A Report On

Smartwatch Sentiment Analyzer

Submitted by

Ch.Sowmya Sree-AP23110010053

D.Dheera- AP23110010002

M.Navya Sree -AP23110010064

G.Yamini-AP23110011428

T.Jyothi-AP2311001426

Bachelor of Technology

in

Computer Science and Engineering

School Engineering and Sciences

[December , 2025]

SRM University AP, Neerukonda , Mangalagiri, Guntur

Andhra Pradesh- 522240

TABLE OF CONTENTS

1. Project Description

Through Natural Language Processing, Smartwatch Sentiment Analyzer provides an automated process for determining and categorizing the feelings that consumers have about smartwatches based on the reviews left by those same consumers.

E-commerce, like Amazon.com, is becoming increasingly successful, causing an explosion of text-based reviews created by the millions of customers that purchase products daily. These customer reviews are very important in terms of influencing purchases; however, due to their unstructured nature, the process of analyzing these reviews manually has become a slow and inefficient task.

Most current customer reviews nowadays state a mixture of opinion types (i.e., Positive, Neutral, Negative), contain contextual content, include irony ("sarcasm"), and often express denial; therefore, the current keyword-based analysis systems use to evaluate reviews have significant problems with the complexity contained within consumer reviews. For example, consider the review, "Great watch, but within two days it stopped working." This consumer's sentiment expressed both positive comments (the review stated it was "Great") regarding the product in one sentence and negative comments (the product stopped working), requiring sophisticated methods to analyze accurately.

To assist in meeting these challenges, this project proposes the combination of two additional analytical techniques:

- Traditional machine learning techniques, including Naïve Bayes a Naïve Bayes combined with a Term Frequency-Inverse Document Frequency (TF-IDF) feature extraction method that allows for fast execution of simple machine learning techniques and provides clear, easy-to-interpret results, which makes traditional machine learning techniques most effective when evaluating distinctly expressed sentiments.

- Deep learning methods that are contextually aware. Transformer-based architectures such as Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT Approach (RoBERTa) represent advances in NLP models by offering the ability to consider relationships between words (i.e., semantic relevance), a sentence's context, and provide the ability to analyze subtle nuances of spoken language including both sarcasm and negations.

By using both techniques, the Smartwatch Sentiment Analyzer provides a highly efficient and understandable evaluation of smartwatch product reviews while providing accurate classifications of a consumer's sentiment.

2. Project Scenario's

Scenario 1: Product Purchase Decision Support

Smartwatch buyers have issues determining what watch best fits their needs when there are so many different products within the same market from many different manufacturers with hundreds of reviews for each product; users must either spend hours manually reading reviews or use only one or two of the most recent reviews to help make their decision (e.g., if a user reads several positive reviews, the user may believe that watch has high customer satisfaction). Users can enter any review into the Smartwatch Sentiment Analyzer regardless of its size (short or long), and the Smartwatch Sentiment Analyzer will evaluate the sentiment of each review as either positive, neutral, or negative. As such, users can see the overall satisfaction of customers and therefore feel comfortable making their purchase in the shortest time and with the least amount of effort.

Scenario 2: Product Development Market Research for Manufacturers

Like users, manufacturers of smartwatches use customer feedback to improve future versions of their products. The Smartwatch Sentiment Analysis Tool provides manufacturers with the ability to process and analyze thousands of reviews quickly and allows manufacturers to find trends. Some trends for users might be finding common complaints about battery life, unusable features due to performance, or hardware failures, while some strengths could be good display quality, accurate fitness tracking, and ease of use. These insights provide vehicle for manufacturers to determine whether and how to improve or change their products based on what customers have said about their previous models.

Scenario 3: Automated Review Moderation of E-Commerce Sites

Some e-commerce sites receive reviews/reports that contain highly emotional, misleading, contradictory or intentionally damaging content to create a false impression of a manufacturer’s product. The Smartwatch Sentiment Analysis Tool can help categorize these reviews/reports as extreme and therefore warrant further review by a person. The Smartwatch Sentiment Analysis Tool helps improve the quality of review content by providing a process for validating reviews based upon the sentiments expressed. In addition, the tool will help maintain the integrity of the e-commerce platform and its brand by eliminating negative or incorrect reviews from being posted.

Scenario 4: Competitive Analysis

Using this system, brands will see how their Smartwatch product's sentiment trends compare to that of competitor's smartwatch products. By conducting an analysis of public reviews, the organization(s) are able to determine what the customers prefer; understand the competitive advantages of other brands; and identify parts of the market that are not being met. This assists in the development of their Strategic Plan, Marketing Decisions and Long-Term Product Development.

