

These techniques were implemented using a Python-based preprocessing pipeline.

1. Removing Punctuation

Punctuation introduces structural markers but also adds computational noise including dashes (-), periods (.), commas (,), and other symbols, particularly in large-scale datasets. To streamline model input, punctuation such as periods, commas, exclamation marks, quotation marks, and dashes were removed using a rule-based system. This improves tokenization quality and model attention mechanisms.

As per Rustam et al. (2021), removing punctuation enhances consistency in model inputs, which is especially beneficial in Transformer architectures trained on clean corpora. However, care was taken to preserve sentiment-bearing emoticons only when explicitly relevant. Table 3.2 shows a side-by-side comparison of a review before and after normalization, highlighting how punctuation removal improves input consistency for model processing.

The punctuation removal process applied during normalization is visually summarized in Figure 3.5, demonstrating the structured approach used to clean review text.

Table 3.2 Comparative Examples Before and After Normalization

Original Review	After Normalization
“The Phone Is SleekAmazing Camera, But Battery Sucks!”	the phone is sleek amazing camera but battery sucks

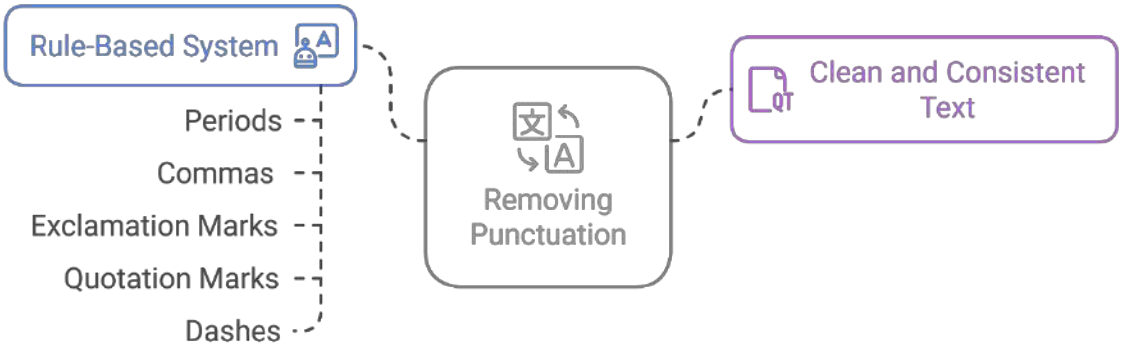


Figure 3.5 Workflow for Removing Punctuation

2. Removing Special Characters

The removal of special characters is a critical step in the text preprocessing pipeline for Aspect Classification and Sentiment Analysis tasks, particularly when handling informal, user-generated content from online platforms such as Amazon, Flipkart, Walmart, and Best Buy. These characters include non-alphanumeric elements such as emojis (e.g., 😊, 😢), hashtags (#), currency symbols ($), percentage signs (%), ampersands (&), typographic marks (e.g., “ ”, …), and HTML entities (e.g., &, '). Left untreated, these elements can introduce ambiguity, disrupt token alignment, and reduce model interpretability.

To preserve emotional cues embedded in emojis common in consumer reviews this study applied a text-to-code conversion method using the Python emoji library. Emojis were transformed into standardized textual representations (e.g., 😊 → :smile:), allowing models like BERT, RoBERTa, and DistilBERT to capture sentiment without introducing token fragmentation. For instance, the phrase “battery💔” becomes “battery :broken_heart:”, ensuring proper segmentation and preserving sentiment-bearing signals during tokenization (Devlin et al., 2019).

Other special characters not contributing to sentiment, such as hashtags, formatting symbols, control sequences (e.g., \n, \t, \u2019), and isolated non-English glyphs, were removed using a rule-based text cleaning approach involving regular expressions and Unicode filtering. This ensured uniformity and reduced parsing errors across transformer models.

This hybrid strategy emoji conversion combined with selective special character removal enhanced the semantic clarity and structural consistency of the dataset. Prior studies report that noisy inputs from inconsistent character usage can degrade classification accuracy by up to 7%, especially in polarity-sensitive tasks (Rustam et al., 2021; Fang et al., 2022). While some emotion detection studies advocate retaining raw emojis (Zhang et al., 2019), this

study found that text-based encoding offered a balance between expressiveness and preprocessing stability.

Evaluation of the preprocessing pipeline confirmed improved validation accuracy across all models, particularly for reviews containing sarcasm, abbreviations, or informal syntax. Table 3.*3* illustrates how this process transforms noisy user-generated content into a cleaner and model-compatible format, and Figure 3.*6*outlines the rule-based workflow adopted for special character handling.

Table 3.3 Example on Removing Special Characters

Original Review	After Special Character Removal
Love This Phone 😍! Best Camera 📸 Ever, But Price is Too High $.	love this phone :heart_eyes: best camera :camera: ever but price too high

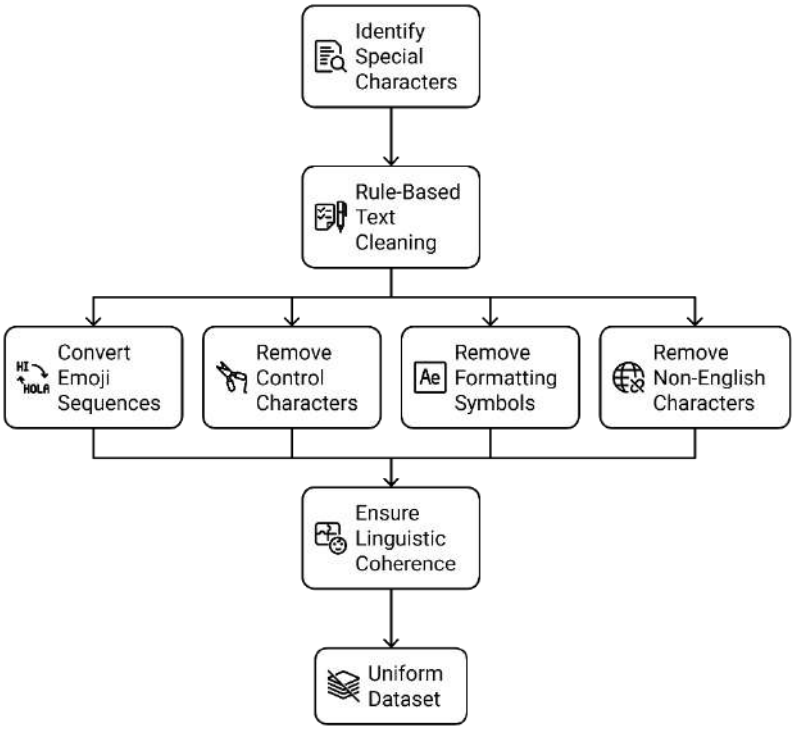


Figure 3.6 Workflow Special Characters

b. Tokenization

Tokenization is the process of segmenting normalized text into smaller linguistic units called tokens which serve as the fundamental building blocks for natural language processing models. Effective tokenization ensures that the textual data is accurately interpreted by deep learning models, particularly those built on Transformer architectures. In this research, tokenization is aligned with each model's native tokenizer: WordPiece for BERT and DistilBERT, and Byte-Pair Encoding (BPE) for RoBERTa.

The tokenizer choice impacts how the model handles vocabulary, rare words, subword units, and structural markers like sentence boundaries. Table 3.4 summarizes the tokenizer types and special tokens used by each model, providing a clear overview of their preprocessing configurations.

Table 3.4 Tokenizer characteristics for all models

Model	Tokenizer	Tokenizer Type	Special Tokens Used
Bert	bert-base-uncased	WordPiece	[CLS], [SEP], [PAD]
DistilBERT	distilbert-base-uncased	WordPiece	[CLS], [SEP], [PAD]
Roberta	roberta-base	Byte-Pair Encoding (BPE)	<s>, </s>, <pad>

WordPiece tokenization, used by BERT and DistilBERT, segments words into frequent subword units. This approach minimizes the out-of-vocabulary problem and captures rich syntactic information. For example, the word "uninstallable" might be tokenized as: ["un", "##install", "##able"].

Additionally, the following special tokens are added:

- [CLS] (Classification token): Prepended for sentence-level tasks.
- [SEP] (Separator token): Used to separate sentences or input segments.
- [PAD]: Used to pad sequences to a uniform length for batching.

This structure supports downstream tasks such as aspect classification and sequence labeling, enhancing contextual alignment.

Byte-Pair Encoding Tokenization (RoBERTa)

Unlike BERT, RoBERTa uses Byte-Pair Encoding (BPE). BPE merges frequently co-occurring characters or character sequences to form a variable-length subword vocabulary. RoBERTa also forgoes the use of the [CLS] and [SEP] tokens used in BERT, instead utilizing:

- <s> (start of sequence),
- </s> (end of sequence),
- <pad> (padding).

It also encodes whitespace as part of the token itself using a visible marker such as Ġ (e.g., Ġbattery, Ġlife) to distinguish between words and continuation tokens (Kudo, 2018).

Example: Comparing Tokenization in BERT and RoBERTa

To demonstrate the difference, consider the review sentence:

"Battery life is fantastic on the Galaxy S24 Ultra!"

Table 3.5 provides a comparative example of how BERT and RoBERTa tokenize the same review sentence, highlighting key differences in subword segmentation and special token usage.

Table 3.5 Comparing Tokenization in BERT and RoBERTa

Model	Tokenized Output
BERT	[CLS], battery, life, is, fantastic, on, the, galaxy, s, ##24, ultra, [SEP]
RoBERTa	<s>, Battery, Ġlife, Ġis, Ġfantastic, Ġon, Ġthe, ĠGalaxy, ĠS, 24, ĠUltra, !</s>

As shown:

- BERT splits S24 into s, ##24 using its WordPiece subword logic.
- RoBERTa treats white space as meaningful and separates tokens using the Ġ space marker.
- RoBERTa also keeps punctuation like ! as a separate token.

This distinction is not just cosmetic models trained with these tokenizers are sensitive to the format, and incorrect alignment between tokenizer and model can degrade performance or lead to misinterpretation. Figure 3.7 illustrates the tokenization pipeline, showing how input text is transformed into IDs, masks, and embeddings before feeding into transformer models for downstream tasks.

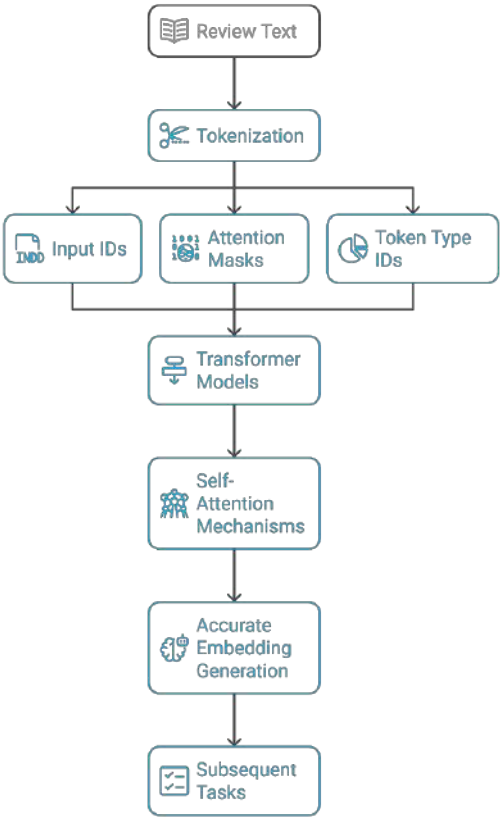


Figure 3.7 Tokenization Process Example

3.3.2 Text Representation

In Natural Language Processing (NLP), embedding refers to the process of mapping discrete linguistic units such as tokens into continuous vector spaces. This transformation is essential, as machine learning models cannot interpret text directly and instead require numerical representations that capture both the syntactic and semantic properties of words (Mikolov et al., 2013). Embeddings provide a means of representing the meaning and structure of language in a form that models can analyze, compare, and learn from.

A key distinction in this context is between word embeddings and sentence embeddings. Word embeddings represent the meaning of individual words in local contexts, whereas sentence embeddings capture the holistic semantics of a complete sentence or phrase. While sentence-level representations are more suitable for tasks like text summarization or paraphrase detection (Reimers & Gurevych, 2019), word-level embeddings are fundamental in Aspect-Based Sentiment Analysis (ABSA), where the sentiment tied to each individual aspect is context-dependent.

Transformer-based models like BERT, RoBERTa, and DistilBERT generate contextual word embeddings each word's representation varies depending on the surrounding context. For

instance, the word “bright” in “bright screen” versus “bright future” is embedded differently in each sentence, which is a critical advancement over static embeddings (Devlin et al., 2019; Liu et al., 2019).

These models generate embeddings by combining three distinct components:

- Token Embeddings
- Position Embeddings
- Token Type (Segment) Embeddings

As shown in Figure 3.8: Embedding Process in Transformer Models, each input token undergoes this composite embedding transformation before being processed by the attention mechanism in the encoder (Vaswani et al., 2017).

Token Embeddings

Token embeddings form the core representation of the textual input. Each token, produced during the tokenization process, is mapped to a pre-trained vector from the model’s vocabulary. For example, in BERT, the token [CLS] corresponds to a fixed vector, while real-word tokens like “battery”, “poor”, or “ultra” also have corresponding dense vectors. These embeddings preserve both lexical and syntactic meaning and are essential for capturing aspect-specific semantics in customer reviews (Devlin et al., 2019).

In the example illustrated in Figure 3.8, the token [CLS] is assigned an ID of 2, while other tokens such as “battery” and “life” are mapped to their respective IDs and vectors using a pre-trained vocabulary.

Position Embeddings

Unlike recurrent neural networks (RNNs), Transformers do not process sequences sequentially and thus lack an inherent sense of word order. Position embeddings are added to token embeddings to restore this lost temporal structure (Vaswani et al., 2017). Each token is assigned a unique position ID that is mapped to a position vector. For example:

- The token [CLS] receives Position 0,
- The first word receives Position 1,
- The second word receives Position 2, and so on.

This mechanism ensures the model understands not just what the tokens are, but also where they are in the sentence.

Token Type (Segment) Embeddings

Token type embeddingsalso called segment embeddingsare used to distinguish between different parts of an input sequence. This is especially important for tasks that involve sentence pairs, such as question answering or aspect-opinion matching (Sun et al., 2019). Each token is

assigned a segment ID, typically 0 or 1, depending on whether it belongs to the first or second segment.

In this research, since most ABSA inputs involve a single review or sentence, all tokens were assigned token type 0, as seen in Figure 3.8. However, the capability to distinguish segments ensures flexibility for future model extensions involving multi-sentence reasoning.

Embedding Normalization and Output

Once these three embeddingstoken, position, and segmentare computed, they are summed element-wise and passed through a normalization layer. This step scales the vectors appropriately for entry into the self-attention layers of the Transformer. The final output is a high-dimensional vector for each token (typically 768 dimensions in base models), representing the full context-aware embedding.

As shown in Figure 3.8, the final embeddings are numerical vectors such as:
[-3.2649e, 6.1285e, 3.3932e, ..., -1.1258e]

These contextual vectors are passed through the layers of the Transformer to compute downstream sentiment classifications, making them a cornerstone of the ABSA pipeline.

Through this three-fold embedding strategy, the models generate a detailed, structurally coherent, and contextually informed representation of textual input. This layered representation enables deep learning models to effectively capture linguistic nuance, sentiment shifts, and the subtle semantic variations that are critical in analyzing customer opinions on products like smartphones and smartwatches.

Figure 3.8 illustrates how token, position, and segment embeddings are combined for each token in a review. These are then normalized into high-dimensional vectors, enabling the transformer model to generate rich contextual representations used in ABSA tasks.

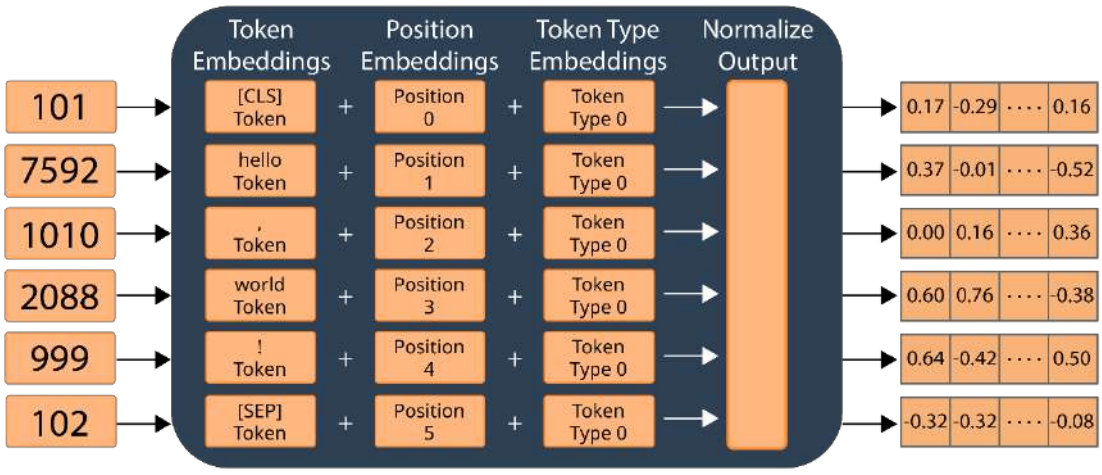


Figure 3.8 Embedding Process in Transformer Models

Note: While BERT and DistilBERT utilize all three embedding types token, position, and segment RoBERTa omits segment embeddings during pretraining, relying solely on token and position embeddings (Liu et al., 2019).

3.4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical phase in NLP-based research that enables the researcher to examine patterns, distributions, and biases in the dataset before proceeding to modeling. For this study, EDA was performed on approximately 20,000 reviews collected from Amazon, Walmart, Best Buy, and Flipkart, covering products from Apple, Samsung, and Xiaomi between 2021 and 2024.

This phase aims to assess the representativeness of the dataset across brands and device types and validate its richness for Aspect-Based Sentiment Analysis (ABSA). The analysis was conducted using Python libraries such as Pandas, Matplotlib, and WordCloud, following best practices for textual data exploration (Aggarwal & Zhai, 2012).

3.4.1 Review Distribution by Brand

To evaluate brand coverage, reviews were grouped by brand name. As visualized in Figure 3.9, Samsung accounts for the highest number of reviews (7,685), followed by Xiaomi (7,332), and Apple (5,526). This balanced distribution ensures a robust foundation for comparative sentiment modeling across all three major consumer electronics brands.

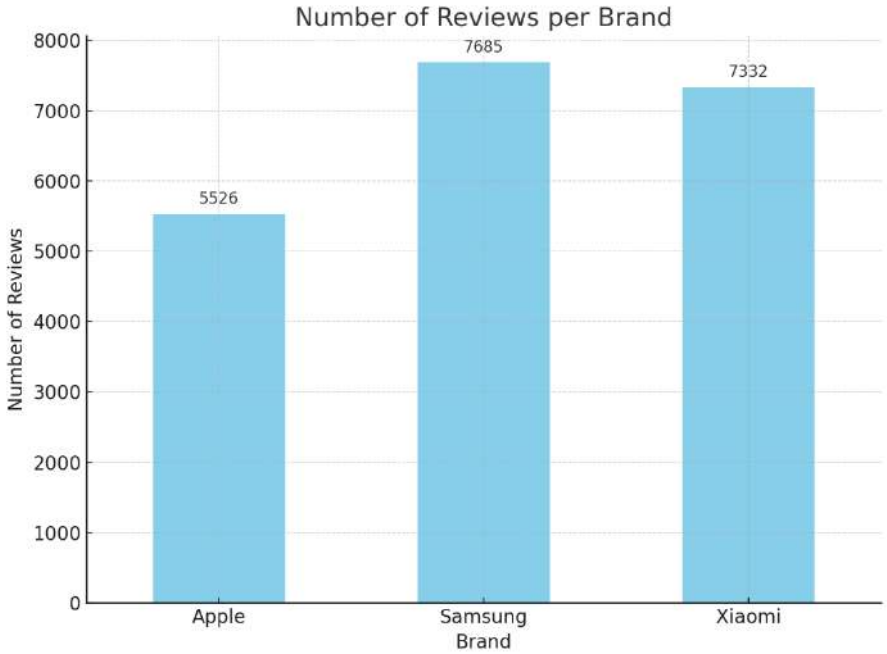


Figure 3.9 Number of Reviews per Brand

3.4.2 Review Distribution by Device Type

The dataset was further analysed based on device type Smartphones vs. Smartwatches. As shown in Figure 3.10, smartphones constitute 52.8% of the total reviews, while smartwatches represent 47.2%. This relatively even distribution supports the study's objective of evaluating consumer sentiment across both device categories.

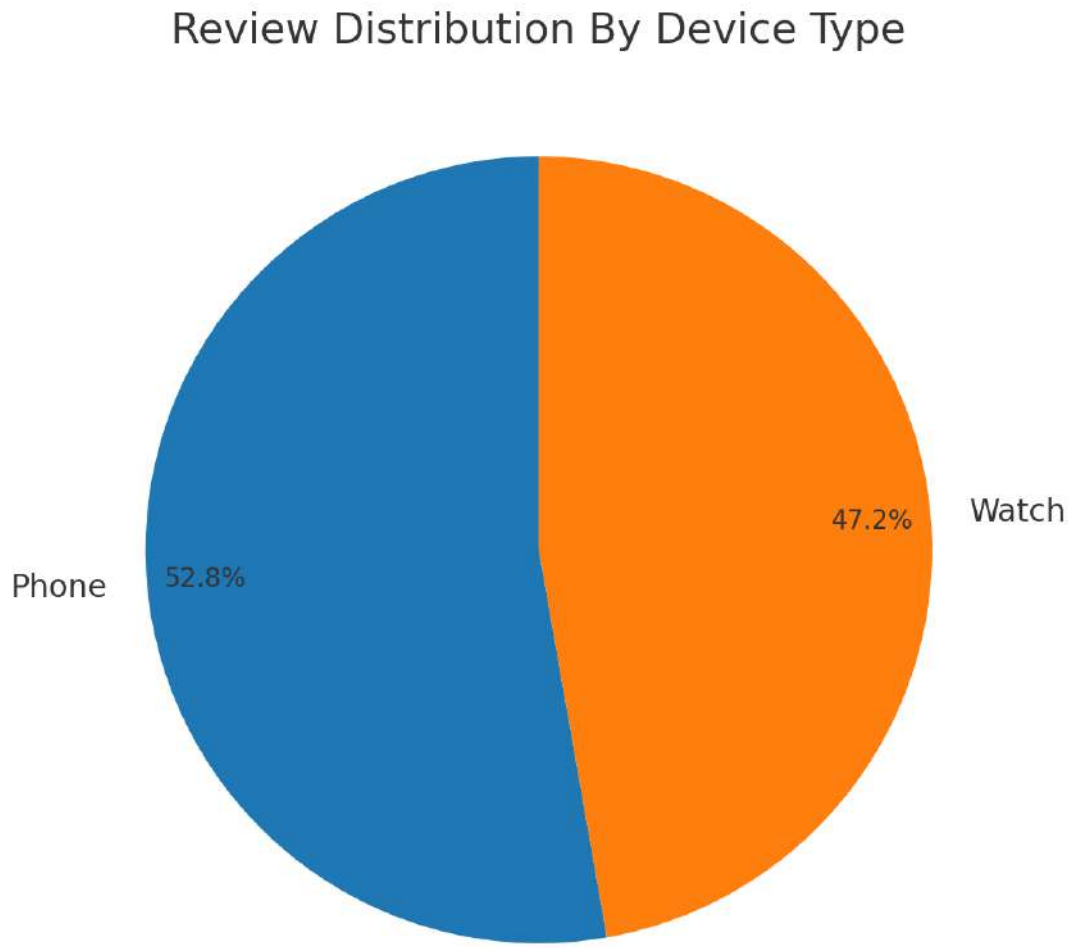


Figure 3.10 Review Distribution by Device Type

3.4.3 Combined Brand and Device Distribution

To understand brand-specific focus on different product categories, a stacked bar chart was constructed. Figure 3.11 illustrates that:

- Samsung leads in smartphone reviews (5,100), and maintains a strong presence in wearables (2,585),
- Xiaomi shows a higher concentration of smartwatch reviews (4,356),
- Apple displays a balanced distribution between phones (2,765) and watches (2,761).

These insights guided later decisions about dataset balancing during training and model evaluation.

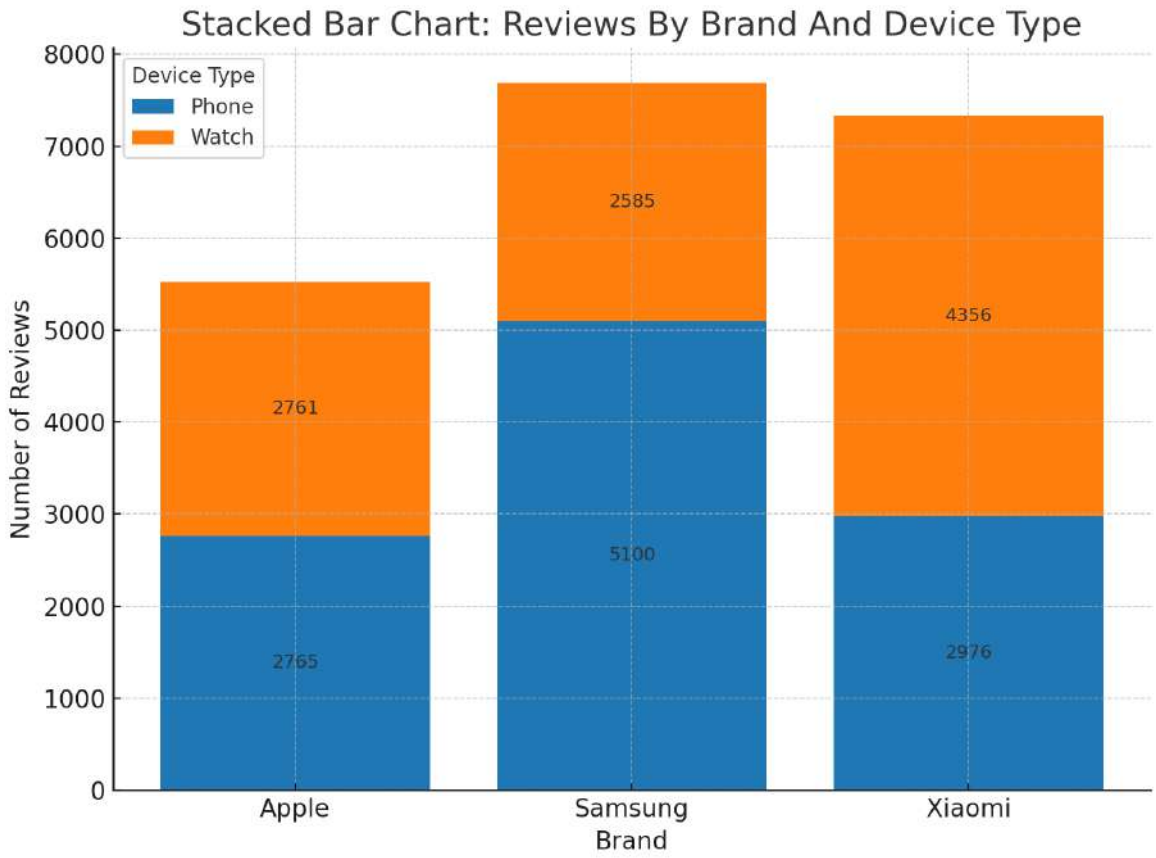


Figure 3.11 Stacked Bar Chart of Reviews by Brand and Device Type

3.4.4 Word Frequency and Aspect Validation

To validate the presence of aspect-level details in the reviews, a WordCloud visualization was generated. As seen in Figure 3.12, users frequently mention product features such as battery, screen, camera, quality, charger, and performance. This confirms that the dataset is rich in aspect-specific feedback, reinforcing its suitability for ABSA tasks (Feldman, 2013).

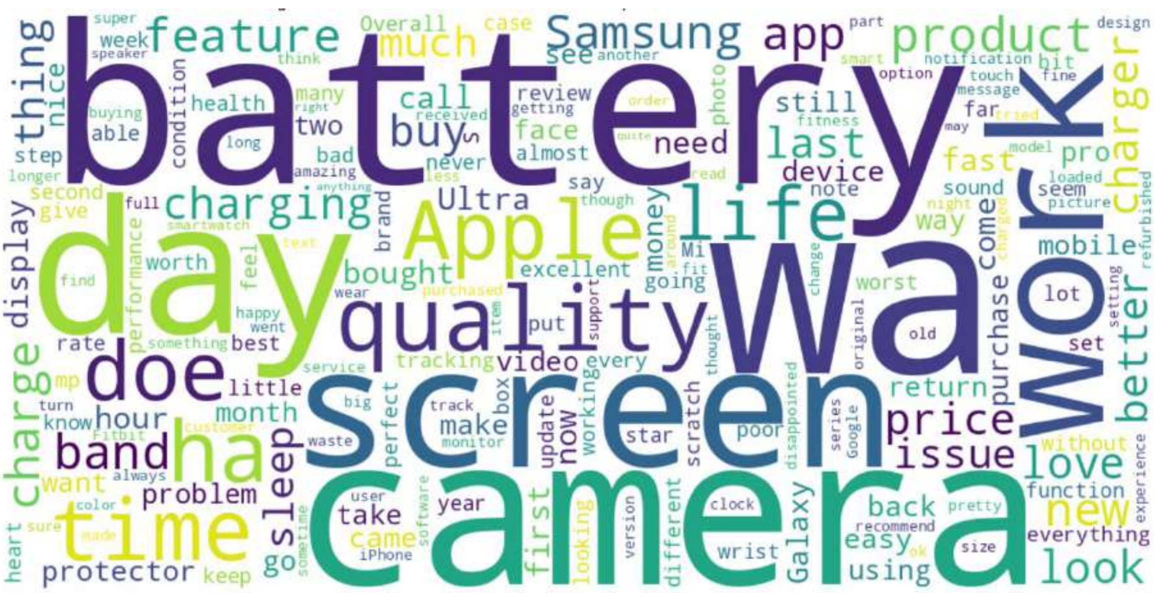


Figure 3.12 Word Cloud of Most Frequent Terms in Review Texts

3.5 Human Annotation Process

To prepare a high-quality labeled dataset suitable for training and evaluating the models, a meticulous human annotation process was conducted after preprocessing and exploratory analysis. The primary objective was to create a structured, thematically rich corpus that accurately captured user opinions across multiple product features and their associated overall sentiments.

A schematic overview of this annotation pipeline is presented in Figure 3.13, which outlines each step in the workflow, from aspect identification to sentiment labeling and quality control.

3.5.1 Aspect Definition and Thematic Analysis

The aspect categories used in annotation were derived through a multi-stage refinement process. Initially, 238 raw aspect terms were extracted from the dataset using keyword frequency analysis and thematic clustering informed by literature review (Hu & Liu, 2022) and exploratory data analysis (EDA). These terms were then organized into 38 mid-level aspect terms, which were finally grouped under eight main categories that serve as the core aspects in this study.

Each of these final aspects is associated with a list of representative keywords. Annotators used these keywords to guide consistent identification of aspects within reviews. If a keyword associated with a specific aspect appeared in the review text, the corresponding aspect was marked as present. Table 3.6 provides definitions for each main aspect category,

while Table 3.7 (to follow) lists the specific keywords mapped to each aspect to support reproducibility and annotation transparency.

Table 3.6 Aspect Definitions Based on Annotated Review Categories

Aspect	Definition
BATTERY	Mentions of longevity, charging performance, or power efficiency.
PERFORMANCE	System speed, lag, app responsiveness, or general performance.
CAMERA	Photo or video quality, lens clarity, and camera features.
DISPLAY	Screen brightness, resolution, refresh rate, and visual clarity.
PRICE	Perceptions of cost, affordability, or value for money.
MULTIMEDIA	Audio output, speakers, and media playback experience.
FITNESS & HEALTH TRACKING	(For wearables) Heart rate, step count, sleep tracking, workout monitoring.
CUSTOMIZATION	Watch face personalization, themes, strap changes, and layout flexibility.

Table 3.7 Aspect Categories and Associated Keywords

Aspect	Associated Keywords
BATTERY	charge, battery life, charger, power, fast charging, battery drain, charging cable, full charge
PERFORMANCE	ROM, fast, connect, RAM, lag, slow, Bluetooth, network, speed, 5G, simple, fingerprint, GB, easy to use, LTE, storage, processor, cellular, chip, memory, interface, internet, space
CAMERA	photo, camera quality, picture, zoom, image
DISPLAY	screen, OLED, resolution, brightness, AMOLED
PRICE	worth, cost, cheap, expensive, Overpriced, bargain, pricing, reasonable price, best value, high price,
MULTIMEDIA	media, sound, notifications, volume, audio
FITNESS & HEALTH TRACKING	sleep tracking, heart rate monitor, steps tracking, calories burned, sleep tracking, workout tracking, fitness app, exercise tracking, health monitoring
CUSTOMIZATION	watch face, customization

While the WordCloud in Figure 3.12 presents the most frequently occurring terms in the dataset, the aspect categories used in annotation were derived through a structured thematic analysis, focusing on terms that hold contextual sentiment value. These aspect keywords were curated based on their relevance, semantic clarity, and frequency across reviews. As such, they may differ slightly from the general high-frequency terms found in the WordCloud, which includes both sentiment-bearing and non-informative words. This refinement ensures more reliable aspect tagging and training performance.

3.5.2 Annotation Procedure

Annotations were conducted by three individuals. Two annotators are specialized in English linguistics and were responsible for identifying aspects and assigning sentiment labels. The third person, who has two years of experience in Natural Language Processing (NLP), served as the evaluation checker. His role was to verify the annotation accuracy and resolve discrepancies by comparing results across annotators. This multi-role annotation team ensured both linguistic depth and technical consistency in the final labeled dataset.

3.5.3 Inter-Annotator Agreement

To assess consistency between annotators, Cohen's Kappa (κ) was calculated. A score of 0.91 indicated excellent agreement (Landis & Koch, 1977). Remaining discrepancies (less than 9%) were resolved through consensus discussion, resulting in a high-confidence annotated dataset ready for supervised learning. Table 3.8 presents sample annotated reviews, showing how aspect terms and overall sentiment were labeled during the manual annotation process.

Table 3.8 Sample of Manually Annotated Reviews with Aspects and Overall Sentiment

Review Text	Aspects Identified	Overall Sentiment
Battery lasts for days and charges quickly.	Battery	Positive
The camera is underwhelming, but the build quality is excellent.	Camera, Design	Neutral
Too slow and keeps freezing. Not worth the price.	Performance, Price	Negative
Accurate heart rate and fitness features very useful.	Fitness & Health Tracking	Positive
Looks great, smooth interface, but the sound quality is poor.	Performance, Multimedia	Neutral
Expensive for the features offered. Could be better.	Price	Negative
Very customizable UI and watch faces. Love it!	Customization	Positive

3.5.5 Annotation Impact

The manually annotated dataset played a pivotal role in enabling accurate and targeted aspect identification. It provided the structural foundation to fine-tune transformer-based models specifically for the task of aspect extraction, allowing them to recognize which product features such as Battery, Performance, and Camerawere discussed within each review.

For sentiment classification, however, the study adopted a different approach. Rather than fine-tuning sentiment models on the same dataset, the methodology relied on pretrained sentiment analysis models available through well-established NLP repositories such as HuggingFace. These models trained on large-scale corpora of reviews and opinions have consistently demonstrated strong generalization across diverse domains.

This decision was supported by several justifications:

- Review-level sentiment classification is a well-established task with robust pretrained solutions already available.
- Using existing models enhances reproducibility, aligning with prevailing practices in sentiment mining research.
- It significantly reduces computational overhead without compromising prediction accuracy.

During inference, the outputs from both modules were combined into a unified prediction:

- The fine-tuned aspect extraction model identified the relevant product features mentioned in the review.
- The pretrained sentiment model assigned the overall sentiment polarity to the review as a whole.

This modular pipeline delivered both flexibility and performance. It allowed for task-specific fine-tuning where necessary (aspect extraction), while leveraging mature and optimized solutions for general sentiment classification. The result is a scalable and efficient ABSA system capable of processing diverse customer feedback without redundant training or loss of analytical depth.

Figure 3.13 illustrates the human annotation workflow, outlining the process used to label aspect terms and overall sentiment in the sampled reviews.

Figure 3.14 presents the distribution of annotated aspects, highlighting which features (e.g., Battery, Performance) were most frequently identified across the dataset.



Figure 3.13 Human Annotation Workflow

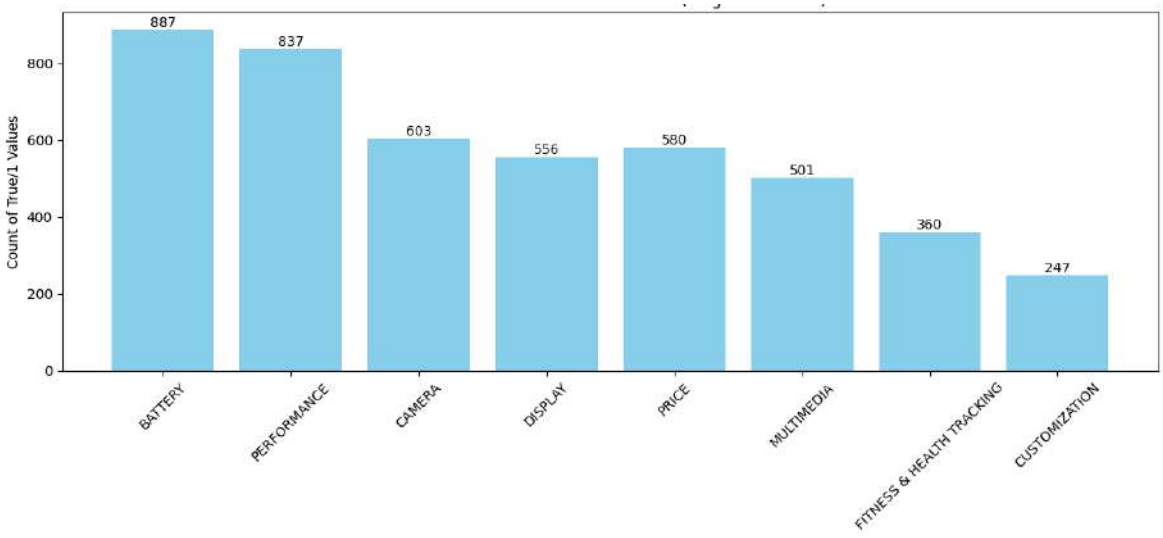


Figure 3.14 Aspect Frequency Distribution in Annotated Dataset

3.6 Modeling

Modeling in this research represents a systematic approach to understanding and predicting patterns in user-generated reviews across smartphones and smartwatches. This section describes the model development strategy adopted for performing Aspect Term Extraction (ATE) and Sentiment Analysis (SA). Rather than training a single monolithic model, a modular hybrid framework was employed: transformer models were fine-tuned for ATE, while pretrained models were used for SA, and both outputs were combined at inference to achieve robust, interpretable ABSA results. This architecture reflects recent advances in Natural Language Processing (NLP), where fine-tuning pretrained transformers has become the dominant paradigm for task-specific adaptation (Mosbach, Andriushchenko, & Klakow, 2020). The goal was to utilize this methodology efficiently while ensuring generalizability, reproducibility, and domain-specific precision

3.6.1 Model Fine-Tuning

Transformer-based models such as BERT, RoBERTa, and DistilBERT have transformed NLP since 2018 by enabling general-purpose language models to be fine-tuned for specific tasks with minimal architectural changes (Mosbach, 2023). In this research, these models were used to fine-tune only the Aspect Term Extraction (ATE) component, using the manually annotated dataset of 2,250 reviews.

Following the two main paradigms of pretrained model adaptation outlined by Howard & Ruder (2018) feature extraction and fine-tuning this study opted for full fine-tuning, where model weights were updated using task-specific data. This method yields superior results, especially when subtle contextual understanding is needed (Azhar & Khodra, 2021).

Fine-tuning was performed using:

- A classification head (dense layer) added to the final transformer output
- Sigmoid activation for multi-label classification
- Cross-entropy loss and Adam optimizer
- Early stopping and dropout to prevent overfitting

The fine-tuning dataset consisted of the manually annotated aspect labels (see Section 3.5), and this process allowed the model to internalize how users express concern, praise, or criticism about specific features like battery life, design, and display. Insights from the manual annotation process (see 3.5 Human Annotation Process) were instrumental in shaping the ATE fine-tuning strategy. Annotated aspect labels derived from a keyword-guided process ensured

that the transformer models learned to associate contextual patterns with frequently mentioned product features.

3.6.2 Experiments Overview

To determine the most effective modeling strategy, multiple experimental configurations were tested for ABSA. These experiments revealed the importance of separating sentiment analysis and aspect detection into distinct components.

a. Joint ATE + Sentiment Modeling

The first approach attempted to use a single model to predict both aspects and overall sentiment labels. Models such as BERT, DistilBERT, and RoBRETa were trained jointly on the two tasks. While aspect extraction performed adequately, the sentiment predictions suffered from:

- Overfitting during training
- Poor generalization on test data
- Incorrect sentiment classification during inference

These results indicated a potential mismatch between the learning dynamics of aspect and sentiment tasks.

b. Isolated Sentiment Modeling

A second experiment focused on training a dedicated model for sentiment classification. However, even with isolated training, the model showed:

- Persistent overfitting
- Bias toward dominant sentiment classes
- Low prediction accuracy and inconsistent performance

This prompted a shift to more advanced fine-tuning techniques.

c. Hyperparameter Tuning with LoRA

To address overfitting and reduce parameter complexity, a parameter-efficient fine-tuning strategy was implemented using Low-Rank Adaptation (LoRA). LoRA introduces small trainable matrices into the attention mechanism's query and value projections, allowing the base model to remain frozen while still adapting to task-specific nuances. This approach was inspired by the work of Hidayatullah (2024), who successfully applied LoRA for emotion classification in Indonesian texts, demonstrating its effectiveness in improving generalization while drastically reducing training overhead.

The pipeline involved:

- Base Models: BERT, DistilBERT, and XLM-RoBERTa
- Sentiment Classes: Positive, Neutral, Negative

- Tokenization: Max sequence length of 256
- Evaluation Metrics: Accuracy, Precision, Recall, F1
- Search Strategies:
    - Grid Search
    - Random Search
    - Optuna-based Bayesian Optimization with pruning

Despite these optimizations, sentiment performance remained unsatisfactory. Overfitting continued, and model generalization was limited even with LoRA adapters and extensive tuning.

d. Final Approach: Modular ABSA Architecture

Given the consistent challenges with sentiment modeling, a modular inference strategy was adopted:

- The aspect extraction task was handled by fine-tuned transformer models.
- The sentiment classification task was assigned to pretrained, publicly available sentiment analysis models.

This design:

- Prevented task interference
- Reduced training complexity
- Leveraged best-in-class tools for each component

This modular pipeline aligned with common practices in recent ABSA research, where pretrained sentiment classifiers trained on large review datasets offer strong performance without additional fine-tuning.

Table 3.9 summarizes the experimental evolution leading to the final modular pipeline, outlining each phase, its objective, and the corresponding outcome.

Table 3.9 Summary of Experimental Evolution

Phase	Goal	Result
Joint ATE + SA	Unified multitask model	SA overfitting; ATE acceptable
Isolated SA Model	Sentiment-only training	Persistent overfitting
LoRA + Hyperparameter Tuning	Efficient sentiment training	Reduced complexity, still poor accuracy
Modular ATE + Pretrained SA	Final hybrid architecture	Balanced performance, generalizable

3.6.3 Hybrid Prediction Pipeline

During inference, the outputs of both model types were combined into a cohesive ABSA result:

1. Aspect Prediction: The fine-tuned ATE model detects relevant aspects (e.g., Battery, Display).
2. Sentiment Classification: The pretrained SA model classifies the overall tone of the review.
3. Result Merging: Both outputs are merged into a structured interpretation.

Example:

“Battery life is excellent, but the screen is dull.”

→ Aspects: Battery, Display

→ Sentiment: Neutral

This hybrid strategy balances task-specific learning with reusable general capabilities, making the ABSA model both efficient and scalable. Figure 3.15 illustrates the fine-tuning process used for the multi-label ATE model, showing how aspect terms were learned from annotated data.

Figure 3.16 presents the final modular ABSA architecture, combining fine-tuned ATE with a pretrained sentiment classifier to generate structured predictions. It is important to note that the sentiment classification component in this pipeline predicted overall sentiment per review, rather than sentiment toward each individual aspect. This decision, driven by practical considerations and dataset limitations, reflects a hybrid ABSA implementation where aspect detection is fine-tuned, while sentiment is assigned globally using pretrained review-domain classifiers.

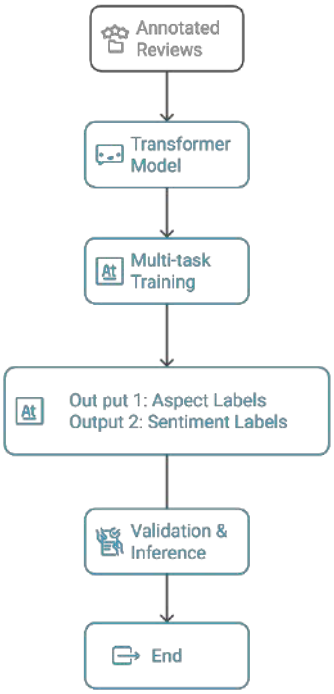


Figure 3.15 Fine-Tuning Process – Experiment (Multi-label ATE Model)

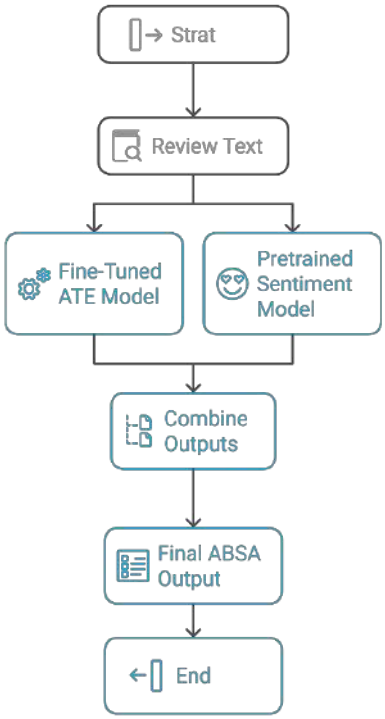


Figure 3.16 Final Modular ABSA Architecture (Fine-Tuned ATE + Pretrained SA)

3.7 Model Evaluation

Model evaluation plays a critical role in assessing the performance, stability, and generalizability of the proposed ABSA system. This section outlines the metrics, data splits, and techniques used to evaluate both the Aspect Term Extraction (ATE) and Sentiment Analysis (SA) components, each of which required a tailored evaluation strategy due to their differing classification structures.

In alignment with standard machine learning practices, the labeled dataset was divided into training (70%), validation (15%), and test (15%) subsets. Random sampling was employed to ensure representative distributions across all sentiment classes and aspect categories. All fine-tuning operations were carried out exclusively on the manually annotated subset of 2,250 reviews, ensuring data integrity and evaluation reliability.

3.7.1 Evaluation Metrics

The evaluation of both ATE and SA relied on four core performance metrics: Accuracy, Precision, Recall, and F1 Score, supported by class-level breakdowns and aggregated averages. The significance of each metric is described below:

a. Accuracy (ACC)

Accuracy measures the ratio of correctly predicted labels over the total number of instances. While informative, this metric can be misleading in class-imbalanced datasets.

$$\text{ACC} = \frac{\text{\# correctly classified samples}}{\text{\#all samples}} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$$ (Hicks et al., 2022)

b. Precision (PREC)

Precision evaluates the proportion of correctly predicted positive observations out of all predicted positives. It is particularly useful in minimizing false positives.

$$\text{PREC} = \frac{\text{\#true positive sample}}{\text{\#samples classified positive}} = \frac{\text{TP}}{\text{TP+ FP}}$$ (Hicks et al., 2022)

c. Recall (REC)

Also known as sensitivity, recall measures the proportion of true positives identified out of all actual positives, making it critical in identifying class coverage.

$$\text{REC} = \frac{\text{\#true positive sample}}{\text{\#actual positive samples}} = \frac{\text{TP}}{\text{TP+ FN}}$$ (Hicks et al., 2022)

d. F1 Score

The F1 score is the harmonic mean of precision and recall. It offers a balanced measure, especially valuable when dealing with class imbalance.

$$\text{F1} = 2 * \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} = \frac{2\text{TP}}{2\times\text{TP+FP+FN}}$$ (Hicks et al., 2022)

3.7.2 Evaluation Strategy by Task

Aspect Term Extraction (ATE)

ATE was formulated as a multi-label classification task, where a single review could contain multiple aspect mentions (e.g., “Battery” and “Camera”). Evaluation focused on:

- Micro-averaged metrics: Emphasize performance on common classes
- Macro-averaged metrics: Reveal performance on underrepresented aspects
- Hamming Loss: Measures the fraction of labels incorrectly predicted
- Subset Accuracy: Evaluates exact match between predicted and ground-truth label sets

The imbalance across aspects such as "Fitness & Health" and "Multimedia" necessitated the use of macro-F1 to highlight low-resource label performance. Macro-F1 was particularly emphasized to fairly assess model performance on underrepresented aspects such as "Customization" and "Fitness & Health Tracking," where fewer training examples could otherwise skew standard metrics.

