Sentiment Analysis of Amazon Reviews: Final Report

Abstract

This project focuses on applying machine learning and deep learning techniques to automate the sentiment analysis of Amazon product reviews. By comparing multiple models on both unbalanced and balanced datasets, I highlight the significant impact of data preprocessing and class distribution on model performance.

Introduction

Sentiment analysis enables businesses to understand customer feedback at scale. In this project, I classified Amazon product reviews into Positive, Neutral, or Negative sentiment. I explored the performance of both traditional machine learning models and deep learning models under different data conditions.

Dataset Description

Source: Amazon product review dataset from Kaggle

Size: ~3000 reviews

Labeling: Star ratings were mapped to sentiment labels:
Positive (>=4 stars)
Neutral (=3 stars)
Negative (<3 stars)

Initial dataset was highly imbalanced, with a majority of Positive reviews.

Preprocessing Steps

Removed null and missing values

Standardized text: lowercase, removed punctuation, stopwords, URLs

For traditional ML models: Applied TF-IDF vectorization (5000 features)

For the LSTM model: Tokenized and padded text sequences (fixed length 100)

Methodology

Traditional Machine Learning Models

In this project, I explored four traditional machine learning models for sentiment classification:

Logistic Regression: A linear model used for classification tasks. It estimates the probability of a class label and works well when the relationship between input features and output labels is linear.

Random Forest: An ensemble method that builds multiple decision trees and averages their results. It captures complex feature interactions and is robust to overfitting.

Naive Bayes: A probabilistic classifier based on Bayes' theorem, assuming feature independence. It performs efficiently on text tasks but may struggle with more complex language patterns.

Linear SVM: A discriminative classifier that finds the optimal hyperplane for class separation. It is strong in high-dimensional spaces but sensitive to overlapping classes.

Deep Learning Model

LSTM (Long Short-Term Memory Network): LSTM networks are designed to capture long-range dependencies in sequential data like text. In this project, I used LSTM with an embedding layer to better understand the sequential context of the reviews. However, its performance depends heavily on data size and diversity.

Handling Class Imbalance

Applied random oversampling to duplicate examples from minority classes (Neutral and Negative).

Balanced all classes to have an equal number of samples.

Retrained all models on the balanced dataset for fairer evaluation.

Experiments and Results

Traditional Models (Unbalanced)

Model	Accuracy	Precision		Recall	F1 Score
Random Forest	0.706	0.703	0.706	0.699	
Logistic Regression		0.703	0.701	0.703	0.696

Naive Bayes	0.691	0.672	0.691	0.673
Linear SVM	0.669	0.675	0.669	0.666

LSTM Model (Unbalanced)

Class	Precision	Recall	F1 Score
Negative	0.48	0.34	0.40
Neutral	0.29	0.27	0.28
Positive	0.73	0.84	0.78
Overall Accuracy		0.65	

Traditional Models (Balanced)

Model	Accuracy	Precision	Recall	F1 Score
Logistic Regression	0.860	0.870	0.860	0.860
Random Forest	0.890	0.910	0.890	0.890
Naive Bayes	0.820	0.830	0.820	0.820
Linear SVM	0.900	0.910	0.900	0.900

LSTM Model (Balanced)

Class	Precision	Recall	F1 Score
Negative	0.73	0.49	0.58
Neutral	0.88	0.56	0.69
Positive	0.55	0.85	0.67
Overall Accuracy		0.65	

Performance Analysis

Balancing the dataset had a substantial impact, especially for the traditional machine learning models. Before balancing, all models were biased toward the Positive class, which was the majority in the unbalanced dataset. This led to lower recall and F1 scores for Neutral and Negative classes.

After applying random oversampling to create a balanced dataset, the traditional models—Logistic Regression, Random Forest, Naive Bayes, and Linear SVM—showed significant improvement. Notably, Linear SVM and Random Forest achieved near 90% accuracy and F1 scores, clearly benefiting from balanced class distributions.

In contrast, while balancing helped the LSTM model improve recall and F1 scores for the Neutral and Negative classes, the overall gain in accuracy was modest. This is likely because deep learning models like LSTM typically require larger and more varied datasets to reach their full potential.

Conclusion

This project demonstrates that while deep learning models offer power and flexibility, traditional machine learning models combined with proper preprocessing and balancing can outperform them, especially on moderately sized datasets. Data preparation had a larger impact on performance than model complexity. Effective preprocessing and class balancing allowed traditional models like Linear SVM and Random Forest to reach very high accuracy and F1 scores without the heavy training costs associated with deep learning.

Future Work

Experiment with transformer-based models like BERT or DistilBERT

Explore synthetic oversampling methods like SMOTE

Add explainability methods (SHAP, LIME) to understand model decisions

Expand dataset size and diversity

Perform detailed hyperparameter tuning for further optimization