Technical Report
Sentiment Analysis on Amazon Baby Products Reviews

Rebeca (Ruijia) Gu

2026.02.22

Contents

1. Project Overview

This project implements a sentiment analysis system for Amazon product reviews in the Baby category. The objective is to classify customer reviews as positive or negative based on their textual content and associated ratings. The study explores both traditional machine learning approaches and modern transfer learning techniques, comparing their performance, computational efficiency, and practical applicability.

2. Dataset Description

2.1 Data Source

The dataset was obtained from the Amazon Review Data repository compiled by Julian McAuley (UCSD).
Source:
Amazon Review Data Repository
http://jmcauley.ucsd.edu/data/amazon/
Category used:

- Baby Products

Each review contains:

- Review text
- Star rating (1–5 stars)

To formulate the sentiment classification problem, ratings were converted into binary labels:

- Low ratings ($\leqslant$ threshold) $\rightarrow$ Negative (class 0)
- High ratings (> threshold) $\rightarrow$ Positive (class 1)

This transformation enabled supervised learning.

2.3 Data Preprocessing

Initial preprocessing steps included:

- Loading 100,000 samples from the JSON file
- Removing entries with null values in either review text or rating
- Resetting dataframe indices after cleaning

3. Exploratory Data Analysis

3.1 Sentiment Distribution

The ratings were converted to binary sentiment labels:

- Negative sentiment (class 0): ratings $\leqslant$ 3
- Positive sentiment (class 1): ratings > 3

Analysis revealed an imbalanced dataset with significantly more positive reviews than negative ones, which has important implications for model training and evaluation.

3.2 Text Analysis

Word Clouds

Visual analysis through word clouds showed distinct linguistic patterns:

- **Negative reviews**: Featured words related to disappointment, functionality issues, and product problems
- **Positive reviews**: Contained terms expressing satisfaction, recommendations, and product benefits

N-gram Analysis

Initial tokenization using simple split approach revealed that the most frequent n-grams were primarily stopwords, confirming the need for proper preprocessing.

4. Text Preprocessing Pipeline

A comprehensive preprocessing function was implemented with the following steps:

1. **Lowercase conversion**: Standardizing text case
2. **Non-alphabetic character removal**: Using regex [^a-z\s] to remove punctuation and symbols
3. **Stopword removal**: Utilizing NLTK's English stopword corpus, with special handling to retain the word "no" for sentiment preservation
4. **Empty review handling**: Removing reviews that became empty after preprocessing

5. Feature Extraction

5.1 TF-IDF Vectorization

The TfidfVectorizer was configured with the following parameters:

- max_df=0.95: Ignore terms that appear in more than 95% of documents
- min_df=3: Ignore terms that appear in fewer than 3 documents
- max_features=2500: Limit vocabulary size to 2500 most important features
- ngram_range=(1,2): Include both unigrams and bigrams (trigrams were tested but introduced noise)

5.2 Feature Analysis with Chi-Square Test

The χ^2 (chi-squared) statistic was employed to analyze the relationship between vocabulary terms and the target variable. Results showed clear sentiment-related signal in the dataset, confirming that bag-of-words representation is a reasonable approach. The top discriminative features included words strongly associated with either positive or negative sentiment.

6. Machine Learning Approach: Logistic Regression with TF-IDF

6.1 Model Configuration

- **Algorithm**: Logistic Regression
- **Class weight**: Balanced (to address dataset imbalance)
- **Hyperparameter tuning**: Grid search over C values [0.01, 0.1, 1, 10, 100, 1000, 10000]

- **Scoring metric**: f1_macro (to ensure equal attention to both classes)
- **Cross-validation**: 5-fold

6.2 Threshold Optimization

Probability thresholds were optimized specifically for the minority class (class 0) to improve precision:

- Thresholds tested: 0.05 to 0.5 (50 values)
- Optimal threshold selected based on precision for class 0

6.3 Results Analysis

Performance Metrics

The classification report revealed:

- **Class 1 (positive)**: High precision and recall
- **Class 0 (negative)**: Lower precision, indicating many false positives

The confusion matrix showed that the model tends to over-predict the majority class, resulting in false negatives for class 1 and false positives for class 0.

ROC-AUC Analysis

The ROC curve demonstrated excellent discriminative power with an AUC of 0.909, indicating that the model has strong ability to distinguish between positive and negative sentiments.