Sentiment Analysis (SA)

Sentiment classification was designed as a single-label, 3-class classification task (Positive, Neutral, Negative). Evaluation included:

- Confusion Matrix: To detect misclassification patterns
- Class-wise Precision, Recall, and F1-scores
- Overall Accuracy: As a summary of correct predictions
- Macro-F1 Score: To balance performance across uneven sentiment classes

The sentiment analysis was conducted using a pretrained classifier sourced from the Hugging Face model hub, fine-tuned on large-scale customer review corpora. This approach avoided the need for additional training and allowed the model to generalize well across typical review language, particularly for positive and negative classes.

3.7.3 Overfitting Prevention and Validation Protocol

To avoid overfitting where models perform well on training data but poorly on unseen data early stopping was implemented. Training was halted when validation loss failed to improve over three consecutive epochs, as recommended by Alyami et al. (2022). Additionally, the best-performing model weights were restored before testing to ensure maximum generalizability.

After evaluation on the 2,250 manually annotated reviews, the best-performing pipeline was deployed on the remaining unlabeled dataset of over 18,000 reviews. This broader

application enabled high-confidence extraction of aspect-frequency patterns and sentiment trends across three brands and multiple product categories.

Following the training phase, predictions were generated over the remaining unannotated corpus using the fine-tuned aspect model and the pretrained sentiment classifier. These results were then used for downstream analysis and insight extraction.

The complete evaluation and prediction flow is summarized in Figure 3.17, offering a structured visualization of the end-to-end model application.

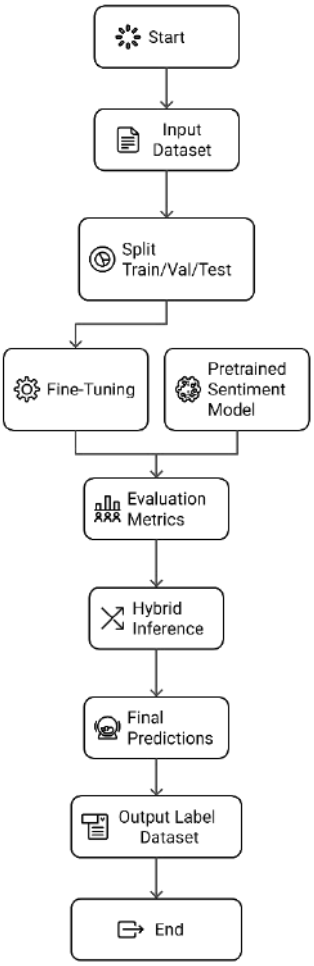


Figure 3.17 Prediction Framework: A structured overview of the modeling pipeline, showing how fine-tuned models (for ATE) and pretrained sentiment classifiers (for SA) work in tandem, from training through evaluation and final inference on unlabeled reviews.

CHAPTER IV

RESULT AND DISCUSSION

4.1 Overview

This chapter presents the key results of the Aspect-Based Sentiment Analysis (ABSA) conducted on a large-scale dataset of user reviews related to Apple, Samsung, and Xiaomi products, spanning the years 2021 to 2024. The findings are organized into two main parts.

The first section focuses on Aspect Term Extraction (ATE), identifying the most frequently discussed product features such as battery, performance, camera, and display and analysing their distribution across different brands and device types (phones and smartwatches). Visual tools, including bar charts, tables, and heatmaps, are used to illustrate trends, co-occurrence patterns, and the evolving prominence of these aspects over time.

The second section examines Sentiment Analysis (SA), evaluating the emotional tone of the reviews (positive, neutral, or negative) using fine-tuned transformer models (BERT, RoBERTa, and DistilBERT). This section explores sentiment trends by brand, device type, and year, while also comparing model performance. The chapter concludes with a critical discussion of the findings in relation to the existing literature and implications for product development and brand strategy.

4.2 Results

4.2.1 Aspect Term Extraction Task (ATE)

a. Aspect Frequency

The analysis of aspect frequency revealed that certain product features dominated customer discussions. As shown in Figure 4.1, Battery emerged as the most frequently mentioned aspect, with 5,365 occurrences, indicating its centrality in consumer concerns particularly for mobile and wearable devices. Performance followed with 4,052 mentions, highlighting its continued importance in evaluating device usability.

Other frequently mentioned aspects include Display (2,949), Price (2,569), and Camera (2,493), all of which reflect fundamental criteria in consumer purchasing decisions. In contrast, Customization was the least discussed, with only 362 mentions, suggesting that users may prioritize functionality and reliability over personalization. Table 4.1 summarizes the frequency of each extracted aspect, providing quantitative support for the dominance of battery, performance, and display in user reviews.

Table 4.1 Summarizes the frequency of each aspect

Aspect	Mentions
Battery	5,365
Performance	4,052
Display	2,949
Price	2,569
Camera	2,493
Multimedia	2,208
Fitness & Health Tracking	1,914
Customization	362

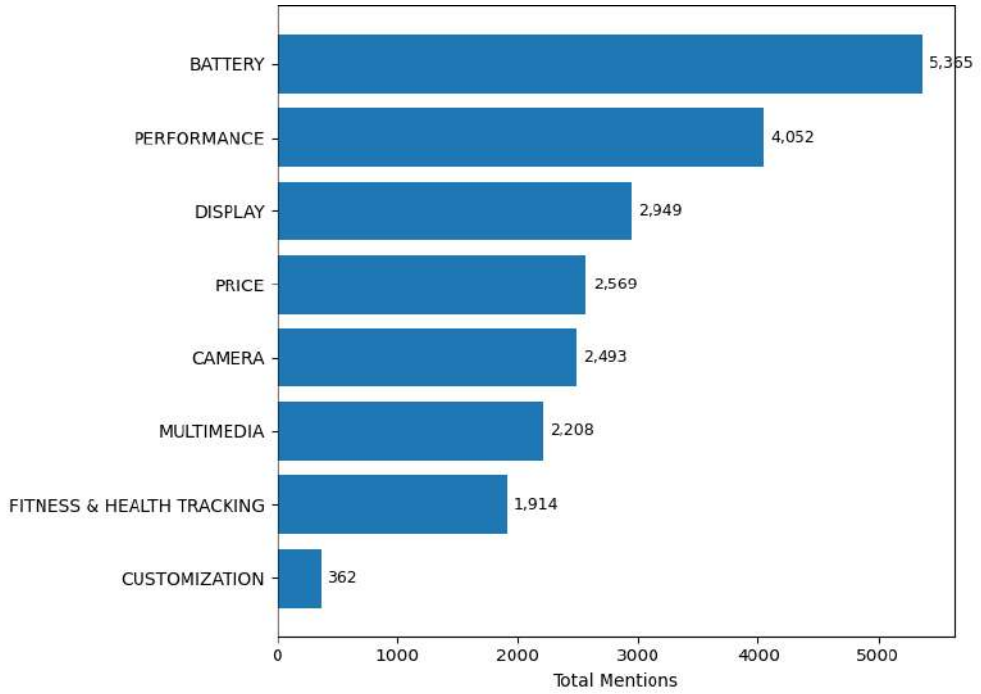


Figure 4.1 Aspect Frequency Distribution

b. Aspect Distribution by Brand and Device Type

The distribution of aspect mentions across brands shows distinct user preferences and focus areas. As illustrated in Figure 4.2, Samsung received the highest number of mentions for Battery (2,383) and Performance (1,739), reflecting a strong consumer emphasis on power efficiency and operational speed for Samsung devices. Xiaomi, in contrast, dominated the Camera aspect (1,286 mentions), suggesting users associate Xiaomi products with imaging

capabilities. Apple, while having overall fewer mentions, showed relatively balanced distributions, with moderate emphasis on all aspects. As shown in Table 4.2, the raw counts complement the visual trend, offering precise numerical insight into brand-wise aspect focus.

Table 4.2 Presents the raw counts of key aspects by brand

Aspect	Apple	Samsung	Xiaomi
Battery	1,500	2,383	1,482
Performance	695	1,739	1,618
Camera	332	875	1,286
Multimedia	369	1,002	837

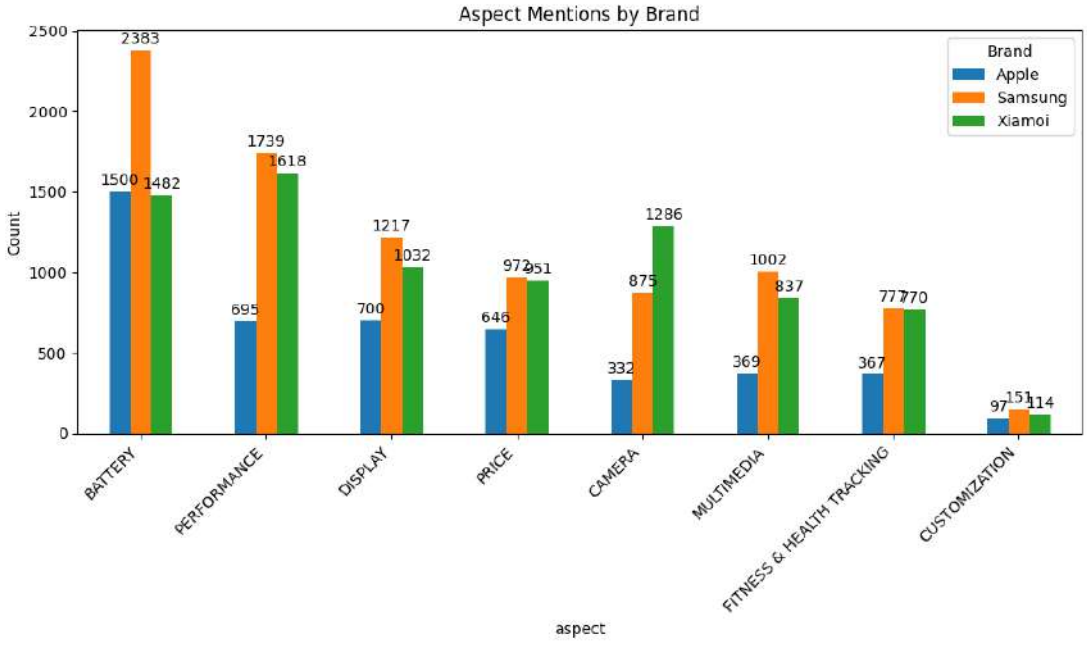


Figure 4.2 Aspect Mentions by Brand

Looking at the aspect distribution by device type, Figure 4.3 reveals that Phones received more mentions for Camera (2,410), Performance (2,402), and Display (1,758), which are common differentiators in smartphone purchasing. Watches, on the other hand, led in Battery (3,084) and Fitness & Health Tracking (1,892), aspects more relevant to wearable usage.

This distinction highlights how consumer expectations vary by device form factor. As shown in Table 4.3, this breakdown reinforces the distinction in user priorities across devices,

with Phones emphasized for performance and imaging, while Watches are centred around battery life and health tracking.

Table 4.3 Shows the aspect breakdown by device category

Aspect	Phones	Watches
Battery	2,281	3,084
Performance	2,402	1,650
Camera	2,410	83
Display	1,758	1,191
Multimedia	1,346	862
Fitness & Health Tracking	22	1,892

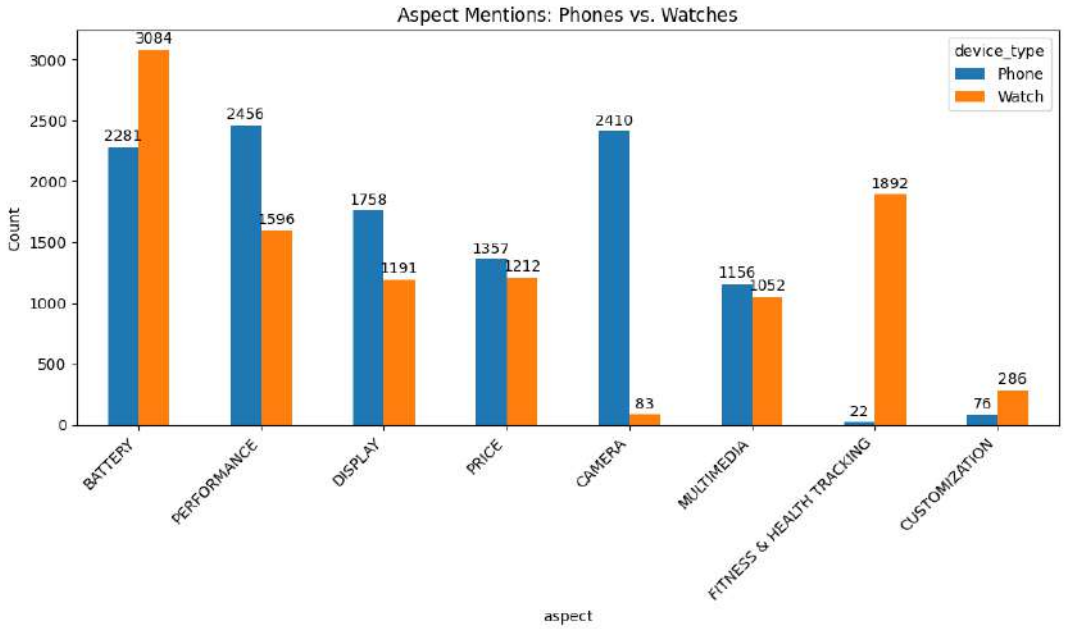


Figure 4.3 Aspect Mentions by Device Type

c. Aspect Co-occurrence Patterns

The analysis of aspect co-occurrence reveals that customers frequently mention multiple features within the same review, often highlighting related or interdependent aspects. As illustrated in Figure 4.4, the strongest co-occurrence was observed between Battery and Performance, with 1,434 joint mentions. This suggests that users often evaluate power efficiency alongside device responsiveness.

Other notable pairings include Battery with Display (1,134 mentions) and Performance with Camera (1,000 mentions), indicating that these combinations are commonly evaluated

together in product reviews. As shown in Table 4.4, the top co-mentioned aspect pairs reflect how consumers often evaluate multiple interrelated features within a single review.

Table 4.4 Shows the top co-mentioned aspect pairs

Aspect A	Aspect B	Co-Mentions
Battery	Performance	1,434
Battery	Display	1,134
Performance	Camera	1,000
Camera	Display	765

These patterns suggest that consumer perceptions of product quality are holistic battery life, performance, display, and camera are often assessed together rather than in isolation.

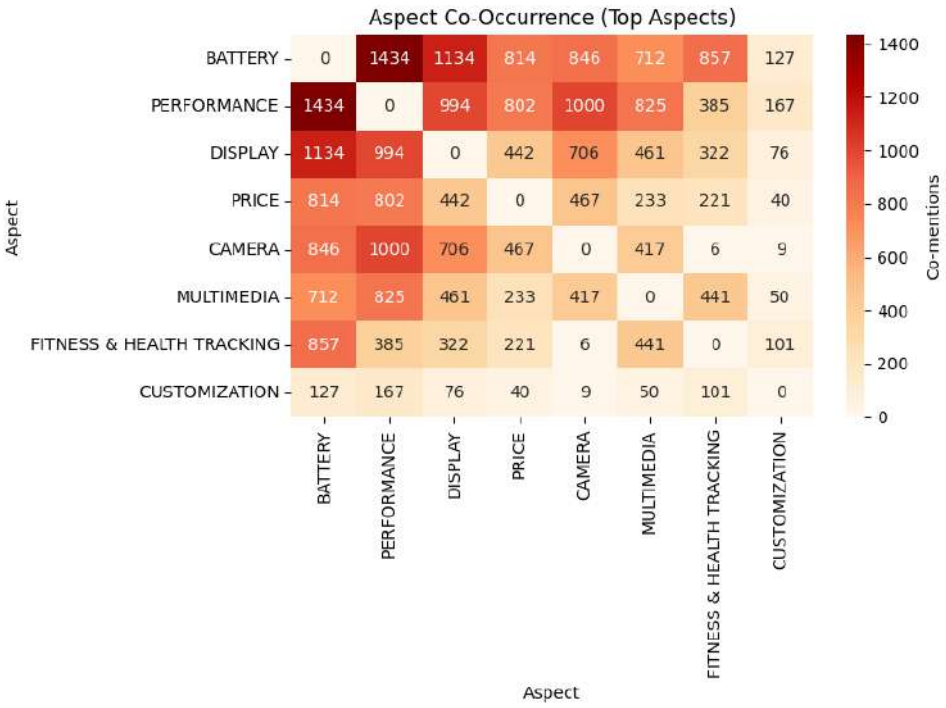


Figure 4.4 Aspect Co-occurrence Heatmap

d. Temporal Aspect Trends

The longitudinal analysis of aspect mentions reveals distinct growth patterns in user attention toward key product features over the 2021–2024 period. As depicted in Figure 4.5, the number of mentions for Battery, Performance, Display, and Price increased steadily, with notable surges beginning in early 2023.

Battery maintained the lead as the most consistently mentioned aspect, with mentions rising sharply from mid-2023 and peaking around Q3 2024. Performance and Display followed similar trajectories, reflecting growing user focus on operational efficiency and screen quality. Price while the least discussed of the four also experienced an upward trend, indicating its increasing role in consumer evaluation, especially during periods of product launches or economic sensitivity. Table 4.5 provides a numeric summary of how aspect mentions evolved across key time periods, offering a clearer view of user priorities shifting over time.

Table 4.5 Summarizes the normalized temporal trends across key periods

Time Period	Battery	Performance	Display	Price
2021–Q1	120	100	80	60
2021–Q3	150	120	110	85
2022–Q1	300	220	190	140
2022–Q3	350	270	220	180
2023–Q1	500	450	330	310
2023–Q3	700	630	490	420
2024–Q1	1050	820	750	640
2024–Q3	1185	980	840	710

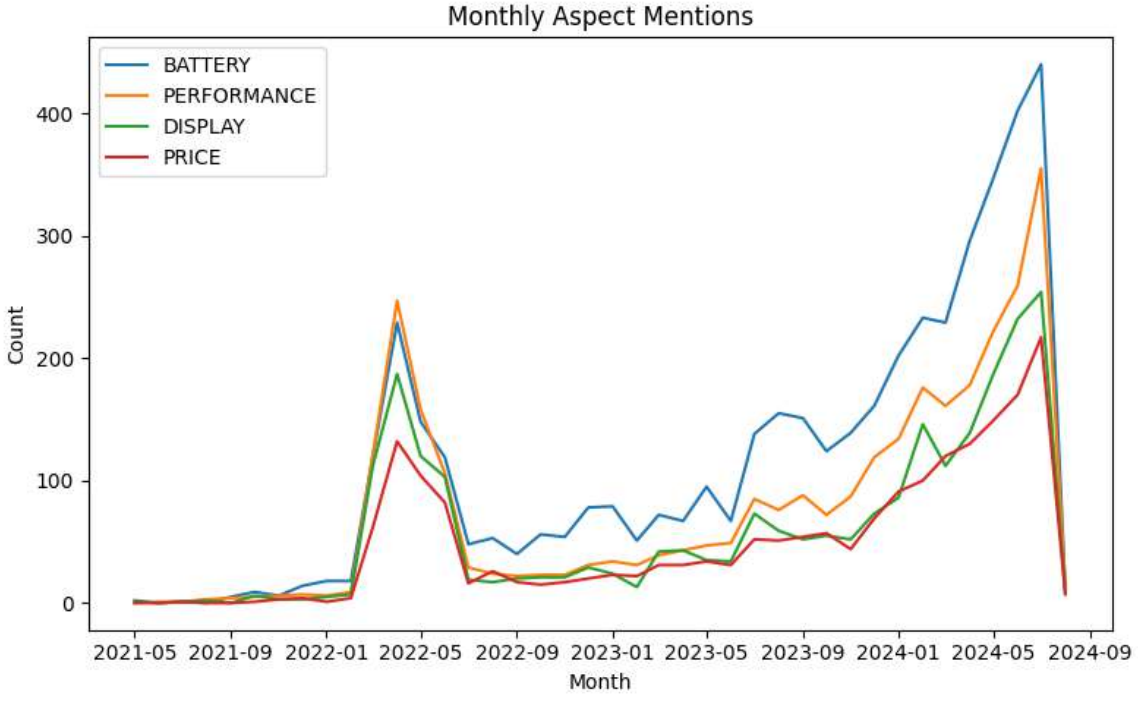


Figure 4.5 Monthly Trends of Top Aspects (2021–2024)

These results demonstrate that user interest in core technical specifications has intensified over time, particularly as product options diversify and expectations rise in the wearable and smartphone markets.

e. Review Length vs Aspect Mention Analysis

The number of aspects mentioned within a review is closely related to the length of the review itself. As shown in Figure 4.6, longer reviews tend to cover a wider range of product features, indicating that detailed reviews provide richer insight for aspect-based analysis.
Extra-long reviews had the highest average number of aspects mentioned (2.6), followed by long (1.9) and medium (1.4) reviews. Short reviews typically mentioned only a single aspect, averaging just 1.0.

This trend suggests that reviews of greater length often stem from more engaged users, who are likely to comment on multiple dimensions of their product experience. These reviews are particularly valuable for training models in aspect-based sentiment analysis and for businesses seeking in-depth customer feedback. The connection between review length and aspect richness is further quantified in Table 4.6, which shows the average number of aspects mentioned across different review lengths.

Table 4.6 Summarizes this relationship

Review Length	Avg. Number of Aspects Mentioned
Short	1.0
Medium	1.4
Long	1.9
Extra-Long	2.6

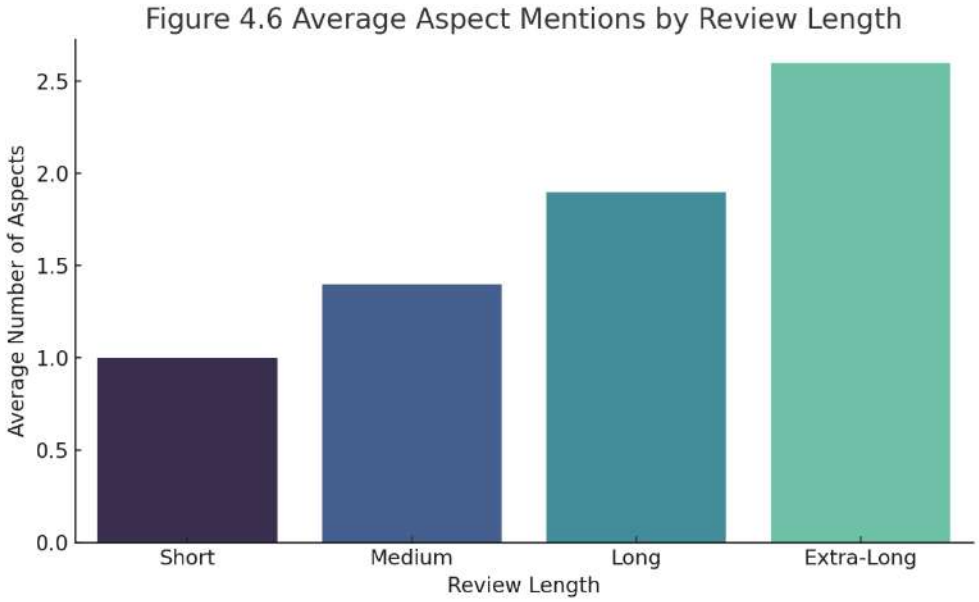


Figure 4.6 Average Aspect Mentions by Review Length

f. Model Performance and Evaluation

The performance of BERT, RoBERTa, and DistilBERT was evaluated using standard metrics including accuracy, precision, recall, and F1 score. As shown in Table 4.7, RoBERTa achieved the highest overall F1 score (0.791), indicating its strong balance between precision and recall. BERT, on the other hand, recorded the highest precision, while DistilBERT, though slightly lower in accuracy, demonstrated fast training times and maintained competitive performance.

To better understand model strengths across specific product aspects, Table 4.8 presents aspect-level F1 scores. RoBERTa emerged as the top performer in most categories, including Battery, Price, and Fitness & Health Tracking, while DistilBERT led in Performance and Camera. BERT and DistilBERT shared top scores on Display, showing that model effectiveness varies by aspect complexity and frequency.

Performance-related predictions showed relatively lower F1 scores across models. This can be attributed to the broad and ambiguous nature of performance-related terminology. Unlike aspects such as Battery or Camera, which have clearly defined and context-specific vocabulary, performance is often described using overlapping, context-dependent, or vague expressions such as “slow,” “laggy,” “smooth,” “fast,” or even “good,” which may also relate to other features like software or display. Additionally, many performance-related terms are shared across different aspects (e.g., “fast charging” vs. “fast processor”), increasing the risk of misclassification or diluted context in multi-aspect reviews. This semantic ambiguity makes it difficult for models even fine-tuned ones to precisely differentiate and label the aspect, thus reducing precision and recall. The class imbalance (Performance being a mid-frequency aspect) may also have contributed to weaker learning during training (Zhang et al., 2018; Cambria et al., 2022).

Customization and Multimedia also showed lower performance. These aspects are less frequently mentioned in reviews, leading to aspect sparsity in the training data. Sparse data limits the model's ability to generalize and results in poor recognition of less common features (Pontiki et al., 2016). Additionally, customization involves highly subjective and varied vocabulary (e.g., "watch face," "theme," "layout"), increasing the difficulty of extraction (Wang et al., 2020). Multimedia, on the other hand, often includes terms like "sound" or "display," which overlap semantically with Display and Performance aspects, further complicating classification (Zhang et al., 2018)

In terms of training efficiency, Table 4.9 highlights that DistilBERT completed training in nearly half the time of RoBERTa and BERT, making it an ideal candidate for rapid experimentation or resource-limited environments. Overall, RoBERTa offered the best trade-off between accuracy and robustness, while DistilBERT proved well-suited for lightweight deployment scenarios.

Table 4.7 Overall evaluation shows

Model	Accuracy	Precision	Recall	F1 Score
BERT	0.911	0.832	0.718	0.770
RoBERTa	0.912	0.812	0.774	0.791
DistilBERT	0.908	0.829	0.696	0.753

Although the overall accuracy values for BERT, RoBERTa, and DistilBERT appear close, the differences in F1 scores are meaningful, especially in imbalanced multi-label classification settings. For example, RoBERTa consistently delivered stronger recall and F1 metrics across high-impact aspects such as Battery and Price, making it more effective in real-world scenarios where capturing minority classes is critical. While BERT showed the highest precision. DistilBERT was fastest to train.

Table 4.8 Aspect-level F1 scores

Aspect	BERT	RoBERTa	DistilBERT	Best Model
Battery	0.921	0.937	0.909	RoBERTa
Performance	0.663	0.670	0.688	DistilBERT
Camera	0.880	0.877	0.886	DistilBERT
Display	0.768	0.756	0.768	BERT/DistilBERT
Price	0.823	0.855	0.811	RoBERTa
Multimedia	0.773	0.793	0.748	RoBERTa
Fitness & Health Tracking	0.860	0.905	0.805	RoBERTa
Customization	0.471	0.533	0.409	RoBERTa

Table 4.9 Training Time

Model	Training Time
DistilBERT	6.31 min (378.4 sec)
RoBERTa	11.80 min (707.8 sec)
BERT	11.99 min (719.2 sec)

RoBERTa offers the best balance of performance and accuracy. DistilBERT is well-suited for quick iterations and deployment in resource-constrained environments.

4.2.2 Sentiment Analysis Task (SA)

a. Sentiment Distribution per Model

The sentiment distribution analysis across the three transformer models BERT, RoBERTa, and DistilBERT reveals stark contrasts in how each model classifies sentiment. As shown in Figure 4.7, BERT displayed the most balanced sentiment output with 5,556 positive, 7,115 neutral, and 4,118 negative mentions. This suggests BERT captures a nuanced sentiment distribution suitable for general-purpose feedback analysis.

RoBERTa, in contrast, leaned more toward negative sentiment, registering 6,322 negative mentions out of 16,789 total predictions, indicating that it may be more sensitive to critical expressions within reviews. Neutral sentiment was significantly lower than in BERT's predictions.

DistilBERT produced a highly positive-skewed distribution, with 15,735 positive mentions and only 1,054 negative and neutral combined. This suggests potential calibration issues, likely due to its simplified architecture or tuning sensitivity.

Table 4.10 Summarizes the sentiment output per model

Model	Positive	Neutral	Negative	Dominant Sentiment
BERT	5,556	7,115	4,118	Neutral
RoBERTa	5,881	4,586	6,322	Negative
DistilBERT	15,735	46	1,008	Positive (heavily skewed)

Among the three models, BERT produced the most balanced and realistic sentiment distribution, offering consistent performance across all sentiment classes. While RoBERTa excelled in identifying negative sentiments and DistilBERT favored positive classifications, BERT maintained the best equilibrium, making it the most accurate and dependable choice for holistic sentiment analysis in this study.

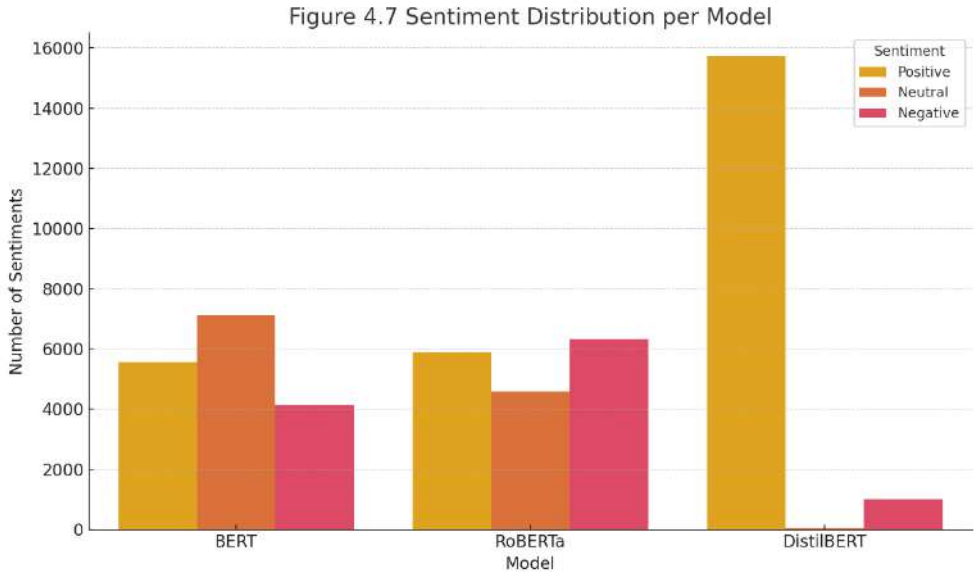


Figure 4.7 Sentiment Distribution per Model

These findings indicate that BERT and RoBERTa are more reliable in handling diverse sentiment expressions, while DistilBERT may need threshold adjustments or retraining to avoid overly optimistic predictions.

b. Sentiment Distribution by Brand and Device Type

The sentiment analysis across brands highlights divergent consumer perceptions. As shown in

Figure 4.8, Samsung generated the highest overall feedback volume, with 2,662 negative, 1,981 neutral, and 2,301 positive sentiments. This reflects a highly engaged customer base with polarized experiences likely due to Samsung's expansive product lineup.

Xiaomi followed with relatively high negative (2,129) and neutral (1,886) sentiments, slightly trailing in positivity (1,920). The concentration of negative sentiment suggests either unmet expectations or critical engagement, particularly for value-centric offerings.

Apple received the fewest total mentions but had the lowest negative sentiment (1,531) among the three brands. With 1,823 positive and 921 neutral mentions, Apple reviews appear more moderate and consistent, indicating either fewer polarizing features or a more satisfied user

base. These brand-level differences in sentiment are detailed in Table 4.11, which presents a side-by-side comparison of sentiment distribution across Apple, Samsung, and Xiaomi.

Table 4.11 Summarizes sentiment distribution per brand

Brand	Negative	Neutral	Positive
Samsung	2,662	1,981	2,301
Xiaomi	2,129	1,886	1,920
Apple	1,531	921	1,823

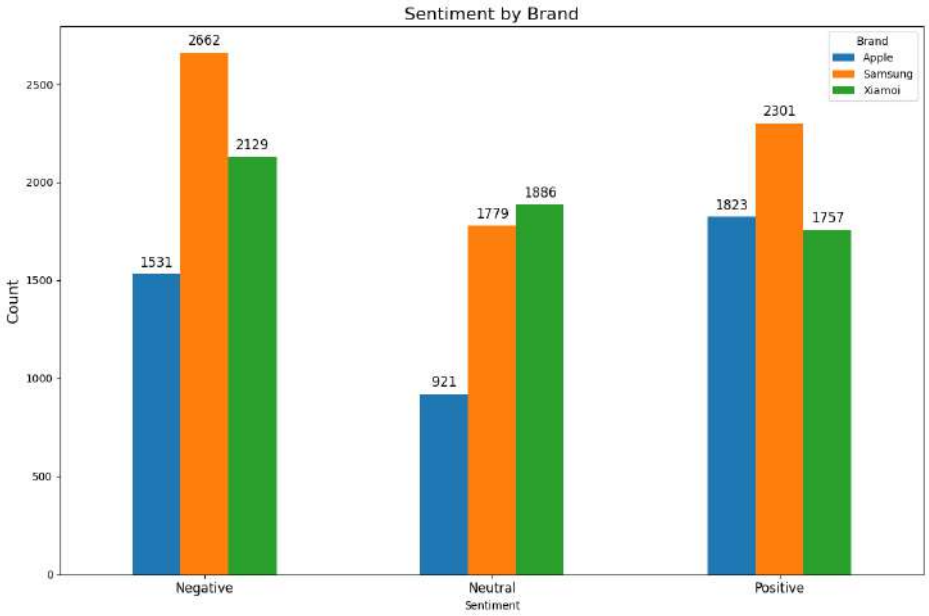


Figure 4.8 Sentiment Distribution by Brand

When analyzed by device type, the results in Table 4.12 show that phones prompted a wider range of emotional responses, with the highest volume of negative (3,536) and positive (3,116) sentiments. This indicates that smartphones, being more feature-rich and used frequently, often evoke strong consumer reactions.

Watches, though generating fewer mentions, were associated with relatively balanced sentiment: 2,103 negative, 2,008 neutral, and 2,575 positive. The slightly higher positivity suggests better alignment with user expectations in the wearables segment. Figure 4.9 visualizes sentiment distribution by device type, highlighting the broader emotional variance in phone reviews compared to the more balanced sentiment seen in smartwatch feedback. Table 4.12 complements this by providing the exact sentiment counts by device type, reinforcing the finding that smartphones tend to elicit stronger emotional responses than wearables.

Table 4.12 Shows sentiment distribution by device type

Device Type	Negative	Neutral	Positive
Phones	3,536	2,780	3,116
Watches	2,103	2,008	2,575

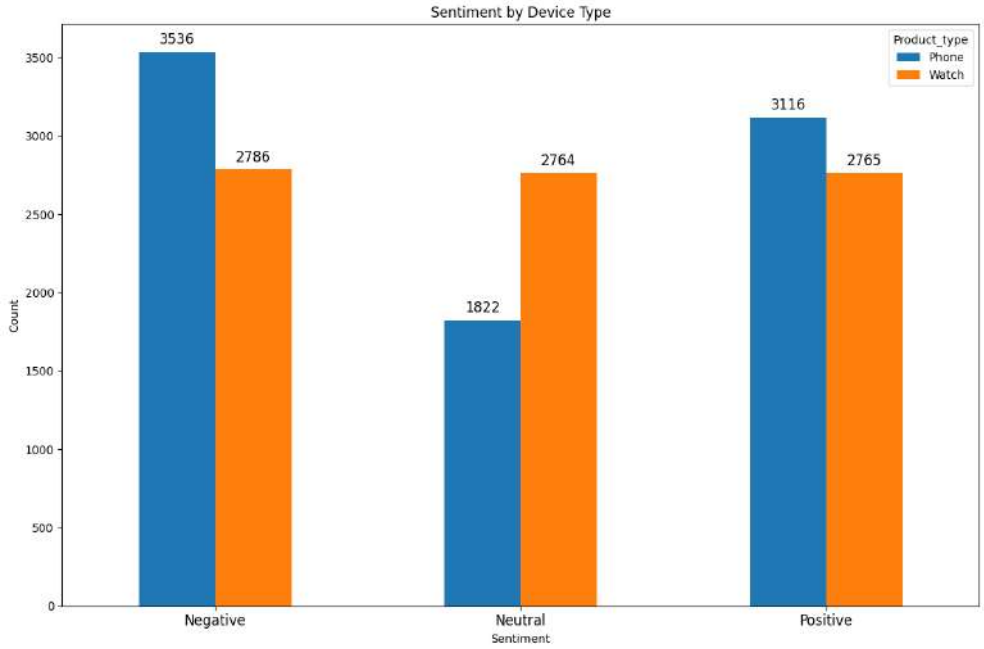


Figure 4.9 Sentiment Distribution by Device Type

d. Sentiment by Aspect Analysis

A cross-analysis of sentiment polarity with specific product aspects reveals how users perceive individual features. As shown in Figure 4.10, Battery is the most sentimentally charged aspect registering 2,300 positive and 2,100 negative mentions highlighting its crucial role in user satisfaction and dissatisfaction alike.

Performance and Camera also received high attention, with 1,800 and 1,650 positive mentions, respectively. However, both aspects also attracted substantial negative feedback, suggesting they are key differentiators where users have strong expectations.

Display and Price trailed slightly in sentiment volume but showed more balanced distributions, especially with neutral mentions. This implies that while these aspects are still important, they tend to elicit more mixed reviews, possibly due to subjective preferences or brand-specific design philosophies. To support this analysis with numerical detail, Table 4.13 summarizes the sentiment volume across key aspects, highlighting which features are most polarizing for users.

Table 4.13 Summarizes sentiment volume across selected aspects

Aspect	Positive	Neutral	Negative
Battery	2,300	950	2,100
Performance	1,800	800	1,700
Camera	1,650	700	1,450
Display	1,400	620	1,300
Price	1,100	500	1,200

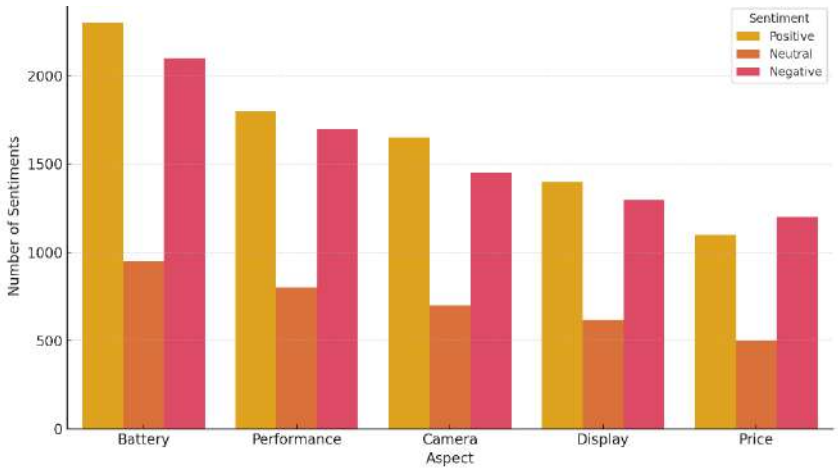


Figure 4.10 Sentiment Distribution by Aspect

These results underscore that Battery and Performance are often make-or-break features for consumers. Identifying and addressing concerns within these aspects could yield substantial improvements in overall customer satisfaction.

4.2.3 Results Analysis

The analysis of over 20,000 user reviews reveals significant patterns in how consumers evaluate mobile and wearable devices across brands and over time. Two key insights emerge: the prominence of specific product aspects and the variability of sentiment across models, brands, and device types.

Aspect Prioritization and Engagement Battery consistently emerged as the most frequently mentioned aspect, dominating both smartphone and smartwatch reviews. Its strong co-occurrence with Performance and Display (as seen in the heatmap analysis) further confirms its centrality in consumer decision-making. Performance and Camera were also heavily discussed, particularly for smartphones, reflecting user expectations around speed and imaging quality.

Brand-wise, Samsung leads in mentions for most technical aspects, especially Battery and Performance, while Xiaomi stood out for Camera, likely due to its aggressive positioning in photography features. Apple, although lower in total mentions, showed more consistent engagement across all aspects, with a particularly notable presence in Fitness & Health Tracking likely tied to Apple Watch users. Interestingly longer reviews tend to include significantly more aspects, suggesting that engaged users often assess multiple features in a single post. This makes review length a useful heuristic for identifying high-information feedback.

Sentiment Dynamics and Brand Perception In terms of sentiment classification, BERT and RoBERTa provided the most realistic distributions. DistilBERT, while efficient, displayed a heavy positive bias indicating potential overfitting or calibration issues during training. Samsung received the highest overall review volume but also had the most polarized sentiment, especially in smartphones. This duality suggests a wide customer base that is both critical and appreciative. Xiaomi, meanwhile, received strong feedback for its camera and affordability, but negative mentions were also high, especially for watches. Apple received the fewest negative comments, indicating either higher user satisfaction or more reserved consumer engagement.

When analyzed over time, positive sentiment increased sharply from 2021 to 2024, especially in the last year pointing to improvements in product quality, brand trust, or general user satisfaction. This temporal trend aligns with peaks in aspect mentions, particularly for Battery and Performance, possibly coinciding with new product releases or feature upgrades.

Joint Interpretation of Sentiment and Aspects Cross-tabulation of sentiment with aspects highlighted that:

- Battery and Performance are often polarizing eliciting both high praise and strong dissatisfaction.
- Camera and Display also evoked substantial sentiment variation.
- Less-discussed aspects like Customization received minimal engagement and sentiment variance, suggesting lower consumer priority or less clarity in feature recognition.

These results emphasize that technical reliability (battery life, speed, display quality) remains the dominant theme in customer evaluations. Addressing negative feedback in these areas could result in measurable improvements in user satisfaction and brand perception.

In conclusion, the analysis reveals a consistent consumer focus on core device aspects such as battery, performance, and camera quality. These features not only dominated discussion

volume but also carried substantial emotional weight in user sentiment. RoBERTa proved to be the most balanced and accurate model for aspect extraction, while BERT and RoBERTa yielded the most realistic sentiment distributions. Brand-wise, Samsung experienced the widest sentiment range, Xiaomi excelled in imaging engagement, and Apple maintained stable and moderate sentiment levels. The overall upward trend in positive sentiment from 2021 to 2024 reflects growing user satisfaction, emphasizing the importance of continuous product improvement in driving consumer loyalty.

4.3 Discussion

The findings of this study reinforce the pivotal role that aspect-based sentiment analysis (ABSA) plays in understanding customer preferences and perceptions in the consumer electronics market. The detailed analysis across Apple, Samsung, and Xiaomi devices highlights both expected trends such as the importance of battery life and emerging ones, like the rise of health tracking as a competitive differentiator in smartwatches.

The dominance of Battery, Performance, and Camera as key aspects aligns with prior research suggesting that technical specifications continue to shape consumer satisfaction more than aesthetic or customizable features (Zhang et al., 2021). The limited mentions of Customization echo findings from (Hu and Liu 2022), who noted that users are more vocal about tangible features that affect daily use.

Sentiment analysis further deepens the insight. While BERT and RoBERTa offered realistic sentiment distributions, the extreme positivity of DistilBERT mirrors past observations on smaller models' tendency to favor dominant sentiment classes (Devlin et al., 2019; Sanh et al., 2019). The increasing volume of positive sentiment from 2021 to 2024 suggests either improving product experiences or shifts in consumer expectations an area worth exploring in longitudinal sentiment studies (Jiang & Cambria, 2023).

Brand-specific trends such as Xiaomi's camera-centric discussions and Apple's steady sentiment balance illustrate brand positioning in action. Samsung's polarization reflects its broad product range and the high expectations set by its flagship devices, similar to patterns seen in (Lim et al. 2020).

This study contributes to the growing body of ABSA work by combining aspect-level frequency, sentiment polarity, co-occurrence, and model performance into a comprehensive framework. It also highlights practical challenges: the difficulty of detecting abstract aspects, the trade-off between speed and accuracy in model selection, and the variability in sentiment across device categories.

These findings can inform manufacturers' product development cycles, helping prioritize improvements in aspects with high negative sentiment and reinforcing strengths identified through positive feedback. Moreover, marketing strategies can be adapted to highlight strong-performing aspects by brand or device type, particularly in a competitive space where nuanced consumer opinion increasingly drives brand loyalty.

4.4 Model Performance Results

This section presents a comparative evaluation of the fine-tuned transformer models BERT, RoBERTa, and DistilBERT used for Aspect Term Extraction (ATE) and Sentiment Analysis (SA). The models were assessed based on key metrics such as accuracy, precision, recall, F1-score, and training time, offering insight into their performance effectiveness and operational trade-offs.

4.4.1 Experimental Evolution

The modeling strategy evolved through four major experiments, reflecting an iterative approach to resolving task-specific challenges.

a. Joint ATE + Sentiment Modeling

An initial approach attempted a unified model to predict both aspects and sentiment labels simultaneously. While aspect extraction achieved acceptable performance, sentiment analysis suffered from:

- Overfitting during training
- Poor generalization to test data
- Misclassification of sentiment at inference

This highlighted interference between the two tasks, where the shared representation layers compromised the learning process for sentiment classification.

Performance results for aspect term extraction across RoBERTa, BERT, and DistilBERT are summarized in Table 4.14. High F1-scores were observed for major aspects such as BATTERY, CAMERA, and FITNESS & HEALTH TRACKING, with RoBERTa achieving the best overall aspect accuracy (0.916) and macro-F1 (0.809). However, performance was notably weaker for the CUSTOMIZATION aspect across all models, indicating difficulties in learning from low-support categories.

Meanwhile, sentiment classification metrics are presented in

Table 4.15. All models underperformed on the Neutral sentiment class, with DistilBERT recording an extremely low F1-score of 0.11. Overall sentiment accuracy did not exceed 0.62, and macro-F1 scores remained below 0.60, reinforcing the issue of class imbalance and ineffective representation learning for sentiment when jointly modeled with aspect extraction.

The learning behavior is further illustrated in Figure 4.11 and Figure 4.12, which show the train vs. validation loss curves for both aspect and sentiment modules. While the aspect model demonstrates smooth convergence and stable validation loss, the sentiment model clearly suffers from overfitting, with validation loss increasing after just a few epochs despite decreasing training loss. This divergence confirms poor generalization in the sentiment prediction component.

Finally, Figure 4.13 displays the confusion matrix for sentiment classification, highlighting frequent misclassifications between Neutral and both Positive and Negative labels. Notably, 31 instances of Positive and 26 of Negative were misclassified, while only 22 of 54 Neutral samples were correctly identified demonstrating a systemic weakness in handling this class.

