Sentiment analysis project

Team Name: SentiLoop

Omar Ajamieh

Donya Albarbarawi

May 31, 2025

Abstract

The exponential growth of Internet based platforms, including social media, blogs, and review sites, has led to a vast influx of user-generated content reflecting opinions and sentiments on everyday topics. Sentiment analysis, also known as opinion mining, is a key Natural Language Processing (NLP) technique used to identify, extract, and analyze subjective information from text data. It plays a vital role in understanding public opinion and is widely used in areas such as product feedback, political forecasting, and customer experience monitoring. Despite its broad applications, sentiment analysis faces several significant challenges, including handling sarcasm, domain dependency, multilingual content, and context understanding. These issues hinder the accurate classification of sentiment polarity and overall opinion interpretation. Addressing these challenges using machine learning and deep learning approaches is essential for building robust and reliable sentiment analysis systems capable of supporting informed decision-making for businesses, governments, and individuals.

1 Introduction

This project focuses on sentiment analysis of customer reviews on Amazon, with the goal of understanding public opinion about products by classifying user sentiments as positive, negative, or neutral. Customer reviews represent a rich source of information that can be leveraged to improve product quality, enhance services, and support data-driven marketing decisions.

In this project, we applied specific machine learning models to process text and accurately analyze sentiment. After training the models on real-world Amazon review data, they demonstrated excellent performance and achieved accurate and effective results in sentiment classification. This success highlights the importance of using artificial intelligence techniques to analyze customer opinions and transform unstructured textual data into actionable insights.

2 Literature Review

Sentiment analysis has become a crucial area of research within natural language processing due to its applications in understanding public opinion across various domains such as e-commerce, social media, and customer feedback analysis. Many studies have focused on using machine learning techniques to improve the accuracy of sentiment classification.

Traditional machine learning models such as logistic regression and support vector machines (SVM), including the Linear Support Vector Classifier (LinearSVC), have been widely used due to their simplicity and effectiveness in text classification tasks. For example, Pang (2002) demonstrated the effectiveness of SVM and logistic regression models in classifying movie reviews with high accuracy. Random Forest Classifiers have also been employed in sentiment analysis tasks, offering robust performance by combining multiple decision trees to improve classification accuracy and reduce overfitting . Studies such as those by Zhang et al. (2018) showed that Random Forest can handle large-scale datasets and noisy text data effectively, making it suitable for sentiment classification in e-commerce reviews .

More recent approaches have explored deep learning models; however, classical machine learning models remain competitive when combined with appropriate feature engineering such as TF-IDF and n-grams . In this project, we utilize LinearSVC, Logistic Regression, and Random Forest classifiers due to their proven performance and efficiency on textual data, particularly for classifying sentiment in Amazon product reviews.

3 Methodology

In this section, we describe the detailed steps followed to collect, preprocess, and analyze the Amazon review data for sentiment classification. We explain how the data was prepared, the models selected and trained, and the evaluation metrics used to assess model performance. The goal is to provide a clear and reproducible description of the methods applied in this project:

- **Data Collection:** The dataset was collected from Amazon product reviews, which includes user ratings and textual feedback. The dataset originally contained multiple columns, including an id column, which was dropped as it was not relevant for the analysis.
- **Data Cleaning and Preprocessing:**To prepare the data for modeling, the ratings were grouped into three sentiment categories: Positive, Negative, and Neutral. A new column named Sentiment was created to reflect these categories.

  The text preprocessing involved the following steps:

  Removing special characters, keeping only alphabetic characters and spaces.

  Converting all text to lowercase.

  Tokenizing the text by splitting it into individual words.

Removing stop words (common words that do not contribute much meaning).

This preprocessing function was applied to the Review column, and tokenization was performed in one step.

- **Feature Engineering:** The sentiment labels were encoded into numeric values to be used . To address class imbalance, oversampling was applied only to the Neutral class. Additionally, the numeric labels could be decoded back to their original sentiment.
- **Model Selection and Training:** Three machine learning classifiers were chosen for this project: Linear Support Vector Classifier (LinearSVC), Logistic Regression, and Random Forest. These models were trained on the preprocessed dataset to classify the sentiment of Amazon reviews.
- **Evaluation Metrics:** The performance of the models was evaluated using several metrics including the Confusion Matrix, Precision, Recall, F1-score, and Accuracy. Cross-validation .

4 Data Description

- **Sources:**The dataset used in this project was obtained from kaggle, Amazon product reviews. It includes customer feedback in the form of text reviews and corresponding numerical ratings. The dataset is publicly available and widely used for sentiment analysis tasks.
- **Size:**The dataset consists of 60,000 entries and 3 columns:

  Id: Unique identifier for each review (dropped during preprocessing)

  Review: Textual content of the customer review

  Rating: Numerical rating from 1 to 5
- **Attributes:** Review: Object (text) – The main input feature containing customer-written reviews.

  Rating: Integer (1–5) – The original rating score provided by the user.