Impact of Neutral Reviews

Comparative analysis with and without rating=3 reviews showed:

- Removing neutral reviews improved performance for the majority class
- However, the model still struggled with the minority class due to class imbalance

7. Transfer Learning Approach: DistilBERT

7.1 Model Selection

DistilBERT-base-uncased was chosen for transfer learning due to:

- Smaller size than BERT-base (40% fewer parameters)
- Retains 97% of BERT's performance
- More efficient for fine-tuning on a single GPU

7.2 Implementation Details

Tokenization

- Maximum sequence length: 128 tokens
- Padding: max_length
- Truncation: enabled

Training Configuration

- **Training epochs**: 3
- **Batch size**: 16 (both train and evaluation)
- **Evaluation strategy**: Per epoch
- **Model selection**: Best model based on f1_macro score

- **Device**: CUDA-enabled GPU

7.3 Results

Classification Performance

The DistilBERT model showed significant improvement:

- **Class 0 precision**: Markedly higher than logistic regression
- **Overall F1-score**: Improved macro F1 score
- **Context understanding**: Better capture of nuanced linguistic patterns

ROC-AUC Comparison

While the transfer learning model achieved lower ROC-AUC than logistic regression, this paradox is explained by:

- Better performance at the chosen decision threshold
- Higher recall for the minority class
- Less well-separated probability distributions across all thresholds

8. Comparative Analysis: Traditional ML vs. Transfer Learning

8.1 Performance Comparison

Aspect	Logistic Regression + TF-IDF	DistilBERT
Class 0 Precision	Low	Significantly Higher
Class 0 Recall	Moderate	Improved
Overall F1	Good	Better
ROC-AUC	0.909	Slightly Lower
Context Understanding	Limited	Rich

8.2 Operational Considerations

Computational Requirements

- **Logistic Regression**:
  - CPU-based training
  - Fast training times
  - Minimal memory requirements
  - Easy deployment
- **DistilBERT**:
  - GPU-dependent (approximately 2 hours training time)
  - High memory consumption
  - Significant computational cost
  - Complex deployment infrastructure

8.3 Trade-off Analysis

Traditional ML Advantages:

- Lower computational cost
- Faster inference
- Easier deployment
- More interpretable

- Stable with smaller datasets

Transfer Learning Advantages:

- Superior context understanding
- Better handling of nuanced language
- Improved minority class performance
- More sophisticated feature representation

9. Challenges and Limitations

9.1 Class Imbalance

The dataset's inherent imbalance negatively affects minority class performance, even with balanced class weights and stratified sampling.

9.2 Bag-of-Words Limitations

- Ignores word order and context
- High dimensionality
- Cannot capture semantic relationships

9.3 Transfer Learning Constraints

- Requires substantial computational resources
- Risk of overfitting with limited data
- Less interpretable than traditional models
- Higher deployment and maintenance costs

10. Conclusions

10.1 Key Findings

1. **Traditional ML Effectiveness**: Logistic Regression with TF-IDF provides solid performance (AUC 0.909) with minimal computational requirements, making it suitable for resource-constrained environments.
2. **Transfer Learning Superiority**: DistilBERT significantly improves minority class detection and captures contextual nuances that traditional methods miss.
3. **Cost-Performance Trade-off**: The choice between approaches should be guided by:
    - Available computational resources
    - Business requirements for minority class accuracy
    - Deployment constraints
    - Expected ROI

10.2 Recommendations

- **For production with limited resources**: Logistic Regression with TF-IDF offers an excellent balance of performance and efficiency
- **For high-stakes applications requiring nuanced understanding**: Transfer learning justifies the additional computational cost
- **For balanced datasets**: Consider ensemble approaches combining both methods

10.3 Future Work

1. Experiment with other Transformer architectures (RoBERTa, ALBERT)
2. Implement data augmentation techniques for minority class
3. Explore ensemble methods combining traditional and deep learning approaches
4. Investigate active learning for efficient annotation of negative reviews
5. Deploy model in production and monitor real-world performance

11. Technical Stack

- **Programming Language**: Python 3
- **Data Processing**: pandas, numpy, re, nltk
- **Machine Learning**: scikit-learn
- **Deep Learning**: transformers, torch, datasets
- **Visualization**: matplotlib, wordcloud
- **Environment**: Google Colab with GPU acceleration