Technical Report

Sentiment Analysis on Automotive Amazon Reviews

Rebeca (Ruijia) Gu

2026.02.22

Contents

1. Introduction

The objective of this project was to build and evaluate a binary sentiment classification model using Amazon automotive product reviews. The sentiment label was derived from the star rating associated with each review.

The project followed a structured Natural Language Processing (NLP) pipeline including:

- Exploratory Data Analysis (EDA)
- Text preprocessing and cleaning
- Feature extraction using TF-IDF
- Model training with Support Vector Machines (SVM)
- Hyperparameter tuning
- Data augmentation via back translation
- Performance evaluation and analysis

The focus was on classical machine learning methods and augmentation strategies.

2. Dataset Description

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:

- Automotive

Each review contains:

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

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

- Low ratings (≤ threshold) → Negative (class 0)
- High ratings (> threshold) → Positive (class 1)

This transformation enabled supervised learning.

3. Exploratory Data Analysis (EDA)

3.1 Rating Distribution

The dataset showed a clear imbalance toward high ratings (4–5 stars). Negative reviews were significantly fewer than positive ones.

This imbalance was expected to influence model performance, particularly recall for the minority class.

3.2 Class Distribution

After binarization:

- Positive class represented the majority.
- Negative class represented a minority.

This imbalance motivated the later use of data augmentation.

3.3 Most Frequent N-grams

The initial n-gram frequency analysis was performed on the raw text before stopword removal. As a result, the most frequent unigrams and bigrams corresponded primarily to common English function words (e.g., “the”, “and”, “of the”).

Although these results do not provide direct sentiment insights, they confirm the typical syntactic structure of English text and justify the need for subsequent preprocessing steps such as stopword removal before feature extraction.

3.4 Word Clouds

Separate word clouds for positive and negative reviews showed:

- Positive reviews emphasized performance and usability.
- Negative reviews emphasized malfunction and dissatisfaction.

This visual analysis confirmed lexical differences between classes.

3.5 Word2Vec Embeddings Visualization

A Word2Vec model was trained on the corpus.

Selected words and their top similar words were projected into 2D space.

Observations:

- Positive adjectives clustered together.
- Negative descriptors formed separate clusters.
- Semantic similarity was preserved in embedding space.

This confirmed that the dataset contains meaningful semantic structure.

4. Data Cleaning and Preprocessing

The following preprocessing steps were applied:

- Lowercasing
- Removal of punctuation
- Stopwords removal
- Tokenization
- Creation of a processed review column

These steps aim to reduce noise while preserving meaningful lexical information.

5. Feature Engineering

TF-IDF (Term Frequency – Inverse Document Frequency) was used to convert textual data into numerical feature vectors.

Advantages:

- Efficient representation
- Suitable for high-dimensional sparse data
- Penalizes overly frequent words

Limitations:

- Ignores word order
- Does not capture contextual meaning
- Treats words independently

6. Model Selection – Support Vector Machine (SVM)

An SVM classifier was selected due to:

- Strong performance in text classification
- Robustness in high-dimensional feature spaces
- Good baseline performance for sentiment analysis

Hyperparameter tuning was conducted using GridSearchCV to optimize the regularization parameter C.

7. Baseline Results

The baseline TF-IDF + SVM model achieved:

- Approximately 80% overall accuracy
- High precision for the positive class
- Lower recall for the negative class

The confusion matrix showed:

- Many negative reviews misclassified as positive
- Clear bias toward the majority class

This confirmed the impact of class imbalance.

8. Data Augmentation – Back Translation

8.1 Motivation

Due to the imbalance in the dataset, data augmentation was applied only to negative reviews to increase minority class representation.

8.2 Method

Back translation was implemented:
English → Spanish → English
Using MarianMT transformer-based translation models.
The goal was to generate paraphrased versions of negative reviews while preserving sentiment.

8.3 Batch Processing

Because translation models are computationally expensive, a batch-based function was implemented.
Instead of translating sentences individually, texts were processed in batches (e.g., batch_size=32), which:

- Reduced overhead
- Improved GPU utilization
- Decreased overall execution time

8.4 Sample Size Limitation

Initially, 5000 negative samples were selected for augmentation. However, processing time was excessive even with GPU acceleration.
Finally, only 2000 samples were augmented to maintain practical execution time.

9. Evaluation After Augmentation

After retraining the SVM model with augmented data, the final evaluation metrics remained almost identical to those obtained without data augmentation.
The confusion matrices before and after augmentation show only negligible differences, and overall performance metrics, including accuracy, macro F1-score, and minority class precision and recall, did not meaningfully change.
In particular, the detection performance for the negative class (class 0) remained stable, with no observable improvement in precision, recall, or F1-score. This indicates that the augmented samples did not introduce additional discriminative information that the TF-IDF + SVM model could effectively exploit.
Therefore, data augmentation via back-translation did not lead to a measurable enhancement in minority class detection in this experimental setting.

10. Analysis of Results

Key observations:

1. Augmentation slightly improved average validation performance.
2. However, minority recall remained relatively weak.
3. The computational cost of back translation was high relative to performance gains.

Possible explanations:

- TF-IDF cannot fully leverage contextual variations introduced by paraphrasing.
- Augmented samples may be lexically similar to original ones.
- Class imbalance might be better addressed using class weighting.

11. Technical Challenges and Lessons Learned

11.1 Computational Cost

Back translation was significantly slower than expected because:

- It is autoregressive.
- It requires two translation passes.
- Generation is sequential.

Even with GPU, generation remained time-consuming.

11.2 Data Alignment Errors

During augmentation, inconsistent sample sizes between features and labels caused training errors.

This highlighted the importance of maintaining strict alignment between X and y after augmentation.

12. Successes

- Successful implementation of a full NLP pipeline.
- Proper hyperparameter tuning.
- Effective use of batching for augmentation.
- Clear experimental comparison between baseline and augmented models.
- Detailed evaluation using confusion matrices and classification reports.

13. Conclusions

1. TF-IDF + SVM provides a strong and efficient baseline.
2. Class imbalance significantly affects minority recall.
3. Back translation introduces variability but is computationally expensive.
4. The improvement from augmentation was limited relative to cost.
5. For classical models, simpler imbalance strategies may be more efficient.

14. Future Work

- Experiment with different thresholds for sentiment labeling.
- Try lighter augmentation techniques (e.g., synonym replacement).
- Explore contextual models in future experiments.

- Compare augmentation against cost-sensitive learning methods.

Final Reflection

This project demonstrates that increasing computational complexity does not necessarily guarantee proportional improvements in model performance.

Careful evaluation of cost-benefit trade-offs is essential in applied NLP tasks. In this case, classical machine learning methods proved efficient, while heavy augmentation strategies offered limited additional benefit.