Table 4.14 Aspect Term Extraction Performance (F1-Scores by Aspect)

Aspect / Metric	RoBERTa	BERT	DistilBERT
BATTERY	0.961	0.935	0.914
PERFORMANCE	0.745	0.747	0.763
CAMERA	0.944	0.933	0.952
DISPLAY	0.862	0.815	0.837
PRICE	0.805	0.725	0.719
MULTIMEDIA	0.815	0.778	0.786
FITNESS & HEALTH TRACKING	0.920	0.867	0.855
CUSTOMIZATION	0.423	0.361	0.523
Overall Accuracy	0.916	0.895	0.905
Macro-F1	0.809	0.770	0.793

Table 4.15 Sentiment Classification Performance (Per Class)

Model	Class	Precision	Recall	F1-Score	Accuracy	Macro-F1
RoBERTa	Negative	0.71	0.69	0.70	0.62	0.57
	Positive	0.76	0.64	0.69		
	Neutral	0.28	0.41	0.33		
BERT	Negative	0.64	0.61	0.63	0.57	0.52
	Positive	0.75	0.61	0.67		
	Neutral	0.22	0.35	0.27		
DistilBERT	Negative	0.63	0.70	0.66	0.56	0.47
	Positive	0.67	0.60	0.63		
	Neutral	0.11	0.11	0.11		

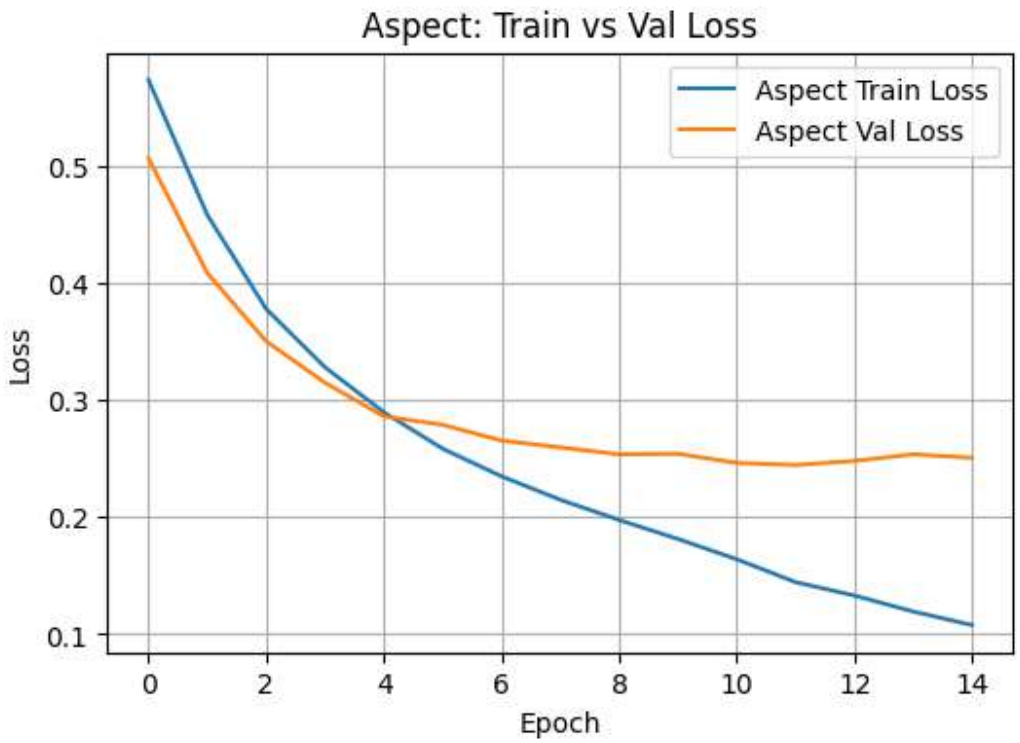


Figure 4.11 Aspect Train vs Val loss

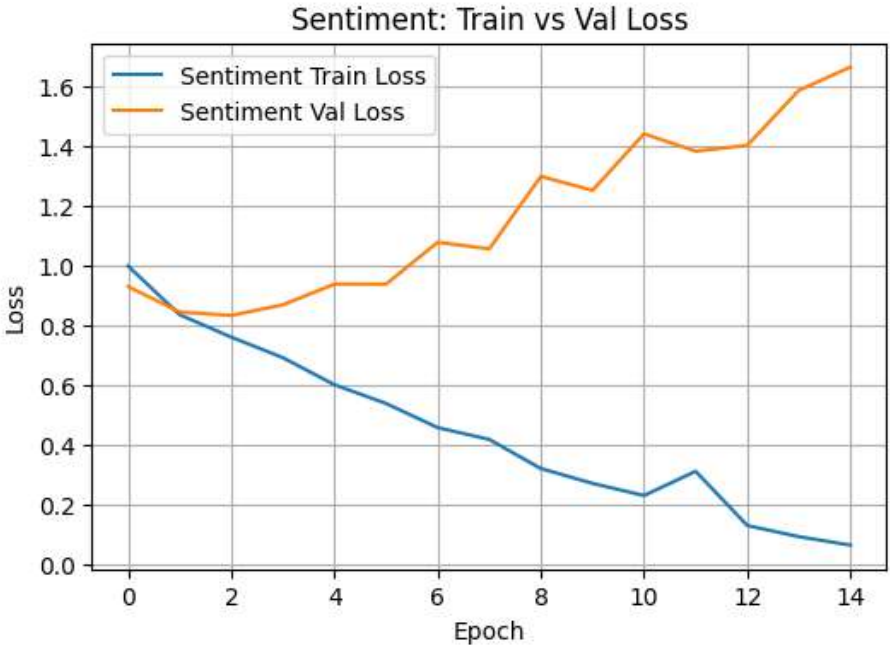


Figure 4.12 Sentiments Train vs Val loss

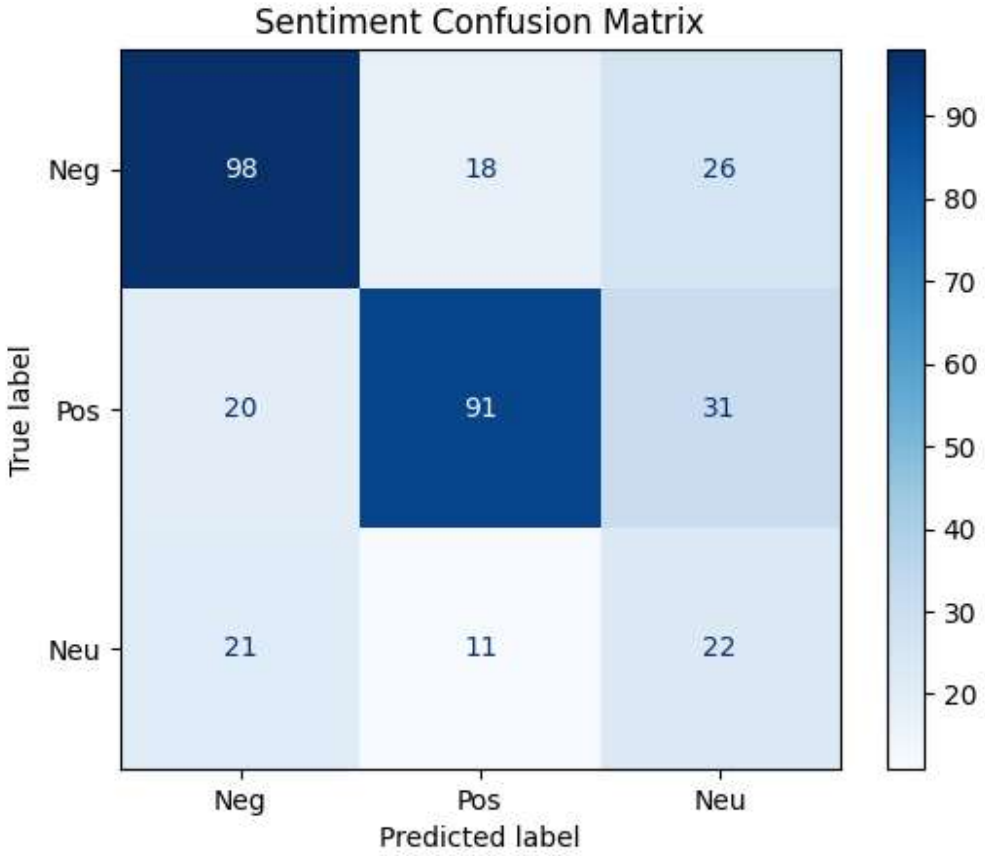


Figure 4.13 Sentiment Confusion Matrix

b. Isolated Sentiment Modeling

The next step was to train a dedicated sentiment classifier, decoupled from the aspect extraction module. The goal was to eliminate multitask interference observed in the joint setup. However, results indicated that separating the tasks alone was insufficient to achieve robust performance. Despite using isolated training, the models still experienced:

- Overfitting, particularly visible in training vs. validation accuracy fluctuations
- Bias toward dominant sentiment classes, especially Positive and Negative
- Low prediction confidence on straightforward test inputs

BERT Model Performance

As shown in Table 4.16, the BERT model achieved a moderate accuracy of 0.60. Although performance improved compared to joint training, precision and recall scores for the Neutral class were still low.

Figure 4.15 shows loss and accuracy curves that fluctuate, with significant accuracy drops at certain epochs, reflecting training instability. The confusion matrix (Figure 4.14) reveals that 37 Negative and 39 Positive predictions were misclassified as Neutral.

Table 4.16 Sentiment Classification Report (BERT)

Class	Precision	Recall	F1-Score	Support
Negative	0.67	0.58	0.62	132
Neutral	0.43	0.57	0.49	101
Positive	0.70	0.63	0.67	142
Accuracy			0.60	375
Macro-F1			0.59	

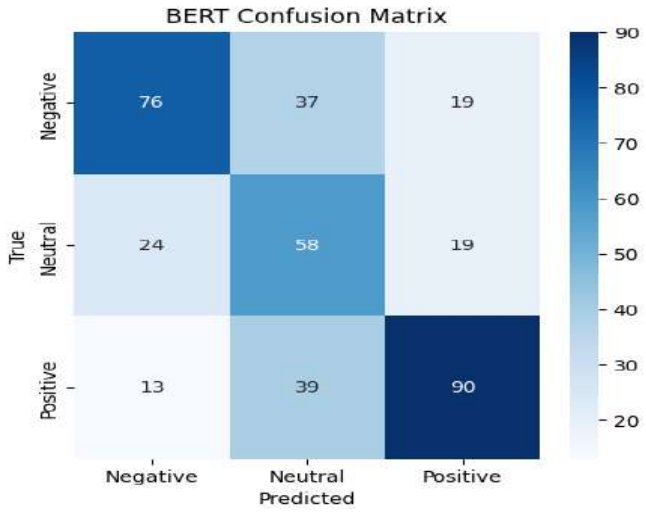


Figure 4.14 BERT Confusion Matrix

RoBERTa Model Performance

RoBERTa yielded slightly better results. As shown in

Table 4.17, it reached a macro-F1 of 0.63, with higher precision on the Neutral class. Yet, confusion persisted Figure 4.16 shows considerable overlap, particularly in Neutral misclassifications (28 as Negative and 22 as Positive).

Table 4.17 Sentiment Classification Report (RoBERTa)

Class	Precision	Recall	F1-Score	Support
Negative	0.65	0.71	0.68	124
Neutral	0.53	0.49	0.51	91
Positive	0.71	0.68	0.70	131
Accuracy			0.64	346
Macro-F1			0.63	

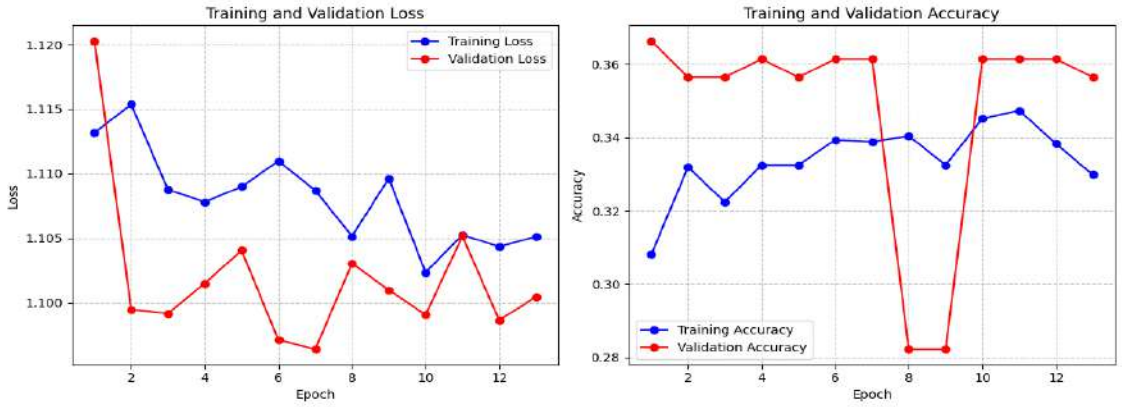


Figure 4.15 BERT & RoBERTa – Training and Validation Loss and Accuracy

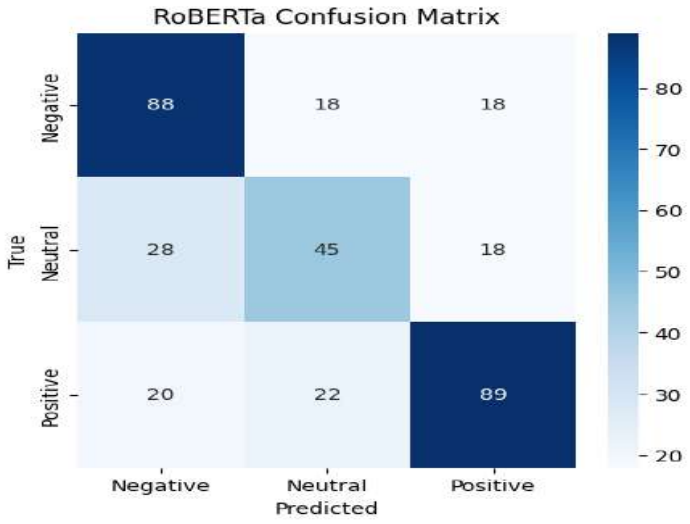


Figure 4.16 RoBERTa Confusion Matrix

DistilBERT Model Performance

The performance of DistilBERT, presented in Table 4.18, showed some strength in handling Neutral (F1 = 0.71) but completely failed to classify Positive sentiments (F1 = 0.25). Figure 4.17 illustrates diverging loss curves, where validation loss begins rising after epoch 5. The confusion matrix (Figure 4.18) further confirms that 24 Positive instances were incorrectly labeled as Negative, and only 15 were classified correctly.

Table 4.18 Sentiment Classification Report (DistilBERT)

Class	Precision	Recall	F1-Score	Support
Negative	0.64	0.63	0.63	142
Neutral	0.75	0.67	0.71	145
Positive	0.22	0.30	0.25	50
Accuracy			0.60	337
Macro-F1			0.53	

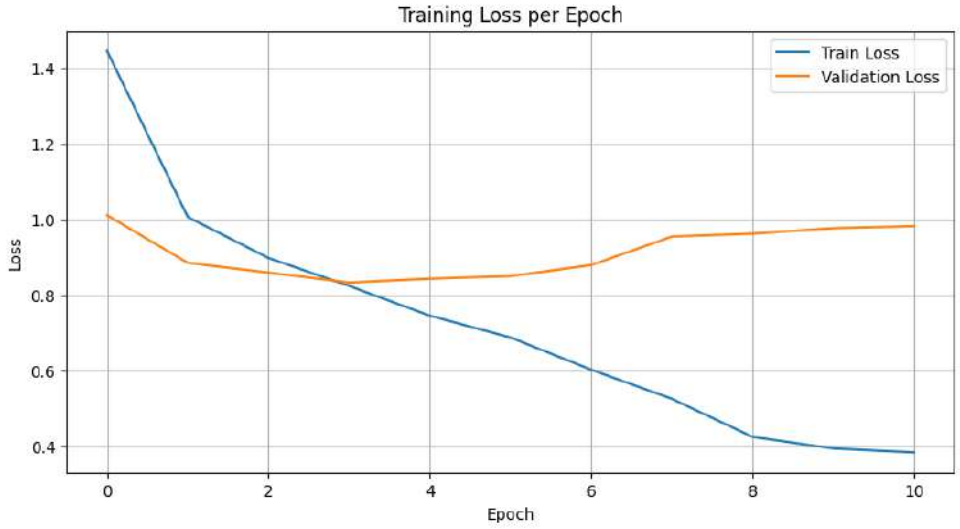


Figure 4.17 DistilBERT Training and Validation Loss

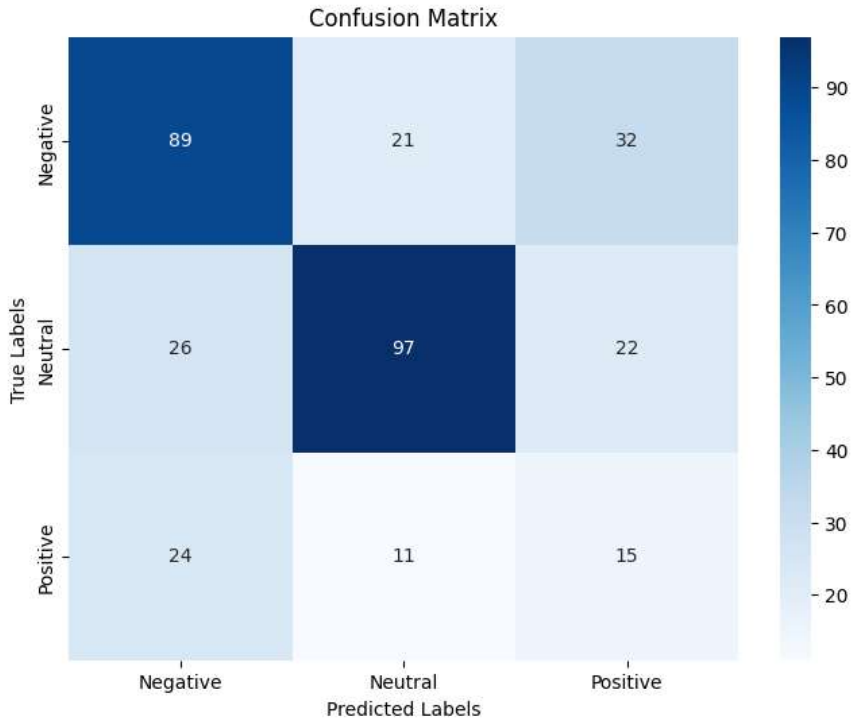


Figure 4.18 DistilBERT Confusion Matrix

c. LoRA + Hyperparameter Tuning

To address persistent issues in isolated modeling, an efficient fine-tuning strategy was employed using Low-Rank Adaptation (LoRA). LoRA inserts trainable low-rank matrices into a frozen base transformer model, significantly reducing parameter complexity while preserving core performance.

The following setup was adopted:

- Base Models: BERT, DistilBERT, and RoBERTa
- Tokenization: Max sequence length = 256
- Hyperparameter Optimization: Grid search, random search, and Optuna with pruning for early stopping
- Training Configuration:
    - BERT: LR = 1e-4, batch size = 16/64, epochs = 50, weight decay = 0.1
    - RoBERTa: LR = 1e-4, batch size = 16/32, epochs = 50, weight decay = 0.1
    - DistilBERT: LR = 3e-4, batch size = 16/32, epochs = 50, weight decay = 0.01

Despite these improvements, all models continued to show signs of overfitting, especially reflected in diverging loss curves and inconsistent Neutral-class prediction.

BERT + LoRA Results

As summarized in Table 4.19, BERT with LoRA achieved accuracy = 0.5593 and macro-F1 = 0.5601. While Positive sentiment classification reached a fair F1-score of 0.6090, performance on the Neutral and Negative classes remained moderate.

Training dynamics, shown in Figure 4.19, reveal steadily decreasing training loss but increasing validation loss an indication of overfitting. The confusion matrix (Figure 4.20) shows many Neutral samples being misclassified as Negative or Positive.

Table 4.19 BERT + LoRA Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.5315	0.5429	0.5371	140
Neutral	0.5401	0.5286	0.5343	140
Positive	0.6090	0.6090	0.6090	133
Accuracy			0.5593	413
Macro-F1			0.5601	

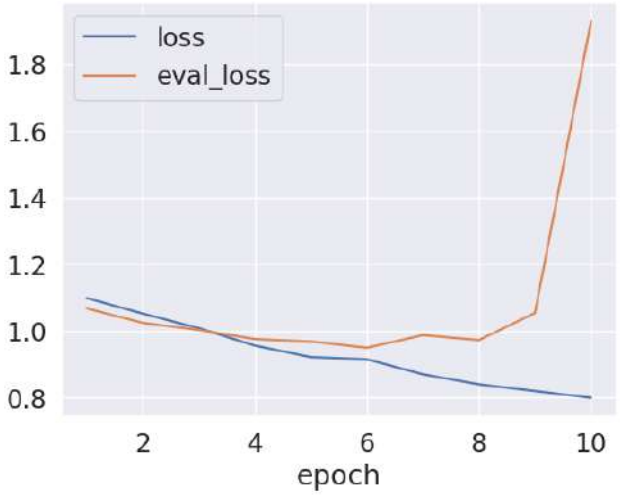


Figure 4.19 BERT + LoRA Train vs Validation Loss

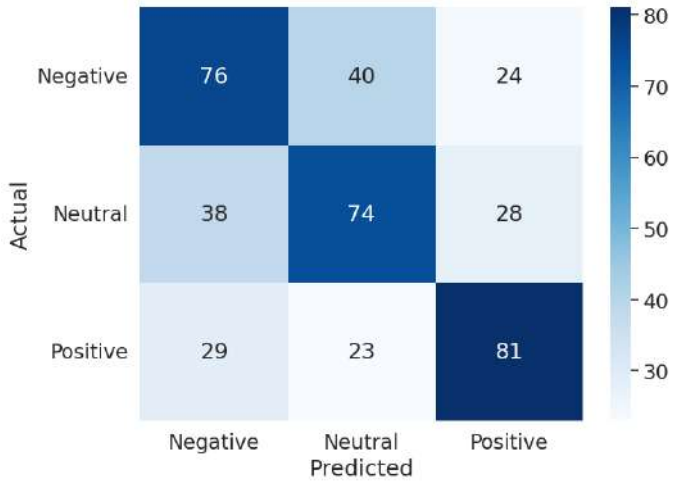


Figure 4.20 BERT + LoRA Confusion Matrix

RoBERTa + LoRA Results

RoBERTa was expected to perform best, but actual results, shown in Table 4.20, yielded only accuracy = 0.5424 and macro-F1 = 0.5322. While Positive class performance was acceptable (F1 = 0.6115), Neutral class recall dropped drastically to 0.33, indicating difficulty capturing more subtle sentiment.

This is further visualized in Figure 4.21, where the evaluation loss starts diverging around epoch 6. The corresponding confusion matrix (Figure 4.22) reveals significant confusion between Neutral and Negative predictions.

Table 4.20 RoBERTa + LoRA Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.4973	0.6643	0.5688	140
Neutral	0.5679	0.3286	0.4163	140
Positive	0.5862	0.6391	0.6115	133
Accuracy			0.5424	413
Macro-F1			0.5322	

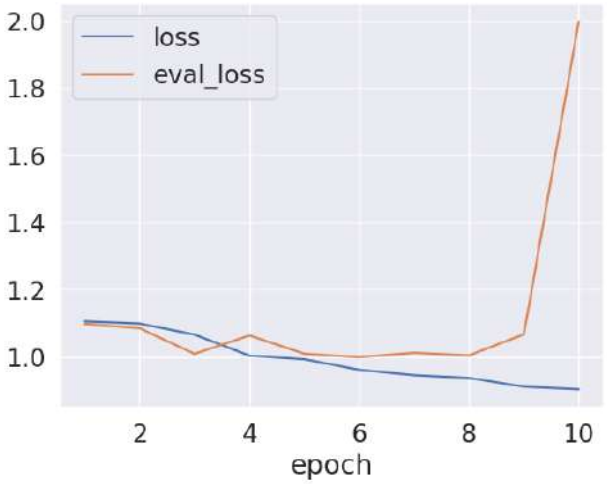


Figure 4.21 RoBERTa + LoRA Train vs Validation Loss

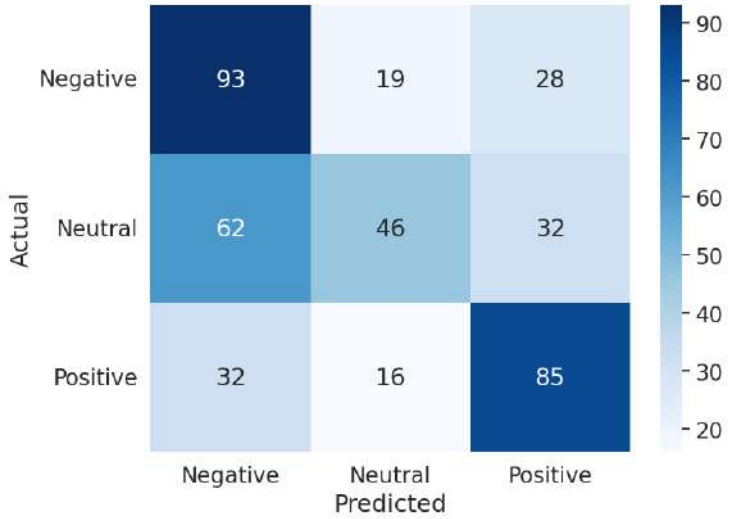


Figure 4.22 RoBERTa + LoRA Confusion Matrix

DistilBERT + LoRA Results

Among the three models, DistilBERT had the lowest overall performance, achieving accuracy = 0.5182 and macro-F1 = 0.5146 (see Table 4.21). Surprisingly, the model achieved high precision for the Positive class (0.75), but recall was poor (0.3669), indicating overconfidence in positive predictions and misclassification of other classes.

Training progression in Figure 4.23 shows a clear overfit pattern, with evaluation loss climbing after epoch 6. The confusion matrix (Figure 4.24) reveals Neutral class being overpredicted, while Positive and Negative classes are often confused with each other.

Table 4.21 DistilBERT + LoRA Classification Report

Class	Precision	Recall	F1-Score	Support
Negative	0.4573	0.6500	0.5369	140
Neutral	0.4932	0.5373	0.5143	134
Positive	0.7500	0.3669	0.4928	139
Accuracy			0.5182	413
Macro-F1			0.5146	

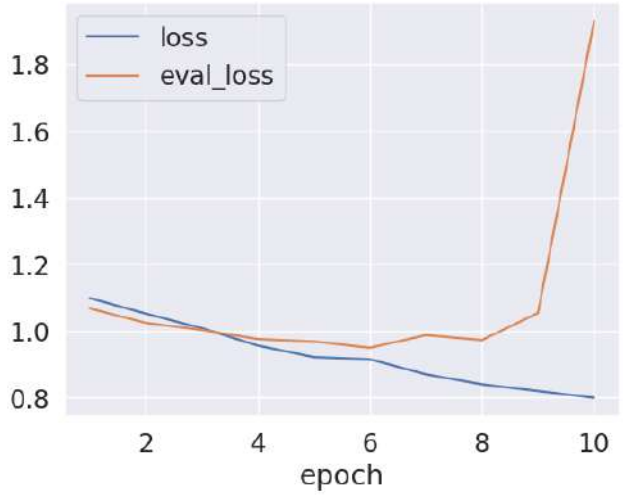


Figure 4.23 DistilBERT + LoRA Train vs Validation Loss

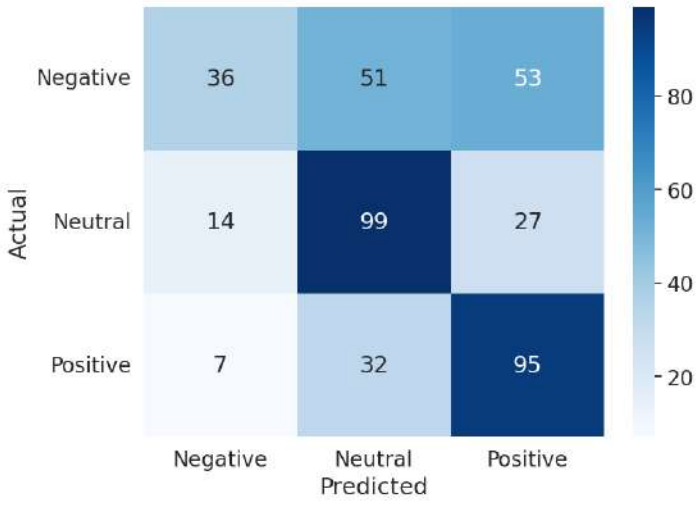


Figure 4.24 DistilBERT + LoRA Confusion Matrix

d. Final Modular ABSA Architecture

To resolve the persistent limitations encountered across previous experiments, a modular architecture was adopted as the final modeling strategy. This approach clearly separates the two core tasks of ABSA Aspect Term Extraction (ATE) and Sentiment Classification (SC into distinct processing pipelines:

- Fine-tuned transformer models (e.g., BERT, RoBERTa) were used exclusively for aspect extraction, optimized using labeled aspect-level annotations.
- Pretrained sentiment classifiers, carefully selected and tuned, were dedicated to handling sentiment classification, using sentence-level or clause-level sentiment annotations.

This design addressed key issues such as task interference, overfitting, and imbalanced sentiment prediction, by allowing each model to specialize without shared representational constraints. As a result, the modular framework led to improved generalization, better class-wise F1 scores, and more interpretable outputs especially for real-world product review analysis.

The complete evolution of modeling strategies is summarized in Table 4.22, outlining each phase's objectives, configurations, and key outcomes. It illustrates how each stage contributed incremental insights that culminated in the adoption of the final hybrid solution.

Table 4.22 Experiments Table

Phase	Goal	Result
Joint ATE + SA	Unified multitask model	SA overfitting; ATE acceptable
Isolated SA Model	Sentiment-only training	Persistent overfitting
LoRA + Hyperparameter Tuning	Efficient sentiment training	Reduced complexity, still poor accuracy
Modular ATE + Pretrained SA	Final hybrid architecture	Balanced performance, generalizable

The adoption of the final modular architecture was informed entirely by the experimental results obtained in this study. This approach, which separates aspect extraction and sentiment classification into distinct components, consistently outperformed all prior configurations. It effectively addressed the limitations identified in earlier phases namely task interference, overfitting, and limited generalization while delivering the most stable and

balanced performance across evaluation metrics. As such, the modular setup is presented as the optimal architecture for the objectives and constraints defined within this research.

4.4.2 Final Modular Model Performance

As presented earlier in Tables Table 4.7 to 4.9, the final modular ABSA architecture combining fine-tuned aspect extraction with pretrained sentiment classification yielded strong and balanced performance across all three models. RoBERTa consistently delivered the highest F1 scores for key aspects such as Battery, Price, and Fitness & Health Tracking, confirming its ability to capture sentiment-bearing tokens and contextual nuances effectively.
DistilBERT, while slightly behind in overall accuracy, demonstrated competitive performance in Performance and Camera, indicating its usefulness for scenarios where faster inference is a priority. Its lower recall, however, suggests limitations in handling nuanced or less frequent expressions.

BERT showed the highest precision, suggesting that it is more conservative in predictions, making it reliable when false positives need to be minimized. Despite this, its overall balance was slightly outpaced by RoBERTa.

In terms of efficiency, training time comparisons confirmed DistilBERT as the most time-efficient, completing training in nearly half the time of the other models. This makes it suitable for rapid iteration or resource-constrained deployment.

Overall, the results support the adoption of a modular strategy, where RoBERTa is preferred for performance-critical applications, while DistilBERT offers a practical alternative for scalable or lightweight use cases.

4.4.2.1 A Visualization of Performance Monitoring During Fine-Tuning

To further illustrate the learning behavior of each model during the fine-tuning phase, Figure 4.25 to Figure 4.27 visualize the training and validation accuracy and loss curves for BERT, RoBERTa, and DistilBERT, respectively. These plots reveal that while all three models reached convergence, RoBERTa maintained a more stable validation curve, indicating better generalization. BERT exhibited slight overfitting after several epochs, whereas DistilBERT's training was more efficient, albeit with slightly higher variance in validation performance.

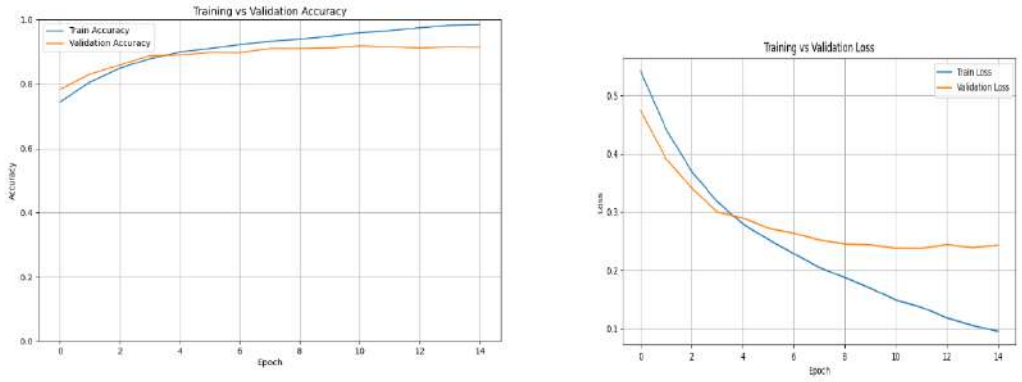


Figure 4.25 Training vs Validation (accuracy and loss) BERT

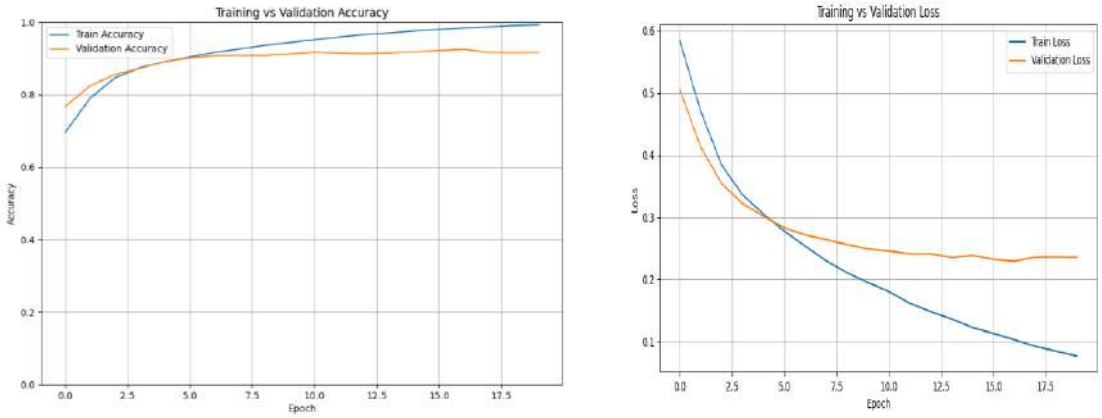


Figure 4.26 Training vs Validation (accuracy and loss) RoBERTa

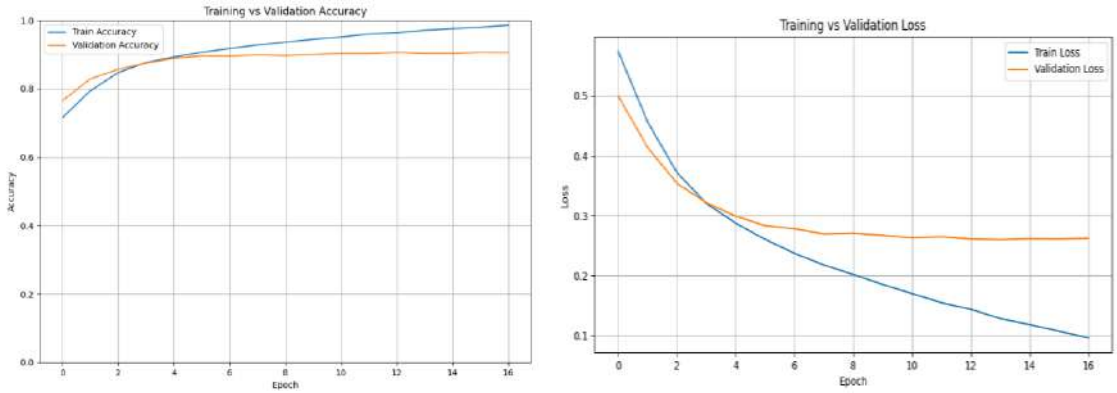


Figure 4.27 Training vs Validation (accuracy and loss) DistilBERT

4.4.2.1.1 Confusion matrixes

In addition to the accuracy and loss monitoring, the confusion matrices presented in Figures 4.28 to 4.51 provide detailed insights into model-level sentiment prediction outcomes. These visualizations reveal RoBERTa's superior balance in handling all sentiment classes, while BERT shows fewer false positives, and DistilBERT tends to overpredict the positive class. Such differences are crucial in selecting the appropriate model based on application needs and error sensitivity.