  Sentiment (created during preprocessing): Categorical – The target label derived from the Rating column. Reviews with a rating of 4 or 5 were labeled Positive, 1 or 2 as Negative, and 3 as Neutral.
- **Preprocessing Steps:** Several preprocessing steps were performed to prepare the dataset for machine learning:

  Column Drop: The Id column was dropped since it does not contribute to sentiment analysis.

  Label Generation: A new Sentiment column was created by grouping rating values into three categories (Positive, Negative, Neutral).

Text Cleaning:

Special characters were removed, retaining only alphabetic characters and spaces.

All text was converted to lowercase.

Reviews were tokenized into words.

Common stopwords were removed.

Handling Missing Data: The dataset had no missing values.

Handling Duplicates: Duplicate entries were checked and removed if found.

Label Encoding: Sentiment labels were encoded into numerical values for training.

Class Balancing: To address class imbalance, oversampling was applied to the underrepresented Neutral class using techniques such as SMOTE.

5 Results and Discussion

This section presents the outputs my models:

5.1 Results

- Random Forest The Random Forest classifier achieved the highest overall performance, with an accuracy of 94.
- Logistic Regression The Logistic Regression model showed a moderate performance with an average cross-validation accuracy of 83.4 Its macro-averaged precision, recall, and F1-score were approximately 81 each.
- LinearSVC The LinearSVC model achieved an average accuracy of around 82.5 based on cross-validation scores. Although slightly lower than Logistic Regression, it still provided competitive results for sentiment classification.

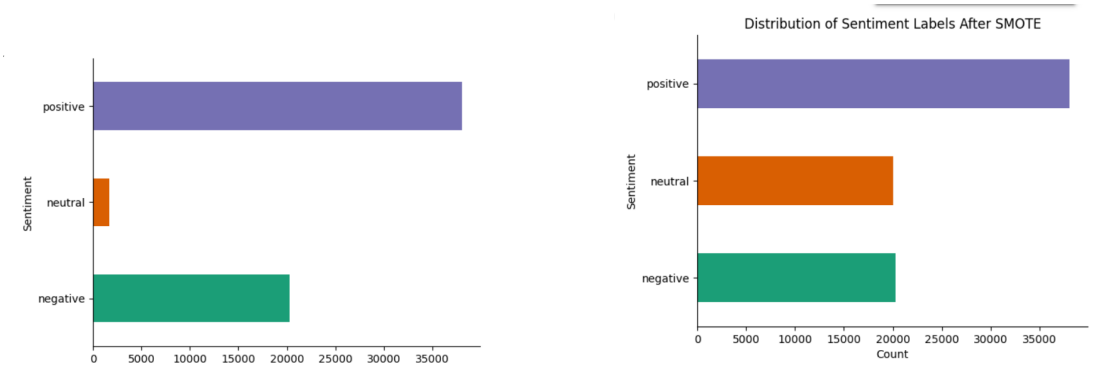
5.2 Discussion

- Challenges: One of the main challenges encountered in this project was dealing with the imbalanced dataset, particularly the underrepresentation of the neutral sentiment class. Although oversampling techniques like SMOTE were applied to balance the classes, handling such imbalance can still affect model performance and generalization. Additionally, training time for some models, especially Random Forest, was longer compared to simpler models like Logistic Regression and LinearSVC.
- Strengths and Weaknesses of Each Model: Random Forest showed the highest accuracy (94) and excellent precision, recall, and F1-scores across all sentiment classes. Its ensemble nature allows it to handle noisy data and complex patterns well. However, it requires more computational resources and longer training time.

Logistic Regression performed moderately well with an average cross-validation accuracy of approximately 83. It is simple, fast, and interpretable but may struggle with capturing complex relationships in the data.

LinearSVC achieved a slightly lower average accuracy (82.5) compared to Logistic Regression. It is efficient for high-dimensional data and text classification but might be sensitive to parameter tuning and data.

6 Tables, Figures, and Code Listings



(a) Distribution of Sentiment Classes (Before)

(b) Distribution of Sentiment Classes (After)

Figure 1: Comparison of Sentiment Class Distributions Before and After Balancing

6.1 Code Listings

```
from sklearn.model_selection import StratifiedKFold,
    cross_val_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,
    confusion_matrix
from sklearn.model_selection import cross_val_predict
import seaborn as sns
import matplotlib.pyplot as plt

model = RandomForestClassifier(n_estimators=100,
    random_state=42)
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state
    =42)
y_pred = cross_val_predict(model, X_resampled, y_resampled,
    cv=skf)
```

```
# Evaluate
print("Classification Report:")
print(classification_report(y_resampled, y_pred,
    target_names=['negative', 'neutral', 'positive']))

# Confusion matrix
cm = confusion_matrix(y_resampled, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
    xticklabels=['neg', 'neu', 'pos'], yticklabels=['neg', '
    neu', 'pos'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix (Cross-Validation)')
plt.show()
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
import numpy as np

# Initialize the SVM model
svc_model = LinearSVC(random_state=42)

# Perform 5-fold cross-validation
scores = cross_val_score(svc_model, X_resampled, y_resampled
    , cv=5, scoring='accuracy')

# Print cross-validation accuracy scores
print("Cross-validation accuracy scores:", scores)
print("Mean accuracy:", np.mean(scores))
from sklearn.metrics import confusion_matrix,
    ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns
fold = 1
for train_index, test_index in skf.split(X_resampled,
    y_resampled):
     X_train, X_test = X_resampled[train_index], X_resampled[
    test_index]
     y_train, y_test = y_resampled[train_index], y_resampled[
    test_index]

     model.fit(X_train, y_train)
     y_pred = model.predict(X_test)
```

```
    cm = confusion_matrix(y_test, y_pred)
    labels = ['negative', 'neutral', 'positive']
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=labels, yticklabels=labels)
    plt.title(f'Confusion Matrix - Fold {fold}')
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.tight_layout()
    plt.show()
    fold += 1
```