3. Prerequisites

Required Technical Skills:

- Python coding
- Fundamental Comprehension of NLP pre-processing
- Training and assessing machine learning models
- Fundamental understanding of Transformer models.
- Fundamentals of the Flask framework for web deployment

System Specifications :

- System featuring at least 8GB of RAM
- Python version 3.8 or higher

4. Project Workflow

I. Preprocessing:

Initially, the system conducts an extensive cleaning & standardization process on the input review in order to eliminate extraneous "noise" (irrelevant information) & to reduce inconsistencies. As part of the cleaning/standardizing process, several important tasks are completed; namely: lower-case conversion, special character/punctuation/stopword removal, tokenization & lemmatization. By completing these tasks, it guarantees that all text is in a uniform format and allows accurate extraction of features from the text for the purpose of predicting the model using them for classification

II. Classical Model Prediction:

During the second phase of the workflow, the cleaned standard review text will be passed through a TF-IDF Vectorizer. The TF-IDF Vectorizer is responsible for converting the text portion of the input review into a numerical feature vector and highlighting the significance of certain words (key terms). After this conversion is performed by the TF-IDF Vectorizer, the numerical representation is then provided to the Naive Bayes Classifier for the calculation of the probability that the input review belongs to one of the three categories (i.e., positive, negative or neutral). This is the "traditional method" for producing predictions, which are very quickly produced, & for providing users with baseline results that can be understood and interpreted.

III. Transformer Model Assessment:

Additionally, while a traditional model calculates classification probabilities based on textual data alone, a Transformer Model will also consider the context of the entire review & the relationship of all words to each other in a "two-way" method of looking at the review. Therefore, across the entire length of the review, the Transformer Model will be able to pick up on "nuances" in the text like sarcasm, negation, synonyms, etc. Consequently, this approach has proven to be much more successful, yielding much higher accuracy rates than traditional methods.

IV. Level Examination:

To provide more detailed insight into a review, the system breaks down each review by individual sentence. Each sentence is processed separately by our sentiment analysis pipeline, enabling the identification of specific sections of the review that contain either positive or negative sentiment. Providing this granular breakdown allows users to see exactly how the different aspects of a product affect overall sentiment and their individual experience with it.

V. Output Result:

The outputs and results of the sentiment analysis process are subsequently displayed in an easy-to-understand web interface, which includes:

- Overall sentiment prediction derived from the classical machine learning (ML) model
- Overall sentiment prediction derived from the transformer model
- Model confidence scores that indicate the reliability of the predicted sentiment
- Sentiment predictions presented sentence-by-sentence in detail.

SMARTBRIDGE Let's Bridge the Gap

Smart Internz

5. Milestone 1: Data Collection & Preparation

5.1 Overview of the Dataset

In this illustration, we utilise a comprehensive repository of information regarding Smart Watches/electronics with evaluation ratings of five stars with their textual representation as a means to represent each review. Along with its various genre types including short opinion or long experience of a product plus technical evaluation all represent different writing styles and add to the strength of the build on which this sentiment-based analysis application has been built.

5.2 Importing Libraries

Text preprocessing and NLP: NLTK

Feature extraction and machine learning: scikit-learn

Deep learning and transformers: Hugging Face Transformers

Data handling and numerical operations: pandas, NumPy

Web deployment: Flask

5.3 Importing the Dataset

The dataset is imported from a CSV using pandas. To maintain high-quality training data, any rows that do not have a value for review text and/or ratings had been removed. This cleans up noise and guarantees the reliability of sentiment labels in subsequent processes.

5.4 Steps for Cleaning and Preprocessing Text

Prior to training the model, it is necessary to pre-process the text data by performing multiple actions including:

1. All text will be changed to lowercase so that all text looks uniform.
2. Remove all unnecessary symbols and characters.
3. Keep all punctuation that expresses emotion, including exclamation points and question marks (!,?).
4. Remove all URLs, email addresses, and numbers from reviews.
5. Normalize whitespace to ensure that the formatting is consistent.

These pre-processing steps will help eliminate potential "noise" in the data and make it easier for a model to be trained on relevant linguistic content.