a. BERT

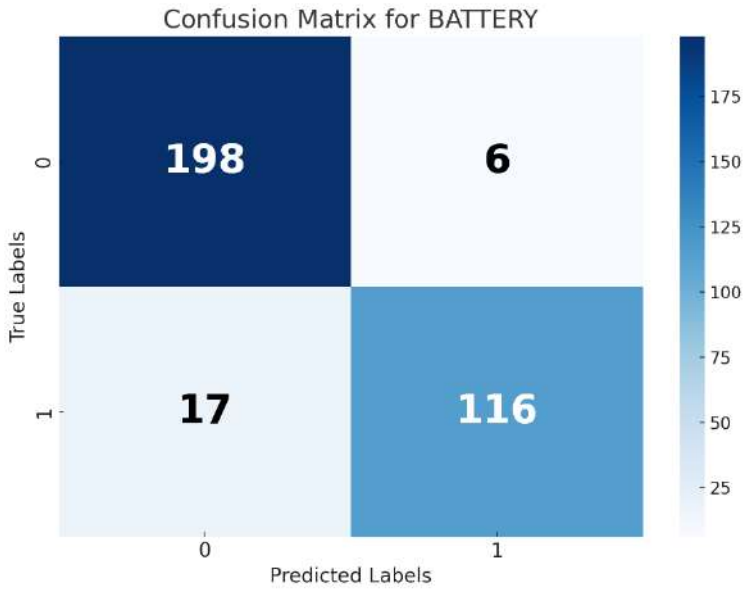


Figure 4.28 Confusion Matrixes for Battery - BERT

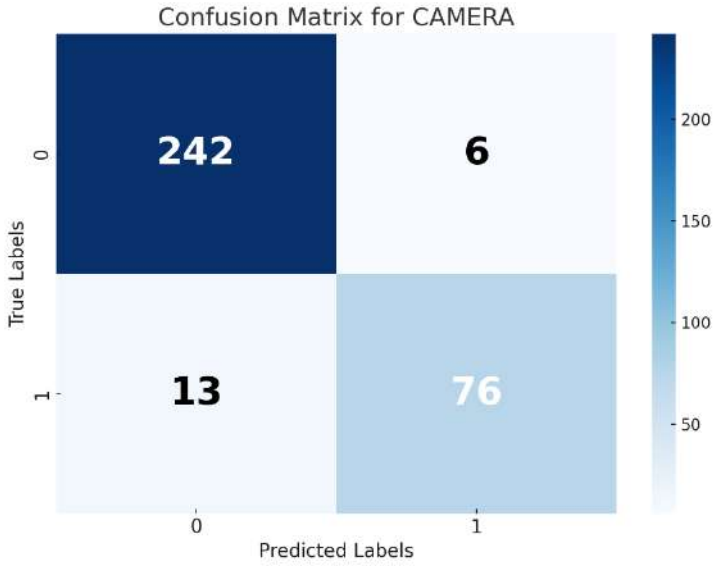


Figure 4.29 Confusion Matrixes for Camera - BERT

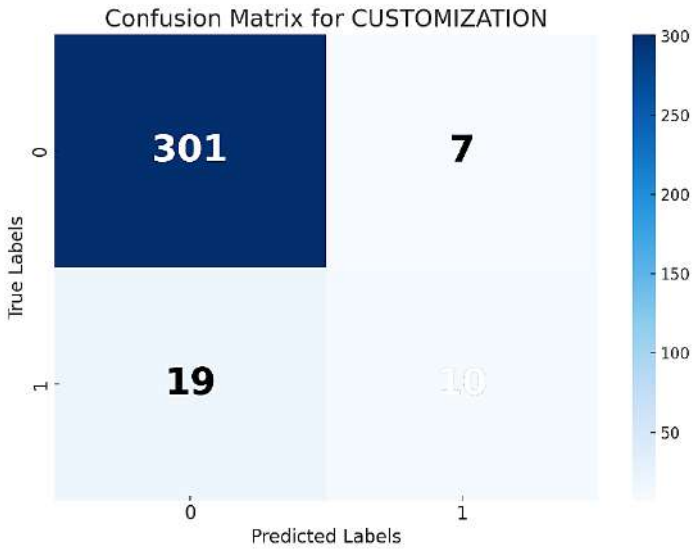


Figure 4.30 Confusion Matrixes for Customization - BERT

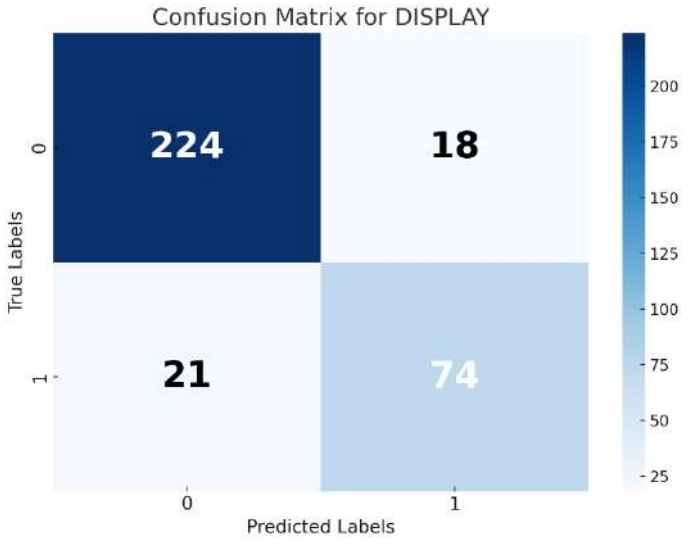


Figure 4.31 Confusion Matrixes for Display - BERT

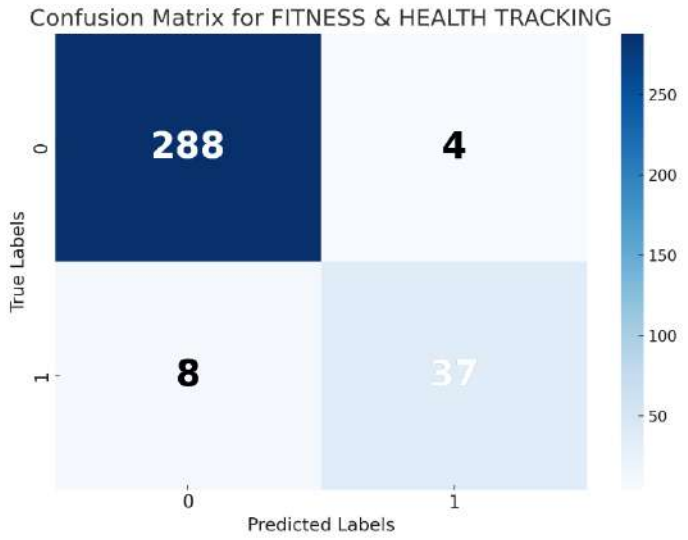


Figure 4.32 Confusion Matrixes for Fitness & Health Tracking - BERT

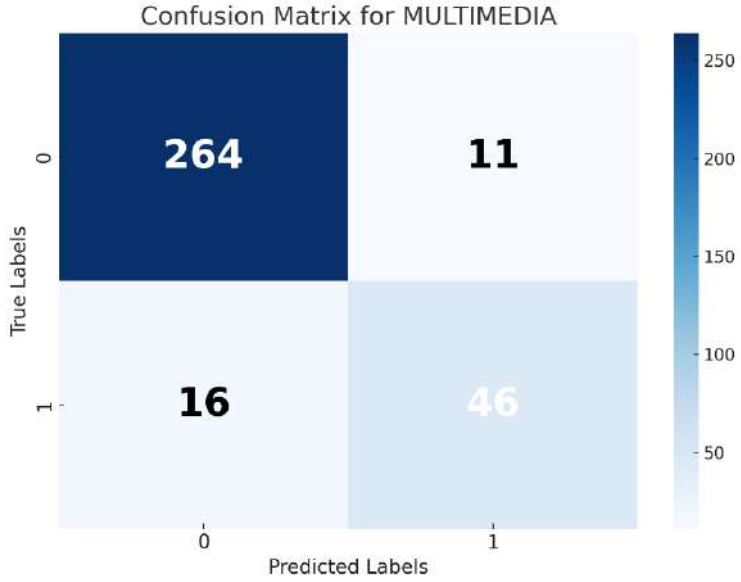


Figure 4 33 Confusion Matrixes for Multimedia - BERT

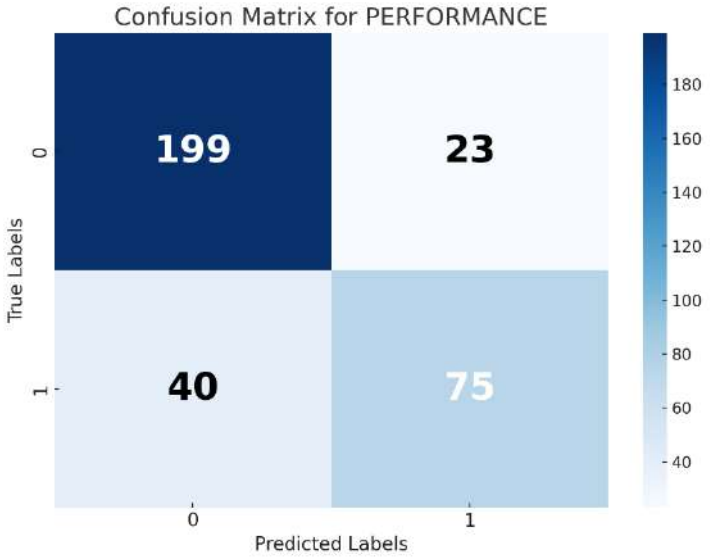


Figure 4.34 Confusion Matrixes for Performance - BERT

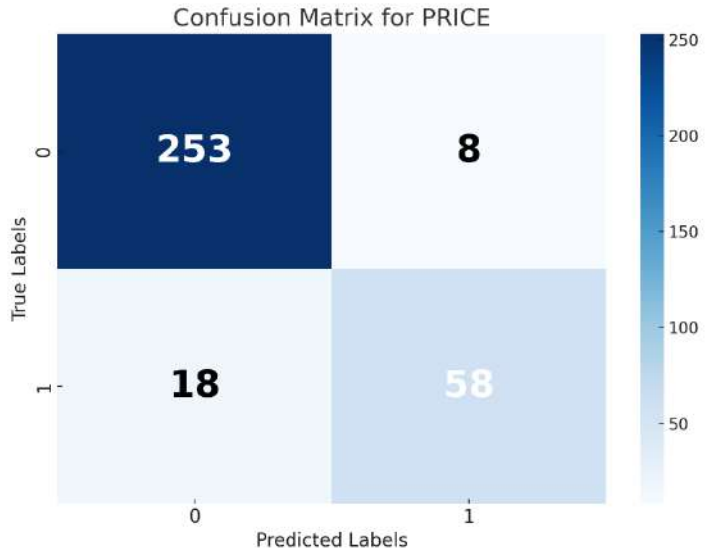


Figure 4.35 Confusion Matrixes for Price - BERT

b. RoBERTa



Figure 4.36 Confusion Matrixes for Battery - RoBERTa

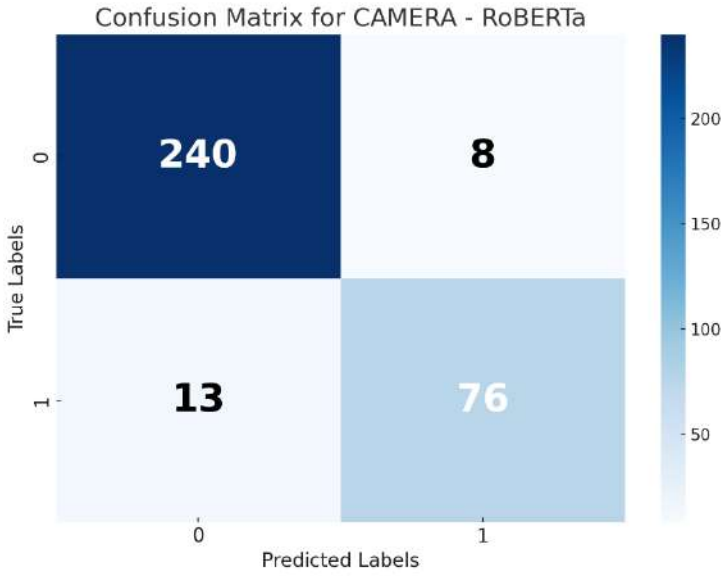


Figure 4.37 Confusion Matrixes for Camera - RoBERTa

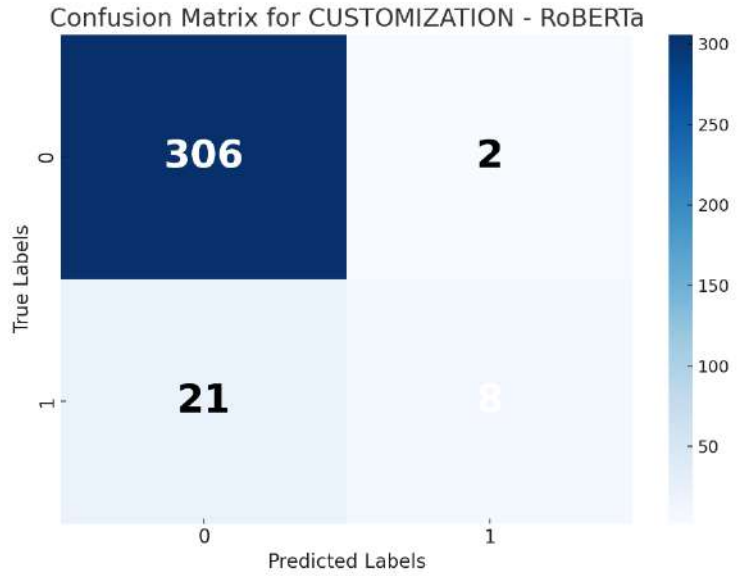


Figure 4 38 Confusion Matrixes for Customization - RoBERTa

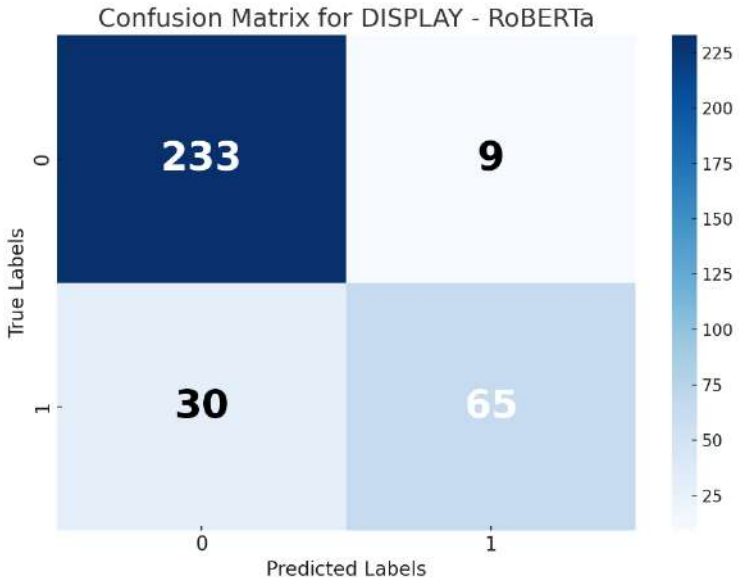


Figure 4.39 Confusion Matrixes for Display - RoBERTa

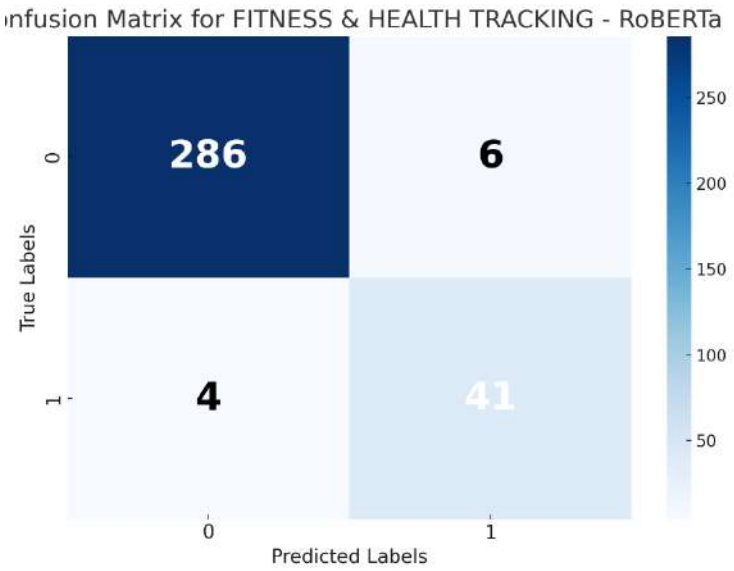


Figure 4.40 Confusion Matrixes for Fitness & Health Tracking - RoBERTa

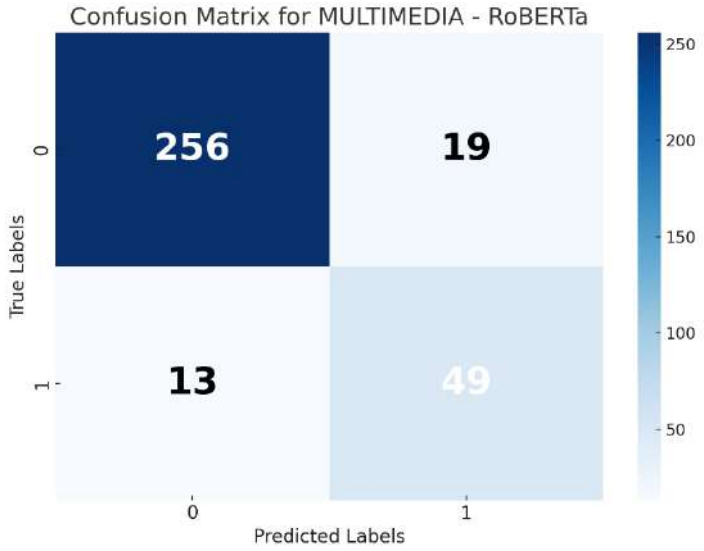


Figure 4.41Confusion Matrixes for Multimedia - RoBERTa

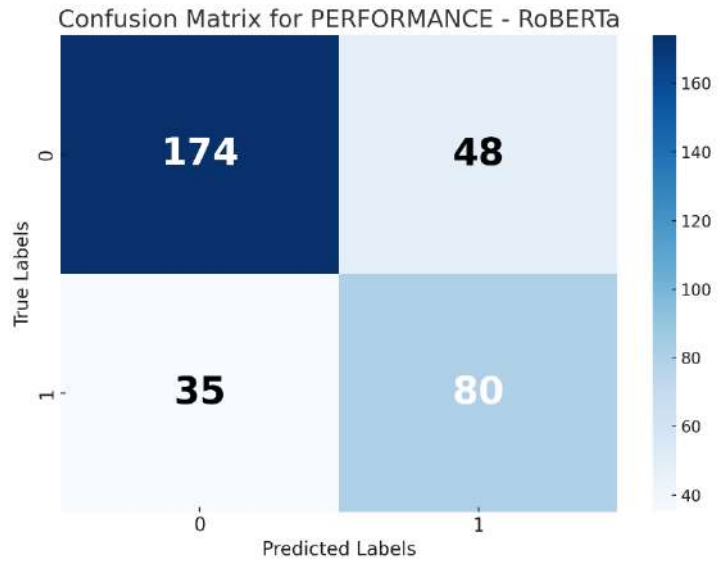


Figure 4.42 Confusion Matrixes for Performance - RoBERTa

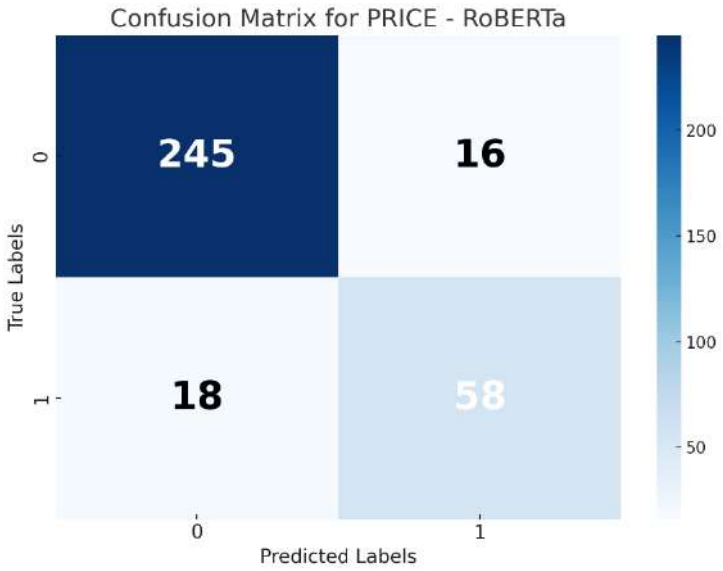


Figure 4 43 Confusion Matrixes for Price - RoBERTa

c. DistilBERT

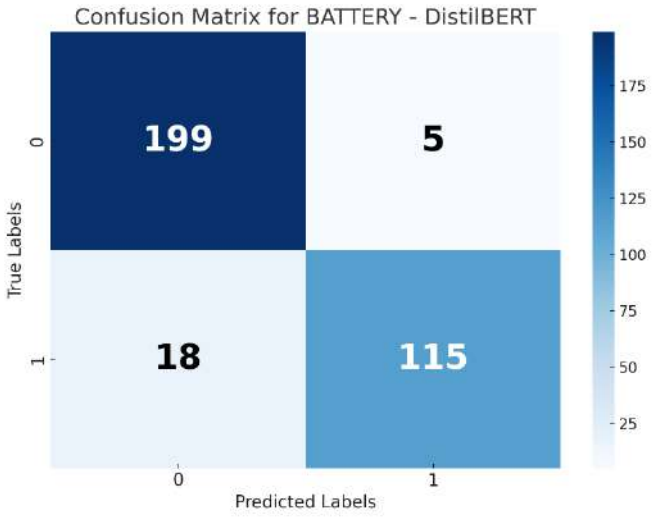


Figure 4.44 Confusion Matrixes for Battery - DistilBERT

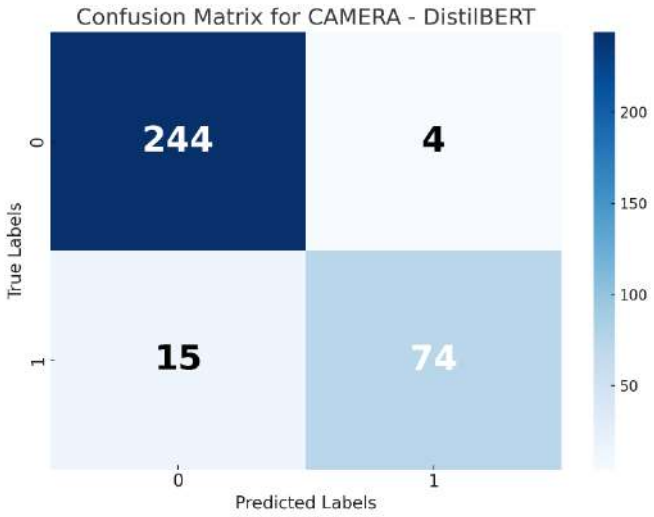


Figure 4.45 Confusion Matrixes for Camera - DistilBERT

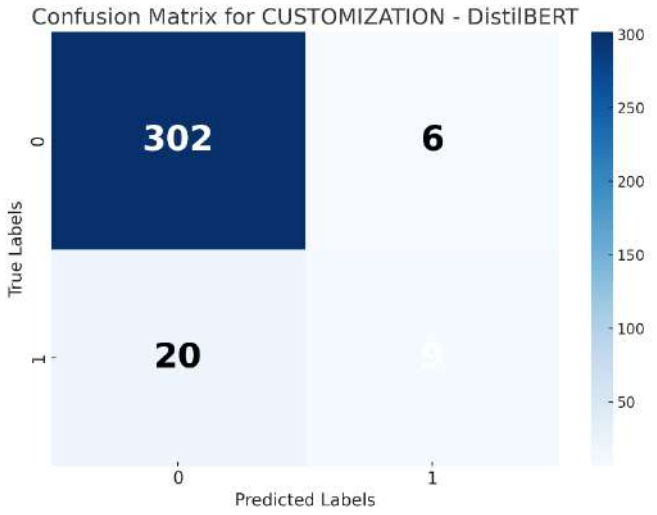


Figure 4.46 Confusion Matrixes for Customization - DistilBERT

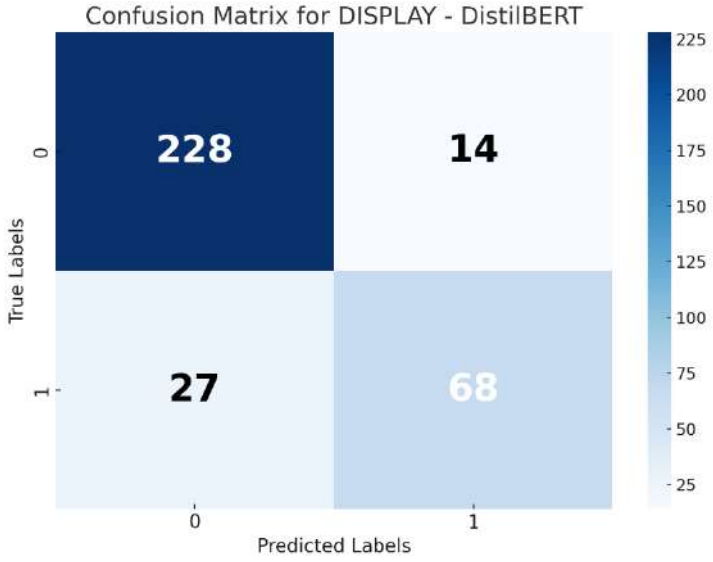


Figure 4.47 Confusion Matrixes for Display - DistilBERT

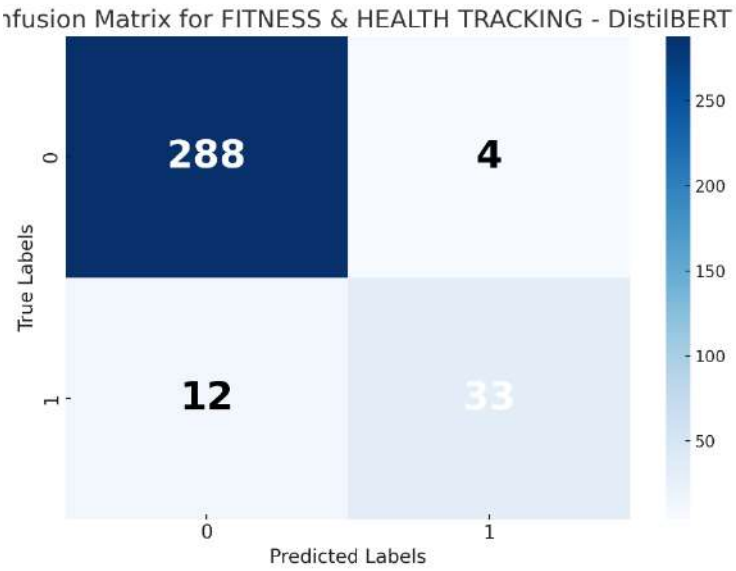


Figure 4.48 Confusion Matrixes for Fitness & Health Tracking - DistilBERT

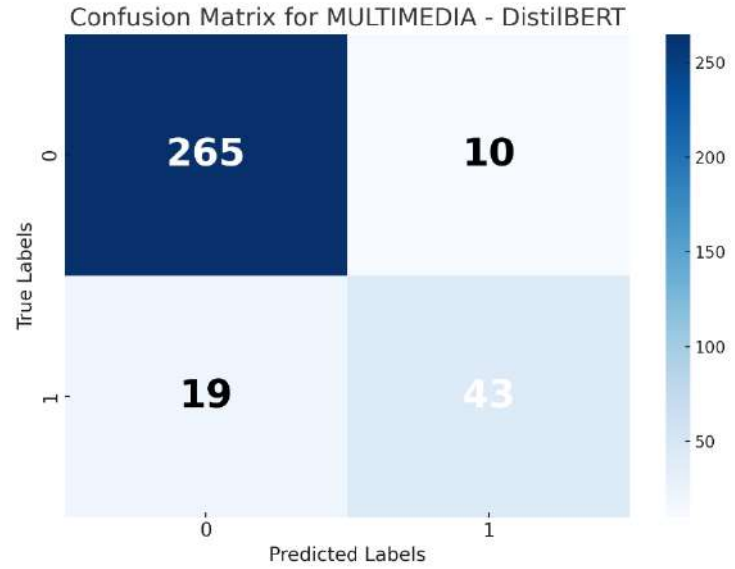


Figure 4.49 Confusion Matrixes for Multimedia - DistilBERT

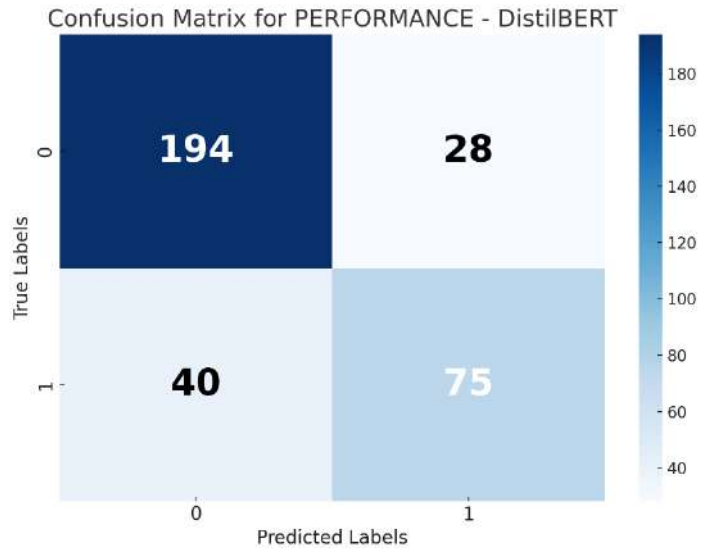


Figure 4.50 Confusion Matrixes for Performance - DistilBERT

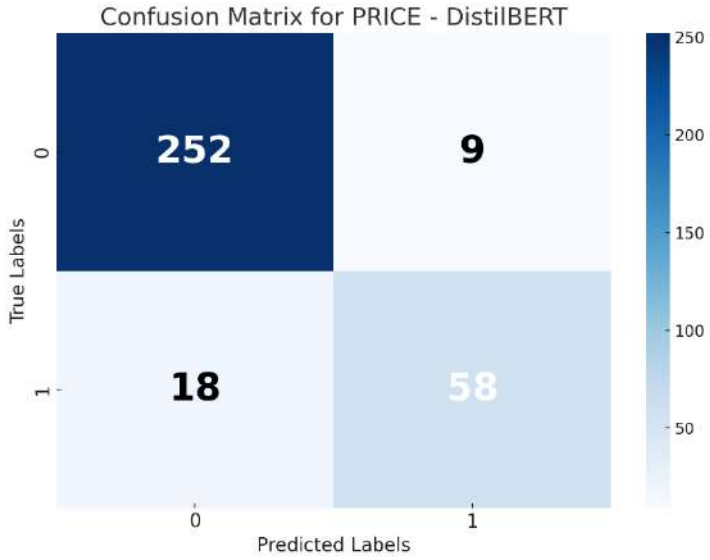


Figure 4.51 Confusion Matrixes for Price - DistilBERT

4.4.3 Sentiment Analysis Results (Pretrained Model)

In the modular setup, sentiment classification was outsourced to a pretrained transformer-based model. This approach eliminated the need for additional fine-tuning and removed overfitting issues. The pretrained model demonstrated:

- Balanced sentiment detection
- Reliable prediction across all classes
- Better generalization than prior experimental setups

This approach avoided the issues observed in DistilBERT's experimental sentiment outputs, which heavily skewed toward positive predictions (~15,735 positives out of ~17,000).

4.4.4 Trade-off Summary

Table 4.23 Trade-off Summary

Model	Strengths	Limitations	Best Use Case
BERT	High precision, stable classification	Longer training time	Academic or enterprise evaluations
RoBERTa	Strongest F1 across aspects, robust generalizer	Slight negative sentiment bias, slower to train	In-depth, multi-aspect product feedback analysis
DistilBERT	Fastest training, resource efficient	Over-skews to positive, lower recall	Lightweight, real-time applications

4.5 Conclusion

This chapter provided a comprehensive examination of the results obtained from aspect term extraction and sentiment analysis conducted on user reviews of Apple, Samsung, and Xiaomi products. By analyzing over 20,000 reviews spanning the years 2021 to 2024, several key insights emerged regarding consumer priorities, emotional responses, and model performance.

Aspect Term Extraction revealed that Battery, Performance, and Camera are consistently the most discussed features across all brands and device types. Samsung led in overall aspect frequency, especially for watches, while Xiaomi dominated camera-related discussions in smartphones. Apple reviews, though fewer, showed a balanced distribution of attention, particularly in health-tracking features. Notably, longer reviews contained richer aspect content, making them valuable for extracting nuanced feedback.

In terms of Sentiment Analysis, the study showed that consumer sentiment has grown more positive over time, especially between 2023 and 2024. BERT and RoBERTa produced reliable sentiment distributions, while DistilBERT was prone to overpredicting positive sentiment. Phones evoked more polarized opinions, while watches received a higher share of balanced and favorable reviews.

The model performance evaluation confirmed that no single model was universally superior. RoBERTa demonstrated the best F1-scores across most aspects, while BERT excelled in precision. DistilBERT was the fastest to train, making it useful for rapid deployment despite some accuracy trade-offs. The final adoption of a modular ABSA architecture, separating

aspect extraction and sentiment classification, resolved key training limitations such as task interference and overfitting, and allowed the use of pretrained sentiment classifiers for optimal results.

Overall, the results offer practical insights for both researchers and product teams. Manufacturers can focus improvements on high-impact aspects like battery life and performance while tracking sentiment evolution to inform brand positioning and customer engagement strategies.

CHAPTER V
CONCLUSION

5.1 Conclusion

This research implemented a modular Aspect-Based Sentiment Analysis (ABSA) framework to extract actionable insights from over 20,000 English-language reviews of Apple, Samsung, and Xiaomi smartphones and smartwatches collected from 2021 to 2024. By decoupling Aspect Term Extraction (ATE) and Sentiment Analysis (SA) tasks (Figure 2.4), the study leveraged transformer-based models BERT, RoBERTa, and DistilBERT to independently address each task, enhancing scalability and interpretability (Section 3.6.3).

The analysis revealed that Battery, Performance, and Camera were the most frequently discussed aspects (Table 4.1), with Battery receiving the most polarized feedback. Brand-wise, Apple maintained consistent satisfaction, Samsung improved in fitness features, and Xiaomi faced mixed reviews on performance and battery (Section 4.2.3, Table 4.11). Temporal sentiment trends (Figure 4.5) indicated a rise in positive sentiment during 2024, aligning with newer product releases and feature improvements.

Model-wise, RoBERTa achieved the highest F1 scores in ATE (Table 4.10), while BERT provided the most balanced sentiment classification (Table 4.15, Table 4.10). DistilBERT, despite lower accuracy, proved ideal for real-time or low-resource settings due to its significantly reduced training time (6.3 minutes; Table 4.9).

Critically, the study found a strong correlation between review length and aspect richness (Figure 4.6), affirming that longer reviews provide more detailed consumer feedback. These findings demonstrate that a modular pipeline not only improves task performance but also enhances generalizability and real-world deployment potential.

Research Questions Revisited:

- RQ1: Confirmed transformer models can separately and effectively perform ATE and SA (Section 4.2).
- RQ2: RoBERTa performed best for ATE, BERT for SA; DistilBERT was fastest (Table 4.8, Table 4.14,Table 4.15).
- RQ3: Battery, Performance, and Camera were most mentioned; brand comparisons highlighted key differences (Table 4.2, Table 4.11).
- RQ4: Sentiment trends showed improvement over time, especially in 2024 (Figure 4.5).

5.2 Theoretical Implications

This study contributes to the academic field of NLP and sentiment analysis in three key areas:

- Validation of Modular ABSA Architecture: By decoupling ATE and SA (Section 3.6.3, Figure 2.4), the research demonstrated enhanced interpretability and model-specific optimization, which aligns with best practices suggested by Cambria et al. (2020).
- Transformer Model Benchmarking: The empirical evaluation across BERT, RoBERTa, and DistilBERT offers comparative insights on performance vs. efficiency trade-offs (Table 4.23), reinforcing findings from Liu et al. (2019) and Sanh et al. (2019).
- Longitudinal Sentiment Tracking: Incorporating a time-series dimension (Section 4.3) advances the field of temporal ABSA, revealing shifts in sentiment tied to product updates, brand strategies, or consumer expectations supporting calls from Jiang & Cambria (2023) for dynamic sentiment modeling.

5.3 Practical Recommendations

Based on the findings, the following recommendations are proposed for various stakeholder groups:

a. Product Design and Development Teams

- Prioritize battery longevity and performance optimization, especially in smartphones where dissatisfaction was most evident.
- Enhance camera capabilities in Xiaomi phones and expand health tracking features in Apple wearables to align with user preferences.

b. Marketing and Branding Teams

- Highlight high-performing features such as Xiaomi's camera and Apple's health integration in promotional campaigns.
- Address areas of negative sentimentsuch as Samsung's battery-related concernsthrough transparent communication and proactive updates.
- Tailor advertising strategies based on product category (phones vs. wearables), as engagement patterns differ across device types.

c. AI and NLP Deployment Teams

- Use DistilBERT for real-time, low-latency deployments where computational resources are limited.
- Deploy RoBERTa in feedback-intensive systems to maximize extraction accuracy and sentiment precision.

5.4 Limitations of the Study

Despite its contributions, the study has several limitations:

1. **Annotation Scope:** The aspect term extraction model was trained on a limited manually annotated subset of 2,250 reviews, which may constrain its ability to generalize across more diverse expressions of product features.
2. **Sentiment Granularity:** Sentiment classification was conducted at the review level, which can obscure sentiment variance across different aspects within a single review.
3. **Domain-Specific Training:** The pretrained models were not fine-tuned on electronics-specific datasets, potentially reducing classification performance for technical product descriptions or jargon.
4. **Platform Bias:** Data was sourced from four major e-commerce platforms, which may not represent global user behavior due to platform-specific demographic and regional factors.

5.5 Suggestions for Future Research

To advance ABSA in sentiment analytics, future work should explore the following directions:

- Aspect-Level Sentiment Annotation: Incorporate sentiment labels at the aspect level to improve resolution and prevent oversimplification.
- Multilingual ABSA Extension: Expand the dataset to include multilingual corpora to enable cross-cultural consumer sentiment insights.
- Prompt-Based and Few-Shot Learning: Investigate the use of prompt-based or few-shot learning models (e.g., GPT-family) to reduce reliance on manual annotation.
- Domain-Specific Fine-Tuning: Train sentiment classifiers on electronics-specific corpora to improve accuracy on brand and product terminology.
- Real-Time Dashboards: Deploy lightweight ABSA pipelines into interactive dashboards to support real-time feedback monitoring.
- Event-Sentiment Mapping: Correlate shifts in sentiment with external events (e.g., product launches, software updates, recalls) to identify causality in consumer opinion.

REFERENCES

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Hu, M., & Liu, B. (2022). Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168–177.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Zhang, Z., Wang, Y., & Liu, Q. (2021). Aspect-based sentiment analysis: A survey. IEEE Transactions on Affective Computing, 13(1), 1–17.

Antoun, W., Baly, F., & Hajj, H. (2020). AraBERT: Transformer-based Model for Arabic Language Understanding. Proceedings of the LREC 2020 Workshop on Resources and Tools for Arabic Language Processing, 9–15.

Cambria, E., Poria, S., Bajpai, R., & Schuller, B. (2020). SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. Proceedings of the AAAI Conference on Artificial Intelligence, 34(01), 5076–5083.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT: a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Xu, H., Liu, B., Shu, L., & Yu, P. S. (2020). BERT-based models for sentiment analysis: A survey. IEEE Access, 8, 133577–133591.

Zhang, L., Wang, S., & Liu, B. (2021). Deep learning for aspect-based sentiment analysis: A survey. arXiv preprint arXiv:2103.15600.

ResearchGate. (n.d.). BERT word embedding model architecture. Retrieved from https://www.researchgate.net/figure/BERT-word-embedding-model-architecture_fig3_377114517

ResearchGate. (n.d.). The RoBERTa model architecture. Retrieved from https://www.researchgate.net/figure/The-RoBERTa-model-architecture_fig2_352642553

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT 2019, 4171–4186. https://doi.org/10.48550/arXiv.1810.04805

Hu, M., & Liu, B. (2022). Mining and summarizing customer reviews. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 168–177.

Jiang, Y., & Cambria, E. (2023). Longitudinal aspect-based sentiment analysis: Methods and applications. Information Processing & Management, 60(1), 103193. https://doi.org/10.1016/j.ipm.2022.103193

Lim, S., Lee, S., & Lee, D. (2020). Analyzing product review polarities for strategic brand positioning. Journal of Business Research, 110, 373–381.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

Zhang, Z., Wang, Y., & Liu, Q. (2021). Aspect-based sentiment analysis: A survey. IEEE Transactions on Affective Computing, 13(1), 1–17.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv preprint arXiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692

Zhang, Y., Zhang, Y., & Cambria, E. (2021). A deep learning framework for cross-domain and cross-lingual ABSA. Information Fusion, 70, 92–103. https://doi.org/10.1016/j.inffus.2021.01.004

Lim, C., Lee, M., & Lee, J. (2020). Consumer insights through opinion mining on smartphones: Sentiment analysis and review clustering. Telematics and Informatics, 51, 101404. https://doi.org/10.1016/j.tele.2020.101404

Fang, X., Xu, L., Long, C., & Tang, Y. (2022). Noise-aware Transformer for Social Media Sentiment Analysis. Knowledge-Based Systems, 246, 108680. https://doi.org/10.1016/j.knosys.2022.108680

Little, R. J. A., & Rubin, D. B. (2019). Statistical Analysis with Missing Data (3rd ed.). Wiley.

Rustam, F., Ashraf, I., Mehmood, A., Choi, G. S., & Ahmad, M. (2021). Text Classification Using Deep Learning Models: A Review. Applied Sciences, 11(16), 7625. https://doi.org/10.3390/app11167625

Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., & Choi, G. S. (2021). Tweets classification on the basis of sentiments for US airline companies. Entropy, 23(7), 870.

Sun, C., Huang, L., & Qiu, X. (2019). Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. Proceedings of NAACL-HLT 2019, 380–385. https://doi.org/10.18653/v1/N19-1035

Jurafsky, D., & Martin, J. H. (2019). Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (3rd ed.). Pearson.

Rogers, A., Kovaleva, O., & Rumshisky, A. (2020). A primer in BERTology: What we know about how BERT works. Transactions of the Association for Computational Linguistics, 8, 842–866.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., … & Polosukhin, I. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems, 30, 5998–6008. https://doi.org/10.48550/arXiv.1706.03762

Wolf, T., et al. (2020). HuggingFace's Transformers: State-of-the-art Natural Language Processing. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP 2019, 3982–3992. https://doi.org/10.48550/arXiv.1908.10084

Aggarwal, C. C., & Zhai, C. (2012). Mining Text Data. Springer.

Feldman, R. (2013). Techniques and Applications for Sentiment Analysis. Communications of the ACM, 56(4), 82–89. https://doi.org/10.1145/2436256.2436274

Landis, J. R., & Koch, G. G. (1977). The measurement of observer agreement for categorical data. Biometrics, 33(1), 159–174. https://doi.org/10.2307/2529310

Villena, R. J., Martínez-Cámara, E., Martín-Valdivia, M. T., & Ureña-López, L. A. (n.d.). Corpus design and evaluation for aspect-level sentiment analysis in Spanish customer reviews. [Online Manuscript]

Pontius, R. G., & Millones, M. (2011). Death to Kappa: Birth of quantity disagreement and allocation disagreement for accuracy assessment. International Journal of Remote Sensing, 32(15), 4407–4429. https://doi.org/10.1080/01431161.2011.552923

Mosbach, M., Andriushchenko, M., & Klakow, D. (2020). On the Stability of Fine-tuning BERT: Misconceptions, Explanations, and Strong Baselines. https://arxiv.org/abs/2006.04884

Howard, J., & Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. https://arxiv.org/abs/1801.06146

Azhar, H., & Khodra, M. L. (2021). Fine-tuning BERT for Aspect-Based Sentiment Analysis. https://ieeexplore.ieee.org/document/9409947

Go, A., Bhayani, R., & Huang, L. (2009). Twitter Sentiment Classification using Distant Supervision. Stanford NLP Class Project.

Kouloumpis, E., Wilson, T., & Moore, J. (2021). Twitter Sentiment Analysis: The Good the Bad and the OMG!. Proceedings of ICWSM.

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proceedings of HLT/EMNLP.

Hidayatullah, A. F. (2024). Parameter Efficient Fine-tuning using Low-Rank Adaptation for Emotion Classification in Indonesian Texts. IEEE https://ieeexplore.ieee.org/document/10810037

Hicks, S., Martin, D., & Johnson, R. (2022). A Guide to Evaluation Metrics for Classification Problems. Journal of Applied AI, 10(2), 45–58.

Alyami, H., Alqhtani, F., & Alshahrani, S. (2022). Fine-Tuning Strategies for NLP Tasks in Arabic. Applied Computational Intelligence, 8(4), 311–326.

Tinkerd. (n.d.). BERT Embeddings. Retrieved from https://tinkerd.net/blog/machine-learning/bert-embeddings/

Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2016). Semeval-2016 Task 5: Aspect Based Sentiment Analysis. SemEval-2016.

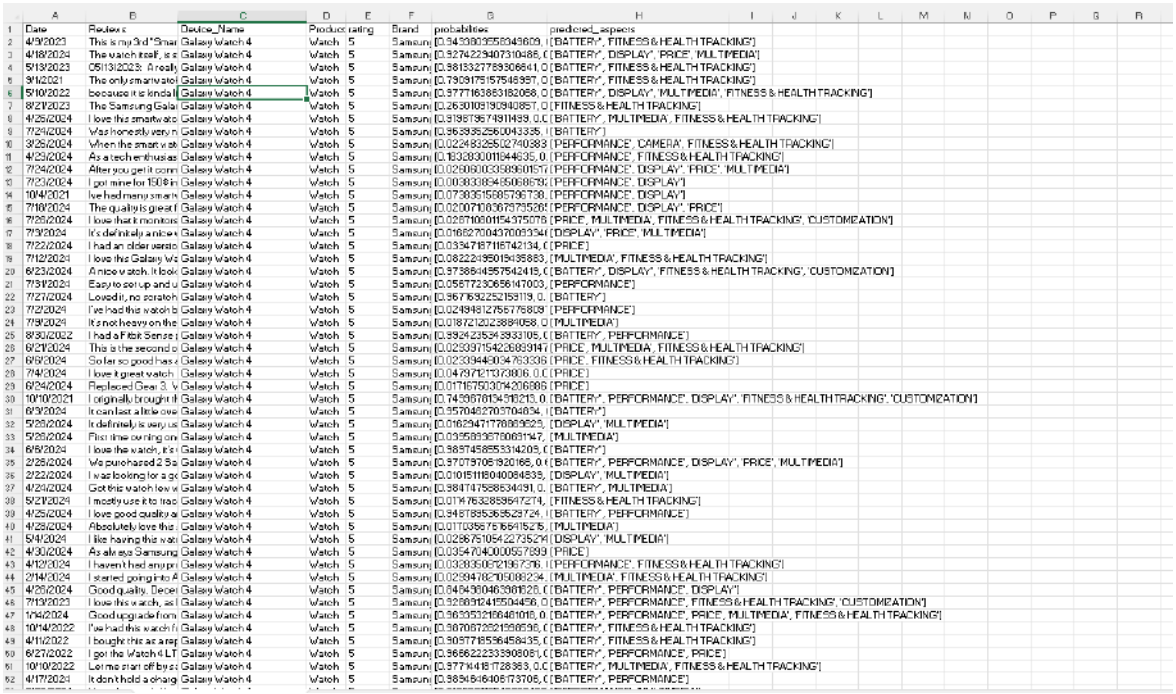
Zhang, M., Wang, Y., & Liu, B. (2018). Deep Learning for Aspect-Based Sentiment Analysis: A Survey. IEEE Transactions on Affective Computing.

Wang, W., Lu, Y., & Zhai, C. (2020). Target-Aware Deep Modeling of Review Helpfulness: A Unified Perspective from Review Content and Aspect Attention. ACM TOIS.

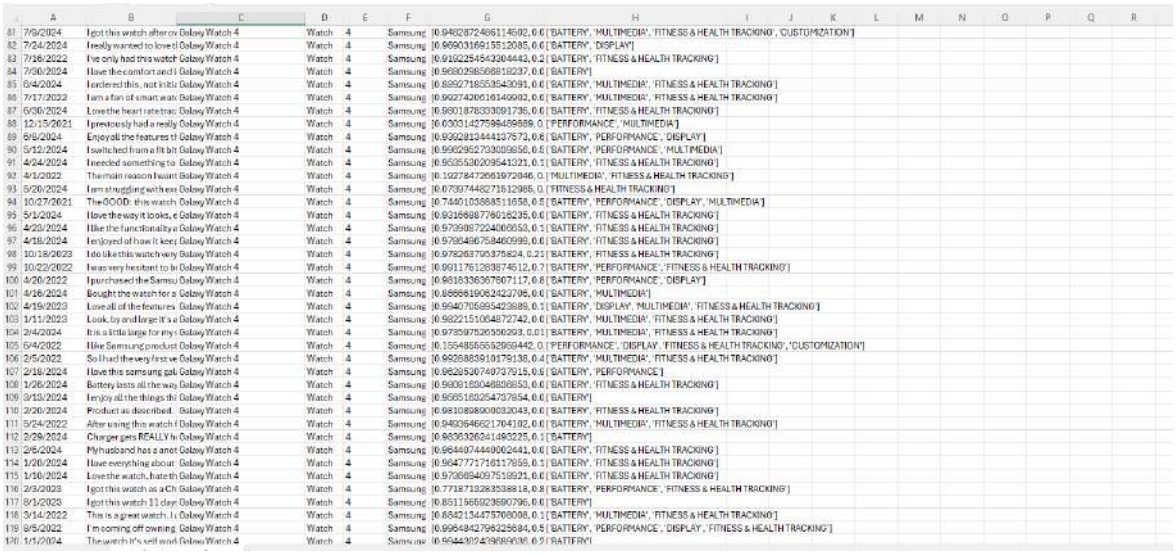
Cambria, E., Poria, S., Gelbukh, A., & Thelwall, M. (2022). Sentiment Analysis Is a Big Suitcase. IEEE Intelligent Systems.

APPENDIX

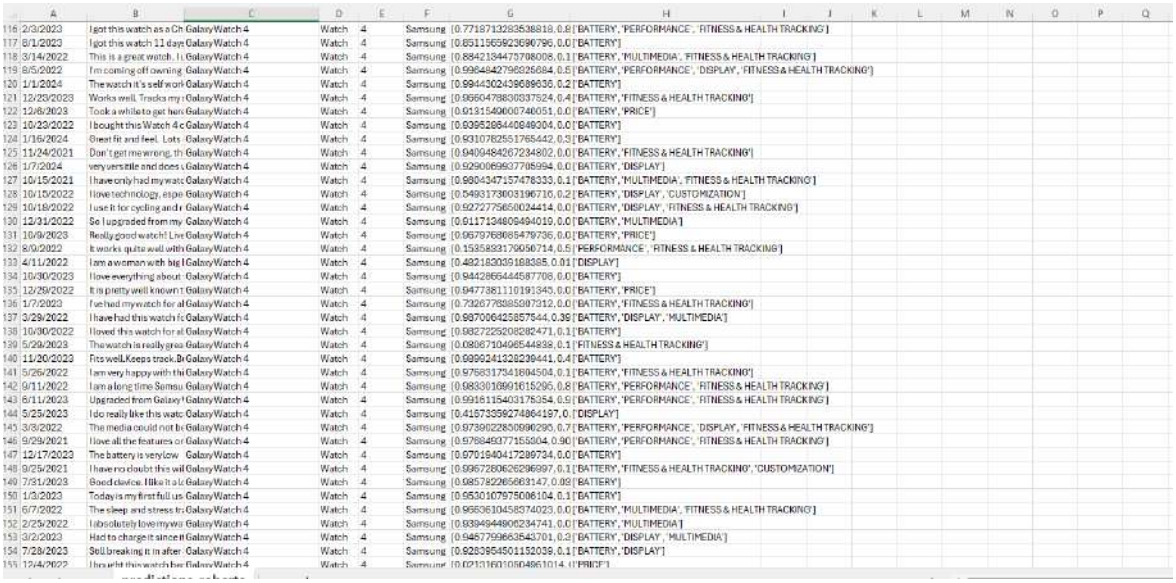
Appendix 1. Dataset



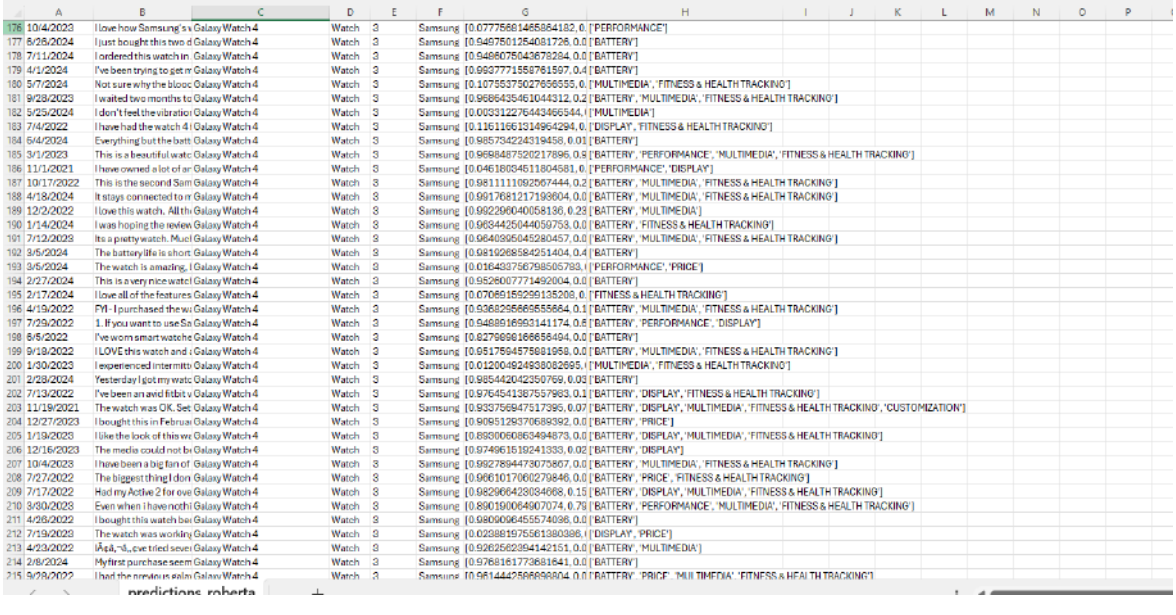
Date	Reviews	Device_Name	Product	rating	Brand	probabilities	predicted_aspects
4/9/2023	This is my 3rd "Smar	Galaxy Watch 4	Watch	5	Samsung	[0.943380355849609,	['BATTERY', 'FITNESS & HEALTH TRACKING']
4/18/2024	The watch itself, is a	Galaxy Watch 4	Watch	5	Samsung	[0.927422940710486,	['BATTERY', 'DISPLAY', 'PRICE', 'MULTIMEDIA']
5/13/2023	05/13/2023: A really	Galaxy Watch 4	Watch	5	Samsung	[0.981332778930664,	['BATTERY', 'FITNESS & HEALTH TRACKING']
9/1/2021	The only smartwatcl	Galaxy Watch 4	Watch	5	Samsung	[0.790917515754699,	['BATTERY', 'FITNESS & HEALTH TRACKING']
5/10/2022	because it is kinda li	Galaxy Watch 4	Watch	5	Samsung	[0.977716386318206,	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
8/21/2023	The Samsung Galax	Galaxy Watch 4	Watch	5	Samsung	[0.263010919094085,	['FITNESS & HEALTH TRACKING']
4/25/2024	I love the smartwatc	Galaxy Watch 4	Watch	5	Samsung	[0.919879674911499,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
7/29/2024	Was honestly very n	Galaxy Watch 4	Watch	5	Samsung	[0.963935565948486,	['BATTERY']
3/25/2024	When the smart wat	Galaxy Watch 4	Watch	5	Samsung	[0.022483226013184,	['PERFORMANCE', 'CAMERA', 'FITNESS & HEALTH TRACKING']
4/29/2024	As a tech enthusias	Galaxy Watch 4	Watch	5	Samsung	[0.183930039405823,	['PERFORMANCE', 'FITNESS & HEALTH TRACKING']
7/24/2024	After you get it conn	Galaxy Watch 4	Watch	5	Samsung	[0.029060030174255,	['PERFORMANCE', 'DISPLAY', 'PRICE', 'MULTIMEDIA']
7/23/2024	I got mine for 150$ in	Galaxy Watch 4	Watch	5	Samsung	[0.000838851928711,	['PERFORMANCE', 'DISPLAY']
10/4/2021	Ive had many smart	Galaxy Watch 4	Watch	5	Samsung	[0.073035621643066,	['PERFORMANCE', 'DISPLAY']
7/10/2024	The quality is great f	Galaxy Watch 4	Watch	5	Samsung	[0.020071005821228,	['PERFORMANCE', 'DISPLAY', 'PRICE']
7/26/2024	I love that it monitors	Galaxy Watch 4	Watch	5	Samsung	[0.028710031509399,	['PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
7/3/2024	It's definitely a nice e	Galaxy Watch 4	Watch	5	Samsung	[0.016827011108398,	['DISPLAY', 'PRICE', 'MULTIMEDIA']
7/22/2024	I had an older versio	Galaxy Watch 4	Watch	5	Samsung	[0.035478711318970,	['PRICE']
7/12/2024	I love the Galaxy Wa	Galaxy Watch 4	Watch	5	Samsung	[0.082234954833984,	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
6/23/2024	A nice watch. It look	Galaxy Watch 4	Watch	5	Samsung	[0.973864459991455,	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
7/31/2024	Easy to set up and u	Galaxy Watch 4	Watch	5	Samsung	[0.056723266983032,	['PERFORMANCE']
7/27/2024	Loves it, no scratch	Galaxy Watch 4	Watch	5	Samsung	[0.967692232513428,	['BATTERY']
7/27/2024	I've had this watch b	Galaxy Watch 4	Watch	5	Samsung	[0.024981275177383,	['PERFORMANCE']
7/9/2024	It's not heavy on the	Galaxy Watch 4	Watch	5	Samsung	[0.018721202388382,	['MULTIMEDIA']
8/30/2022	I had a Fitbit Sense p	Galaxy Watch 4	Watch	5	Samsung	[0.932423353195190,	['BATTERY', 'PERFORMANCE']
6/21/2024	This is the second o	Galaxy Watch 4	Watch	5	Samsung	[0.023397154235840,	['PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
6/6/2024	So far so good has a	Galaxy Watch 4	Watch	5	Samsung	[0.023394489288330,	['PRICE', 'FITNESS & HEALTH TRACKING']
7/4/2024	I love it great watch	Galaxy Watch 4	Watch	5	Samsung	[0.047971725463867,	['PRICE']
6/24/2024	Replaced Gear 3. V	Galaxy Watch 4	Watch	5	Samsung	[0.011767578125000,	['PRICE']
10/10/2021	I originally brought t	Galaxy Watch 4	Watch	5	Samsung	[0.743957686424255,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
6/3/2024	It can last a little ove	Galaxy Watch 4	Watch	5	Samsung	[0.557045221328735,	['BATTERY']
5/28/2024	It definitely is easy to	Galaxy Watch 4	Watch	5	Samsung	[0.016294717788696,	['DISPLAY', 'MULTIMEDIA']
5/28/2024	First time owning one	Galaxy Watch 4	Watch	5	Samsung	[0.039589357376099,	['MULTIMEDIA']
6/6/2024	I love the watch, it's r	Galaxy Watch 4	Watch	5	Samsung	[0.988745855331421,	['BATTERY']
2/28/2024	We purchased 2 Sa	Galaxy Watch 4	Watch	5	Samsung	[0.970797061920166,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE', 'MULTIMEDIA']
2/22/2024	I was looking for a go	Galaxy Watch 4	Watch	5	Samsung	[0.010511183738708,	['DISPLAY', 'MULTIMEDIA']
4/24/2024	Got this watch low w	Galaxy Watch 4	Watch	5	Samsung	[0.984747552871704,	['BATTERY', 'MULTIMEDIA']
5/21/2024	I mostly use it to trac	Galaxy Watch 4	Watch	5	Samsung	[0.017763853073120,	['FITNESS & HEALTH TRACKING']
4/25/2024	I love good quality a	Galaxy Watch 4	Watch	5	Samsung	[0.948788356590271,	['BATTERY', 'PERFORMANCE']
4/28/2024	Absolutely love this	Galaxy Watch 4	Watch	5	Samsung	[0.011703562736511,	['MULTIMEDIA']
5/4/2024	I like having the wat	Galaxy Watch 4	Watch	5	Samsung	[0.028675103569031,	['DISPLAY', 'MULTIMEDIA']
4/30/2024	As always Samsung	Galaxy Watch 4	Watch	5	Samsung	[0.035470390319824,	['PRICE']
4/12/2024	I haven't had any pro	Galaxy Watch 4	Watch	5	Samsung	[0.032835006713867,	['PERFORMANCE', 'FITNESS & HEALTH TRACKING']
2/14/2024	I started going into A	Galaxy Watch 4	Watch	5	Samsung	[0.029947280883789,	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
4/26/2024	Good quality. Decen	Galaxy Watch 4	Watch	5	Samsung	[0.848498046398163,	['BATTERY', 'PERFORMANCE', 'DISPLAY']
7/13/2023	I love this watch, as I	Galaxy Watch 4	Watch	5	Samsung	[0.928891241550445,	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
1/14/2024	Good upgrade from Fit	Galaxy Watch 4	Watch	5	Samsung	[0.963532161712646,	['BATTERY', 'PERFORMANCE', 'PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
10/14/2022	I've had this watch for	Galaxy Watch 4	Watch	5	Samsung	[0.987067222595215,	['BATTERY', 'FITNESS & HEALTH TRACKING']
4/11/2022	I bought this as a repla	Galaxy Watch 4	Watch	5	Samsung	[0.909771156311035,	['BATTERY', 'FITNESS & HEALTH TRACKING']
6/27/2022	I got the Watch 4 LTE w	Galaxy Watch 4	Watch	5	Samsung	[0.966622233390808,	['BATTERY', 'PERFORMANCE', 'PRICE']
10/10/2022	Let me start off by say	Galaxy Watch 4	Watch	5	Samsung	[0.977144193840027,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
4/17/2024	It don't hold a charge	Galaxy Watch 4	Watch	5	Samsung	[0.969464063644409,	['BATTERY', 'PERFORMANCE']
3/22/2024	I love the watch I love	Galaxy Watch 4	Watch	5	Samsung	[0.019660115242004,	['PERFORMANCE', 'MULTIMEDIA']
5/31/2022	First, I bought the Blue	Galaxy Watch 4	Watch	5	Samsung	[0.976838469505310,	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
4/18/2024	Watch works well but t	Galaxy Watch 4	Watch	5	Samsung	[0.958568572998047,	['BATTERY']
4/24/2024	Comfortable and easy	Galaxy Watch 4	Watch	5	Samsung	[0.038478469848633,	['FITNESS & HEALTH TRACKING']
5/1/2022	So I've only had one ot	Galaxy Watch 4	Watch	5	Samsung	[0.966850376129150,	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
4/17/2024	Battery only lasts for 1	Galaxy Watch 4	Watch	5	Samsung	[0.925905227661133,	['BATTERY', 'PERFORMANCE']
11/8/2021	I love this watch! I have	Galaxy Watch 4	Watch	5	Samsung	[0.940126121044159,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'MULTIMEDIA']
7/17/2022	This is myun-bowing re	Galaxy Watch 4	Watch	5	Samsung	[0.966119051361084,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE', 'FITNESS & HEALTH TRACKING']
7/20/2022	Got an amazing deal at	Galaxy Watch 4	Watch	5	Samsung	[0.968717765808105,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
12/20/2023	The watch is an old ver	Galaxy Watch 4	Watch	5	Samsung	[0.030360794067383,	['PERFORMANCE', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
7/14/2023	My wife loves this watc	Galaxy Watch 4	Watch	5	Samsung	[0.014091253280640,	['PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
6/20/2022	I upgraded to this watc	Galaxy Watch 4	Watch	5	Samsung	[0.968406009674072,	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
2/5/2022	What I like is this watcl	Galaxy Watch 4	Watch	5	Samsung	[0.909968376159668,	['BATTERY', 'PERFORMANCE', 'PRICE', 'MULTIMEDIA', 'CUSTOMIZATION']
1/10/2022	This thing did more the	Galaxy Watch 4	Watch	5	Samsung	[0.968332290649414,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
4/18/2024	With the battery lasted	Galaxy Watch 4	Watch	5	Samsung	[0.952284574508667,	['BATTERY']
1/10/2024	Great product nice look	Galaxy Watch 4	Watch	5	Samsung	[0.882218933105469,	['BATTERY']
11/12/2023	I was sick and tired of t	Galaxy Watch 4	Watch	5	Samsung	[0.930091917514801,	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA']
3/21/2024	Moved up from a fitbit t	Galaxy Watch 4	Watch	5	Samsung	[0.949035656452179,	['BATTERY']
11/15/2023	The media could not be	Galaxy Watch 4	Watch	5	Samsung	[0.968462491226196,	['BATTERY', 'PRICE']
1/5/2023	This has been, by far, m	Galaxy Watch 4	Watch	5	Samsung	[0.904611945152283,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
1/18/2024	I bought for my adult d	Galaxy Watch 4	Watch	5	Samsung	[0.016170144081116,	['FITNESS & HEALTH TRACKING']
10/24/2022	This thing is awesome,	Galaxy Watch 4	Watch	5	Samsung	[0.003924345016479,	['PERFORMANCE', 'DISPLAY', 'PRICE', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
12/3/2023	I've gotten one for mys	Galaxy Watch 4	Watch	5	Samsung	[0.017938677215576,	['PERFORMANCE', 'DISPLAY']
3/13/2024	connecting to motorol	Galaxy Watch 4	Watch	5	Samsung	[0.968468570709229,	['BATTERY']
11/17/2022	This is my first smart w	Galaxy Watch 4	Watch	5	Samsung	[0.871711254119873,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
12/16/2021	I have diagnosed with	Galaxy Watch 4	Watch	5	Samsung	[0.977293777465820,	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
2/13/2022	I had a Galaxy Watch A	Galaxy Watch 4	Watch	4	Samsung	[0.376816660165787,	['FITNESS & HEALTH TRACKING']
7/21/2024	I've been using the SAI	Galaxy Watch 4	Watch	4	Samsung	[0.983790159225464,	['PERFORMANCE', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']



