Smart Watch Sentiment Analyzer

Team 9 Genovate

Team Members	
AP23110010053	Ch.Sowmya Sree
AP23110010002	D.Dheera
AP23110011448	G.Yamini
AP23110011426	T.Jyothi
AP23110010064	M.Navya sree

Problem Statement

- Online smartwatch reviews on Amazon are large in volume and unstructured.
- Manual analysis is slow and cannot capture overall customer sentiment effectively.
- Traditional bag-of-words models struggle with context, sarcasm, and negation (e.g., *“not good”*).

Objectives

Build a Smartwatch Sentiment Analyzer to classify reviews as Positive / Negative / Neutral.

Compare Naive Bayes (classical ML) with a Transformer model (RoBERTa/BERT).

Provide a simple web interface for instant sentiment prediction.

Offer sentence-level insights to show which parts of a review are positive or negative.

Proposed Solution

- Use Amazon gadget review dataset as a proxy for smartwatch reviews.
- Convert star ratings → sentiment labels and train a Naive Bayes + TF-IDF model.
- Use a pre-trained Transformer model (RoBERTa/BERT) for contextual sentiment understanding.
- Deploy both models in a Flask web application with a clean, watch-themed UI.

Workflow

1. Data Collection
2. Data Preprocessing
3. Sentiment Labeling
4. Feature Extraction
5. Model Training
6. Model Evaluation
7. Model Selection
8. Web App Development
9. Deployment

Dataset

- Source: Amazon gadget reviews dataset (includes smartwatch-like devices).
- Columns used: reviews.text, reviews.rating, product metadata.
- Several thousand reviews available for training/testing.
- Sentiment labels based on ratings:

1) 1–2 → Negative
2) 3 → Neutral
3) 4–5 → Positive

Methodology

- Classical ML Approach: Apply TF-IDF vectorization and train baseline models such as Logistic Regression, Random Forest, and SVM.
- Transformer Approach: Use a BERT tokenizer and fine-tune a pre-trained BERT model to capture contextual sentiment.
- Data Splitting: Divide the dataset into training and testing sets to ensure objective and fair performance comparison.
- Model Evaluation: Assess all models using accuracy, precision, recall, and F1-score.

Technologies Used

Programming : Python for preprocessing, modeling, and backend logic.

Backend : Flask REST API for /analyze and HTML rendering.

ML / NLP : scikit-learn (TF-IDF, Naive Bayes, metrics), pandas, numpy, transformers (Hugging Face models + tokenizer) and nltk (sentence splitting)

Frontend: HTML, CSS (watch-themed UI) and JavaScript (fetch API calls & dynamic rendering)

Key Findings

- Classical approaches work well for clear and straightforward sentiment patterns.
- Transformer-based models provide better understanding of context, tone, sarcasm, and mixed opinions.
- Contextual models reduce misclassification in reviews with subtle or complex phrasing.

Results

Model	Precision	Recall	F1-Score	Accuracy
Naive Bayes	0.78–0.82	0.75–0.80	0.76–0.81	77–82%
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%
SVM	0.83–0.88	0.81–0.86	0.82–0.87	83–88%
RoBERTa Transformer	0.88–0.93	0.87–0.92	0.87–0.92	88–92%

Conclusion

- Built a complete end-to-end Smartwatch Sentiment Analyzer—from preprocessing to web deployment.
- Demonstrated how:
  a. Naive Bayes + TF-IDF gives a strong classical baseline.
  b. Transformer models provide deeper, context-based understanding.
- Web app allows easy sentiment analysis and model comparison for any review.

Future Scope

- Expand dataset and add multi-language support.
- Fine-tune Transformer specifically for smartwatch reviews.
- Add aspect-based sentiment analysis (battery, display, fitness tracking).
- Add interpretability features (highlight influential words).
- Deploy cloud-based API for integration into e-commerce dashboards.

Thank you