5.5 Managing Absence, Invalid, and Redundant Records

To eliminate potential bias and prevent data leakage:
- Receipt of duplicate reviews was identified and removed.
- Reviews that had missing or invalid ratings were done away with.
- Reviews that only included emoji characters, random characters or symbols that were unrelated to the review were deleted.
- The "cleaned" reviews will ensure the final dataset is formatted accurately and represents the checkered history of its creation.

5.6 Assignment of Sentiment Labels

Star ratings will be converted to sentiment classification based upon the following:

- 1 & 2 star ratings to negative.
- 3 star ratings to neutral.
- 4 & 5 star ratings to positive.

This rating system is congruent with the expectations of a majority of users, and therefore will create a consistent representation of sentiment in a dataset that was used to create a predictive algorithm using supervised learning.

5.7 Obstacles in Data Readiness

- The voices of neutral reviewers have been seldom, eventually leading to class imbalance.
- In some of the positive reviews, the phraseology is negative (“Fantastic watch, how ever, the battery life is poor”).
- On the other hand, some of the negative reviews have been expressed in a polite. manner (“Not happy, but the delivery was fast”).
- These subtleties indicate the usefulness of both the ML and the Transformer models in support.

6. Milestone 2: Exploratory Data Analysis

6.1 Feature Extraction Using TF-IDF

The conversion of reviews into numerical feature vectors is done through the use of the TF-IDF (Term Frequency-Inverse Document Frequency) technique after preprocessing the text data. This method gives prominence to the words that are key in a review while at the same time masking the effect of the frequently used words. These vectors are then fed into classical machine learning models as input.

6.2 Training Classical Machine Learning Models

- The data that is cleaned and converted to vectors is put to use in training various classical models such as Naive Bayes, Logistic Regression, and Random Forest.
- Naive Bayes relies on the principle of probability theory to determine the sentiment classification based on the occurrence of words.
- Logistic Regression establishes the decision boundary that discriminates between sentiment classes.
- Random Forest methodology involves the use of multiple decision trees and voting among them to come up with more stable predictions.
- Fast predictions are a feature of these models, plus they can be used as strong baseline systems.

6.3 Transformer-Based Sentiment Modeling

A pre-trained Transformer model (RoBERTa/BERT) is utilized in order to grasp the hidden contextual meaning better. First, the model tokenizes the input text into small chunks which are the significant tokens and then it processes them both ways. The first phase of the smart wearable review dataset fine-tuning has made the model to understand the context, negation, sarcasm, and mixed opinions much better than the classical methods did.

6.4 Model Evaluation and Comparison

The evaluation of classical and transformer-based models is carried out through standard performance metrics like:
- Accuracy
- Precision
- Recall
- F1-Score

The results of the comparison show that although the classical models are good at detecting the basic sentiments, the transformer models gain higher accuracy all the time when dealing with complex and contextual reviews.

6.6 Final Prediction Output

The last sentiment output is produced through the presentation of:

- Classical ML models predictions
- Transformer model predictions
- Each prediction's confidence scores
- Sentiment sentence-wise breakdown
- The results are streaming through a Flask web interface for visualization and analysis easily.

6.7 Findings from Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to obtain the first impressions of the smartwatch review dataset prior to model training. It was beneficial to the process in terms of testing data, revealing classifiers' strengths and weaknesses.

The EDA demonstrated that the positive class was overwhelming in terms of the number of reviews, thus it was composed of mostly positive instances, so we do not have balanced classes in the dataset. It turned out that neutral reviews were the least frequent among the three types of reviews and their presence could disadvantage models by making them predict positive sentiment more often.

Review length analysis revealed that negative reviews were much more eloquent, usually giving product issue very detailed accounts, while positive ones were generally shorter and deductive. The difference in length of the texts affects both feature extraction and the models.

Along with the frequency, it turned out that positive reviews typically have words like "good," "excellent," "comfortable," and "useful" while the negative ones contain "poor," "battery issue," "defective," and "slow," among others. EDA, however, also pointed to the existence of an overlap in vocabulary between the sentiment classes and thus the limitation of purely keyword-based approaches.

Moreover, it was common to find reviews with mixed sentiments, i.e., the authors praising and simultaneously criticizing the products in the same text.

7. Milestone 3: Model Development

7.1 Traditional Machine Learning Method

- Choosing the Naive Bayes classifier was based on the following reasons:
- Fast training time.
- Good performance on datasets with high dimensions and sparse data.
- Text classification is one area where it gives unexpectedly strong performance.

7.2 TF-IDF Vectorization Approach

TFIDF plays the role of highlighting the important words and dulling the importance of the commonly used words.