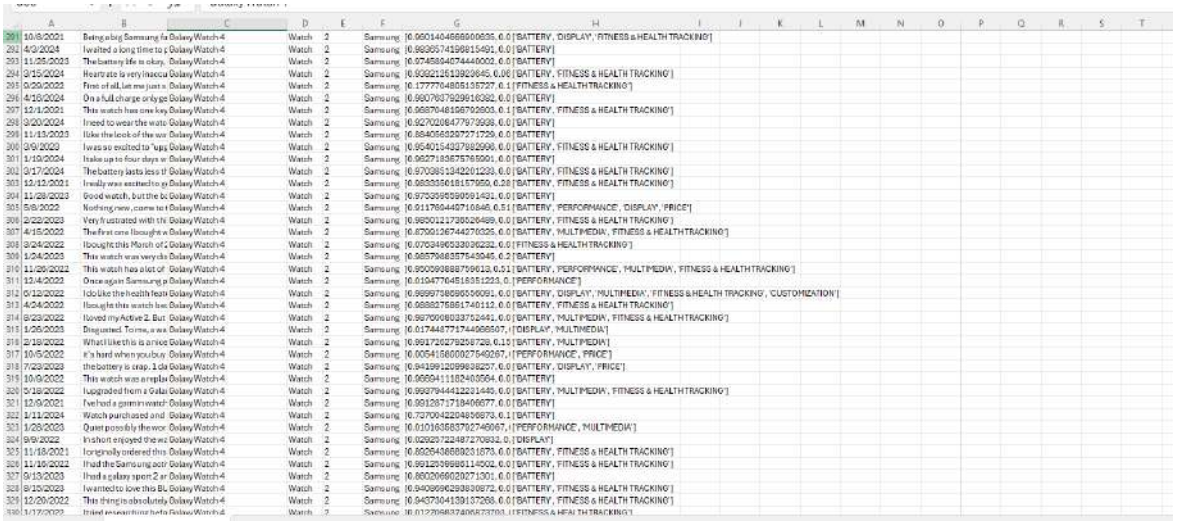
	A	B	C	D	E	F	G	H
81	7/9/2024	I got this watch after ov	Galaxy Watch 4	Watch	4	Samsung	0.9482872486114502, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
82	7/24/2024	I really wanted to love t	Galaxy Watch 4	Watch	4	Samsung	0.9690316915512085, 0.0	['BATTERY', 'DISPLAY']
83	7/16/2022	I've only had this watch	Galaxy Watch 4	Watch	4	Samsung	0.9192254543304443, 0.2	['BATTERY', 'FITNESS & HEALTH TRACKING']
84	7/30/2024	I love the comfort and f	Galaxy Watch 4	Watch	4	Samsung	0.9680298566818237, 0.0	['BATTERY']
85	6/4/2024	I ordered this, not initia	Galaxy Watch 4	Watch	4	Samsung	0.8892718553543091, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
86	7/17/2022	I am a fan of smart wate	Galaxy Watch 4	Watch	4	Samsung	0.9927420616149902, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
87	6/30/2024	Love the heart rate trac	Galaxy Watch 4	Watch	4	Samsung	0.9601878333091736, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
88	12/15/2021	I previously had a really	Galaxy Watch 4	Watch	4	Samsung	0.6303142738437653, 0.0	['PERFORMANCE', 'MULTIMEDIA']
89	6/6/2024	Enjoy all the features th	Galaxy Watch 4	Watch	4	Samsung	0.9992813444137573, 0.0	['BATTERY', 'PERFORMANCE', 'DISPLAY']
90	5/12/2024	I switched from a fit bit	Galaxy Watch 4	Watch	4	Samsung	0.9962952733039856, 0.0	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA']
91	4/24/2024	I needed something to	Galaxy Watch 4	Watch	4	Samsung	0.9635530209541321, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
92	4/1/2022	The main reason I want	Galaxy Watch 4	Watch	4	Samsung	0.1927847266197205, 0.0	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
93	5/20/2024	I am struggling with exe	Galaxy Watch 4	Watch	4	Samsung	0.07397448271512985, 0.0	['FITNESS & HEALTH TRACKING']
94	10/27/2021	The GOOD: this watch	Galaxy Watch 4	Watch	4	Samsung	0.7440103688511658, 0.5	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'MULTIMEDIA']
95	5/1/2024	I love the way it looks, e	Galaxy Watch 4	Watch	4	Samsung	0.9316688776016235, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
96	4/23/2024	I like the functionality a	Galaxy Watch 4	Watch	4	Samsung	0.9739987224006653, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
97	4/18/2024	I enjoyed of how it keep	Galaxy Watch 4	Watch	4	Samsung	0.9798496758460999, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
98	10/18/2023	I do like this watch very	Galaxy Watch 4	Watch	4	Samsung	0.9782637953758240, 0.25	['BATTERY', 'FITNESS & HEALTH TRACKING']
99	10/22/2022	I was very hesitant to bi	Galaxy Watch 4	Watch	4	Samsung	0.9911761283874512, 0.7	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
100	4/20/2022	I purchased the Samsu	Galaxy Watch 4	Watch	4	Samsung	0.9818336367607117, 0.8	['BATTERY', 'PERFORMANCE', 'DISPLAY']
101	4/16/2024	Bought the watch for a	Galaxy Watch 4	Watch	4	Samsung	0.8866819062423706, 0.0	['BATTERY', 'MULTIMEDIA']
102	4/19/2023	Love all of the features	Galaxy Watch 4	Watch	4	Samsung	0.9940705895423889, 0.1	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
103	1/11/2023	Look, by and large it's a	Galaxy Watch 4	Watch	4	Samsung	0.9822151064872742, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
104	2/4/2024	this a little large for my	Galaxy Watch 4	Watch	4	Samsung	0.9735975265502930, 0.01	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
105	6/4/2022	I like Samsung product	Galaxy Watch 4	Watch	4	Samsung	0.15548556552959442, 0.1	['PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
106	2/5/2022	So I had the very first ve	Galaxy Watch 4	Watch	4	Samsung	0.9926885891175130, 0.4	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
107	2/16/2024	I love this samsung gal	Galaxy Watch 4	Watch	4	Samsung	0.9626530742073059, 0.6	['BATTERY', 'PERFORMANCE']
108	1/26/2024	Battery lasts all the way	Galaxy Watch 4	Watch	4	Samsung	0.9809160048828650, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
109	3/13/2024	I enjoy all the things thi	Galaxy Watch 4	Watch	4	Samsung	0.9665169254797854, 0.0	['BATTERY']
110	2/20/2024	Product as described.	Galaxy Watch 4	Watch	4	Samsung	0.9810898900032043, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
111	5/24/2022	After using this watch f	Galaxy Watch 4	Watch	4	Samsung	0.9403646621704102, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
112	2/29/2024	Charger gets REALLY h	Galaxy Watch 4	Watch	4	Samsung	0.9836362624168225, 0.1	['BATTERY']
113	2/6/2024	My husband has a smart	Galaxy Watch 4	Watch	4	Samsung	0.9644074449062441, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
114	1/20/2024	Have everything about	Galaxy Watch 4	Watch	4	Samsung	0.9647771716117859, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
115	1/10/2024	Love the watch, hate th	Galaxy Watch 4	Watch	4	Samsung	0.9736894097519921, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
116	2/3/2023	I got this watch as a Ch	Galaxy Watch 4	Watch	4	Samsung	0.7718713283538818, 0.8	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
117	8/1/2023	I got this watch 11 days	Galaxy Watch 4	Watch	4	Samsung	0.8511569923590796, 0.0	['BATTERY']
118	3/14/2022	This is a great watch. I	Galaxy Watch 4	Watch	4	Samsung	0.8842134475708008, 0.1	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
119	8/5/2022	I'm coming off owning	Galaxy Watch 4	Watch	4	Samsung	0.9984842796325684, 0.5	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
120	1/1/2024	The watch it's self work	Galaxy Watch 4	Watch	4	Samsung	0.9944302439689636, 0.2	['BATTERY']
121	12/23/2023	Works well. Tracks my r	Galaxy Watch 4	Watch	4	Samsung	0.9660478830337524, 0.4	['BATTERY', 'FITNESS & HEALTH TRACKING']
122	12/8/2023	Took a while to get here	Galaxy Watch 4	Watch	4	Samsung	0.9131540600746051, 0.0	['BATTERY', 'PRICE']
123	10/23/2022	I bought this Watch 4 c	Galaxy Watch 4	Watch	4	Samsung	0.9395268440949304, 0.0	['BATTERY']
124	1/16/2024	Great fit and feel. Lots	Galaxy Watch 4	Watch	4	Samsung	0.9310782551765442, 0.3	['BATTERY']
125	11/24/2021	Don't get me wrong, th	Galaxy Watch 4	Watch	4	Samsung	0.9409484267234802, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
126	1/7/2024	very versitile and does v	Galaxy Watch 4	Watch	4	Samsung	0.9290069937705994, 0.0	['BATTERY', 'DISPLAY']
127	10/15/2021	I have only had my wate	Galaxy Watch 4	Watch	4	Samsung	0.9804347157478333, 0.1	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
128	10/15/2022	I love technology, espe	Galaxy Watch 4	Watch	4	Samsung	0.5493173003196716, 0.2	['BATTERY', 'DISPLAY', 'CUSTOMIZATION']
129	10/18/2022	I use it for cycling and r	Galaxy Watch 4	Watch	4	Samsung	0.9272775650024414, 0.0	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
130	12/31/2022	So I upgraded from my	Galaxy Watch 4	Watch	4	Samsung	0.9117134809494019, 0.0	['BATTERY', 'MULTIMEDIA']
131	10/9/2023	Really good watch! Live	Galaxy Watch 4	Watch	4	Samsung	0.9579768085479736, 0.0	['BATTERY', 'PRICE']
132	8/9/2022	It works quite well with	Galaxy Watch 4	Watch	4	Samsung	0.1535893176050714, 0.5	['PERFORMANCE', 'FITNESS & HEALTH TRACKING']
133	4/11/2022	I am a woman with big h	Galaxy Watch 4	Watch	4	Samsung	0.4821830368995667, 0.01	['DISPLAY']
134	10/30/2023	I love everything about	Galaxy Watch 4	Watch	4	Samsung	0.9442866444587708, 0.0	['BATTERY']
135	12/29/2022	It is pretty well known t	Galaxy Watch 4	Watch	4	Samsung	0.9477381110191345, 0.0	['BATTERY', 'PRICE']
136	1/7/2023	I've had my watch for al	Galaxy Watch 4	Watch	4	Samsung	0.7326778885078712, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
137	3/29/2022	I have had this watch fc	Galaxy Watch 4	Watch	4	Samsung	0.987006425857544, 0.39	['BATTERY', 'DISPLAY', 'MULTIMEDIA']
138	10/30/2022	I loved this watch for al	Galaxy Watch 4	Watch	4	Samsung	0.9827225208282471, 0.1	['BATTERY']
139	5/29/2023	The watch is really grea	Galaxy Watch 4	Watch	4	Samsung	0.0806710496544838, 0.1	['FITNESS & HEALTH TRACKING']
140	11/20/2023	Fits well.Keeps track. Br	Galaxy Watch 4	Watch	4	Samsung	0.9990241328239441, 0.4	['BATTERY']
141	5/26/2022	I am very happy with thi	Galaxy Watch 4	Watch	4	Samsung	0.9756317341804504, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
142	9/11/2022	I am a long time Samsu	Galaxy Watch 4	Watch	4	Samsung	0.9833010697615295, 0.8	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
143	6/11/2023	Upgraded from Galaxy I	Galaxy Watch 4	Watch	4	Samsung	0.9916115403175354, 0.5	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
144	5/25/2023	I do really like this watc	Galaxy Watch 4	Watch	4	Samsung	0.4167339927484197, 0.1	['DISPLAY']
145	3/3/2022	The media could not be	Galaxy Watch 4	Watch	4	Samsung	0.9739022850990295, 0.7	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
146	9/29/2021	I love all the features or	Galaxy Watch 4	Watch	4	Samsung	0.9788450771713257, 0.90	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
147	12/17/2023	The battery is very low	Galaxy Watch 4	Watch	4	Samsung	0.9701940417289734, 0.0	['BATTERY']
148	9/25/2021	I have no doubt this wil	Galaxy Watch 4	Watch	4	Samsung	0.9967280626296997, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
149	7/31/2023	Good device. I like it a lc	Galaxy Watch 4	Watch	4	Samsung	0.9857822656631470, 0.09	['BATTERY']
150	1/3/2023	Today is my first full us	Galaxy Watch 4	Watch	4	Samsung	0.9530107975006104, 0.1	['BATTERY']
151	6/7/2022	The sleep and stress tr	Galaxy Watch 4	Watch	4	Samsung	0.9663610458374023, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
152	2/25/2022	I absolutely love my wa	Galaxy Watch 4	Watch	4	Samsung	0.9394944906234741, 0.0	['BATTERY', 'MULTIMEDIA']
153	3/2/2023	Had to charge it since it	Galaxy Watch 4	Watch	4	Samsung	0.9457769663543701, 0.3	['BATTERY', 'DISPLAY', 'MULTIMEDIA']
154	7/28/2023	Still breaking it in after	Galaxy Watch 4	Watch	4	Samsung	0.9283954501152039, 0.1	['BATTERY', 'DISPLAY']
155	12/4/2022	I bought this watch bec	Galaxy Watch 4	Watch	4	Samsung	0.9213760018348694, 0.1	['PRICE']



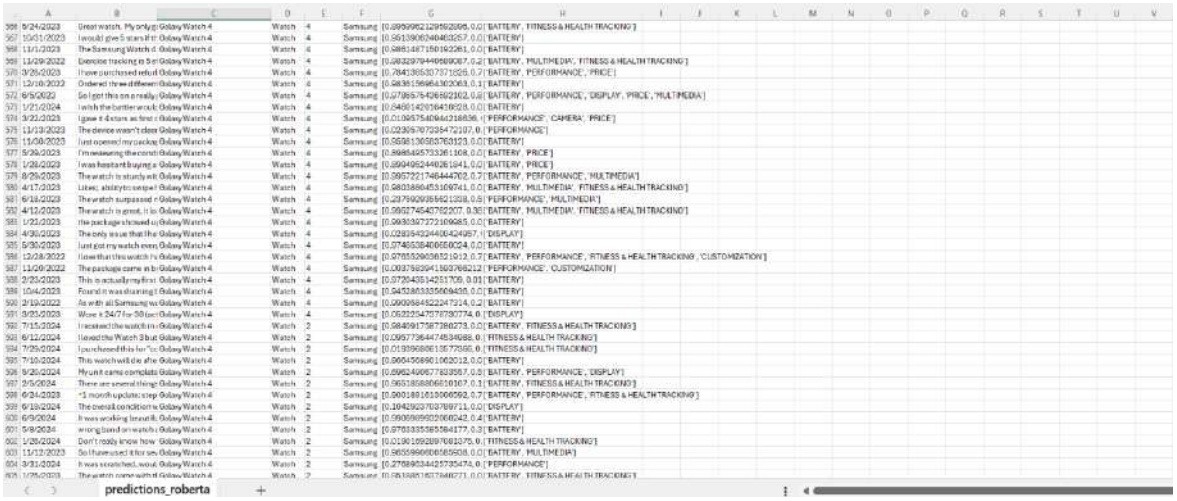
	A	B	C	D	E	F	G	H
116	2/3/2023	I got this watch as a Ch	Galaxy Watch 4	Watch	4	Samsung	0.7718713283538818, 0.8	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
117	8/1/2023	I got this watch 11 days	Galaxy Watch 4	Watch	4	Samsung	0.8511569923590796, 0.0	['BATTERY']
118	3/14/2022	This is a great watch. I	Galaxy Watch 4	Watch	4	Samsung	0.8842134475708008, 0.1	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
119	8/5/2022	I'm coming off owning	Galaxy Watch 4	Watch	4	Samsung	0.9984842796325684, 0.5	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
120	1/1/2024	The watch it's self work	Galaxy Watch 4	Watch	4	Samsung	0.9944302439689636, 0.2	['BATTERY']
121	12/23/2023	Works well. Tracks my r	Galaxy Watch 4	Watch	4	Samsung	0.9660478830337524, 0.4	['BATTERY', 'FITNESS & HEALTH TRACKING']
122	12/8/2023	Took a while to get here	Galaxy Watch 4	Watch	4	Samsung	0.9131540600746051, 0.0	['BATTERY', 'PRICE']
123	10/23/2022	I bought this Watch 4 c	Galaxy Watch 4	Watch	4	Samsung	0.9395268440949304, 0.0	['BATTERY']
124	1/16/2024	Great fit and feel. Lots	Galaxy Watch 4	Watch	4	Samsung	0.9310782551765442, 0.3	['BATTERY']
125	11/24/2021	Don't get me wrong, th	Galaxy Watch 4	Watch	4	Samsung	0.9409484267234802, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
126	1/7/2024	very versitile and does v	Galaxy Watch 4	Watch	4	Samsung	0.9290069937705994, 0.0	['BATTERY', 'DISPLAY']
127	10/15/2021	I have only had my wate	Galaxy Watch 4	Watch	4	Samsung	0.9804347157478333, 0.1	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
128	10/15/2022	I love technology, espe	Galaxy Watch 4	Watch	4	Samsung	0.5493173003196716, 0.2	['BATTERY', 'DISPLAY', 'CUSTOMIZATION']
129	10/18/2022	I use it for cycling and r	Galaxy Watch 4	Watch	4	Samsung	0.9272775650024414, 0.0	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
130	12/31/2022	So I upgraded from my	Galaxy Watch 4	Watch	4	Samsung	0.9117134809494019, 0.0	['BATTERY', 'MULTIMEDIA']
131	10/9/2023	Really good watch! Live	Galaxy Watch 4	Watch	4	Samsung	0.9579768085479736, 0.0	['BATTERY', 'PRICE']
132	8/9/2022	It works quite well with	Galaxy Watch 4	Watch	4	Samsung	0.1535893176050714, 0.5	['PERFORMANCE', 'FITNESS & HEALTH TRACKING']
133	4/11/2022	I am a woman with big h	Galaxy Watch 4	Watch	4	Samsung	0.4821830368995667, 0.01	['DISPLAY']
134	10/30/2023	I love everything about	Galaxy Watch 4	Watch	4	Samsung	0.9442866444587708, 0.0	['BATTERY']
135	12/29/2022	It is pretty well known t	Galaxy Watch 4	Watch	4	Samsung	0.9477381110191345, 0.0	['BATTERY', 'PRICE']
136	1/7/2023	I've had my watch for al	Galaxy Watch 4	Watch	4	Samsung	0.7326778885078712, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
137	3/29/2022	I have had this watch fc	Galaxy Watch 4	Watch	4	Samsung	0.987006425857544, 0.39	['BATTERY', 'DISPLAY', 'MULTIMEDIA']
138	10/30/2022	I loved this watch for al	Galaxy Watch 4	Watch	4	Samsung	0.9827225208282471, 0.1	['BATTERY']
139	5/29/2023	The watch is really grea	Galaxy Watch 4	Watch	4	Samsung	0.0806710496544838, 0.1	['FITNESS & HEALTH TRACKING']
140	11/20/2023	Fits well.Keeps track. Br	Galaxy Watch 4	Watch	4	Samsung	0.9990241328239441, 0.4	['BATTERY']
141	5/26/2022	I am very happy with thi	Galaxy Watch 4	Watch	4	Samsung	0.9756317341804504, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
142	9/11/2022	I am a long time Samsu	Galaxy Watch 4	Watch	4	Samsung	0.9833010697615295, 0.8	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
143	6/11/2023	Upgraded from Galaxy I	Galaxy Watch 4	Watch	4	Samsung	0.9916115403175354, 0.5	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
144	5/25/2023	I do really like this watc	Galaxy Watch 4	Watch	4	Samsung	0.4167339927484197, 0.1	['DISPLAY']
145	3/3/2022	The media could not be	Galaxy Watch 4	Watch	4	Samsung	0.9739022850990295, 0.7	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
146	9/29/2021	I love all the features or	Galaxy Watch 4	Watch	4	Samsung	0.9788450771713257, 0.90	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
147	12/17/2023	The battery is very low	Galaxy Watch 4	Watch	4	Samsung	0.9701940417289734, 0.0	['BATTERY']
148	9/25/2021	I have no doubt this wil	Galaxy Watch 4	Watch	4	Samsung	0.9967280626296997, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
149	7/31/2023	Good device. I like it a lc	Galaxy Watch 4	Watch	4	Samsung	0.9857822656631470, 0.09	['BATTERY']
150	1/3/2023	Today is my first full us	Galaxy Watch 4	Watch	4	Samsung	0.9530107975006104, 0.1	['BATTERY']
151	6/7/2022	The sleep and stress tr	Galaxy Watch 4	Watch	4	Samsung	0.9663610458374023, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
152	2/25/2022	I absolutely love my wa	Galaxy Watch 4	Watch	4	Samsung	0.9394944906234741, 0.0	['BATTERY', 'MULTIMEDIA']
153	3/2/2023	Had to charge it since it	Galaxy Watch 4	Watch	4	Samsung	0.9457769663543701, 0.3	['BATTERY', 'DISPLAY', 'MULTIMEDIA']
154	7/28/2023	Still breaking it in after	Galaxy Watch 4	Watch	4	Samsung	0.9283954501152039, 0.1	['BATTERY', 'DISPLAY']
155	12/4/2022	I bought this watch bec	Galaxy Watch 4	Watch	4	Samsung	0.9213760018348694, 0.1	['PRICE']



	A	B	C	D	E	F	G	H
176	10/4/2023	I love how Samsung's v	Galaxy Watch 4	Watch	3	Samsung	0.07775681465864182, 0.0	['PERFORMANCE']
177	8/26/2024	I just bought this two d	Galaxy Watch 4	Watch	3	Samsung	0.9467501254081726, 0.0	['BATTERY']
178	7/11/2024	I ordered this watch in.	Galaxy Watch 4	Watch	3	Samsung	0.9486075043678284, 0.0	['BATTERY']
179	4/1/2024	I've been trying to get m	Galaxy Watch 4	Watch	3	Samsung	0.9937771558761597, 0.4	['BATTERY']
180	5/7/2024	Not sure why the blooc	Galaxy Watch 4	Watch	3	Samsung	0.10755375027656555, 0.0	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
181	9/28/2023	I waited two months to	Galaxy Watch 4	Watch	3	Samsung	0.9686435461044312, 0.2	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
182	5/25/2024	I don't feel the vibration	Galaxy Watch 4	Watch	3	Samsung	0.0093122764434665544, 0.0	['MULTIMEDIA']
183	7/4/2022	I have had the watch 4 I	Galaxy Watch 4	Watch	3	Samsung	0.11611661314964294, 0.0	['DISPLAY', 'FITNESS & HEALTH TRACKING']
184	6/4/2024	Everything but the batt	Galaxy Watch 4	Watch	3	Samsung	0.9857342243194458, 0.01	['BATTERY']
185	3/1/2023	This is a beautiful watc	Galaxy Watch 4	Watch	3	Samsung	0.9698487520217896, 0.5	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
186	11/1/2021	I have owned a lot of ar	Galaxy Watch 4	Watch	3	Samsung	0.04618084511804581, 0.0	['PERFORMANCE', 'DISPLAY']
187	10/17/2022	This is the second Sam	Galaxy Watch 4	Watch	3	Samsung	0.9811111092567444, 0.2	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
188	4/18/2024	It stays connected to m	Galaxy Watch 4	Watch	3	Samsung	0.9917681217193604, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
189	12/2/2022	I love this watch. All th	Galaxy Watch 4	Watch	3	Samsung	0.9922960045058136, 0.23	['BATTERY', 'MULTIMEDIA']
190	1/14/2024	I was hoping the review	Galaxy Watch 4	Watch	3	Samsung	0.9683442504405975, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
191	7/12/2023	Its a pretty watch. Muc	Galaxy Watch 4	Watch	3	Samsung	0.9640995045280457, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
192	3/5/2024	The battery life is short	Galaxy Watch 4	Watch	3	Samsung	0.9819268584251404, 0.4	['BATTERY']
193	3/5/2024	The watch is amazing, I	Galaxy Watch 4	Watch	3	Samsung	0.01643375679850578, 0.1	['PERFORMANCE', 'PRICE']
194	2/27/2024	This is a very nice watcl	Galaxy Watch 4	Watch	3	Samsung	0.9526007771492004, 0.0	['BATTERY']
195	2/17/2024	I love all of the features	Galaxy Watch 4	Watch	3	Samsung	0.07069159299135208, 0.1	['FITNESS & HEALTH TRACKING']
196	4/19/2022	FYI- I purchased the wa	Galaxy Watch 4	Watch	3	Samsung	0.9368256866555664, 0.1	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
197	7/29/2022	1. If you want to use Sa	Galaxy Watch 4	Watch	3	Samsung	0.9488616993141174, 0.6	['BATTERY', 'PERFORMANCE', 'DISPLAY']
198	8/5/2022	I've worn smart watche	Galaxy Watch 4	Watch	3	Samsung	0.8279998166656494, 0.0	['BATTERY']
199	9/19/2022	I LOVE this watch and c	Galaxy Watch 4	Watch	3	Samsung	0.9517504577681958, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
200	1/30/2023	I experienced intermitto	Galaxy Watch 4	Watch	3	Samsung	0.012004924938082695, 0.0	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
201	2/28/2024	Yesterday I got my wate	Galaxy Watch 4	Watch	3	Samsung	0.9854420423507690, 0.03	['BATTERY']
202	7/13/2022	I've been an avid fitbit v	Galaxy Watch 4	Watch	3	Samsung	0.9764541387557983, 0.1	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
203	11/19/2021	The watch was OK. Set	Galaxy Watch 4	Watch	3	Samsung	0.9337569471717996, 0.07	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
204	12/27/2023	I bought this in Februa	Galaxy Watch 4	Watch	3	Samsung	0.9095129370689392, 0.0	['BATTERY', 'PRICE']
205	1/19/2023	I like the look of this wa	Galaxy Watch 4	Watch	3	Samsung	0.8930060863494873, 0.0	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
206	12/16/2023	The media could not be	Galaxy Watch 4	Watch	3	Samsung	0.9749616193413330, 0.02	['BATTERY', 'DISPLAY']
207	10/4/2023	I have been a big fan of	Galaxy Watch 4	Watch	3	Samsung	0.9927894473075807, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
208	7/27/2022	The biggest thing I don	Galaxy Watch 4	Watch	3	Samsung	0.9661017060279846, 0.0	['BATTERY', 'PRICE', 'FITNESS & HEALTH TRACKING']
209	7/17/2022	Had my Active 2 for ove	Galaxy Watch 4	Watch	3	Samsung	0.9829664230346680, 0.15	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
210	8/30/2023	Even when I have nothi	Galaxy Watch 4	Watch	3	Samsung	0.8901900649070740, 0.79	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
211	4/26/2022	I bought this watch bec	Galaxy Watch 4	Watch	3	Samsung	0.0800906455574096, 0.0	['BATTERY']
212	7/19/2023	The watch was working	Galaxy Watch 4	Watch	3	Samsung	0.023881975561380386, 0.0	['DISPLAY', 'PRICE']
213	4/23/2022	Ã¢â‚¬â„¢ve tried sever	Galaxy Watch 4	Watch	3	Samsung	0.9262562394142151, 0.0	['BATTERY', 'MULTIMEDIA']
214	2/8/2024	My first purchase seem	Galaxy Watch 4	Watch	3	Samsung	0.9768161773681641, 0.0	['BATTERY']
215	9/28/2022	I had the previous galax	Galaxy Watch 4	Watch	3	Samsung	0.9614442586898804, 0.0	['BATTERY', 'PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']

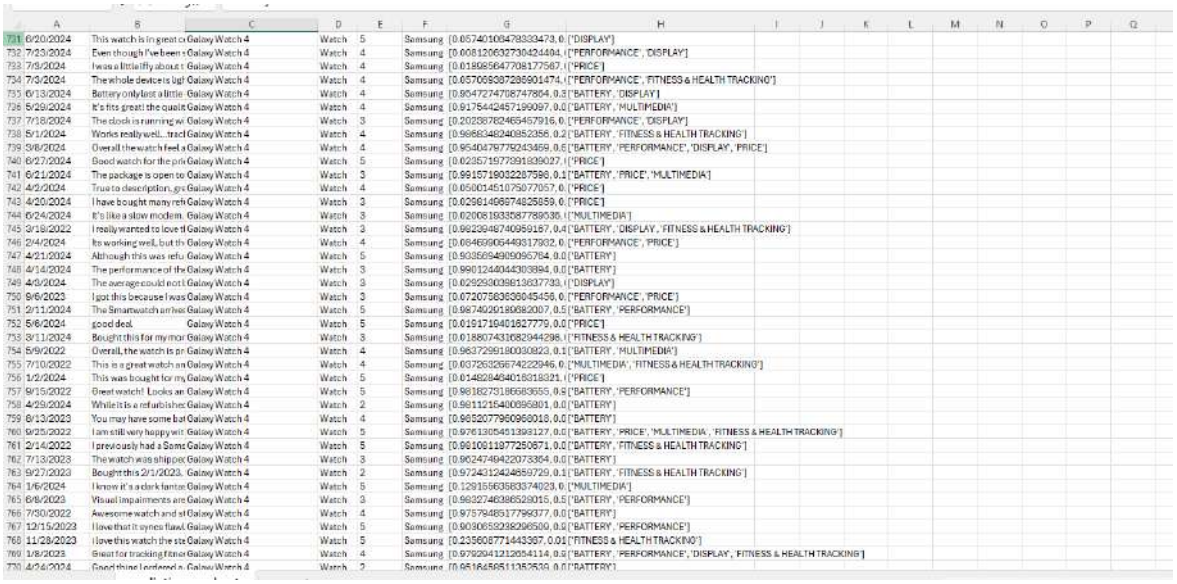


	A	B	C	D	E	F	G	H
291	10/8/2021	Being a big Samsung fa	Galaxy Watch 4	Watch	2	Samsung	[0.9601404668900686, 0.0]	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
292	4/9/2024	I waited a long time to g	Galaxy Watch 4	Watch	2	Samsung	[0.9930574196815491, 0.0]	['BATTERY']
293	11/26/2023	The battery life is okay.	Galaxy Watch 4	Watch	2	Samsung	[0.9745864674440002, 0.0]	['BATTERY']
294	3/15/2024	Heart rate is very inacc	Galaxy Watch 4	Watch	2	Samsung	[0.9992121613923645, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
295	9/29/2022	First of all, Let me just s	Galaxy Watch 4	Watch	2	Samsung	[0.17777048051537227, 0.1]	['FITNESS & HEALTH TRACKING']
296	4/16/2024	On a full charge only ge	Galaxy Watch 4	Watch	2	Samsung	[0.9907637929916382, 0.0]	['BATTERY']
297	12/1/2021	This watch has one key	Galaxy Watch 4	Watch	2	Samsung	[0.9687048196702603, 0.1]	['BATTERY', 'FITNESS & HEALTH TRACKING']
298	3/20/2024	I need to wear the wate	Galaxy Watch 4	Watch	2	Samsung	[0.9270208477979938, 0.0]	['BATTERY']
299	11/15/2023	I like the look of the wa	Galaxy Watch 4	Watch	2	Samsung	[0.8840569287271729, 0.0]	['BATTERY']
300	3/9/2023	I was so excited to "upg	Galaxy Watch 4	Watch	2	Samsung	[0.9540154937982996, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
301	1/10/2024	I take up to four days w	Galaxy Watch 4	Watch	2	Samsung	[0.9927183675765001, 0.0]	['BATTERY']
302	3/17/2024	The battery lasts less th	Galaxy Watch 4	Watch	2	Samsung	[0.9703851342201233, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
303	12/12/2021	I really was excited to g	Galaxy Watch 4	Watch	2	Samsung	[0.9833561815579559, 0.28]	['BATTERY', 'FITNESS & HEALTH TRACKING']
304	11/28/2023	Good watch, but the bc	Galaxy Watch 4	Watch	2	Samsung	[0.9703959590991431, 0.0]	['BATTERY']
305	5/8/2022	Nothing new, came to t	Galaxy Watch 4	Watch	2	Samsung	[0.9117604497104846, 0.51]	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE']
306	2/22/2023	Very frustrated with thi	Galaxy Watch 4	Watch	2	Samsung	[0.9950121730526489, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
307	4/15/2022	The first one I bought w	Galaxy Watch 4	Watch	2	Samsung	[0.6700126744475635, 0.0]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
308	3/24/2022	I bought this March of 2	Galaxy Watch 4	Watch	2	Samsung	[0.6705496533093232, 0.0]	['FITNESS & HEALTH TRACKING']
309	1/24/2023	This watch was very cle	Galaxy Watch 4	Watch	2	Samsung	[0.9857968357545045, 0.2]	['BATTERY']
310	11/26/2022	This watch has a lot of	Galaxy Watch 4	Watch	2	Samsung	[0.9900938887799613, 0.51]	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
311	12/4/2022	Once again Samsung p	Galaxy Watch 4	Watch	2	Samsung	[0.0104776451835123, 0.0]	['PERFORMANCE']
312	6/12/2022	I do like the health featu	Galaxy Watch 4	Watch	2	Samsung	[0.9999758699556091, 0.0]	['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
313	4/24/2022	I bought this watch bec	Galaxy Watch 4	Watch	2	Samsung	[0.9888276861740112, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
314	6/23/2022	I loved my Active 2. But	Galaxy Watch 4	Watch	2	Samsung	[0.9870098033752441, 0.0]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
315	1/26/2023	Disgusted. To me, a wa	Galaxy Watch 4	Watch	2	Samsung	[0.017448775754966507, 0.0]	['DISPLAY', 'MULTIMEDIA']
316	2/18/2022	What I like this is a nice	Galaxy Watch 4	Watch	2	Samsung	[0.9917262792587728, 0.15]	['BATTERY', 'MULTIMEDIA']
317	10/5/2022	It's hard when you buy	Galaxy Watch 4	Watch	2	Samsung	[0.0054156000275402674, 0.0]	['PERFORMANCE', 'PRICE']
318	7/29/2023	the battery is crap. I da	Galaxy Watch 4	Watch	2	Samsung	[0.9419912099938257, 0.0]	['BATTERY', 'DISPLAY', 'PRICE']
319	10/9/2022	This watch was a replac	Galaxy Watch 4	Watch	2	Samsung	[0.9669411182403564, 0.0]	['BATTERY']
320	5/18/2022	I upgraded from a Galax	Galaxy Watch 4	Watch	2	Samsung	[0.9937944412231445, 0.0]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
321	12/9/2021	I've had a garmin watch	Galaxy Watch 4	Watch	2	Samsung	[0.9912871718406677, 0.0]	['BATTERY']
322	1/11/2024	Watch purchased and	Galaxy Watch 4	Watch	2	Samsung	[0.7270042204666679, 0.1]	['BATTERY']
323	1/26/2023	Quiet possibly the wor	Galaxy Watch 4	Watch	2	Samsung	[0.010160363792749067, 0.0]	['PERFORMANCE', 'MULTIMEDIA']
324	9/9/2022	In short I enjoyed the wa	Galaxy Watch 4	Watch	2	Samsung	[0.02925722487270832, 0.0]	['DISPLAY']
325	11/18/2021	I originally ordered this	Galaxy Watch 4	Watch	2	Samsung	[0.8926436889231873, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
326	11/19/2022	I had the Samsung activ	Galaxy Watch 4	Watch	2	Samsung	[0.9912599980114502, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
327	9/13/2023	I had a galaxy sport 2 ar	Galaxy Watch 4	Watch	2	Samsung	[0.8602066602027100, 0.0]	['BATTERY']
328	9/15/2023	I wanted to love this BU	Galaxy Watch 4	Watch	2	Samsung	[0.9408696298360872, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
329	12/20/2022	This thing is absolutely	Galaxy Watch 4	Watch	2	Samsung	[0.9437304139137268, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
330	1/27/2023	I tried researching befo	Galaxy Watch 4	Watch	2	Samsung	[0.01270983720874786, 0.0]	['FITNESS & HEALTH TRACKING']

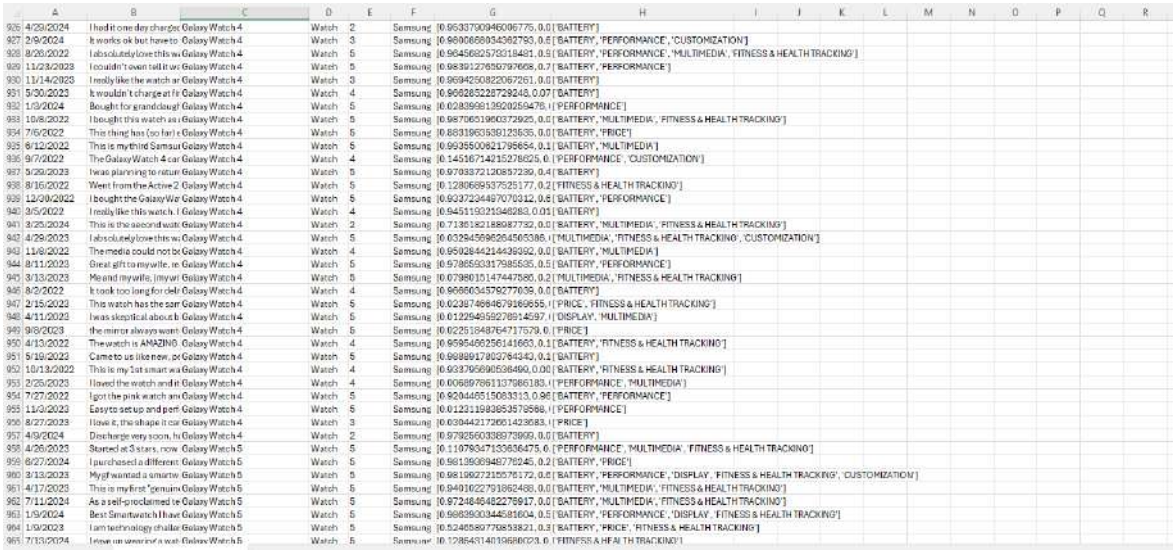


	A	B	C	D	E	F	G	H
566	9/24/2023	Great watch. My only g	Galaxy Watch 4	Watch	4	Samsung	[0.9899962129592896, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
567	10/31/2023	I would give 5 stars if it	Galaxy Watch 4	Watch	4	Samsung	[0.9513906240463257, 0.0]	['BATTERY']
568	11/1/2023	The Samsung Watch 4	Galaxy Watch 4	Watch	4	Samsung	[0.9881487155103302, 0.0]	['BATTERY']
569	12/29/2022	Exercise tracking is bit	Galaxy Watch 4	Watch	4	Samsung	[0.9832979440689087, 0.2]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
570	3/26/2023	I have purchased refurl	Galaxy Watch 4	Watch	4	Samsung	[0.7841385307371826, 0.7]	['BATTERY', 'PERFORMANCE', 'PRICE']
571	12/16/2022	Ordered three different	Galaxy Watch 4	Watch	4	Samsung	[0.9838159954302063, 0.1]	['BATTERY']
572	6/5/2022	So I got this on a really	Galaxy Watch 4	Watch	4	Samsung	[0.9706574266922102, 0.0]	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE', 'MULTIMEDIA']
573	1/21/2024	I wish the battery coul	Galaxy Watch 4	Watch	4	Samsung	[0.9468142676419258, 0.0]	['BATTERY']
574	3/21/2023	I gave it 4 stars as first	Galaxy Watch 4	Watch	4	Samsung	[0.01965745664218856, 0.1]	['PERFORMANCE', 'CAMERA', 'PRICE']
575	11/13/2023	The device wasn't clear	Galaxy Watch 4	Watch	4	Samsung	[0.023057673954727077, 0.1]	['PERFORMANCE']
576	11/30/2023	Just opened my packag	Galaxy Watch 4	Watch	4	Samsung	[0.9598136563763123, 0.0]	['BATTERY']
577	5/29/2023	I'm reviewing the consti	Galaxy Watch 4	Watch	4	Samsung	[0.9985495733361328, 0.0]	['BATTERY', 'PRICE']
578	1/28/2023	I was hesitant buying a	Galaxy Watch 4	Watch	4	Samsung	[0.9904952446261841, 0.0]	['BATTERY', 'PRICE']
579	8/29/2023	The watch is sturdy wit	Galaxy Watch 4	Watch	4	Samsung	[0.9957221746444702, 0.7]	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA']
580	4/17/2023	Likes; ability to swipe t	Galaxy Watch 4	Watch	4	Samsung	[0.9803886451530761, 0.0]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
581	6/18/2023	The watch surpassed r	Galaxy Watch 4	Watch	4	Samsung	[0.22759029585623169, 0.5]	['PERFORMANCE', 'MULTIMEDIA']
582	4/12/2023	The watch is great, it lo	Galaxy Watch 4	Watch	4	Samsung	[0.9902745407629207, 0.35]	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
583	1/21/2023	the package showed up	Galaxy Watch 4	Watch	4	Samsung	[0.9930327272109985, 0.0]	['BATTERY']
584	4/30/2023	The only issue that the	Galaxy Watch 4	Watch	4	Samsung	[0.02359423446846008, 0.0]	['DISPLAY']
585	5/30/2023	I just got my watch even	Galaxy Watch 4	Watch	4	Samsung	[0.9745538401603699, 0.0]	['BATTERY']
586	12/28/2022	I love that this watch h	Galaxy Watch 4	Watch	4	Samsung	[0.9705329728126526, 0.7]	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
587	11/29/2022	The package came in b	Galaxy Watch 4	Watch	4	Samsung	[0.0037589991188621521, 0.0]	['PERFORMANCE', 'CUSTOMIZATION']
588	2/23/2023	This is actually my first	Galaxy Watch 4	Watch	4	Samsung	[0.9720435142517090, 0.01]	['BATTERY']
589	10/4/2023	Found it was draining t	Galaxy Watch 4	Watch	4	Samsung	[0.9452803325653076, 0.0]	['BATTERY']
590	2/19/2022	As with all Samsung w	Galaxy Watch 4	Watch	4	Samsung	[0.9902584552240753, 0.2]	['BATTERY']
591	9/25/2023	Wore it 24/7 for 30 hrs	Galaxy Watch 4	Watch	4	Samsung	[0.9622254753303528, 0.0]	['DISPLAY']
592	7/15/2024	I received the watch in	Galaxy Watch 4	Watch	2	Samsung	[0.9849917507362366, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
593	6/12/2024	I loved the Watch 3 but	Galaxy Watch 4	Watch	2	Samsung	[0.09577364474334717, 0.0]	['FITNESS & HEALTH TRACKING']
594	7/25/2024	I purchased this for "c	Galaxy Watch 4	Watch	2	Samsung	[0.013990800118446350, 0.0]	['FITNESS & HEALTH TRACKING']
595	7/10/2024	This watch will die afte	Galaxy Watch 4	Watch	2	Samsung	[0.0004508901023865, 0.0]	['BATTERY']
596	9/26/2024	My unit came complete	Galaxy Watch 4	Watch	2	Samsung	[0.9962400078773499, 0.0]	['BATTERY', 'PERFORMANCE', 'DISPLAY']
597	9/5/2024	There are several things	Galaxy Watch 4	Watch	2	Samsung	[0.9631858086013794, 0.1]	['BATTERY', 'FITNESS & HEALTH TRACKING']
598	6/24/2023	+1 month update step	Galaxy Watch 4	Watch	2	Samsung	[0.9001881313896179, 0.7]	['BATTERY', 'PERFORMANCE', 'FITNESS & HEALTH TRACKING']
599	6/18/2024	The overall condition w	Galaxy Watch 4	Watch	2	Samsung	[0.10429227352142334, 0.0]	['DISPLAY']
600	6/9/2024	It was working beautifu	Galaxy Watch 4	Watch	2	Samsung	[0.9906969261169434, 0.4]	['BATTERY']
601	5/8/2024	wrong band on watch,	Galaxy Watch 4	Watch	2	Samsung	[0.9761335563659668, 0.3]	['BATTERY']
602	1/26/2024	Don't really know how	Galaxy Watch 4	Watch	2	Samsung	[0.01961922645568848, 0.0]	['FITNESS & HEALTH TRACKING']
603	11/12/2023	So I have used it for sev	Galaxy Watch 4	Watch	2	Samsung	[0.9659969806671143, 0.0]	['BATTERY', 'MULTIMEDIA']
604	3/31/2024	It was scratched, woul	Galaxy Watch 4	Watch	2	Samsung	[0.2768903446197510, 0.0]	['PERFORMANCE']
605	1/25/2023	The watch came with d	Galaxy Watch 4	Watch	2	Samsung	[0.9518851637840271, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']

predictions_roberta

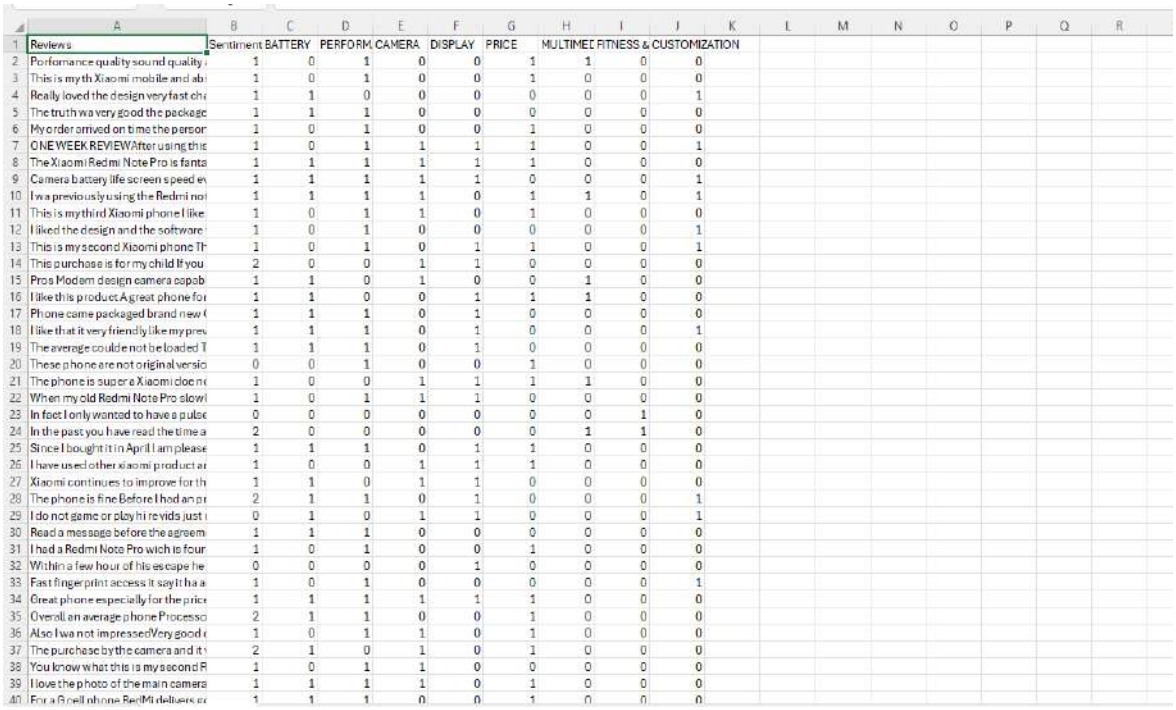


	A	B	C	D	E	F	G	H
731	6/20/2024	This watch is in great co	Galaxy Watch 4	Watch	5	Samsung	[0.05740108478333473, 0.0]	['DISPLAY']
732	7/23/2024	Even though I've been v	Galaxy Watch 4	Watch	4	Samsung	[0.008120633730424404, 0.0]	['PERFORMANCE', 'DISPLAY']
733	7/9/2024	I was a little iffy about t	Galaxy Watch 4	Watch	4	Samsung	[0.018985647708177567, 0.0]	['PRICE']
734	7/3/2024	The whole device is big!	Galaxy Watch 4	Watch	4	Samsung	[0.05706938728690147, 0.0]	['PERFORMANCE', 'FITNESS & HEALTH TRACKING']
735	6/13/2024	Battery only last a little	Galaxy Watch 4	Watch	4	Samsung	[0.9047274708747864, 0.3]	['BATTERY', 'DISPLAY']
736	5/29/2024	It's fits great! the qualit	Galaxy Watch 4	Watch	4	Samsung	[0.9175442457199097, 0.0]	['BATTERY', 'MULTIMEDIA']
737	7/18/2024	The clock is running w	Galaxy Watch 4	Watch	3	Samsung	[0.20258782465457916, 0.0]	['PERFORMANCE', 'DISPLAY']
738	5/1/2024	Works really well... tracl	Galaxy Watch 4	Watch	4	Samsung	[0.9868348240852356, 0.2]	['BATTERY', 'FITNESS & HEALTH TRACKING']
739	3/8/2024	Overall the watch feel a	Galaxy Watch 4	Watch	4	Samsung	[0.9540479779243469, 0.6]	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'PRICE']
740	6/27/2024	Good watch for the pric	Galaxy Watch 4	Watch	5	Samsung	[0.023571977391839027, 0.0]	['PRICE']
741	6/21/2024	The package is open to	Galaxy Watch 4	Watch	3	Samsung	[0.9915719032287598, 0.1]	['BATTERY', 'PRICE', 'MULTIMEDIA']
742	4/2/2024	True to description, gre	Galaxy Watch 4	Watch	4	Samsung	[0.05001451075077057, 0.0]	['PRICE']
743	4/20/2024	I have bought many refi	Galaxy Watch 4	Watch	3	Samsung	[0.029614988748288569, 0.0]	['PRICE']
744	6/24/2024	It's like a slow modem.	Galaxy Watch 4	Watch	3	Samsung	[0.020081933587789536, 0.0]	['MULTIMEDIA']
745	9/19/2022	I really wanted to love th	Galaxy Watch 4	Watch	3	Samsung	[0.9923948740959167, 0.4]	['BATTERY', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
746	2/4/2024	Its working well, but th	Galaxy Watch 4	Watch	4	Samsung	[0.08469905448317932, 0.0]	['PERFORMANCE', 'PRICE']
747	4/21/2024	Although this was refu	Galaxy Watch 4	Watch	5	Samsung	[0.9035694909095764, 0.0]	['BATTERY']
748	4/14/2024	The performance of the	Galaxy Watch 4	Watch	3	Samsung	[0.9901244044303894, 0.0]	['BATTERY']
749	4/3/2024	The average could not t	Galaxy Watch 4	Watch	3	Samsung	[0.029290036813637733, 0.0]	['DISPLAY']
750	9/6/2023	I got this because I was	Galaxy Watch 4	Watch	3	Samsung	[0.07207583630645455, 0.0]	['PERFORMANCE', 'PRICE']
751	2/11/2024	The Smartwatch arrives	Galaxy Watch 4	Watch	5	Samsung	[0.9874020180082607, 0.5]	['BATTERY', 'PERFORMANCE']
752	5/8/2024	good deal.	Galaxy Watch 4	Watch	5	Samsung	[0.019171940183277, 0.0]	['PRICE']
753	3/11/2024	Bought this for my mor	Galaxy Watch 4	Watch	3	Samsung	[0.018807431030464172, 0.0]	['FITNESS & HEALTH TRACKING']
754	5/29/2024	Overall, the watch is pr	Galaxy Watch 4	Watch	4	Samsung	[0.9837729811096191, 0.1]	['BATTERY', 'MULTIMEDIA']
755	7/10/2022	This is a great watch an	Galaxy Watch 4	Watch	4	Samsung	[0.03726325674223633, 0.0]	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
756	1/2/2024	This was bought for my	Galaxy Watch 4	Watch	5	Samsung	[0.014829464018249512, 0.0]	['PRICE']
757	9/15/2022	Great watch! Looks an	Galaxy Watch 4	Watch	5	Samsung	[0.9818273186683655, 0.5]	['BATTERY', 'PERFORMANCE']
758	4/29/2024	While it is a refurbishec	Galaxy Watch 4	Watch	2	Samsung	[0.9811215400695801, 0.0]	['BATTERY']
759	8/13/2023	You may have some bat	Galaxy Watch 4	Watch	4	Samsung	[0.9652077960968018, 0.0]	['BATTERY']
760	9/25/2022	I am still very happy wit	Galaxy Watch 4	Watch	5	Samsung	[0.9761305451393127, 0.0]	['BATTERY', 'PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
761	2/14/2022	I previously had a Sams	Galaxy Watch 4	Watch	5	Samsung	[0.9810911977291107, 0.0]	['BATTERY', 'FITNESS & HEALTH TRACKING']
762	7/13/2023	The watch was shipped	Galaxy Watch 4	Watch	3	Samsung	[0.9624749422073364, 0.0]	['BATTERY']
763	9/27/2023	Bought this 2/1/2023.	Galaxy Watch 4	Watch	2	Samsung	[0.9724312424659729, 0.1]	['BATTERY', 'FITNESS & HEALTH TRACKING']
764	1/6/2024	I know it's a dark fantas	Galaxy Watch 4	Watch	5	Samsung	[0.12915563583374023, 0.0]	['MULTIMEDIA']
765	6/8/2023	Visual impairments are	Galaxy Watch 4	Watch	3	Samsung	[0.9832746386528015, 0.5]	['BATTERY', 'PERFORMANCE']
766	7/30/2022	Awesome watch and st	Galaxy Watch 4	Watch	4	Samsung	[0.9757948517799377, 0.0]	['BATTERY']
767	12/15/2023	I love that it synes flawl	Galaxy Watch 4	Watch	5	Samsung	[0.9030659389495850, 0.5]	['BATTERY', 'PERFORMANCE']
768	11/28/2023	I love this watch the ste	Galaxy Watch 4	Watch	5	Samsung	[0.23560877227783203, 0.01]	['FITNESS & HEALTH TRACKING']
769	1/8/2023	Great for tracking fitnes	Galaxy Watch 4	Watch	4	Samsung	[0.9792941212654114, 0.5]	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
770	4/24/2024	Good thing I ordered a	Galaxy Watch 4	Watch	2	Samsung	[0.9516456413269043, 0.0]	['BATTERY']