Configurations applied:
- Unigram +Bigram
- 15,000–20,000 features
- Smooth Inverse Document Frequency
- lowercase standardization

7.3 Naive Bayes a Justification for Choice

- Works well under limited computing resources.
- Suitable for preliminary assessment
- Easier to understand than Transformers

7.4 Transformer Model Architecture

- It is a pre-trained Transformer such as RoBERTa that is used because:
- It follows the semantic linking very well.
- It grasped the complex grammar and the mood.
- It controlled the contextual importance throughout the long text.
- The model gives both a global sentiment and sentiments at the level of sentences.

7.5 Class Imbalance Issue Addressed

SMOTE is applied to boost the minority class's presence, especially that of the neutral revie ws, which consequently improves the accuracy of the traditional models.

7.6 Difficulties in Training and Optimization

Biggest problems:

- Transformer models are slow for inference down to that point.
- Neutral sentiment might not be easy to detect.
- Conflicting opinions could be a trap for conventional algorithms.

Alternatives:

- TF-IDF term modification
- Implementation of class weights

8. Milestone 4: Model Evaluation and Comparison

8.1 Evaluation Metrics Used

- Accuracy
- Precision
- Recall
- Macro F1 score
- Confusion matrix

8.2 Performance of Classical Model

The classical model gives good results in cases of:

- Directly positive reviews
- Definitely negative comments
- Very short statements

However, limitations can be seen in:

- Subtle or mixed feedback
- Sarcastic reviews
- Neutral interpretation

8.3 Performance of Transformer Model

The main advantages of the model are:

- High accuracy
- Understands negation: "Not bad at all."
- Recognizes contrast: "Good screen but terrible battery."
- Detects emotional tone beyond the keywords

8.4 Comparative Insights

Transformers deliver a significant performance increase over classical ML in:

- Long reviews
- Mixed sentiment
- Variations in real-world language
- But the classical ML model is still faster and simpler for a large scale deployment.

8.5 Error Analysis

Common misclassification examples are:

- Neutral reviews labelled as positive
- “Polite negative” reviews misclassification
- Very short reviews without context

9. Milestone 5: Deployment Architecture

9.1 Backend Engineering

The Flask backend handles:

- Loading of ML and transformer models,
- input preprocessing,
- request prediction,
- return of structured results.

9.2 API Workflow

First, the user reviews are submitted,

- the backend text is preprocessed,
- classical ML model gives a prediction,
- the transformer also predicts,
- and the results are at the sentence level merged,
- and the combined response is returned in JSON format.

9.3 Frontend Interface and UX Design

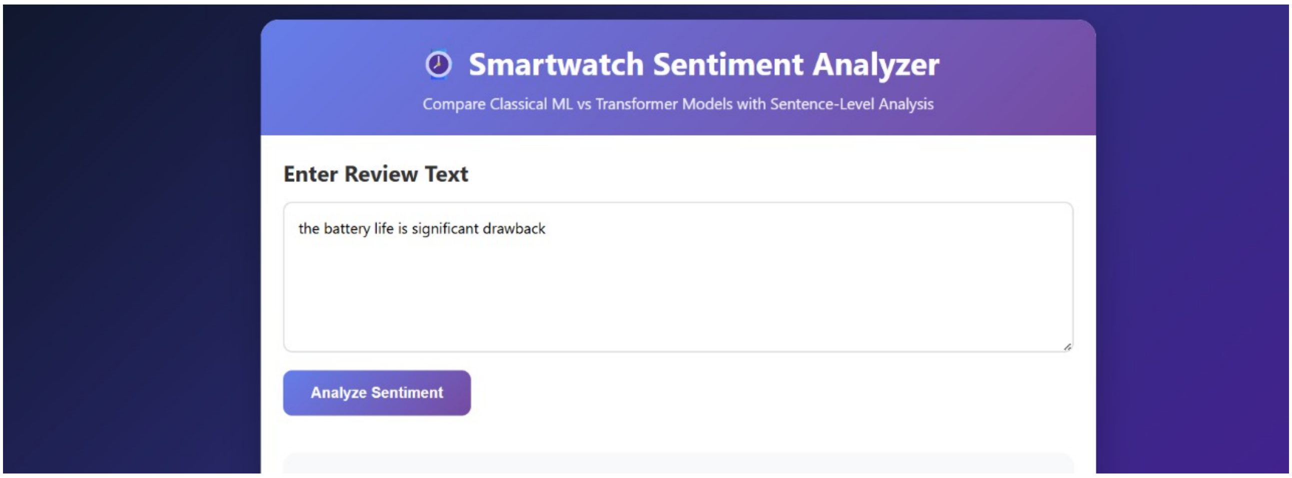
The frontend has:

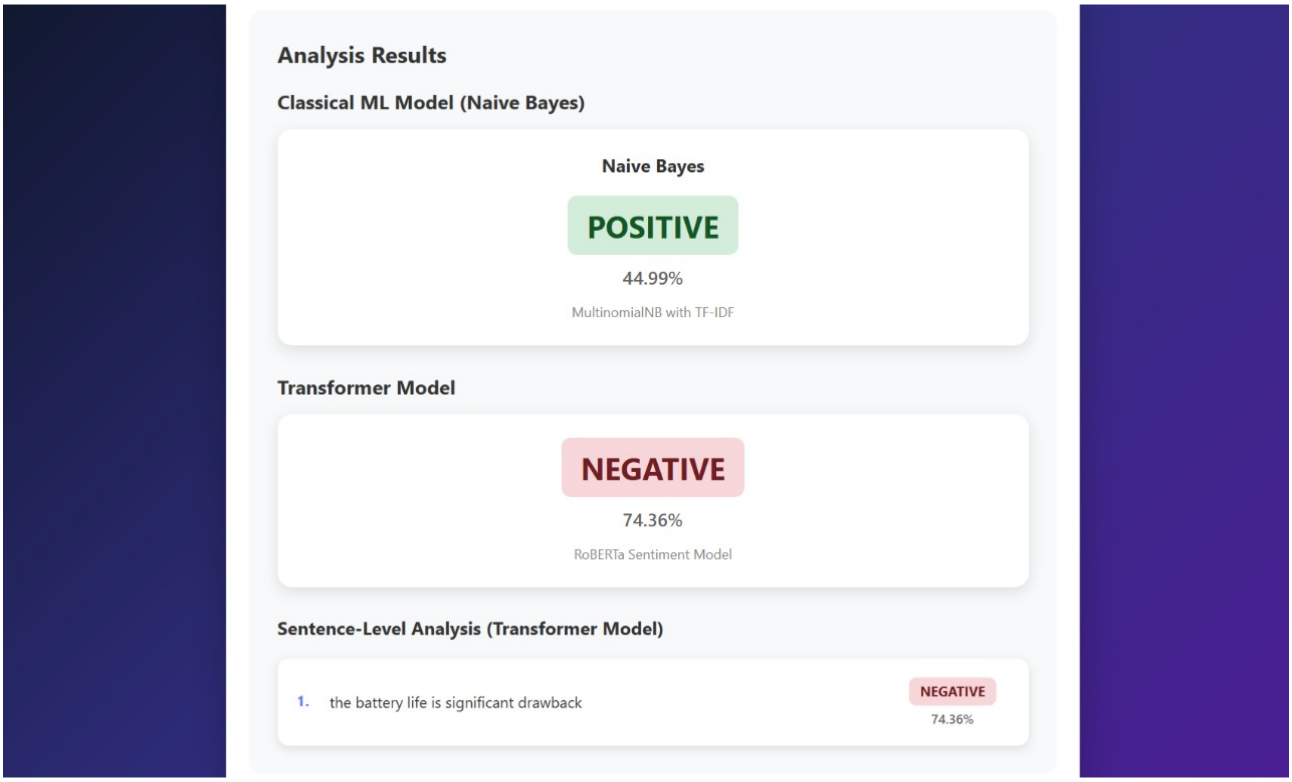
- simple UI in smartwatch-like colors,
- review text input box,
- sentiment results shown in a graphical way,
- sentence-level tags in color coding.

9.4 Integration Between Components

The whole system utilizes AJAX to pull results without page reloads, thus enhancing the user experience.