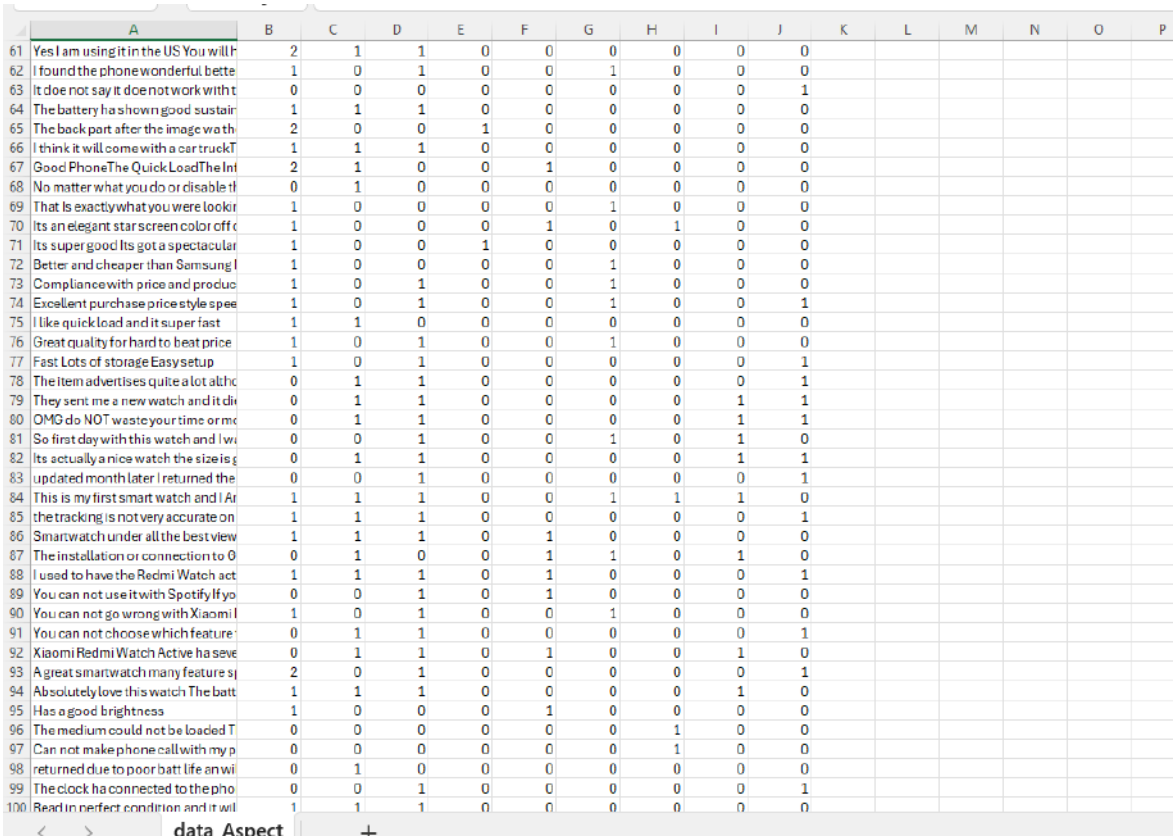


	A	B	C	D	E	F	G	H
926	4/29/2024	I had it one day charge	Galaxy Watch 4	Watch	2	Samsung	[0.9633790946096775, 0.0	['BATTERY']
927	2/9/2024	It works ok but have to	Galaxy Watch 4	Watch	3	Samsung	[0.9690668034362793, 0.6	['BATTERY', 'PERFORMANCE', 'CUSTOMIZATION']
928	8/26/2022	I absolutely love this w	Galaxy Watch 4	Watch	5	Samsung	[0.9645682573318481, 0.5	['BATTERY', 'PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
929	11/23/2023	I couldn't even tell it w	Galaxy Watch 4	Watch	5	Samsung	[0.9839127659797668, 0.7	['BATTERY', 'PERFORMANCE']
930	11/14/2023	I really like the watch ar	Galaxy Watch 4	Watch	3	Samsung	[0.9694260822057261, 0.0	['BATTERY']
931	5/30/2023	It wouldn't charge at fir	Galaxy Watch 4	Watch	4	Samsung	[0.9662852287292248, 0.07	['BATTERY']
932	1/3/2024	Bought for granddaugh	Galaxy Watch 4	Watch	5	Samsung	[0.6280896139220459478, 0	['PERFORMANCE']
933	10/8/2022	I bought this watch as a	Galaxy Watch 4	Watch	5	Samsung	[0.9870651960372925, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
934	7/6/2022	This thing has (so far) e	Galaxy Watch 4	Watch	5	Samsung	[0.8831965398025535, 0.0	['BATTERY', 'PRICE']
935	8/12/2022	This is my third Samsur	Galaxy Watch 4	Watch	5	Samsung	[0.9935500621795654, 0.1	['BATTERY', 'MULTIMEDIA']
936	9/7/2022	The Galaxy Watch 4 car	Galaxy Watch 4	Watch	4	Samsung	[0.1455671421527862 5, 0.	['PERFORMANCE', 'CUSTOMIZATION']
937	5/29/2023	I was planning to return	Galaxy Watch 4	Watch	5	Samsung	[0.9703372120857239, 0.4	['BATTERY']
938	8/16/2022	Went from the Active 2	Galaxy Watch 4	Watch	5	Samsung	[0.1280689537525177, 0.2	['FITNESS & HEALTH TRACKING']
939	12/30/2022	I bought the Galaxy Wa	Galaxy Watch 4	Watch	5	Samsung	[0.9337234497070312, 0.6	['BATTERY', 'PERFORMANCE']
940	3/5/2022	I really like this watch. I	Galaxy Watch 4	Watch	4	Samsung	[0.9451190185546875, 0.01	['BATTERY']
941	3/25/2024	This is the second watc	Galaxy Watch 4	Watch	2	Samsung	[0.7136182188987732, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
942	4/29/2023	I absolutely love this w	Galaxy Watch 4	Watch	5	Samsung	[0.0329456901550293 6, ['MULTIMEDIA', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']	
943	11/8/2022	The media could not be	Galaxy Watch 4	Watch	4	Samsung	[0.9592844211483002, 0.0	['BATTERY', 'MULTIMEDIA']
944	8/11/2023	Great gift to my wife, re	Galaxy Watch 4	Watch	5	Samsung	[0.9786593317985535, 0.5	['BATTERY', 'PERFORMANCE']
945	3/13/2023	Me and my wife, (my wi	Galaxy Watch 4	Watch	5	Samsung	[0.0798015147447586, 0.2	['MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
946	8/2/2022	It took too long for deliv	Galaxy Watch 4	Watch	4	Samsung	[0.9666634583473206 9, 0.0	['BATTERY']
947	2/15/2023	This watch has the sam	Galaxy Watch 4	Watch	5	Samsung	[0.0239748846793746 95, ['PRICE', 'FITNESS & HEALTH TRACKING']	
948	4/11/2023	I was skeptical about th	Galaxy Watch 4	Watch	5	Samsung	[0.0122949552726745 97, ['DISPLAY', 'MULTIMEDIA']	
949	9/8/2023	the mirror always want	Galaxy Watch 4	Watch	5	Samsung	[0.0225194835472106 79, 0.	['PRICE']
950	4/13/2022	The watch is AMAZING	Galaxy Watch 4	Watch	4	Samsung	[0.9595459365463257 8, 0.1	['BATTERY', 'FITNESS & HEALTH TRACKING']
951	5/19/2023	Came to us like new, pr	Galaxy Watch 4	Watch	5	Samsung	[0.9989917802810669 9, 0.1	['BATTERY']
952	10/13/2022	This is my 1st smart wa	Galaxy Watch 4	Watch	4	Samsung	[0.9337958908081055 0, 0.0	['BATTERY', 'FITNESS & HEALTH TRACKING']
953	2/25/2023	I loved the watch and it	Galaxy Watch 4	Watch	4	Samsung	[0.0068976611137390 137, ['PERFORMANCE', 'MULTIMEDIA']	
954	7/27/2022	I got the pink watch an	Galaxy Watch 4	Watch	5	Samsung	[0.9204455184936523 4, 0.96	['BATTERY', 'PERFORMANCE']
955	11/3/2023	Easy to set up and pair	Galaxy Watch 4	Watch	5	Samsung	[0.0122319889068603 52, ['PERFORMANCE']	
956	8/27/2023	I love it, the shape it car	Galaxy Watch 4	Watch	3	Samsung	[0.0394421720504760 74, ['PRICE']	
957	4/9/2024	Discharge very soon, hi	Galaxy Watch 4	Watch	2	Samsung	[0.9792560386657715 6, 0.0	['BATTERY']
958	4/26/2023	Started at 3 stars, now	Galaxy Watch 5	Watch	5	Samsung	[0.1107834815979004 75, 0.	['PERFORMANCE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
959	8/27/2024	I purchased a different	Galaxy Watch 5	Watch	5	Samsung	[0.9813909530639648 4, 0.2	['BATTERY', 'PRICE']
960	3/13/2023	My gf wanted a smartw	Galaxy Watch 5	Watch	5	Samsung	[0.9819927215576172, 0.6	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING', 'CUSTOMIZATION']
961	4/17/2023	This is my first "genuine	Galaxy Watch 5	Watch	5	Samsung	[0.9491022491455078 1, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
962	7/11/2024	As a self-proclaimed te	Galaxy Watch 5	Watch	5	Samsung	[0.9724846482276917, 0.0	['BATTERY', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING']
963	1/9/2024	Best Smartwatch I have	Galaxy Watch 5	Watch	5	Samsung	[0.9983903360366821 3, 0.5	['BATTERY', 'PERFORMANCE', 'DISPLAY', 'FITNESS & HEALTH TRACKING']
964	1/9/2023	I am technology challer	Galaxy Watch 5	Watch	5	Samsung	[0.5246580719947815 4, 0.3	['BATTERY', 'PRICE', 'FITNESS & HEALTH TRACKING']
965	7/13/2024	I gave up wearing a wat	Galaxy Watch 5	Watch	5	Samsung	[0.1286471402645111 3, 0.	['FITNESS & HEALTH TRACKING']

Appendix 2. Human Annotated dataset

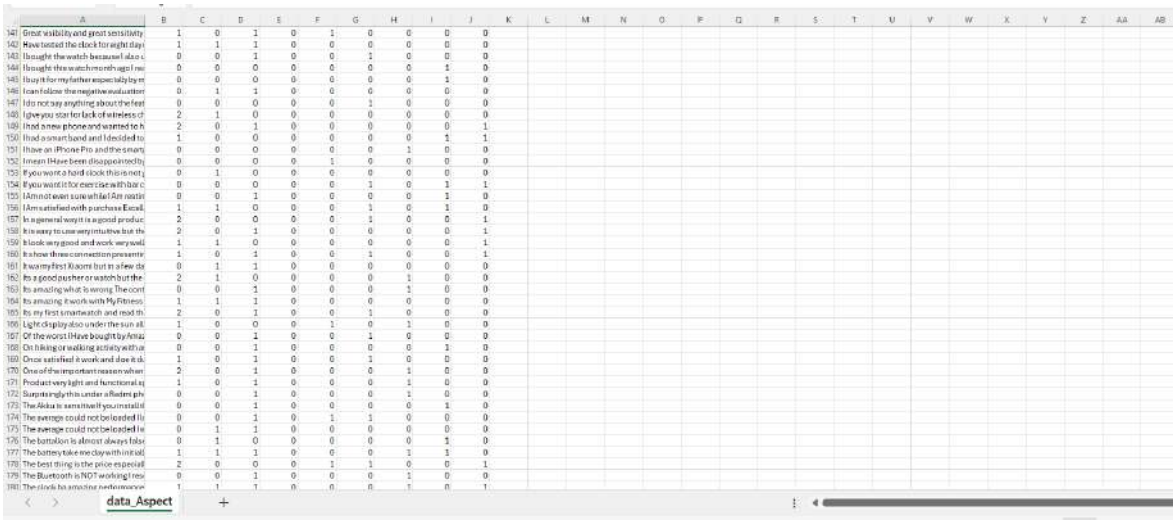


	A	B	C	D	E	F	G	H	I	J	K
1	Reviews	Sentiment	BATTERY	PERFORM	CAMERA	DISPLAY	PRICE	MULTIMEI	FITNESS &	CUSTOMIZATION	
2	Performance quality sound quality	1	0	1	0	0	1	1	0	0	
3	This is my th Xiaomi mobile and ab	1	0	1	0	0	1	0	0	0	
4	Really loved the design very fast ch	1	1	0	0	0	0	0	0	1	
5	The truth wa very good the packag	1	1	1	0	0	0	0	0	0	
6	My order arrived on time the persor	1	0	1	0	0	1	0	0	0	
7	ONE WEEK REVIEWAfter using this	1	0	1	1	1	1	0	0	1	
8	The Xiaomi Redmi Note Pro is fanta	1	1	1	1	1	1	0	0	0	
9	Camera battery life screen speed ev	1	1	1	1	1	0	0	0	1	
10	I wa previously using the Redmi not	1	1	1	1	0	1	1	0	1	
11	This is my third Xiaomi phone I like	1	0	1	1	0	1	0	0	0	
12	I liked the design and the software	1	0	1	0	0	0	0	0	1	
13	This is my second Xiaomi phone Th	1	0	1	0	1	1	0	0	1	
14	This purchase is for my child If you	2	0	0	1	1	0	0	0	0	
15	Pros Modern design camera capab	1	1	0	1	0	0	1	0	0	
16	I like this product A great phone for	1	1	0	0	1	1	1	0	0	
17	Phone came packaged brand new (	1	1	1	0	1	0	0	0	0	
18	I like that it very friendly like my prev	1	1	1	0	1	0	0	0	1	
19	The average could not be loaded T	1	1	1	0	1	0	0	0	0	
20	These phone are not original versio	0	0	1	0	0	1	0	0	0	
21	The phone is super e Xiaomi cloe n	1	0	0	1	1	1	1	0	0	
22	When my old Redmi Note Pro slowl	1	0	1	1	1	0	0	0	0	
23	In feet I only wanted to have a pulse	0	0	0	0	0	0	0	1	0	
24	In the past you have read the time a	2	0	0	0	0	0	1	1	0	
25	Since I bought it in April I am please	1	1	1	0	1	1	0	0	0	
26	I have used other xiaomi product ar	1	0	0	1	1	1	0	0	0	
27	Xiaomi continues to improve for th	1	1	0	1	1	0	0	0	0	
28	The phone is fine Before I had an pr	2	1	1	0	1	0	0	0	1	
29	I do not game or play hi revids just	0	1	0	1	1	0	0	0	1	
30	Read a message before the agreem	1	1	1	0	0	0	0	0	0	
31	I had a Redmi Note Pro wich is four	1	0	1	0	0	1	0	0	0	
32	Within a few hour of his escape he	0	0	0	0	1	0	0	0	0	
33	Fast finger print access It say It ha a	1	0	1	0	0	0	0	0	1	
34	Great phone especially for the price	1	1	1	1	1	1	0	0	0	
35	Overall an average phone Processo	2	1	1	0	0	1	0	0	0	
36	Also I wa not impressed Very good c	1	0	1	1	0	1	0	0	0	
37	The purchase by the camera and it v	2	1	0	1	0	1	0	0	0	
38	You know what this is my second R	1	0	1	1	0	0	0	0	0	
39	I love the photo of the main camera	1	1	1	1	0	1	0	0	0	
40	For a 6 cell phone RedMi delivers ex	1	1	1	0	0	1	0	0	0	

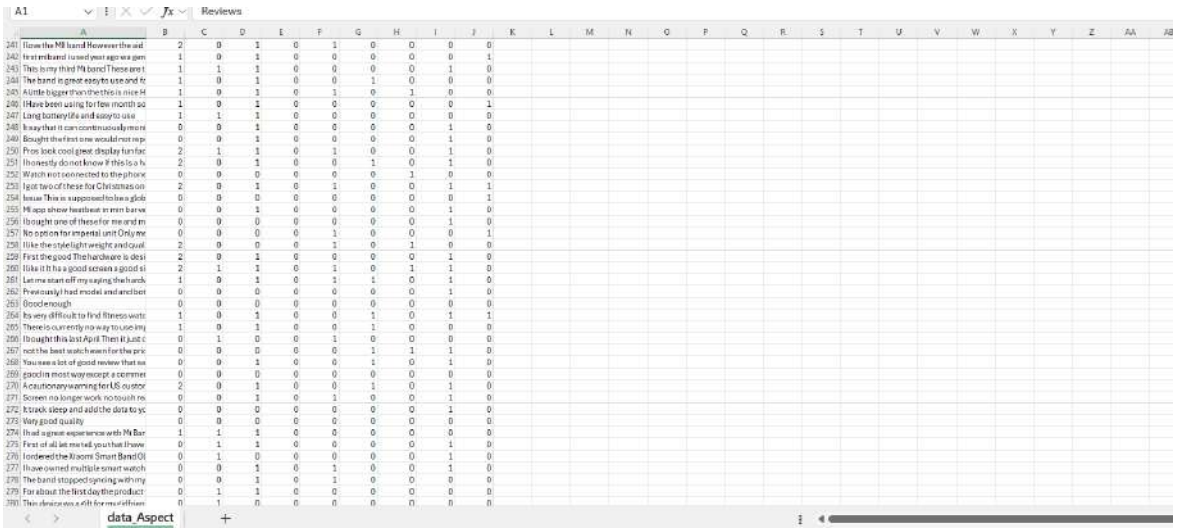


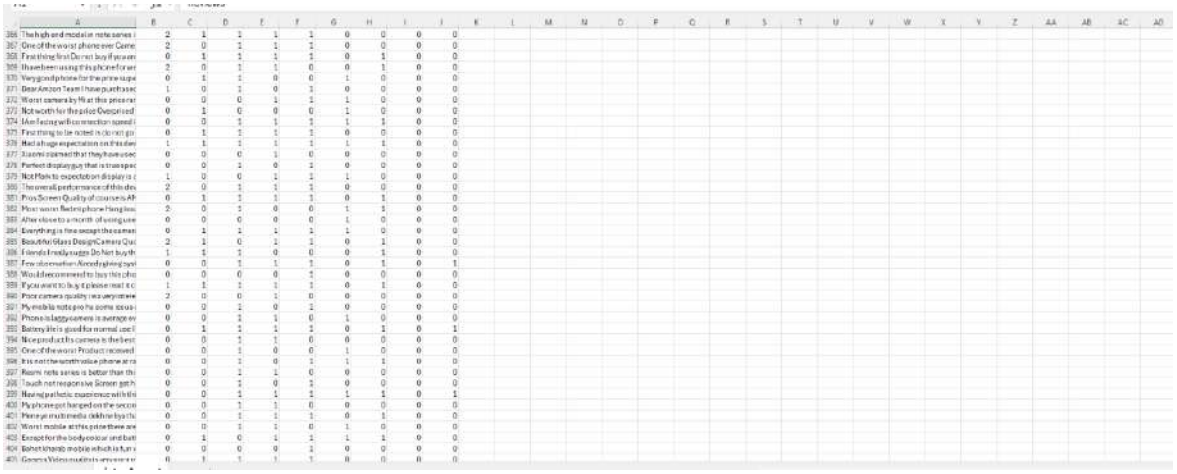
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
61	Yes I am using it in the US You will h	2	1	1	0	0	0	0	0	0						
62	I found the phone wonderful bette	1	0	1	0	0	1	0	0	0						
63	It doe not say it doe not work with t	0	0	0	0	0	0	0	0	1						
64	The battery ha shown good sustair	1	1	1	0	0	0	0	0	0						
65	The back part after the image wa th	2	0	0	1	0	0	0	0	0						
66	I think it will come with a car truckT	1	1	1	0	0	0	0	0	0						
67	Good PhoneThe Quick LoadThe Inf	2	1	0	0	1	0	0	0	0						
68	No matter what you do or disable th	0	1	0	0	0	0	0	0	0						
69	That Is exactly what you were lookir	1	0	0	0	0	1	0	0	0						
70	Its an elegant star screen color off	1	0	0	0	1	0	1	0	0						
71	Its super good Its got a spectacular	1	0	0	1	0	0	0	0	0						
72	Better and cheaper than Samsung I	1	0	0	0	0	1	0	0	0						
73	Compliance with price and produc	1	0	1	0	0	1	0	0	0						
74	Excellent purchase price style spee	1	0	1	0	0	1	0	0	1						
75	I like quick load and it super fast	1	1	0	0	0	0	0	0	0						
76	Great quality for hard to beat price	1	0	1	0	0	1	0	0	0						
77	Fast Lots of storage Easy setup	1	0	1	0	0	0	0	0	1						
78	The item advertises quite a lot alth	0	1	1	0	0	0	0	0	1						
79	They sent me a new watch and it di	0	1	1	0	0	0	0	1	1						
80	OMG do NOT waste your time or mo	0	1	1	0	0	0	0	1	1						
81	So first day with this watch and I wa	0	0	1	0	0	1	0	1	0						
82	Its actually a nice watch the size is g	0	1	1	0	0	0	0	1	1						
83	updated month later I returned the	0	0	1	0	0	0	0	0	1						
84	This is my first smart watch and I Ar	1	1	1	0	0	1	1	1	0						
85	the tracking is not very accurate on	1	1	1	0	0	0	0	0	1						
86	Smartwatch under all the best view	1	1	1	0	1	0	0	0	0						
87	The installation or connection to 0	0	1	0	0	1	1	0	1	0						
88	I used to have the Redmi Watch act	1	1	1	0	1	0	0	0	1						
89	You can not use it with Spotify If yo	0	0	1	0	1	0	0	0	0						
90	You can not go wrong with Xiaomi I	1	0	1	0	0	1	0	0	0						
91	You can not choose which feature	0	1	1	0	0	0	0	0	1						
92	Xiaomi Redmi Watch Active ha seve	0	1	1	0	1	0	0	1	0						
93	A great smartwatch many feature s	2	0	1	0	0	0	0	0	1						
94	Absolutely love this watch The batt	1	1	1	0	0	0	0	1	0						
95	Has a good brightness	1	0	0	0	1	0	0	0	0						
96	The medium could not be loaded T	0	0	0	0	0	0	1	0	0						
97	Can not make phone call with my p	0	0	0	0	0	0	1	0	0						
98	returned due to poor batt life an wil	0	1	0	0	0	0	0	0	0						
99	The clock ha connected to the pho	0	0	1	0	0	0	0	0	1						
100	Read in perfect condition and it wil	1	1	1	0	0	0	0	0	0						

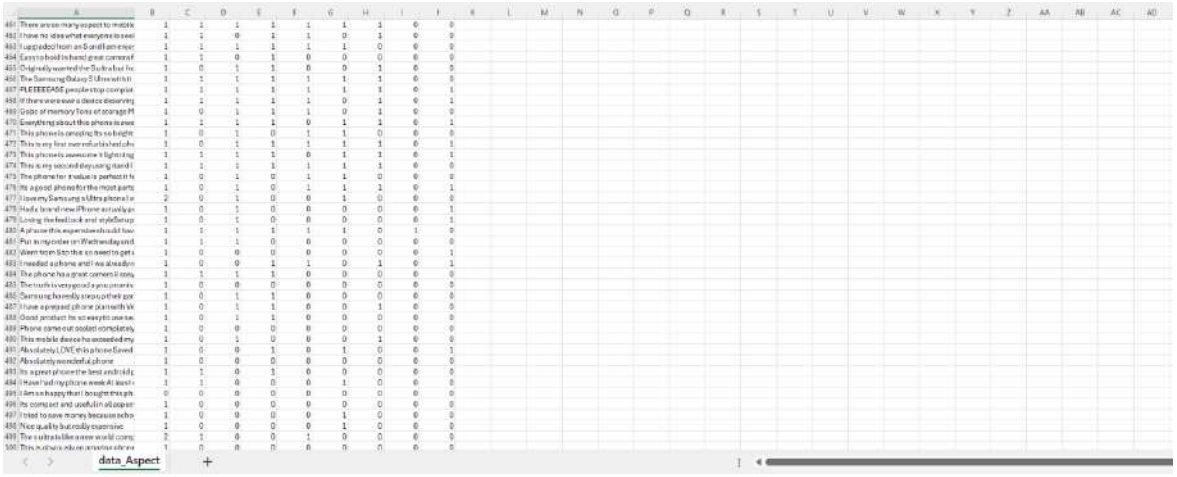
data_Aspect +

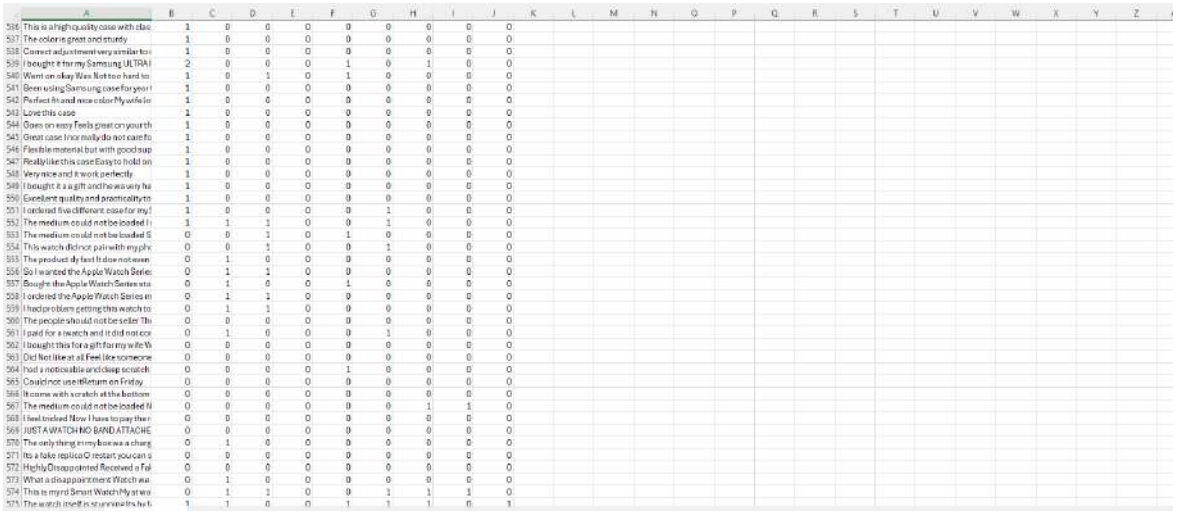


data_Aspect +

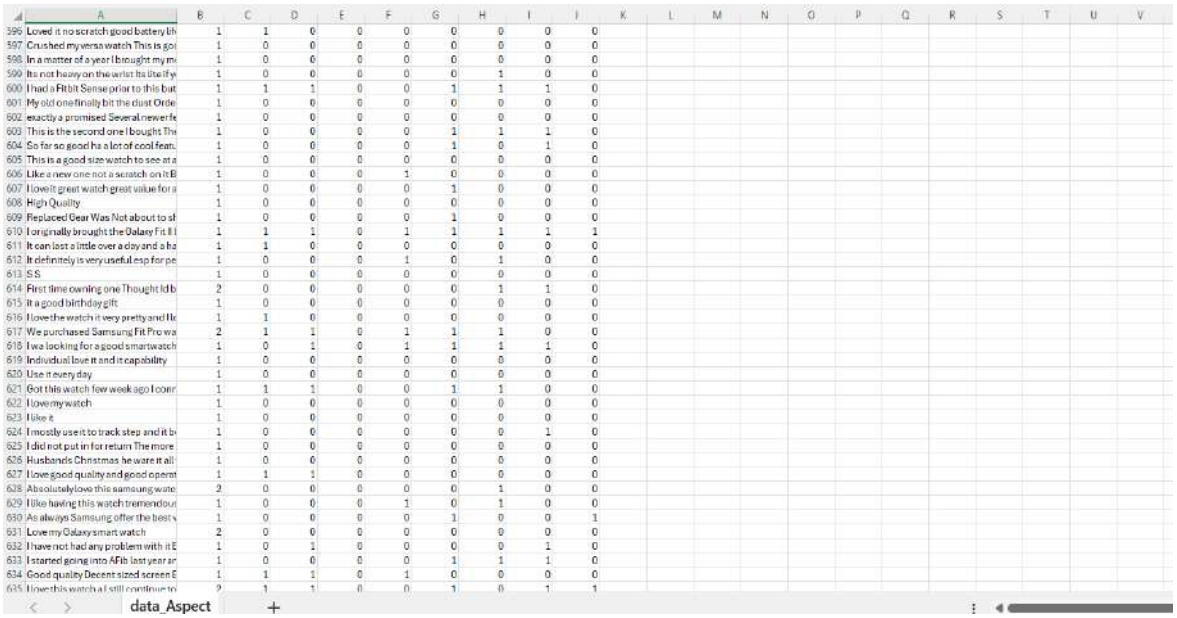

A1
Reviews
data_Aspect




data_Aspect

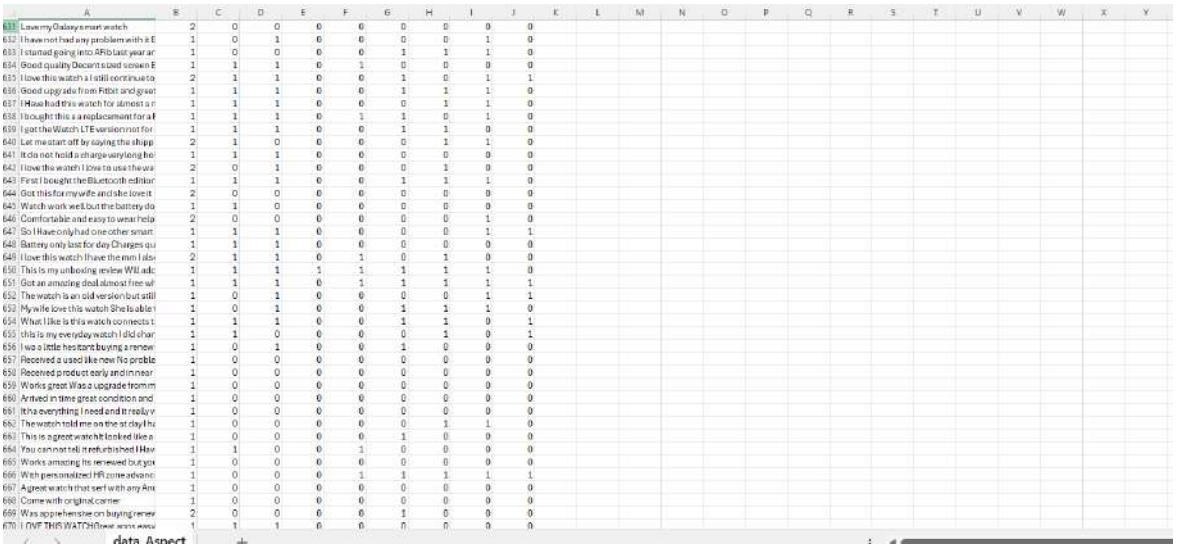


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
536	This is a high quality case with clas	1	0	0	0	0	0	0	0	0																
537	The color is great and sturdy	1	0	0	0	0	0	0	0	0																
538	Correct adjustment very similar to	1	0	0	0	0	0	0	0	0																
539	I bought it for my Samsung ULTRA	2	0	0	0	1	0	1	0	0																
540	Went on okay Was flat to e hard to	1	0	1	0	1	0	0	0	0																
541	Been using Samsung case for year t	1	0	0	0	0	0	0	0	0																
542	Perfect fit and nice color My wife lo	1	0	0	0	0	0	0	0	0																
543	Love this case	1	0	0	0	0	0	0	0	0																
544	Goes on easy Feels great on your th	1	0	0	0	0	0	0	0	0																
545	Great case I normally do not care fo	1	0	0	0	0	0	0	0	0																
546	Flexible material but with good sup	1	0	0	0	0	0	0	0	0																
547	Really like this case Easy to hold on	1	0	0	0	0	0	0	0	0																
548	Very nice and it work perfectly	1	0	0	0	0	0	0	0	0																
549	I bought it a a gift and he was very ha	1	0	0	0	0	0	0	0	0																
550	Excellent quality and practicality to	1	0	0	0	0	0	0	0	0																
551	I ordered five different case for my	1	0	0	0	0	1	0	0	0																
552	The medium could not be loaded I	1	1	1	0	0	1	0	0	0																
553	The medium could not be loaded S	0	0	1	0	1	0	0	0	0																
554	This watch did not pair with my phy	0	0	1	0	0	1	0	0	0																
555	The product dy fast It dow not wan	0	1	0	0	0	0	0	0	0																
556	So I wanted the Apple Watch Series	0	1	1	0	0	0	0	0	0																
557	Bought the Apple Watch Series xta	0	1	0	0	1	0	0	0	0																
558	I ordered the Apple Watch Series m	0	1	1	0	0	0	0	0	0																
559	I had problem getting this watch to	0	1	1	0	0	0	0	0	0																
560	The people should not be seller Th	0	0	0	0	0	0	0	0	0																
561	I paid for a swatch and it did not co	0	1	0	0	0	1	0	0	0																
562	I bought this for a gift for my wife W	0	0	0	0	0	0	0	0	0																
563	Did Not like at all Feel like someone	0	0	0	0	0	0	0	0	0																
564	had a noticeable and deep scratch	0	0	0	0	1	0	0	0	0																
565	Could not use it Return on Friday	0	0	0	0	0	0	0	0	0																
566	It come with a scratch at the bottom	0	0	0	0	0	0	0	0	0																
567	The medium could not be loaded N	0	0	0	0	0	0	1	1	0																
568	I feel tricked Now I have to pay the	0	0	0	0	0	0	0	0	0																
569	JUST A WATCH NO BAND ATTACHE	0	0	0	0	0	0	0	0	0																
570	The only thing is my box was a charg	0	1	0	0	0	0	0	0	0																
571	Its a fake replica Q restart you can s	0	0	0	0	0	0	0	0	0																
572	Highly Disappointed Received a Fak	0	0	0	0	0	0	0	0	0																
573	What a disappointment Watch wa	0	1	0	0	0	0	0	0	0																
574	This is myrd Smart Watch My at wa	0	1	1	0	0	1	1	1	0																
575	The watch itself is stunning Its hyb	1	1	0	0	1	1	1	0	1																



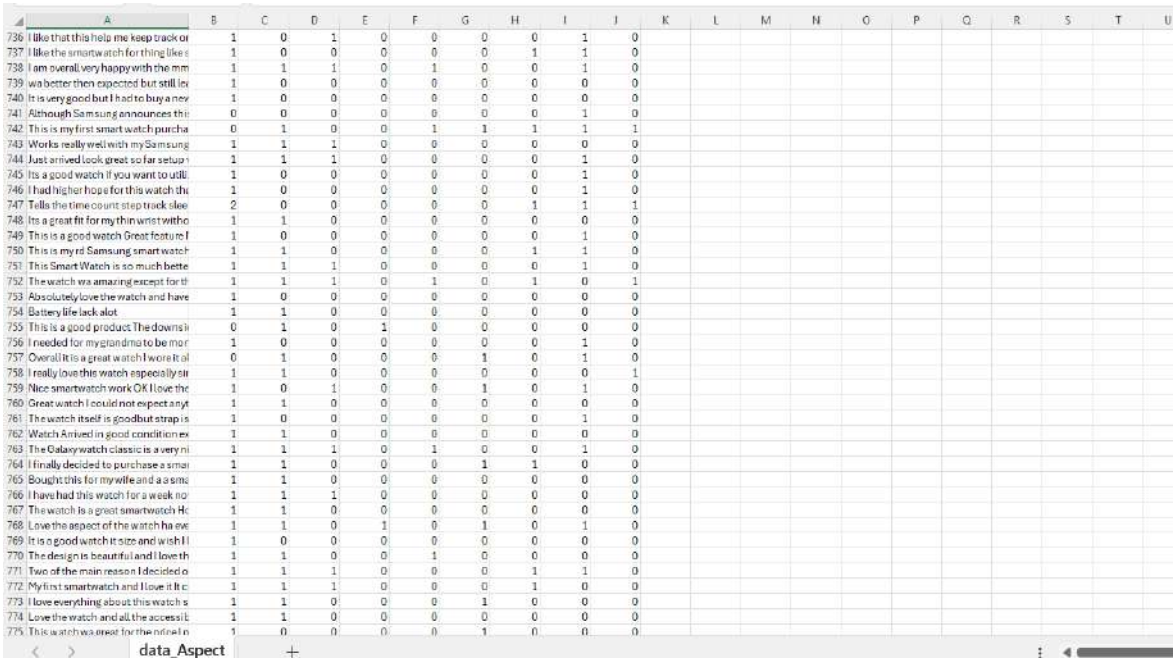
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
596	Loved it no scratch good battery lif	1	1	0	0	0	0	0	0	0												
597	Crushed my versa watch This is go	1	0	0	0	0	0	0	0	0												
598	In a matter of a year I brought my m	1	0	0	0	0	0	0	0	0												
599	Its not heavy on the wrist Its like if y	1	0	0	0	0	0	1	0	0												
600	I had a Fitbit Sense prior to this bat	1	1	1	0	0	1	1	1	0												
601	My old one finally bit the dust Orde	1	0	0	0	0	0	0	0	0												
602	exactly a promised Several new wfe	1	0	0	0	0	0	0	0	0												
603	This is the second one I bought The	1	0	0	0	0	1	1	1	0												
604	So far so good ha a lot of cool featu	1	0	0	0	0	1	0	1	0												
605	This is a good size watch to see at a	1	0	0	0	0	0	0	0	0												
606	Like a new one not a scratch on it B	1	0	0	0	1	0	0	0	0												
607	I love it great watch great value for a	1	0	0	0	0	1	0	0	0												
608	High Quality	1	0	0	0	0	0	0	0	0												
609	Replaced Gear Was Not about to sh	1	0	0	0	0	1	0	0	0												
610	I originally brought the Galaxy Fit II I	1	1	1	0	1	1	1	1	1												
611	It can last a little over a day and a ha	1	1	0	0	0	0	0	0	0												
612	It definitely is very useful esp for pe	1	0	0	0	1	0	1	0	0												
613	S S	1	0	0	0	0	0	0	0	0												
614	First time owning one Thought Id b	2	0	0	0	0	0	1	1	0												
615	It a good birthday gift	1	0	0	0	0	0	0	0	0												
616	I love the watch it very pretty and Ho	1	1	0	0	0	0	0	0	0												
617	We purchased Samsung Fit Pro wa	2	1	1	0	1	1	1	0	0												
618	I wa looking for a good smart watch	1	0	1	0	1	1	1	1	0												
619	Individual love it and it capability	1	0	0	0	0	0	0	0	0												
620	Use it every day	1	0	0	0	0	0	0	0	0												
621	Got this watch few week ago I conn	1	1	1	0	0	1	1	0	0												
622	I love my watch	1	0	0	0	0	0	0	0	0												
623	I like it	1	0	0	0	0	0	0	0	0												
624	I mostly use it to track step and it b	1	0	0	0	0	0	0	1	0												
625	I did not put in for return The more	1	0	0	0	0	0	0	0	0												
626	Husbands Christmas he ware it all	1	0	0	0	0	0	0	0	0												
627	I love good quality and good operat	1	1	1	0	0	0	0	0	0												
628	Absolutely love this samsung wate	2	0	0	0	0	0	1	0	0												
629	I like having this watch tremendou	1	0	0	0	1	0	1	0	0												
630	As always Samsung offer the best v	1	0	0	0	0	1	0	0	1												
631	Love my Galaxy smart watch	2	0	0	0	0	0	0	0	0												
632	I have not had any problem with it E	1	0	1	0	0	0	0	1	0												
633	I started going into AFib last year ar	1	0	0	0	0	1	1	1	0												
634	Good quality Decent sized screen E	1	1	1	0	1	0	0	0	0												
635	I love this watch a I still continue to	2	1	1	0	0	1	0	1	1												

data_Aspect



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
631	Love my Galaxy smart watch	2	0	0	0	0	0	0	0	0															
632	I have not had any problem with it E	1	0	1	0	0	0	0	1	0															
633	I started going into AFib last year ar	1	0	0	0	0	1	1	1	0															
634	Good quality Decent sized screen E	1	1	1	0	1	0	0	0	0															
635	I love this watch a I still continue to	2	1	1	0	0	1	0	1	1															
636	Good upgrade from Fitbit and great	1	1	1	0	0	1	1	1	0															
637	I Have had this watch for almost a m	1	1	1	0	0	0	1	1	0															
638	I bought this a a replacement for a f	1	1	1	0	1	1	0	1	0															
639	I got the Watch LTE version not for	1	1	1	0	0	1	1	0	0															
640	Let me start off by saying the shipp	2	1	0	0	0	0	1	1	0															
641	It do not hold a charge very long ho	1	1	1	0	0	0	0	0	0															
642	I love this watch I love to use the wa	2	0	1	0	0	0	1	0	0															
643	First I bought the Bluetooth edition	1	1	1	0	0	1	1	1	0															
644	Got this for my wife and she love it	2	0	0	0	0	0	0	0	0															
645	Watch work well but the battery do	1	1	0	0	0	0	0	0	0															
646	Comfortable and easy to wear Hela	2	0	0	0	0	0	0	1	0															
647	So I Have only had one other smart	1	1	1	0	0	0	0	1	1															
648	Battery only last for day Charges qu	1	1	1	0	0	0	0	0	0															
649	I love this watch I have the mm I als	2	1	1	0	1	0	1	0	0															
650	This is my unboxing review Will add	1	1	1	1	1	1	1	1	0															
651	Got an amazing deal almost free wh	1	1	1	0	1	1	1	1	1															
652	The watch is an old version but still	1	0	1	0	0	0	0	1	1															
653	My wife love this watch She is able t	1	0	1	0	0	1	1	1	0															
654	What I like is this watch connects t	1	1	1	0	0	1	1	0	1															
655	this is my everyday watch I did char	1	1	0	0	0	0	1	0	1															
656	I wa a little hesitant buying a renew	1	0	1	0	0	1	0	0	0															
657	Received a used like new No proble	1	0	0	0	0	0	0	0	0															
658	Received product early and in near	1	0	0	0	0	0	0	0	0															
659	Works great Was a upgrade from m	1	0	0	0	0	0	0	0	0															
660	Arrived in time great condition and	1	0	0	0	0	0	0	0	0															
661	It ha everything I need and it really w	1	0	0	0	0	0	0	0	0															
662	The watch told me on the st day I ha	1	0	0	0	0	0	1	1	0															
663	This is a great watch It looked like a	1	0	0	0	0	1	0	0	0															
664	You can not tell it refurbished I Hav	1	1	0	0	1	0	0	0	0															
665	Works amazing Its renewed but you	1	0	0	0	0	0	0	0	0															
666	With personalized HR zone advanc	1	0	0	0	1	1	1	1	1															
667	A great watch that serf with any An	1	0	0	0	0	0	0	0	0															
668	Come with original carrier	1	0	0	0	0	0	0	0	0															
669	Was apprehensive on buying renew	2	0	0	0	0	1	0	0	0															
670	I LOVE THIS WATCH love apps awa	1	1	1	0	0	0	0	0	0															

data_Aspect



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
736	I like that this help me keep track or	1	0	1	0	0	0	0	1	0											
737	I like the smartwatch for thing like s	1	0	0	0	0	0	1	1	0											
738	I am overall very happy with the mm	1	1	1	0	1	0	0	1	0											
739	wa better then expected but still lea	1	0	0	0	0	0	0	0	0											
740	It is very good but I had to buy a nev	1	0	0	0	0	0	0	0	0											
741	Although Samsung announces thi	0	0	0	0	0	0	0	1	0											
742	This is my first smart watch purcha	0	1	0	0	1	1	1	1	1											
743	Works really well with my Samsung	1	1	1	0	0	0	0	0	0											
744	Just arrived look great so far setup	1	1	1	0	0	0	0	1	0											
745	Its a good watch if you want to utili	1	0	0	0	0	0	0	1	0											
746	I had higher hope for this watch thi	1	0	0	0	0	0	0	1	0											
747	Tells the time count step track slee	2	0	0	0	0	0	1	1	1											
748	Its a great fit for my thin wrist witho	1	1	0	0	0	0	0	0	0											
749	This is a good watch Great feature I	1	0	0	0	0	0	0	1	0											
750	This is my rd Samsung smart watch	1	1	0	0	0	0	1	1	0											
751	This Smart Watch is so much bette	1	1	1	0	0	0	0	1	0											
752	The watch wa amazing except for th	1	1	1	0	1	0	1	0	1											
753	Absolutely love the watch and have	1	0	0	0	0	0	0	0	0											
754	Battery life lack alot	1	1	0	0	0	0	0	0	0											
755	This is a good product The downs li	0	1	0	1	0	0	0	0	0											
756	I needed for my grandma to be mor	1	0	0	0	0	0	0	1	0											
757	Overall it is a great watch I wore it al	0	1	0	0	0	1	0	1	0											
758	I really love this watch especially sir	1	1	0	0	0	0	0	0	1											
759	Nice smartwatch work OK I love the	1	0	1	0	0	1	0	1	0											
760	Great watch I could not expect anyt	1	1	0	0	0	0	0	0	0											
761	The watch itself is goodbut strap is	1	0	0	0	0	0	0	1	0											
762	Watch Arrived in good condition ex	1	1	0	0	0	0	0	0	0											
763	The Galaxy watch classic is a very ni	1	1	1	0	1	0	0	1	0											
764	I finally decided to purchase a smar	1	1	0	0	0	1	1	0	0											
765	Bought this for my wife and a a sma	1	1	0	0	0	0	0	0	0											
766	I have had this watch for a week no	1	1	1	0	0	0	0	0	0											
767	The watch is a great smartwatch Hc	1	1	0	0	0	0	0	0	0											
768	Love the aspect of the watch ha eve	1	1	0	1	0	1	0	1	0											
769	It is a good watch it size and wish I I	1	0	0	0	0	0	0	0	0											
770	The design is beautiful and I love th	1	1	0	0	1	0	0	0	0											
771	Two of the main reason I decided o	1	1	1	0	0	0	1	1	0											
772	My first smartwatch and I love it It c	1	1	1	0	0	0	1	0	0											
773	I love everything about this watch s	1	1	0	0	0	1	0	0	0											
774	Love the watch and all the accessib	1	1	0	0	0	0	0	0	0											
775	This watch wa great for the price I o	1	0	0	0	0	1	0	0	0											

data_Aspect

Appendix 3. Remove Punctuation & Special Character Removal

```
# Step 2: Data Cleaning Function
def clean_text(text):
    # Handle non-string values (e.g., NaN or float)
    if not isinstance(text, str):
        return ""

    # Remove URLs
    text = re.sub(r'http\S+|www\S+|https\S+', '', text, flags=re.MULTILINE)
    # Remove HTML tags
    text = re.sub(r'<.*?>', '', text)
    text = re.sub(r'<.*?ðŸ•ðŸ«ðŸ¤ðŸ€™>', '', text)

    # Remove special characters and numbers, keep letters and spaces
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = re.sub(r'[ðŸ•ðŸ’ðŸðŸ«ðŸ¤ðŸ€™]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[•]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[ðŸ]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[Ÿ]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[ð]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[¤]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[»]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'\s+', ' ', text).strip()  # Normalize whitespace
    text = re.sub(r'[..]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[€™]', '', text)  # Remove specific symbols/emojis
    text = re.sub(r'[]', '', text)  # Remove specific symbols/emojis



    # Remove extra whitespace
    text = " ".join(text.split())
```