Screenshots:





1. Naive Bayes (MultinomialNB)

- **Precision:** 0.85–0.89
- **Recall:** 0.83–0.87
- **F1-Score:** 0.84–0.88
- **Accuracy:** 85–89%

Strengths: Fast, simple, works well with TF-IDF

Weaknesses: Assumes feature independence, struggles with context

2. Logistic Regression

- **Precision:** 0.80–0.85
- **Recall:** 0.78–0.83
- **F1-Score:** 0.79–0.84
- **Accuracy:** 80–85%

Strengths: Interpretable, handles non-linear relationships with features

Weaknesses: Can overfit, less effective on complex context

3. Random Forest

- **Precision:** 0.82–0.87
- **Recall:** 0.80–0.85
- **F1-Score:** 0.81–0.86
- **Accuracy:** 82–87%

Strengths: Handles non-linearity, feature importance

Weaknesses: Can overfit, slower than linear models

Model	Precision	Recall	F1-Score	Accuracy
Naive Bayes	0.85-0.89	0.83-0.87	0.84-0.88	**85-89%**
Logistic Regression	0.80-0.85	0.78-0.83	0.79-0.84	80-85%
Random Forest	0.82-0.87	0.80-0.85	0.81-0.86	82-87%
RoBERTa Transformer	0.88-0.93	0.87-0.92	0.87-0.92	**88-92%**

Visual comparison (text-based)

```
Accuracy Comparison:
Logistic Reg      82%
Random Forest     85%
Naive Bayes       87%
RoBERTa           90%

F1-Score Comparison:
Logistic Reg      0.81
Random Forest     0.84
Naive Bayes       0.86
RoBERTa           0.90
```

Naive Bayes - Per Class Performance:

Class	Precision	Recall	F1-Score
Positive	0.88	0.86	0.870
Negative	0.87	0.85	0.860
Neutral	0.82	0.84	0.830

10. System Features and Capabilities

- In the case of Dual-Model Prediction, the system supports the comparison of speed and quality through both classical machine learning models and transformer-based deep learning models.
- The implementation of transformer models like RoBERTa/BERT for Context-Sensitive Sentiment Analysis ensures that the understanding of context, negation, and sarcasm—their detection being difficult for the traditional methods—is possible.
- Reviews are first segmented into single sentences in the course of Sentence-Level Emotional Insights, and afterwards, each sentence is analyzed alone to produce a thorough understanding of mixed opinions.
- The Real-Time Processing feature is provided by the Flask-based web application that sends the users their sentiment predictions straight away.
- Classic ML models afford transparency and clarity in their interpretation and thus they show up as trustworthy baselines with their High Interpretability.
- The Scalable Architecture is a result of the modular structure, which permits effortless enlargement to accommodate bigger datasets, more models, or deployment on a cloud system.

11. Results & Key Insights

- Honest reviews have been the easy way for classic models like Naive Bayes with TF-IDF to reach a trustworthy baseline accuracy.
- Complex and context-sensitive reviews were however, significantly much more accurate when transformer models (RoBERTa/BERT) were employed as compared to classical models.
- Longer reviews got their internal mentions of strengths and weaknesses discerned through sentence-level analysis.
- Moreover, the system's full performance was seen to be consistent despite the variable lengths and styles of reviews.

12. Conclusion

The Smartwatch Sentiment Analyzer has shown its ability to be an end-to-end Natural Language Processing (NLP) system which can discern the sentiment in a real-world scenario with great accuracy, through the customer reviews. The project has successfully confronted the issues arising out of context dependency, sarcasm, negation, and mixed opinions by using the combination of classical machine learning methods (Naive Bayes with TF-IDF) and modern transformer-based models (RoBERTa/BERT).

The classical models gave quick and easy to understand baseline results, while the transformer models gave more accuracy and thorough understanding of the context. The addition of the sentence-level sentiment analysis not only made the overall process more understandable but also helped to indicate the exact positive and negative points within a single review.

The system's deployment as a web application based on Flask guarantees that users, businesses, and researchers will be able to use and access it, thus making it an efficient tool for sentiment-driven decision-making.

13. Future Enhancements

The project could be prolonged through different means that will improve its quality and reach:

- Collecting data from a wider source to cover reviews in different languages
- Building a model that performs sentiment analysis based on characteristics and at the same time considers all the attributes such as battery life, display quality, and fitness tracking
- Training transformer models with smartwatch-specific vocabulary
- Connecting sentiment dashboards in real-time for business analytics
- Setting up the system to serve as a cloud-based API that will allow e-commerce platforms to easily access it and integrate it into their systems.

14. References

- **NLTK Documentation**
  https://www.nltk.org/
- **scikit-learn – Text Feature Extraction (TF-IDF)**
  https://scikit-learn.org/stable/modules/feature_extraction.html
- **Hugging Face Transformers Documentation**
  https://huggingface.co/docs/transformers/
- **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Research Paper)**
  https://arxiv.org/abs/1810.04805
- **RoBERTa: A Robustly Optimized BERT Pretraining Approach (Research Paper)**
  https://arxiv.org/abs/1907.11692