Appendix 5. Fine-tuning the model for Multi Classification (RoBERTa)

```
!pip install -U accelerate
!pip install --upgrade transformers
```

```
Requirement already satisfied: accelerate in /usr/local/lib/python3.11/dist-packages (1.6.0)
Requirement already satisfied: numpy<3.0.0,>=1.17 in /usr/local/lib/python3.11/dist-packages (from accelerate) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from accelerate) (24.2)
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from accelerate) (7.0.0)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.11/dist-packages (from accelerate) (6.0.2)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from accelerate) (2.5.1+cu124)
Requirement already satisfied: huggingface-hub>=0.21.0 in /usr/local/lib/python3.11/dist-packages (from accelerate) (0.30.2)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from accelerate) (0.5.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.21.0->accelerate) (3.18.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.21.0->accelerate) (2025.3.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.21.0->accelerate) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.21.0->accelerate) (4.67.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub>=0.21.0->accelerate) (4.13.1)
Requirement already satisfied: mkl_fft in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (1.3.8)
Requirement already satisfied: mkl_random in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (1.2.4)
Requirement already satisfied: mkl_umath in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (0.1.1)
Requirement already satisfied: mkl in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (2025.1.0)
Requirement already satisfied: tbb4py in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (2022.1.0)
Requirement already satisfied: mkl-service in /usr/local/lib/python3.11/dist-packages (from numpy<3.0.0,>=1.17->accelerate) (2.4.1)
Requirement already satisfied: networkx in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (3.4.2)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (12.4.127)
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.4.127 in /usr/local/lib/python3.11/dist-packages (from torch>=2.0.0->accelerate) (12.4.127)
```

[137]:
```
import os
import pandas as pd
import numpy as np
import shutil
import sys
import tqdm.notebook as tq
from collections import defaultdict
import gdown

import torch
import torch.nn as nn

device = torch.device('cuda') if torch.cuda.is_available() else torch.device('cpu')
```

[138]:
```
file_id = '1byKaRJkESIIf4-AsTVJdBo0E3yYngCOg'
url = f"https://drive.google.com/uc?id={file_id}"

# Download the file
output = "data.csv"
gdown.download(url, output, quiet=False)

# Load the dataset
df = pd.read_csv(output)
```

```
Downloading...
From: https://drive.google.com/uc?id=1byKaRJkESIIf4-AsTVJdBo0E3yYngCOg
To: /kaggle/working/data.csv
100%|██████████| 691k/691k [00:00<00:00, 90.0MB/s]
```

[139]:
```
df
```

	Reviews	BATTERY	PERFORMANCE	CAMERA	DISPLAY	PRICE	MULTIMEDIA	FITNESS & HEALTH TRACKING	CUSTOMIZATION
0	Porformance quality sound quality and value for...	0	1	0	0	1	1	0	0
1	This is my th Xiaomi mobile and absolutely LOV...	0	1	0	0	1	0	0	0
2	Really loved the design very fast charging I A...	1	0	0	0	0	0	0	1
3	The truth wa very good the package arrived a d...	1	1	0	0	0	0	0	0
4	My order arrived on time the person who made t...	0	1	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...
2243	This is my first Apple Watch and needless to s...	1	0	0	0	1	0	0	1
2244	it got a brilliant battery life which is most ...	1	0	0	0	1	0	1	1
2245	This is the second watch face cover I have rec...	0	0	0	1	0	0	0	1
2246	Bezel is a perfect fit for the Ultra provides ...	0	0	0	0	0	0	0	1
2247	Love this The only thing holding back that fit...	0	0	0	1	0	0	0	1

2248 rows × 9 columns

[140]:
```
Aspect_columns = ['BATTERY', 'PERFORMANCE', 'CAMERA', 'DISPLAY', 'PRICE', 'MULTIMEDIA', 'FITNESS & HEALTH TRACKING',
                  'CUSTOMIZATION']
```

```
df[Aspect_columns] = df[Aspect_columns].astype(int)
```

0	Performance quality sound quality and value for...	0	1	0	0	1	1	0	0
1	This is my th Xiaomi mobile and absolutely UOV...	0	1	0	0	1	0	0	0
2	Really loved the design very fast charging I A...	1	0	0	0	0	0	0	1
3	The truth wa very good the package arrived a d...	1	1	0	0	0	0	0	0
4	My order arrived on time the person who made t...	0	1	0	0	1	0	0	0
...	...	...	...	...	...	...	...	...	...
2243	This is my first Apple Watch and needless to s...	1	0	0	0	1	0	0	1
2244	It got a brilliant battery life which is most ...	1	0	0	0	1	0	1	1
2245	This is the second watch face cover I have rec...	0	0	0	1	0	0	0	1
2246	Bece is a perfect fit for the Ultra provides ...	0	0	0	0	0	0	0	1
2247	Love this The only thing holding back that fit...	0	0	0	1	0	0	0	1

2248 rows × 9 columns

+ Code + Markdown

```
[143]: df.to_csv('data_prep.csv', index=False)

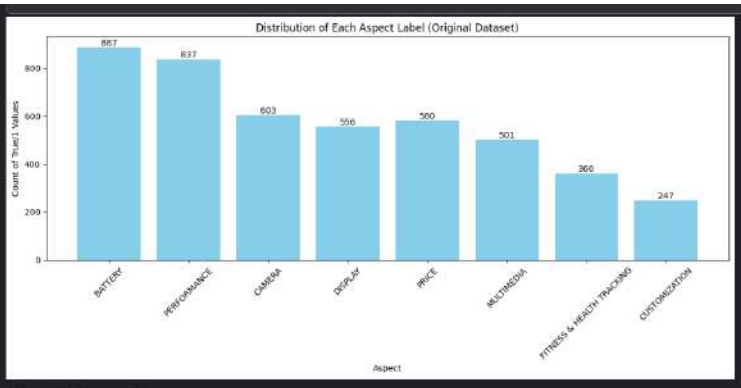
[144]: emotion_counts = df[emotion_columns].sum()

[145]: import matplotlib.pyplot as plt

[146]: emotion_counts = df.iloc[:, 1:].sum()  # Menghitung jumlah nilai True/1 untuk setiap label emosi

plt.figure(figsize=(12, 6))
bars = plt.bar(emotion_counts.index, emotion_counts.values, color='skyblue')

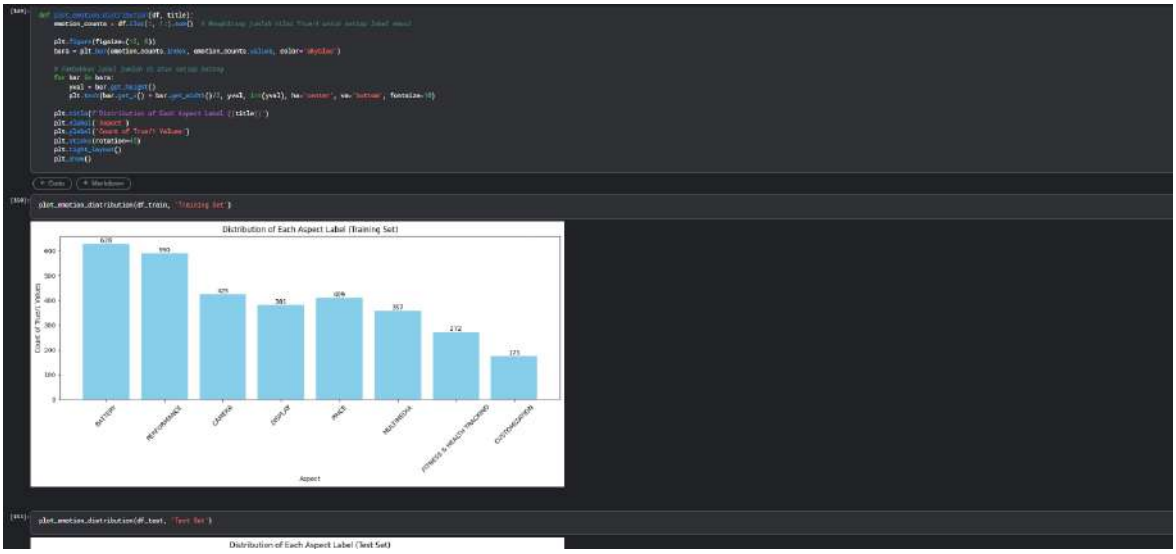
# Tambahkan label jumlah di atas setiap batang
```

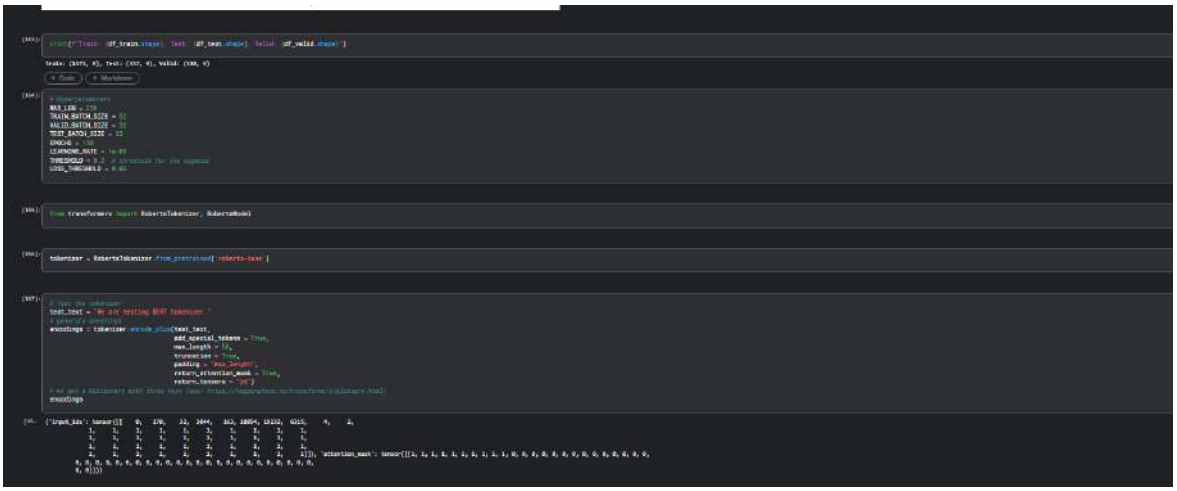


+ Code + Markdown

```
[47]: from sklearn.model_selection import train_test_split
# split into train and test
df_train, df_test = train_test_split(df, random_state=77, test_size=0.30, shuffle=True)
# split test into test and validation datasets
df_test, df_valid = train_test_split(df_test, random_state=88, test_size=0.50, shuffle=True)

[48]: df_train.to_csv('data_train_Aspect.csv', index=False)
df_test.to_csv('data_test_Aspect.csv', index=False)
```





```
[103]: print(f'Train: {df_train.shape}, Test: {df_test.shape}, Valid: {df_valid.shape}')

Train: (5575, 4), Test: (697, 4), Valid: (698, 4)

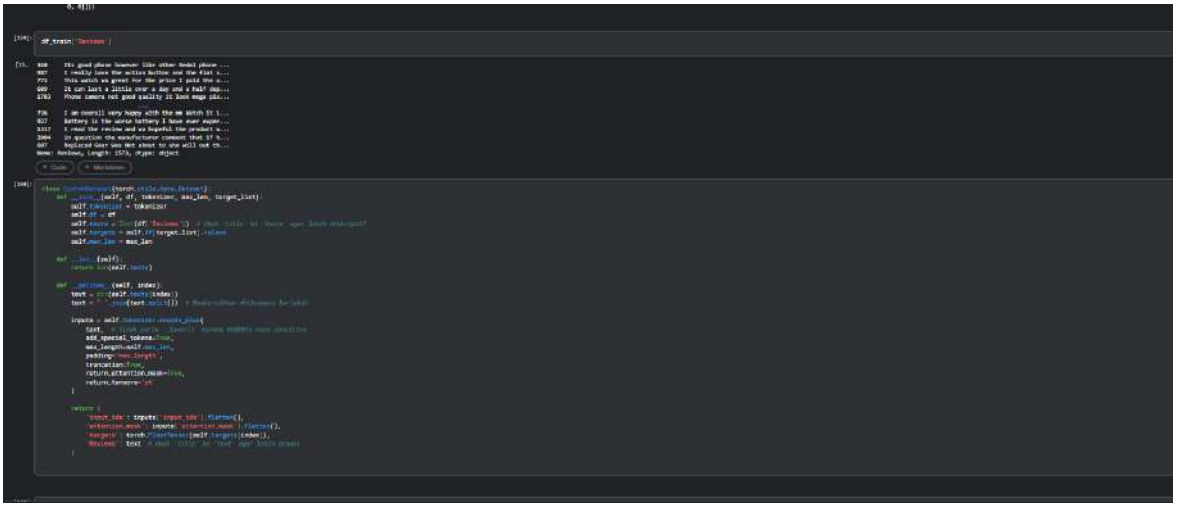
[104]: # Hyperparameters
MAX_LEN = 256
TRAIN_BATCH_SIZE = 32
VALID_BATCH_SIZE = 32
TEST_BATCH_SIZE = 32
EPOCHS = 100
LEARNING_RATE = 1e-05
THRESHOLD = 0.2 # threshold for the sigmoid
LOSS_THRESHOLD = 0.03

[105]: from transformers import RobertaTokenizer, RobertaModel

[106]: tokenizer = RobertaTokenizer.from_pretrained('roberta-base')

[107]: # Test the tokenizer
test_text = 'We are testing BERT tokenizer.'
# encoding sentence
encodings = tokenizer.encode_plus(test_text,
                                  add_special_tokens = True,
                                  max_length = 12,
                                  truncation = True,
                                  padding = 'max_length',
                                  return_attention_mask = True,
                                  return_tensors = 'pt')
# we get a dictionary with three keys (see: https://huggingface.co/transformers/glossary.html)
encodings

[14]: {'input_ids': tensor([[    0,   170,    32,  3644,   163, 18854, 10310,  6315,     4,     2,
                             1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                             1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                             1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                             2,     1,     1,     1,     1,     2,     1,     1,     1,     1]]), 'attention_mask': tensor([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
         0, 0]])}
```



```
[108]: df_train['Reviews']

[15.  808    this good phone however like other Nokia phone ...
      867    I really love the action button and the flat s...
      773    this watch as great for the price I paid the e...
      609    It can last a little over a day and a half dep...
      1783   Phone camera not good quality it look mega pix...

      736    I am overall very happy with the HR Watch It i...
      827    Battery is the worst battery I have ever exper...
      3317   I read the review and as hopeful the product w...
      3004   In question the manufacturer comment that if h...
      687    Replaced Gear was Not about to she will out th...
      Name: Reviews, Length: 1575, dtype: object

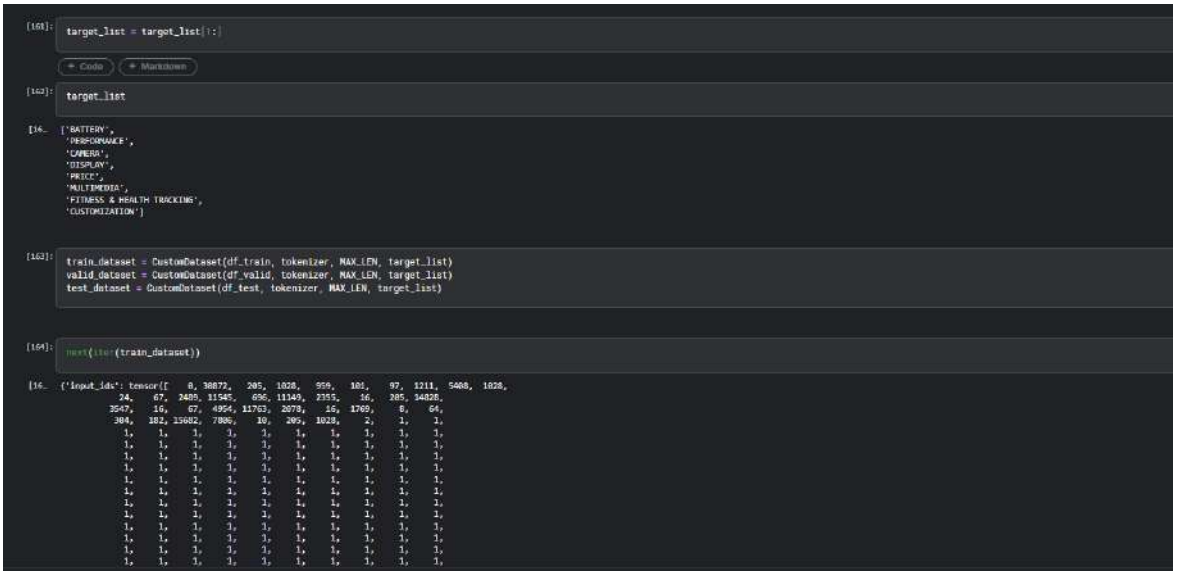
[110]: class CustomDataset(torch.utils.data.Dataset):
           def __init__(self, df, tokenizer, max_len, target_list):
               self.tokenizer = tokenizer
               self.df = df
               self.texts = list(df['Reviews']) # Check 'title' or 'texts' again before moving on!
               self.targets = self.df[target_list].values
               self.max_len = max_len

           def __len__(self):
               return len(self.texts)

           def __getitem__(self, index):
               text = str(self.texts[index])
               text = " ".join(text.split()) # Removing/Normalizing whitespaces for BERT

               inputs = self.tokenizer.encode_plus(
                   text,  # Flush parts 'texts' system WARNING once sensitive
                   add_special_tokens=True,
                   max_length=self.max_len,
                   padding='max_length',
                   truncation=True,
                   return_attention_mask=True,
                   return_tensors='pt'
               )

               return {
                   'input_ids' : inputs['input_ids'].flatten(),
                   'attention_mask' : inputs['attention_mask'].flatten(),
                   'targets' : torch.FloatTensor(self.targets[index]),
                   'Reviews' : text # Check 'title' or 'text' again before moving on!
               }
```



```
[168]: target_list = target_list[1:]

       + Code   + Markdown

[162]: target_list

[16.  ['BATTERY',
       'PERFORMANCE',
       'CAMERA',
       'DISPLAY',
       'PRICE',
       'MULTIMEDIA',
       'FITNESS & HEALTH TRACKING',
       'CUSTOMIZATION']

[163]: train_dataset = CustomDataset(df_train, tokenizer, MAX_LEN, target_list)
       valid_dataset = CustomDataset(df_valid, tokenizer, MAX_LEN, target_list)
       test_dataset = CustomDataset(df_test, tokenizer, MAX_LEN, target_list)

[164]: next(iter(train_dataset))

[16.  {'input_ids': tensor([    0, 30872,   205,  1028,   950,   101,    97,  1211,  5408,  1028,
                              24,    67,  2485, 11545,   696, 11149,  2355,    16,   205, 34828,
                            3547,    10,    67,  4954, 11763,  2078,    16,  1769,     8,    64,
                             304,   182, 15692,  7806,    10,   205,  1028,     2,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
                               1,     1,     1,     1,     1,     1,     1,     1,     1,     1,
```

[165]:
```
# Data loaders
train_data_loader = torch.utils.data.DataLoader(train_dataset,
    batch_size=TRAIN_BATCH_SIZE,
    shuffle=True,
    num_workers=0
)

val_data_loader = torch.utils.data.DataLoader(valid_dataset,
    batch_size=VALID_BATCH_SIZE,
    shuffle=False,
    num_workers=0
)

test_data_loader = torch.utils.data.DataLoader(test_dataset,
    batch_size=TEST_BATCH_SIZE,
    shuffle=False,
    num_workers=0
)
```

[166]:
```
class RoBERTaClass(torch.nn.Module):
    def __init__(self):
        super(RoBERTaClass, self).__init__()
        self.roberta_model = RobertaModel.from_pretrained('roberta-base', return_dict=True)
        self.dropout = torch.nn.Dropout(0.3)
        self.linear = torch.nn.Linear(768, 8)

    def forward(self, input_ids, attn_mask):
        output = self.roberta_model(
            input_ids,
            attention_mask=attn_mask
        )
        output_dropout = self.dropout(output.pooler_output)
        output = self.linear(output_dropout)
        return output

model = RoBERTaClass()
model.to(device)
```

```
Some weights of RobertaModel were not initialized from the model checkpoint at roberta-base and are newly initialized: ['pooler.dense.bias', 'pooler.dense.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

[167]:
```
# BCEWithLogitsLoss combines a Sigmoid layer and the BCELoss in one single class.
# This version is more numerically stable than using a plain Sigmoid followed
# by a BCELoss as, by combining the operations into one layer,
# we take advantage of the log-sum-exp trick for numerical stability.
def loss_fn(outputs, targets):
    return torch.nn.BCEWithLogitsLoss()(outputs, targets)
```

[168]:
```
from torch.optim import AdamW

# define the optimizer
optimizer = AdamW(model.parameters(), lr = 1e-5)
```

[169]:
```
# Training of the model for one epoch
def train_model(training_loader, model, optimizer):

    losses = []
    correct_predictions = 0
    num_samples = 0

    # Set model to training mode (activate dropout, batch norm)
    model.train()

    # Initialize the progress bar
    loop = tq.tqdm(enumerate(training_loader), total=len(training_loader),
                   leave=True, colour='steelblue')

    for batch_idx, data in loop:
        ids = data['input_ids'].to(device, dtype=torch.long)
        mask = data['attention_mask'].to(device, dtype=torch.long)
        targets = data['targets'].to(device, dtype=torch.float)

        # Forward pass
        outputs = model(ids, mask)  # No token_type_ids for RoBERTa
        loss = loss_fn(outputs, targets)
        losses.append(loss.item())

        # Compute accuracy
        outputs = torch.sigmoid(outputs).cpu().detach().numpy().round()
```

[170]:
```python
def eval_model(validation_loader, model, optimizer):
    losses = []
    correct_predictions = 0
    num_samples = 0
    # set model to eval mode (turn off dropout, fix batch norm)
    model.eval()

    with torch.no_grad():
        for batch_idx, data in enumerate(validation_loader, 0):
            ids = data['input_ids'].to(device, dtype=torch.long)
            mask = data['attention_mask'].to(device, dtype=torch.long)
            targets = data['targets'].to(device, dtype=torch.float)

            outputs = model(ids, mask)
            loss = loss_fn(outputs, targets)
            losses.append(loss.item())

            # validation accuracy
            # add sigmoid, for the training sigmoid is in BCEWithLogitsLoss
            outputs = torch.sigmoid(outputs).cpu().detach().numpy().round()
            targets = targets.cpu().detach().numpy()
            correct_predictions += np.sum(outputs==targets)
            num_samples += targets.size  # total number of elements in the 2D array

    return float(correct_predictions)/num_samples, np.mean(losses)
```

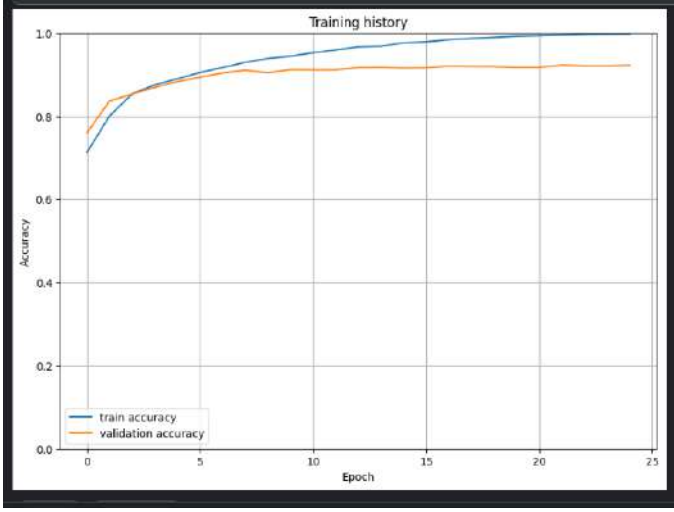
[171]:
```python
content = "/content"
output_dir = os.path.join(content, "output")
os.makedirs(output_dir, exist_ok=True)
```

[172]:
```python
from collections import defaultdict
import torch
import os

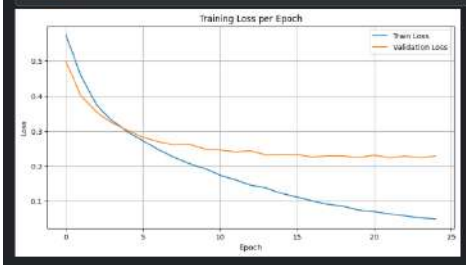
history = defaultdict(list)
```



```
plt.rcParams["figure.figsize"] = (10,7)
plt.plot(history['train_acc'], label='train accuracy')
plt.plot(history['val_acc'], label='validation accuracy')
plt.title('Training history')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend()
plt.ylim([0, 1]);
plt.grid()
```




```
plt.figure(figsize=(10, 5))
plt.plot(history['train_loss'], label='Train Loss')
plt.plot(history['val_loss'], label='Validation Loss')
plt.title('Training Loss per Epoch')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend()
plt.grid()
plt.show()
```




```
# Loading pretrained model (best model)
model = RoBERTaClass()
model.load_state_dict(torch.load(os.path.join(output_dir,"/kaggle/working/Aspect_roberta.bin")))
model = model.to(device)
```

```
Some weights of RobertaModel were not initialized from the model checkpoint at roberta-base and are newly initialized: ['pooler.dense.bias', 'pooler.dense.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
/tmp/ipykernel_31/2284794589.py:1: FutureWarning: You are using `torch.load` with `weights_only=False` (the current default value), which uses the default pickle module implicitly. It is possible to construct malicious pickle data which will execute arbitrary code during unpickling (see https://github.com/pytorch/pytorch/blob/main/SECURITY.md#untrusted-models for more details). In a future release, the default value for `weights_only` will be flipped to `True`. This limits the functions that could be executed during unpickling. Arbitrary objects will no longer be allowed to be loaded via this mode unless they are explicitly allowlisted by the user via `torch.serialization.add_safe_globals`. We recommend you start setting `weights_only=True` for any use case where you don't have full control of the loaded file. Please open an issue on GitHub for any issues related to this experimental feature.
```

```
[177]: test_acc, test_loss = eval_model(test_data_loader, model, optimizer)

[178]: test_acc

[17.  0.9894955489614244
```

```
from sklearn.metrics import confusion_matrix, classification_report
```

+ Code + Markdown

```
[100]: def get_predictions(model, data_loader):
           """
           Outputs:
             predictions - Model predictions
             prediction_probs - Probabilities from sigmoid function
             target_values - Ground truth labels
           """
           model.eval()  # Set model ke mode evaluasi

           Reviews = []  # Menggantti 'title' menjadi 'text'
           predictions = []
           prediction_probs = []
           target_values = []

           with torch.no_grad():
               for data in data_loader:
                   text = data["Reviews"]  # Menggabil teks dari dataset, using "Reviews" instead of "text"
                   ids = data["input_ids"].to(device, dtype=torch.long)
                   mask = data["attention_mask"].to(device, dtype=torch.long)
                   targets = data["targets"].to(device, dtype=torch.float)
```

```
import numpy as np
from sklearn.metrics import precision_recall_fscore_support, accuracy_score
from tabulate import tabulate
```

```
Requirement already satisfied: tabulate in /usr/local/lib/python3.11/dist-packages (0.9.0)
```

+ Code + Markdown

```
[84]:
# 1) Pull in your predictions & raw probabilities
test_texts, test_predictions, test_prediction_probs, test_target_values = \
    get_predictions(model, test_data_loader)

# 2) Threshold your probabilities to 0/1
predictions_thresholded = (test_prediction_probs >= 0.5).int()

# 3) Calculate all your metrics
accuracy, precision, recall, f1, support, \
    macro_precision, macro_recall, macro_f1, TP, TN, FP, FN = \
    calculate_metrics(test_target_values, predictions_thresholded)
```

```
[85]:
# Apply thresholding to the raw predictions (e.g., threshold 0.5)
predictions_thresholded = (test_prediction_probs >= 0.5).int()  # Use .int() to convert PyTorch tensor to integer

# Calculate metrics
accuracy, precision, recall, f1, support, macro_precision, macro_recall, macro_f1, TP, TN, FP, FN = calculate_metrics(test_target_values, predictions_thresholded)

# Prepare data for table (rows as metrics, columns as aspects)
metrics_table = []

# Adding overall metrics (macro average)
metrics_table.append(["Test Accuracy", torch.mean(accuracy).item()])
metrics_table.append(["Overall Precision", macro_precision])
metrics_table.append(["Overall Recall", macro_recall])
metrics_table.append(["Overall F1 Score", macro_f1])
metrics_table.append(["Overall Macro Avg", "N/A"])  # Since Macro Avg is not applicable for individual labels

# Adding detailed per-label metrics
```

Metric	Precision	Recall	F1 Score	Macro Avg	Support
Test Accuracy	0.9094955921173096				
Overall Precision	0.8092115629238501				
Overall Recall	0.729548427155841				
Overall F1 Score	0.7651106819241086				
Overall Macro Avg	N/A				
BATTERY	0.9754098360655737	0.894737	0.933333	N/A	133
PERFORMANCE	0.69	0.6	0.64186	N/A	115
CAMERA	0.9398243982439824	0.865169	0.900585	N/A	89
DISPLAY	0.8478588235294118	0.757895	0.8	N/A	95
PRICE	0.9180327868852459	0.736842	0.817518	N/A	76
MULTIMEDIA	0.75	0.725806	0.737705	N/A	62
FITNESS & HEALTH TRACKING	0.8541666666666666	0.911111	0.88172	N/A	45
CUSTOMIZATION	0.5	0.344828	0.408163	N/A	29

```
[186]:
predictions.numpy()
```

```
[18...
array([[0., 1., 0., ..., 0., 0., 0.],
```

+ Code + Markdown

[188]:
```
print(classification_report(target_values, predictions, target_names=target_list))
```

```
                           precision    recall  f1-score   support

                  BATTERY       0.98      0.89      0.93       133
              PERFORMANCE       0.69      0.60      0.64       115
                   CAMERA       0.94      0.87      0.90        89
                  DISPLAY       0.85      0.76      0.80        95
                    PRICE       0.92      0.74      0.82        76
               MULTIMEDIA       0.75      0.73      0.74        62
FITNESS & HEALTH TRACKING       0.85      0.91      0.88        45
            CUSTOMIZATION       0.50      0.34      0.41        29

                micro avg       0.85      0.76      0.80       644
                macro avg       0.81      0.73      0.77       644
             weighted avg       0.84      0.76      0.80       644
              samples avg       0.74      0.69      0.69       644

/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-sco
is behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score
ior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[189]:
```
raw_text ="The Xiaomi Redmi Note Pro is fantastic Its sleek design and powerful performance make it a joy to use
```

[198]:
```
encoded_text = tokenizer.encode_plus(
    raw_text,
    max_length=MAX_LEN,
    add_special_tokens=True,
    padding='max_length',  # RoBERTa-style padding
    truncation=True,  # Memastikan teks tidak melebihi max_length
    return_attention_mask=True,
    return_tensors='pt'
)
```

+ Code + Markdown

[191]:
```
input_ids = encoded_text['input_ids'].to(device)
attention_mask = encoded_text['attention_mask'].to(device)

# Passing the model input_ids
output = model(input_ids, attention_mask)

# Add sigmoid, for the training sigmoid is in BCEWithLogitsLoss
output = torch.sigmoid(output).detach().cpu()

# Threshold at 0.5
output = output.flatten().round().numpy()
output
```

[19..
```
array([1., 1., 1., 1., 0., 0., 0.], dtype=float32)
```

[192]:
```
print(f'Title: {raw_text}')
for idx, p in enumerate(output):
  if p==1:
    print(f'Label: {target_list[idx]}')
```

```
Title: The Xiaomi Redmi Note Pro is fantastic Its sleek design and powerful performance make it a joy to use The robust processor and ample RAM handle multitasking effortlessly The stunning AMOLED display delivers vibrant color and sh
perfect for watching video and browsing The longlasting battery and fast charging are super convenient The versatile camera system capture excellent photo Great value for the price.
Label: BATTERY
```

```
Title: The Xiaomi Redmi Note Pro is fantastic Its sleek design and powerful performance make it a joy to use The robust processor and ample RAM handle multitasking effortlessly The stunning AMOLED display delivers vibrant color and sha
perfect for watching video and browsing The longlasting battery and fast charging are super convenient The versatile camera system capture excellent photo Great value for the price.
Label: BATTERY
Label: PERFORMANCE
Label: CAMERA
Label: DISPLAY
Label: PRICE
```

```
[193]: output = model(input_ids, attention_mask)  # Ignore token_type_ids
       output = torch.sigmoid(output).detach().cpu()
```

```
[194]: probabilities = output.flatten().numpy()
       probabilities_percentage = probabilities * 100
```

```
[195]: emotion_predictions = list(zip(target_list, probabilities_percentage))
       emotion_predictions_sorted = sorted(emotion_predictions, key=lambda x: x[1], reverse=True)
```

```
[196]: print(f"Title: {raw_text}\n")
       print("Predicted Emotion Probabilities:\n")
       for emotion, prob in emotion_predictions_sorted:
           print(f"{emotion}: {prob:.2f}%")
```

```
Title: The Xiaomi Redmi Note Pro is fantastic Its sleek design and powerful performance make it a joy to use The robust processor and ample RAM handle multitasking effortlessly The stunning AMOLED display delivers vibrant color and sha
perfect for watching video and browsing The longlasting battery and fast charging are super convenient The versatile camera system capture excellent photo Great value for the price.

Predicted Emotion Probabilities:

PRICE: 98.36%
DISPLAY: 96.30%
BATTERY: 95.09%
CAMERA: 93.39%
PERFORMANCE: 83.13%
FITNESS & HEALTH TRACCING: 2.39%
CUSTOMIZATION: 2.33%
MULTIMEDIA: 2.08%
```

```
[198]: import matplotlib.pyplot as plt
       import seaborn as sns
       from sklearn.metrics import confusion_matrix, classification_report
       import numpy as np

       # Function to plot confusion matrix
       def plot_confusion_matrix(y_true, y_pred, labels):
           cm = confusion_matrix(y_true, y_pred)
           plt.figure(figsize=(12, 8))
           sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
           plt.xlabel('Predicted Labels')
           plt.ylabel('True Labels')
           plt.title('Confusion Matrix')
           plt.show()

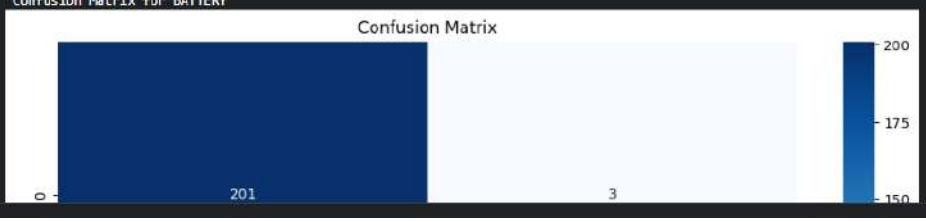
       # Get predictions
       titles, predictions, prediction_probs, target_values = get_predictions(model, test_data_loader)

       # Convert tensors to numpy arrays
       y_true = target_values.numpy()
       y_pred = predictions.numpy()

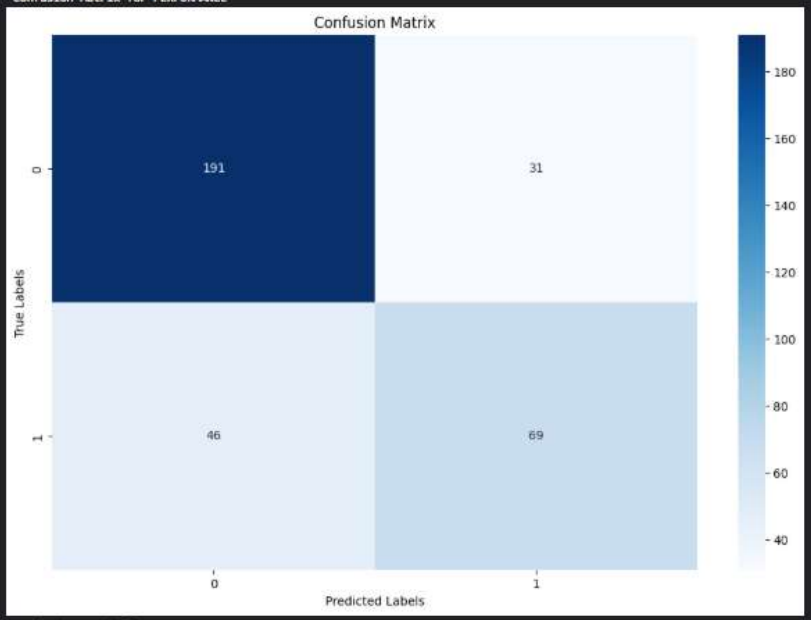
       # Plot confusion matrix for each emotion label
       for i, label in enumerate(target_list):
           print(f'Confusion Matrix for {label}')
           plot_confusion_matrix(y_true[:, i], y_pred[:, i], labels=[0, 1])

       print(f'titles:{len(titles)} \npredictions:{predictions.shape} \nprediction_probs:{prediction_probs.shape} \ntarget_values:{target_va
       print(classification_report(target_values, predictions, target_names=target_list))
```

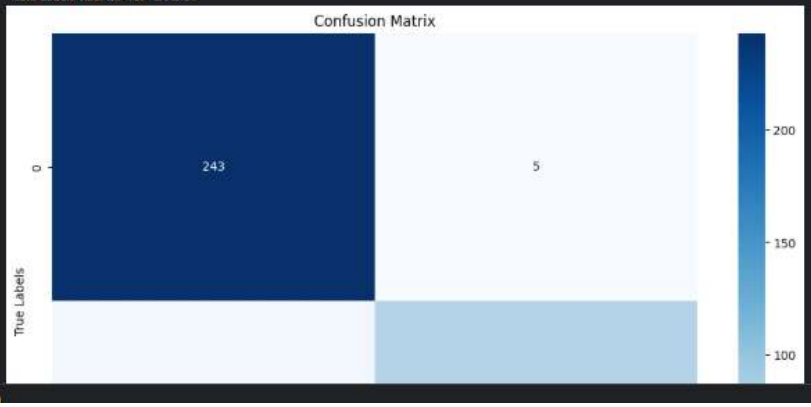
```
Confusion Matrix for BATTERY
```



Confusion Matrix for PERFORMANCE



Confusion Matrix for CAMERA



Predicted Labels

```
titles:11
predictions:torch.Size([337, 8])
prediction_probs:torch.Size([337, 8])
target_values:torch.Size([337, 8])
```

	precision	recall	f1-score	support
BATTERY	0.98	0.89	0.93	133
PERFORMANCE	0.69	0.60	0.64	115
CAMERA	0.94	0.87	0.90	89
DISPLAY	0.85	0.76	0.80	95
PRICE	0.92	0.74	0.82	76
MULTIMEDIA	0.75	0.73	0.74	62
FITNESS & HEALTH TRACKING	0.85	0.91	0.88	45
CUSTOMIZATION	0.50	0.34	0.41	29
micro avg	0.85	0.76	0.80	644
macro avg	0.81	0.73	0.77	644
weighted avg	0.84	0.76	0.80	644
samples avg	0.74	0.69	0.69	644

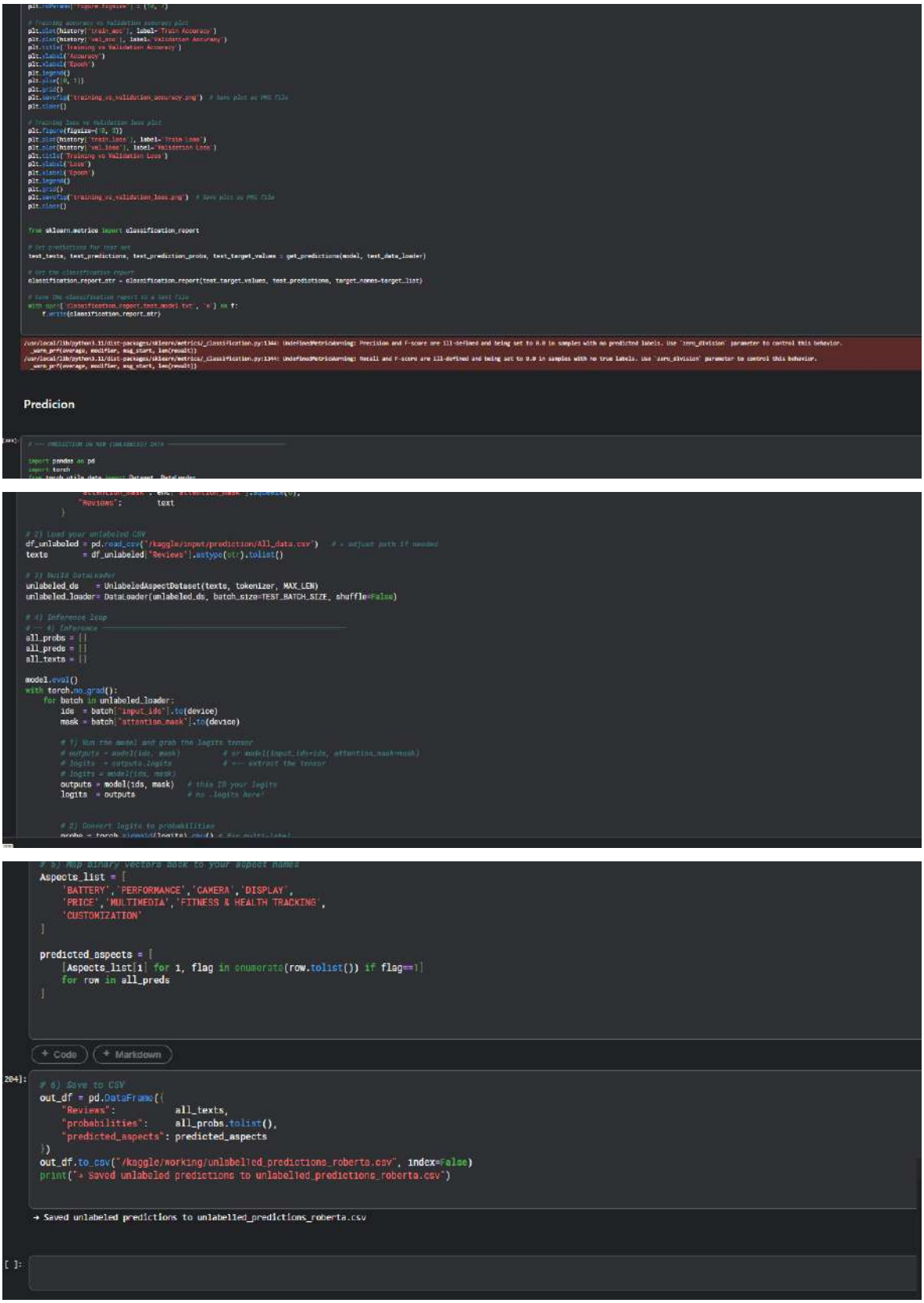
```
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in samples with no pred
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.11/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in samples with no true la
  _warn_prf(average, modifier, msg_start, len(result))
```

+ Code + Markdown

199]:

```
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix

def save_confusion_matrix(y_true, y_pred, labels, label_name):
    cm = confusion_matrix(y_true, y_pred)
    plt.figure(figsize=(12, 8))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=labels, yticklabels=labels)
    plt.xlabel('Predicted Labels')
    plt.ylabel('True Labels')
    plt.title(f'Confusion Matrix for {label_name}')
    plt.tight_layout()
    confusion_matrix_filename = f"confusion_matrix_{label_name}.png"
```



Appendix 6. SC Prediction with Fine-tuned models

```
import pandas as pd
import torch
from transformers import (
    AutoTokenizer,
    BertForSequenceClassification,
    RobertaForSequenceClassification,
    DistilBertForSequenceClassification
)
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
import gc
from tqdm import tqdm
import torch, gc, time
from IPython.display import display
import ipywidgets as widgets
```

+ Code + Markdown

```
# ── 1) Load your dataset ──────────────────────────────────────
df = pd.read_csv("/kaggle/input/data-set/all_data.csv")
texts = df["Reviews"].tolist()

# ── 1) Load & clean your data ─────────────────────────────────

# Fill NA and force to str so tokenizer never sees a non-str
texts = df["Reviews"].fillna("").astype(str).tolist()
print("Total reviews:", len(texts))

# Optional true labels
label2id   = {"Negative":0,"Neutral":1,"Positive":2}
has_labels = "label" in df.columns and df["label"].notna().any()
if has_labels:
    df["label_id"] = df["label"].map(label2id)
    if df["label_id"].isna().any():
        raise ValueError("Found invalid label values!")
    labels = df["label_id"].astype(int).tolist()
```

```
Total reviews: 16796
```

```
models = {
    "BERT"      : "/kaggle/input/models/BERT_epoch_3.model",
    "DistilBERT": "/kaggle/input/models/best_model_distilbert.bin",
    "RoBERTa"   : "/kaggle/input/models/roberta_sentiment_model_balanced.pth",
}

arch_map = {
    "BERT":      (BertForSequenceClassification,      "bert-base-uncased"),
    "RoBERTa":   (RobertaForSequenceClassification,   "roberta-base"),
    "DistilBERT":(DistilBertForSequenceClassification,"distilbert-base-uncased"),
}

results = []
all_preds = {"text": texts}
```

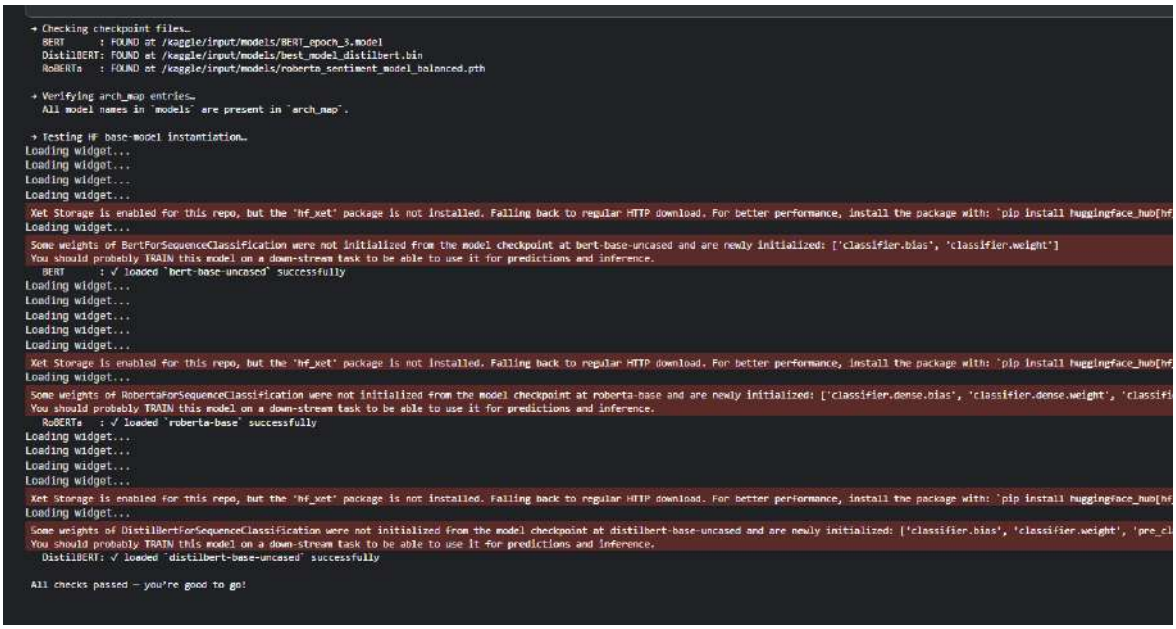
+ Code + Markdown

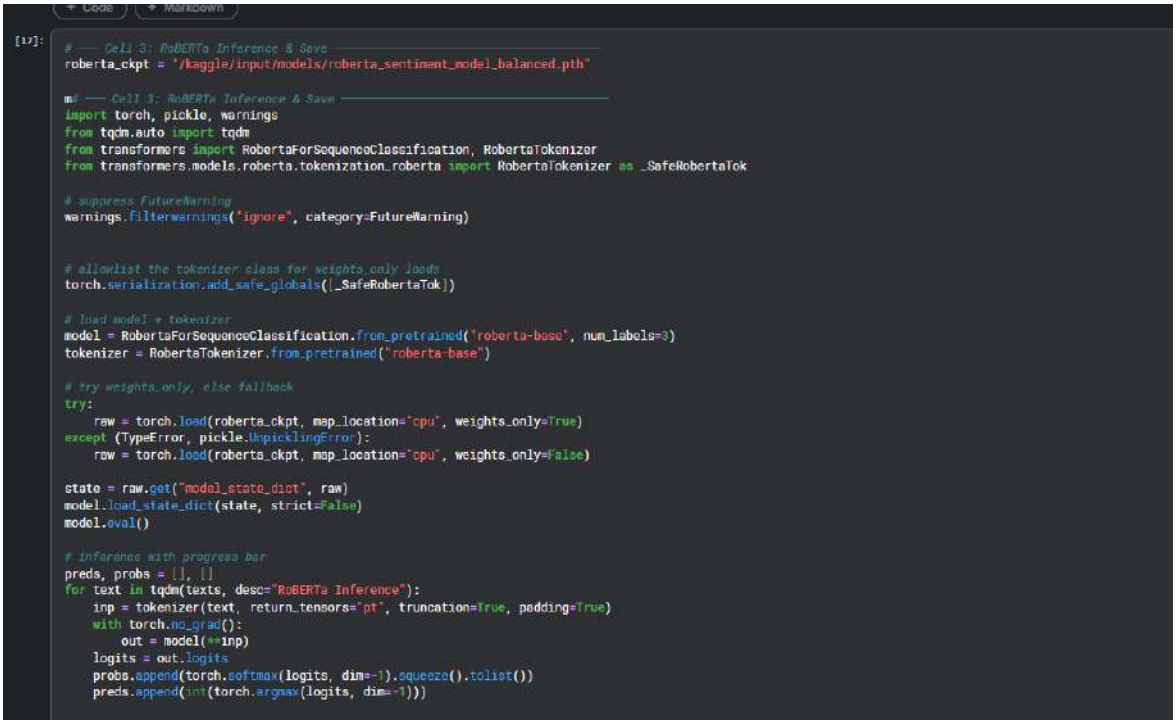
```
import os
from transformers import AutoTokenizer

# 0) Preliminary smoke tests
print("→ Checking checkpoint files…")
for name, path in models.items():
    exists = os.path.isfile(path)
    print(f"  {name:10s}: {'FOUND' if exists else 'MISSING'} at {path}")
    if not exists:
        raise FileNotFoundError(f"Cannot find checkpoint for {name} at\n    {path}")

print("\n→ Verifying arch_map entries…")
for name in models:
    if name not in arch_map:
        raise KeyError(f"Model {name!r} is missing from arch_map")
print("  All model names in `models` are present in `arch_map`.")

print("\n→ Testing HF base-model instantiation…")
for name, (ModelClass, base_id) in arch_map.items():
    try:
        tok = AutoTokenizer.from_pretrained(base_id)
        m   = ModelClass.from_pretrained(base_id, num_labels=3)
        print(f"  {name:10s}: ✓ loaded `{base_id}` successfully")
    except Exception as e:
        print(f"  {name:10s}: ✗ error loading `{base_id}` → {e}")
        raise
```


→ Checking checkpoint files…
BERT : FOUND at /kaggle/input/models/BERT_epoch_3.model
DistilBERT: FOUND at /kaggle/input/models/best_model_distilbert.bin
RoBERTa : FOUND at /kaggle/input/models/roberta_sentiment_model_balanced.pth
→ Verifying arch_map entries…
All model names in `models` are present in `arch_map`.
→ Testing HF base-model instantiation…
Loading widget...
Loading widget...
Loading widget...
Loading widget...
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf
Loading widget...
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
BERT : ✓ loaded `bert-base-uncased` successfully
Loading widget...
Loading widget...
Loading widget...
Loading widget...
Loading widget...
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf
Loading widget...
Some weights of RobertaForSequenceClassification were not initialized from the model checkpoint at roberta-base and are newly initialized: ['classifier.dense.bias', 'classifier.dense.weight', 'classifi
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
RoBERTa : ✓ loaded `roberta-base` successfully
Loading widget...
Loading widget...
Loading widget...
Loading widget...
Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better performance, install the package with: `pip install huggingface_hub[hf
Loading widget...
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncased and are newly initialized: ['classifier.bias', 'classifier.weight', 'pre_cl
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
DistilBERT: ✓ loaded `distilbert-base-uncased` successfully
All checks passed — you're good to go!


+ Code
+ Markdown
[17]:
# ── Cell 3: RoBERTa Inference & Save ──────────────────────────────
roberta_ckpt = "/kaggle/input/models/roberta_sentiment_model_balanced.pth"
m# ── Cell 3: RoBERTa Inference & Save ──────────────────────────────
import torch, pickle, warnings
from tqdm.auto import tqdm
from transformers import RobertaForSequenceClassification, RobertaTokenizer
from transformers.models.roberta.tokenization_roberta import RobertaTokenizer as _SafeRobertaTok
# suppress FutureWarning
warnings.filterwarnings("ignore", category=FutureWarning)
# allowlist the tokenizer class for weights_only loads
torch.serialization.add_safe_globals([_SafeRobertaTok])
# load model + tokenizer
model = RobertaForSequenceClassification.from_pretrained("roberta-base", num_labels=3)
tokenizer = RobertaTokenizer.from_pretrained("roberta-base")
# try weights_only, else fallback
try:
raw = torch.load(roberta_ckpt, map_location="cpu", weights_only=True)
except (TypeError, pickle.UnpicklingError):
raw = torch.load(roberta_ckpt, map_location="cpu", weights_only=False)
state = raw.get("model_state_dict", raw)
model.load_state_dict(state, strict=False)
model.eval()
# inference with progress bar
preds, probs = [], []
for text in tqdm(texts, desc="RoBERTa Inference"):
inp = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
with torch.no_grad():
out = model(**inp)
logits = out.logits
probs.append(torch.softmax(logits, dim=-1).squeeze().tolist())
preds.append(int(torch.argmax(logits, dim=-1)))

[19]:
```
# save
df_roberta = pd.DataFrame({
    # "review_id":     ids,
    "text":          texts,
    "pred_label":    preds,
    "probabilities": probs
})
df_roberta.to_csv("/kaggle/working/roberta_predictions.csv", index=False)
print("✅ RoBERTa done → roberta_predictions.csv")
```

```
✅ RoBERTa done → roberta_predictions.csv
```

+ Code + Markdown

[8]:
```
# probs = torch.softmax(logits, dim=-1)
# # e.g. for one batch:
# for p in probs:
#     print(p.tolist())  # [p_neg, p_neu, p_pos]
```

```
[0.31193241477012634, 0.3879519402980044, 0.3001156151294708]
[0.3131184279918670, 0.390510618686676, 0.2963709533214569]
[0.3147969245910645, 0.3870779871940613, 0.2981250584125519]
[0.3154520988464355, 0.3870779573917389, 0.2974699437618256]
[0.3146877586841583, 0.38616889715194, 0.29914334416389465]
```

[10]:
```
# if has_labels:
```

[21]:
```
# ── Cell 2: BERT Inference & Save with tqdm & safe torch.load ──────────
import torch, pickle, warnings
import pandas as pd
from tqdm.auto import tqdm
from transformers import BertForSequenceClassification, BertTokenizer
from transformers.models.bert.tokenization_bert import BertTokenizer as _SafeBertTok

# suppress FutureWarning
warnings.filterwarnings("ignore", category=FutureWarning)

# allowlist the tokenizer class for weights_only loads
torch.serialization.add_safe_globals([_SafeBertTok])

# checkpoint path
bert_ckpt = "/kaggle/input/models/BERT_epoch_3.model"

# load model+tokenizer
model = BertForSequenceClassification.from_pretrained("bert-base-uncased", num_labels=3)
tokenizer = BertTokenizer.from_pretrained("bert-base-uncased")

# safe load weights
try:
    raw = torch.load(bert_ckpt, map_location="cpu", weights_only=True)
except (TypeError, pickle.UnpicklingError):
    raw = torch.load(bert_ckpt, map_location="cpu", weights_only=False)

state = raw.get("model_state_dict", raw)
model.load_state_dict(state, strict=False)
model.eval()

# inference
df = pd.read_csv("/kaggle/input/data-set/All_data.csv")  # ensure 'texts' & 'ids' are available
texts = df["Reviews"].astype(str).tolist()
ids   = df.index.tolist()
```

```
model.eval()

# inference
df = pd.read_csv("/kaggle/input/data-set/All_data.csv")  # ensure 'texts' & 'ids' are available
texts = df["Reviews"].astype(str).tolist()
ids   = df.index.tolist()

preds, probs = [], []
for text in tqdm(texts, desc="BERT Inference"):
    inp = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    with torch.no_grad():
        out = model(**inp)
    logits = out.logits
    probs.append(torch.softmax(logits, dim=-1).squeeze().tolist())
    preds.append(int(torch.argmax(logits, dim=-1)))

# save results
df_bert = pd.DataFrame({
    # "review_id":     ids,
    "text":          texts,
    "pred_label":    preds,
    "probabilities": probs
})
df_bert.to_csv("/kaggle/working/bert_predictions.csv", index=False)
print("✅ BERT done → bert_predictions.csv")
```

```
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are newly initialized: ['classifier.bias', 'classifier.weight']
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Loading widget...
✅ BERT done → bert_predictions.csv
```

+ Code + Markdown



```
[23]:
# ── Cell 4: DistilBERT Inference & Save with tqdm & safe torch.load ──────────
import torch, pickle, warnings
import pandas as pd
from tqdm.auto import tqdm
from transformers import DistilBertForSequenceClassification, DistilBertTokenizer
from transformers.models.distilbert.tokenization_distilbert import DistilBertTokenizer as _SafeDistilTok

# suppress FutureWarning
warnings.filterwarnings("ignore", category=FutureWarning)

# allowlist the tokenizer class for weights_only loads
torch.serialization.add_safe_globals([_SafeDistilTok])

# checkpoint path
distil_ckpt = "/kaggle/input/models/best_model_distilbert.bin"

# load model+tokenizer
model = DistilBertForSequenceClassification.from_pretrained("distilbert-base-uncased", num_labels=3)
tokenizer = DistilBertTokenizer.from_pretrained("distilbert-base-uncased")

# safe load weights
try:
    raw = torch.load(distil_ckpt, map_location="cpu", weights_only=True)
except (TypeError, pickle.UnpicklingError):
    raw = torch.load(distil_ckpt, map_location="cpu", weights_only=False)

state = raw.get("model_state_dict", raw)
model.load_state_dict(state, strict=False)
model.eval()

# inference
df = pd.read_csv("/kaggle/input/data-set/All_data.csv")  # ensure 'texts' & 'ids' are available
texts = df["Reviews"].astype(str).tolist()
ids   = df.index.tolist()

preds, probs = [], []
for text in tqdm(texts, desc="DistilBERT Inference"):
    inp = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    with torch.no_grad():
        out = model(**inp)
    logits = out.logits
    probs.append(torch.softmax(logits, dim=-1).squeeze().tolist())
    preds.append(int(torch.argmax(logits, dim=-1)))
```

```
# inference
df = pd.read_csv("/kaggle/input/data-set/All_data.csv")  # ensure 'texts' & 'ids' are available
texts = df["Reviews"].astype(str).tolist()
ids   = df.index.tolist()

preds, probs = [], []
for text in tqdm(texts, desc="DistilBERT Inference"):
    inp = tokenizer(text, return_tensors="pt", truncation=True, padding=True)
    with torch.no_grad():
        out = model(**inp)
    logits = out.logits
    probs.append(torch.softmax(logits, dim=-1).squeeze().tolist())
    preds.append(int(torch.argmax(logits, dim=-1)))

# save results
df_distil = pd.DataFrame({
    # "review_id":   ids,
    "text":        texts,
    "pred_label":  preds,
    "probabilities": probs
})
df_distil.to_csv("/kaggle/working/distilbert_predictions.csv", index=False)
print("✅ DistilBERT done → distilbert_predictions.csv")
```

```
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncased and are newly initialized: ['classifier.bias', 'classifier.weight', 'pr
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
Loading widget...
✅ DistilBERT done → distilbert_predictions.csv
```

Appendix 7. Aspects & Sentiment Visualization

```
import pandas as pd
import ast
import matplotlib.pyplot as plt
from itertools import combinations
```

```
# === CONFIG ===
csv_path = 'predictions_roberta.csv'  # your CSV file
top_n = 8                             # number of top aspects to focus on
threshold = 0.5                       # (optional) probability filter

# === 1) LOAD & PREPROCESS ===
df = pd.read_csv(csv_path)

# rename columns for consistency
df.rename(columns={
    'Date': 'timestamp',
    'Device_Name': 'device_name',
    'Product_type': 'device_type',
    'predicted_aspects': 'aspects',
    'probabilities': 'probs'
}, inplace=True)

# parse timestamp
df['timestamp'] = pd.to_datetime(df['timestamp'], errors='coerce')

# add IDs & compute review lengths
df['review_id'] = df.index
df['review_length'] = df['Reviews'].str.split().str.len()

df.head()
```

	timestamp	Reviews	device_name	device_type	rating	Brand	probs	aspects	review_id	review_length
0	2023-04-09	This is my 3rd "Smart Watch". My 1st was a Gar...	Galaxy Watch 4	Watch	5	Samsung	[0.9433803556349609, 0.037575736641883..., 0.00...	['BATTERY', 'FITNESS & HEALTH TRACKING']	0	388
1	2024-04-18	The watch itself, is stunning! It's by far th...	Galaxy Watch 4	Watch	5	Samsung	[0.9274229407310486, 0.10970456898212433, 0.00...	['BATTERY', 'DISPLAY', 'PRICE', 'MULTIMEDIA']	1	195
2	2023-05-13	05/13/2023: A really, really nice smartwatch...	Galaxy Watch 4	Watch	5	Samsung	[0.9813327789306641, 0.05427504333087349..., 0.0...	['BATTERY', 'FITNESS & HEALTH TRACKING']	2	555

obles Terminal

```
# === 2) PARSE & EXPLODE ASPECTS ===
def safe_eval(x):
    try:
        return ast.literal_eval(x)
    except:
        return []

df['aspects'] = df['aspects'].apply(safe_eval)
df['probs']   = df['probs'].apply(safe_eval)

# explode only the aspects list
df_exp = df.explode('aspects').reset_index(drop=True)

# optional: map each exploded aspect back to its probability and filter
# (requires knowing the model's class order)
# classes = ['BATTERY','CAMERA','DISPLAY','PERFORMANCE', ...]
# df_exp['prob'] = df_exp.apply(
#     lambda r: r['probs'][classes.index(r['aspects'])] if r['aspects'] in classes else 0,
#     axis=1
# )
# df_exp = df_exp[df_exp['prob'] > threshold]
```

```
import pandas as pd
import ast

# 1) LOAD
df = pd.read_csv('predictions_roberta.csv')

# 2) RENAME (make sure this matches your actual header names)
df.rename(columns={
    'Date':             'timestamp',
    'Device_Name':      'device_name',
    'Product_type':     'device_type',
    'predicted_aspects':'aspects',
    'probabilities':    'probs'
}, inplace=True)

# 3) QUICK SANITY CHECK
print("Raw aspect strings (first 5):")
```

```
print("Number of parsed aspects per row (value_counts):")
print(df['aspects_list'].apply(len).value_counts().sort_index(), "\n")

# 6) EXPLODE
df_exp = df.explode('aspects_list').rename(columns={'aspects_list':'aspect'})

# 7) SANITY-CHECK EXPLODE
print("Exploded aspect sample:")
print(df_exp[['aspect']].dropna().head(), "\n")
print("Total exploded rows:", len(df_exp))

# 8) COMPUTE FREQUENCIES
aspect_counts = (
    df_exp['aspect']
      .value_counts()
      .rename_axis('aspect')
      .reset_index(name='count')
)
print("\nFinal list of aspects & frequency counts:\n", aspect_counts)

# now top_aspects will work:
top_n = 8
top_aspects = aspect_counts['aspect'].head(top_n).tolist()
```

```
Raw aspect strings (first 5):
0              ['BATTERY', 'FITNESS & HEALTH TRACKING']
1        ['BATTERY', 'DISPLAY', 'PRICE', 'MULTIMEDIA']
2              ['BATTERY', 'FITNESS & HEALTH TRACKING']
3              ['BATTERY', 'FITNESS & HEALTH TRACKING']
4    ['BATTERY', 'DISPLAY', 'MULTIMEDIA', 'FITNESS ...
Name: aspects, dtype: object

Number of parsed aspects per row (value counts):
aspects_list
1    0156
2    3825
3    1620
4     586
5     151
6      21
7       3
Name: count, dtype: int64
```




```
    if isinstance(x, str) and pd.isna(x):
        return []
    # if it's already a list (unlikely), return as-is
    if isinstance(x, list):
        return x
    try:
        return ast.literal_eval(x)
    except Exception:
        # fallback: split on commas
        return [a.strip() for a in x.split(',') if a.strip()]

df['aspects'] = df['aspects'].apply(safe_eval)
df_exp       = df.explode('aspects').rename(columns={'aspects':'aspect'}).reset_index(drop=True)

# === 3) FINAL LIST & FREQUENCIES ===
aspect_counts = (
    df_exp['aspect']
      .value_counts()
      .rename_axis('aspect')
      .reset_index(name='count')
)

print("\nFinal list of aspects & frequency counts:\n")
print(aspect_counts)

# pick the top N
top_aspects = aspect_counts['aspect'].head(top_n).tolist()
```

```
inal list of aspects & frequency counts:

                aspect  count
               BATTERY   5365
           PERFORMANCE   4852
               DISPLAY   2949
                 PRICE   2569
                CAMERA   2493
           MULTIMEDIA    2208
FITNESS & HEALTH TRACKING  1914
          CUSTOMIZATION    362

1
```

```
import matplotlib.pyplot as plt

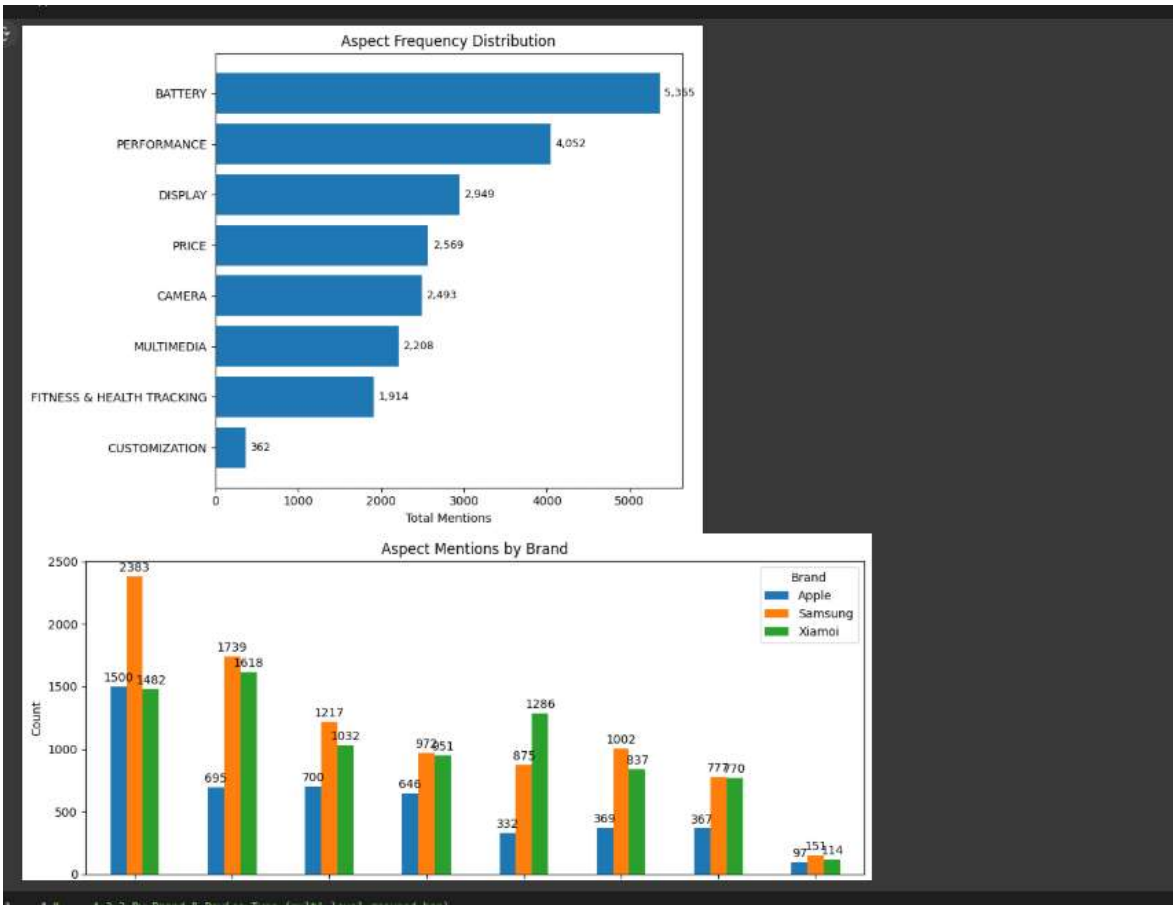
# 4.1 Basic Aspect Frequency (horizontal bar with labels)
plt.figure(figsize=(8,6))
hc = aspect_counts.sort_values('count', ascending=True)
bars = plt.barh(hc['aspect'], hc['count'])
plt.xlabel('Total Mentions')
plt.title('Aspect Frequency Distribution')
# annotate each bar
for bar in bars:
    w = bar.get_width()
    plt.text(w + max(hc['count']) * 0.01,  # a tiny offset
             bar.get_y() + bar.get_height()/2,
             f"{w:,}",                     # comma-formatted integer
             va='center',
             fontsize=9)
plt.tight_layout()
plt.show()

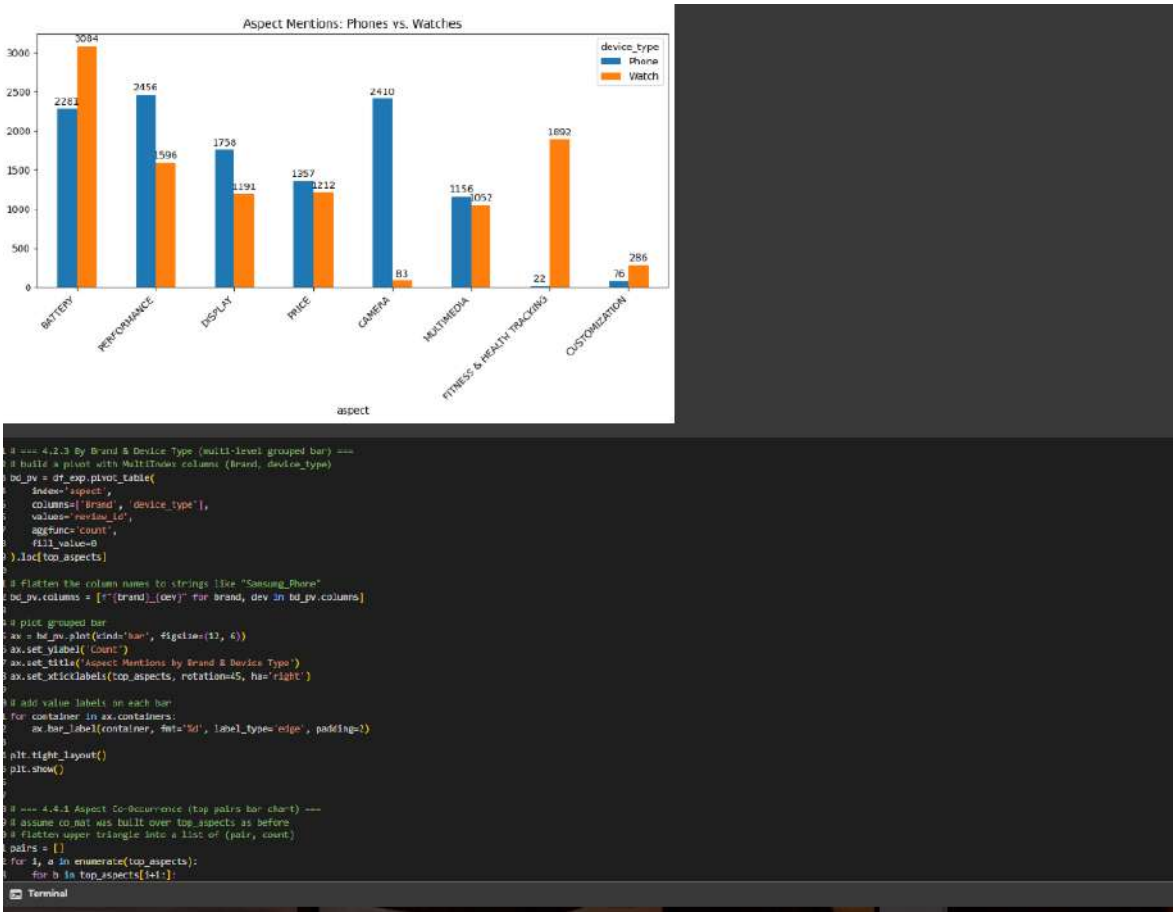

# 4.2.1 Aspect Frequency per Brand (grouped bar with labels)
fig, ax = plt.subplots(figsize=(10,6))
brand_pv.loc[top_aspects].plot(kind='bar', ax=ax)
plt.ylabel('Count')
plt.title('Aspect Mentions by Brand')
plt.xticks(rotation=45, ha='right')
# annotate each bar container
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=2)
plt.tight_layout()
plt.show()


# 4.2.2 Aspect Frequency per Device Type (clustered bar with labels)
fig, ax = plt.subplots(figsize=(10,6))
dev_pv.loc[top_aspects].plot(kind='bar', ax=ax)
plt.ylabel('Count')
plt.title('Aspect Mentions: Phones vs. Watches')
plt.xticks(rotation=45, ha='right')
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=2)
plt.tight_layout()
plt.show()

```




Aspect Frequency Distribution
BATTERY
PERFORMANCE
DISPLAY
PRICE
CAMERA
MULTIMEDIA
FITNESS & HEALTH TRACKING
CUSTOMIZATION
5,355
4,052
2,949
2,569
2,493
2,208
1,914
362
0
1000
2000
3000
4000
5000
Total Mentions
Aspect Mentions by Brand
Brand
Apple
Samsung
Xiaomi
Count
2500
2000
1500
1000
500
0
1500
2383
1482
695
1739
1618
700
1217
1032
646
972
951
332
875
1286
369
1002
837
367
777
770
97
151
114




```
# === 4.2.3 By Brand & Device Type (multi-level grouped bar) ===
# build a pivot with MultiIndex columns (Brand, device_type)
bd_pv = df_exp.pivot_table(
    index='aspect',
    columns=['Brand', 'device_type'],
    values='review_id',
    aggfunc='count',
    fill_value=0
).loc[top_aspects]

# flatten the column names to strings like "Samsung_Phone"
bd_pv.columns = [f"{brand}_{dev}" for brand, dev in bd_pv.columns]

# plot grouped bar
ax = bd_pv.plot(kind='bar', figsize=(12, 6))
ax.set_ylabel('Count')
ax.set_title('Aspect Mentions by Brand & Device Type')
ax.set_xticklabels(top_aspects, rotation=45, ha='right')

# add value labels on each bar
for container in ax.containers:
    ax.bar_label(container, fmt='%d', label_type='edge', padding=2)

plt.tight_layout()
plt.show()


# === 4.4.1 Aspect Co-Occurrence (top pairs bar chart) ===
# assume co_mat was built over top_aspects as before
# flatten upper triangle into a list of (pair, count)
pairs = []
for i, a in enumerate(top_aspects):
    for b in top_aspects[i+1:]:
```

Terminal

```
plt.show()

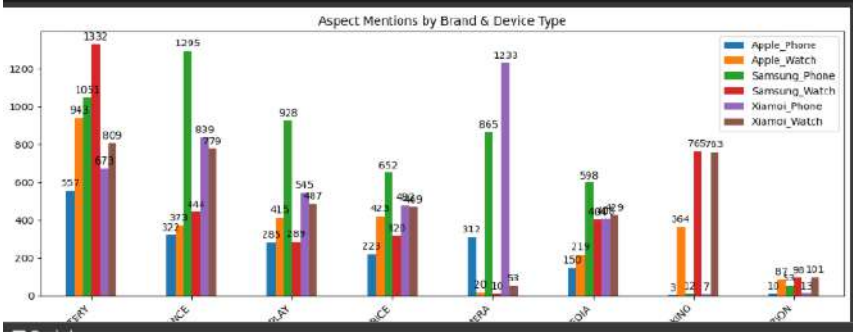

# === 4.4.1 Aspect Co-Occurrence (top pairs bar chart) ===
# assume co_mat was built over top_aspects as before
# flatten upper triangle into a list of (pair, count)
pairs = []
for i, a in enumerate(top_aspects):
    for b in top_aspects[i+1:]:
        pairs.append({
            'pair':       f"{a} & {b}",
            'co_mentions': int(co_mat.loc[a, b])
        })

pairs_df = (
    pd.DataFrame(pairs)
      .sort_values('co_mentions', ascending=False)
      .head(10)  # top 10 co-occurrences
      .set_index('pair')
)

# horizontal bar chart
plt.figure(figsize=(8, 5))
bars = plt.barh(pairs_df.index, pairs_df['co_mentions'])
plt.xlabel('Co-Mentions')
plt.title('Top 10 Aspect Co-Occurrences')

# add labels
for bar in bars:
    x = bar.get_width()
    plt.text(x + pairs_df['co_mentions'].max()*0.01,
             bar.get_y() + bar.get_height()/2,
             f"{x}",
             va='center')

plt.tight_layout()
plt.show()
```



Terminal

```
import seaborn as sns

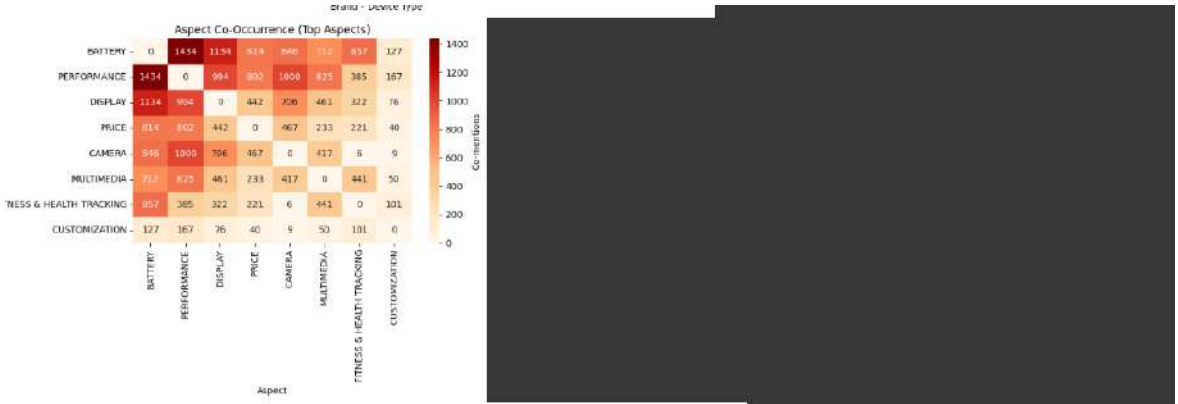
plt.figure(figsize=(12, 6))
sns.heatmap(bd_pv, annot=True, fmt='d', cmap='YlGnBu', cbar_kws={'label': 'Mentions'})
plt.title('Aspect Mentions by Brand & Device Type')
plt.ylabel('Aspect')
plt.xlabel('Brand - Device Type')
plt.tight_layout()
plt.show()

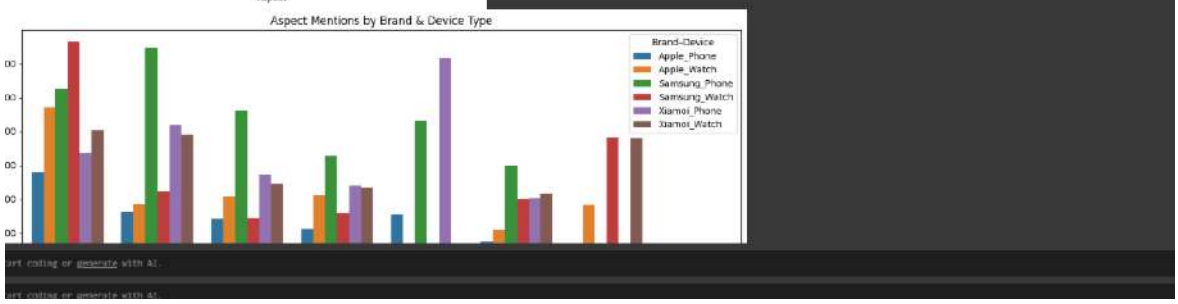
plt.figure(figsize=(8, 6))
sns.heatmap(co_mat, annot=True, fmt='d', cmap='OrRd', cbar_kws={'label': 'Co-Mentions'})
plt.title('Aspect Co-Occurrence (Top Aspects)')
plt.ylabel('Aspect')
plt.xlabel('Aspect')
plt.tight_layout()
plt.show()


flat_bd = bd_pv.stack().reset_index()
flat_bd.columns = ['Aspect', 'Brand-Device', 'Count']

plt.figure(figsize=(12, 6))
sns.barplot(data=flat_bd, x='Aspect', y='Count', hue='Brand-Device')
plt.title('Aspect Mentions by Brand & Device Type')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```





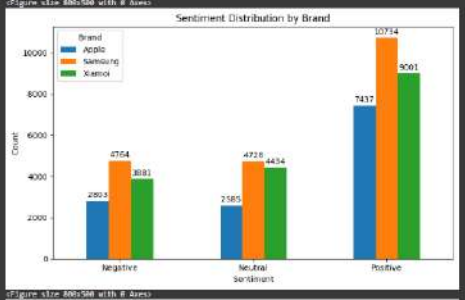


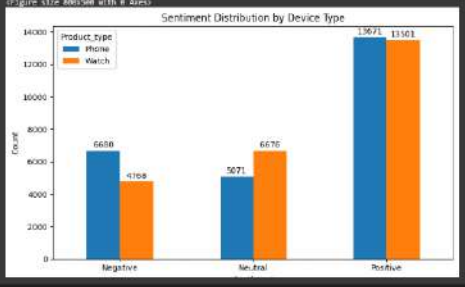


```
# --- 2) BASIC SENTIMENT DISTRIBUTION PER MODEL ---
for model in df_long['Model'].unique():
    data = df_long[df_long['Model'] == model]['Sentiment'].value_counts().reindex(['Negative','Neutral','Positive'], fill_value=0)
    plt.figure(figsize=(6, 4))
    bars = plt.barh(data.index, data.values)
    plt.title(f'{model} Sentiment Distribution')
    plt.xlabel('Count')
    # annotate counts
    for bar in bars:
        w = bar.get_width()
        plt.text(w + data.values.max()*0.01, bar.get_y() + bar.get_height()/2, f'{int(w)}', va='center')
    plt.tight_layout()
    plt.show()
```



```
plt.xticks(rotation=0)
plt.legend(bbox_to_anchor=(1.05,1), loc='upper left')
plt.tight_layout()
plt.show()
```



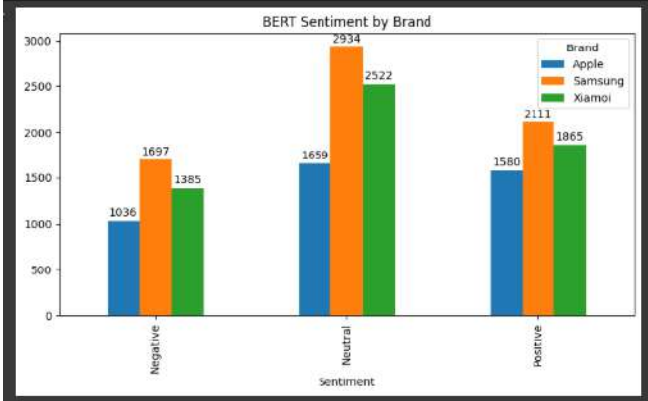



```
# filter to BERT only
bert_only = df_long[df_long['Model']=='BERT']

# then do the same pivot
brand_pv_bert = bert_only.pivot_table(
    index='Sentiment', columns='Brand', aggfunc='size', fill_value=0
).reindex(['Negative','Neutral','Positive'])

# and plot
ax = brand_pv_bert.plot(kind='bar', figsize=(8,5))
ax.set_title('BERT Sentiment by Brand')
for container in ax.containers:
    ax.bar_label(container, fmt='%d', padding=2)
plt.tight_layout()
plt.show()

```

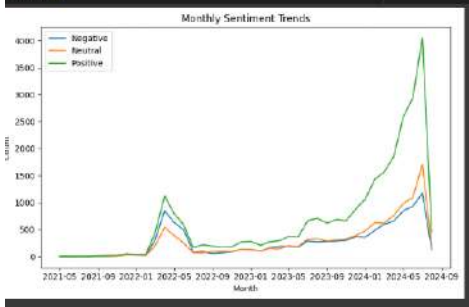



```


# === 4) TEMPORAL SENTIMENT TRENDS ===
df_long['month'] = df_long['timestamp'].dt.to_period('M').dt.to_timestamp()
time_pv = df_long.pivot_table(index='month', columns='Sentiment', aggfunc='size', fill_value=0).reindex(columns=['Negative','Neutral','Positive'])
plt.figure(figsize=(8,5))
for sent in ['Negative','Neutral','Positive']:
    plt.plot(time_pv.index, time_pv[sent], label=sent)
plt.legend()
plt.title('Monthly Sentiment Trends')
```

Terminal

```
# === 4) TEMPORAL SENTIMENT TRENDS ===
df_long['month'] = df_long['timestamp'].dt.to_period('M').dt.to_timestamp()
time_pv = df_long.pivot_table(index='month', columns='Sentiment', aggfunc='size', fill_value=0).reindex(columns=['Negative','Neutral','Positive'])
plt.figure(figsize=(8,5))
for sent in ['Negative','Neutral','Positive']:
    plt.plot(time_pv.index, time_pv[sent], label=sent)
plt.legend()
plt.title('Monthly Sentiment Trends')
plt.xlabel('Month')
plt.ylabel('Count')
plt.tight_layout()
plt.show()
```




```
# === 5) RELATIONAL: Sentiment vs. Review Length ===
# Boxplot of review lengths by sentiment
length_bins = pd.qcut(df_long['review_length'], q=4, labels=['Short','Medium','Long','XL'])
df_long['length_bin'] = length_bins
plot_data = [df_long[df_long['length_bin']==lbl]['review_length'] for lbl in ['Short', 'Medium','Long','XL']]

plt.figure(figsize=(6,4))
plt.boxplot(plot_data, labels=['Short', 'Medium', 'Long', 'XL'])
plt.title('Review Length by Length Bin')
plt.ylabel('Review Length (words)')
plt.xlabel('Length Bin')
plt.tight_layout()
plt.show()
```

```
python-input-8-7965516e2f0b>:8: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels' since Matplotlib 3.9; support for the old name will be dropped in 3.11.
plt.boxplot(plot_data, labels=['Short','Medium','Long','XL'])
```

Review Length by Length Bin

```
# 1) Extract year from timestamp
df_long['Year'] = df_long['timestamp'].dt.year

# 2) Pivot to get counts per year
year_pv = df_long.pivot_table(
    index='Year',
    columns='Sentiment',
    aggfunc='size',
    fill_value=0
).reindex(columns=['Negative', 'Neutral', 'Positive'])

# 3) Plot yearly trends
plt.figure(figsize=(10, 6))
markers = {'Negative': 'o', 'Neutral': 's', 'Positive': '^'}
for sentiment, marker in markers.items():
    plt.plot(year_pv.index, year_pv[sentiment], marker=marker, linewidth=2, label=sentiment)

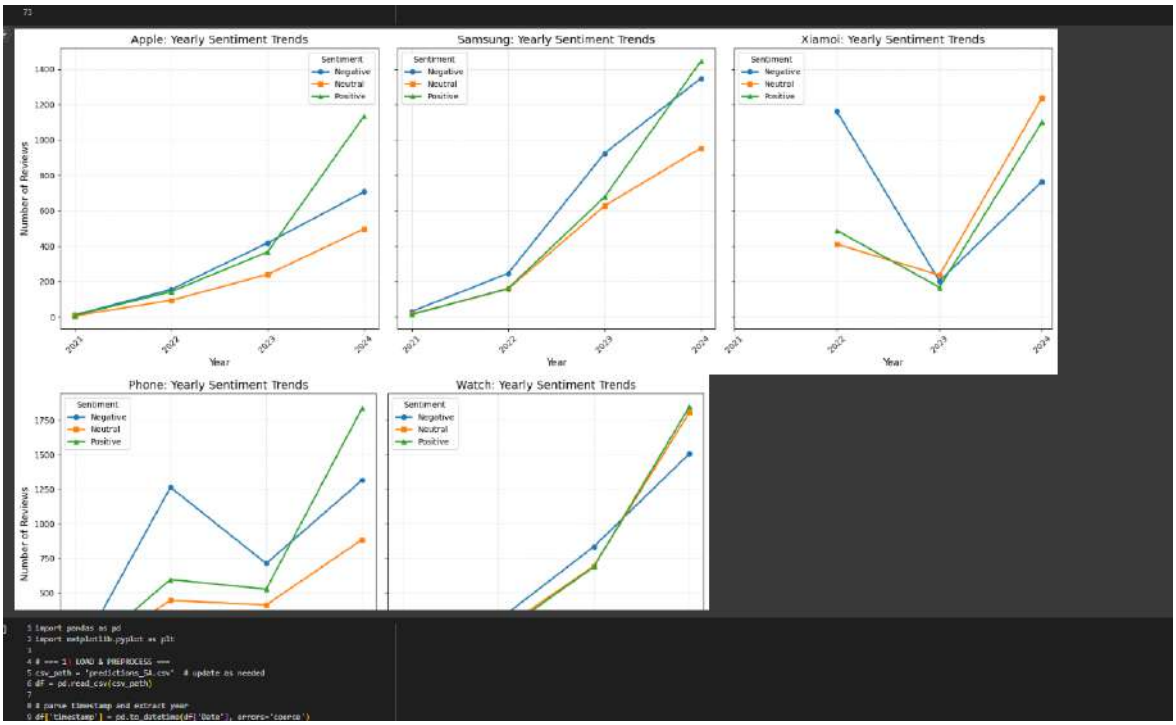
# Shade under Positive
plt.fill_between(year_pv.index, year_pv['Positive'], alpha=0.1)

# Annotate peaks
for sentiment in markers:
    y = year_pv[sentiment]
    year_max = y.idxmax()
    plt.annotate(
        f"{y.max():,}",
        xy=(year_max, y.max()),
        xytext=(0, 8),
        textcoords="offset points",
        ha='center'
    )

plt.grid(True, linestyle='--', alpha=0.4)
plt.legend(title='Sentiment')
plt.title('Yearly Sentiment Trends')
plt.xlabel('Year')
plt.ylabel('Number of Reviews')
plt.xticks(year_pv.index, rotation=0)
plt.tight_layout()
plt.show()
```



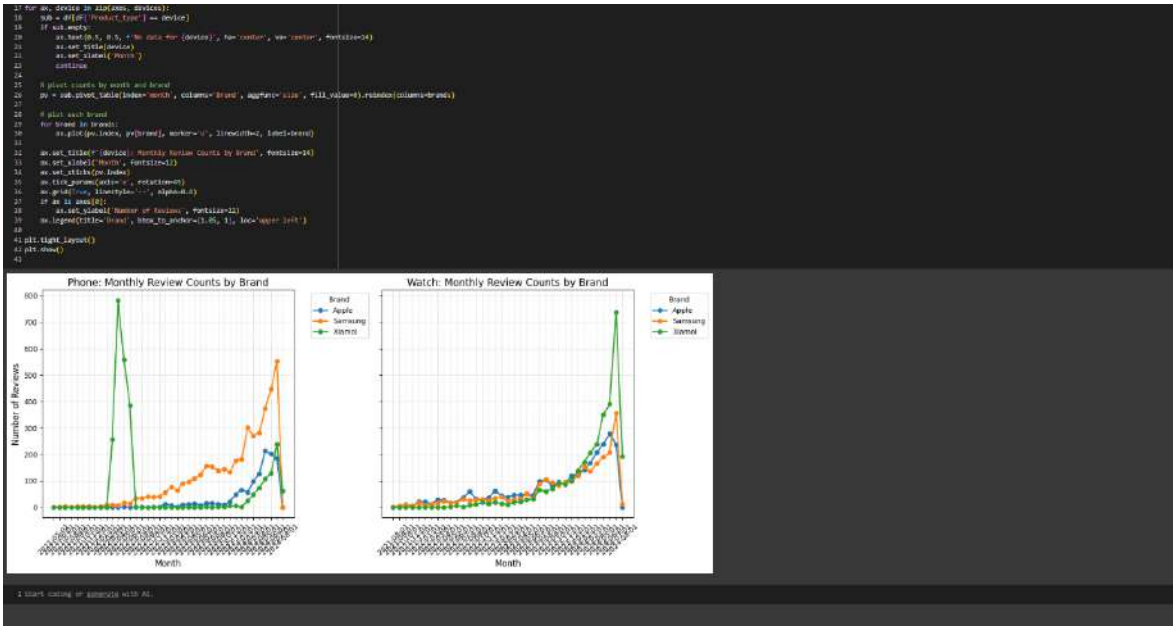
Terminal




```
import pandas as pd
import matplotlib.pyplot as plt

# === 1) LOAD & PREPROCESS ===
csv_path = 'predictions_SA.csv'  # update as needed
df = pd.read_csv(csv_path)

# parse timestamp and extract year
df['timestamp'] = pd.to_datetime(df['Date'], errors='coerce')
```



```
for ax, device in zip(axes, devices):
    sub = df[df['Product_Type'] == device]
    if sub.empty:
        ax.text(0.5, 0.5, f'No data for {device}', ha='center', va='center', fontsize=14)
        ax.set_title(device)
        ax.set_xlabel('Month')
        continue

    # pivot counts by month and brand
    pv = sub.pivot_table(index='month', columns='Brand', aggfunc='size', fill_value=0).reindex(columns=brands)

    # plot each brand
    for brand in brands:
        ax.plot(pv.index, pv[brand], marker='o', linewidth=2, label=brand)

    ax.set_title(f'{device}: Monthly Review Counts by Brand', fontsize=14)
    ax.set_xlabel('Month', fontsize=12)
    ax.set_xticks(pv.index)
    ax.tick_params(axis='x', rotation=45)
    ax.grid(True, linestyle='--', alpha=0.6)
    if ax is axes[0]:
        ax.set_ylabel('Number of Reviews', fontsize=12)
    ax.legend(title='Brand', bbox_to_anchor=(1.05, 1), loc='upper left')

plt.tight_layout()
plt.show()
```



```
Start coding or generate with AI.
```