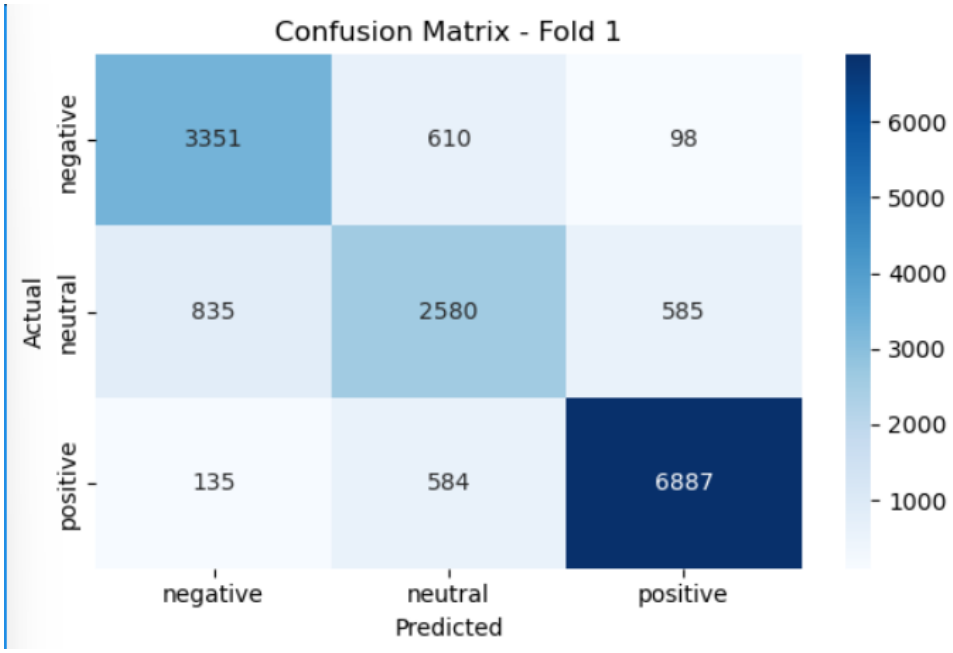


Figure 2: Confusion Matrix for Random Forest

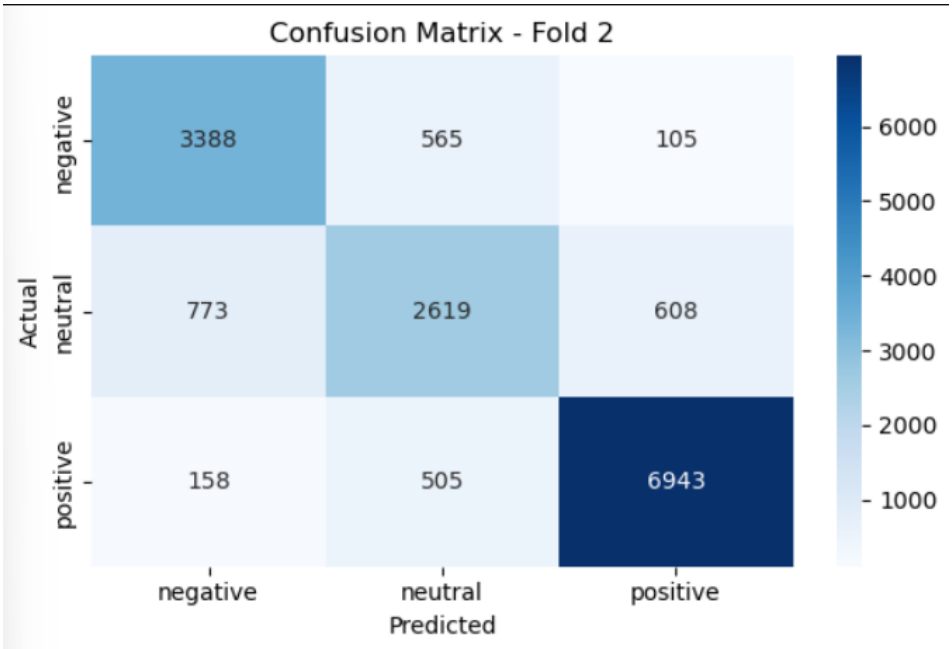


Figure 3: Confusion Matrix for Logistic Regression

Review Vectors

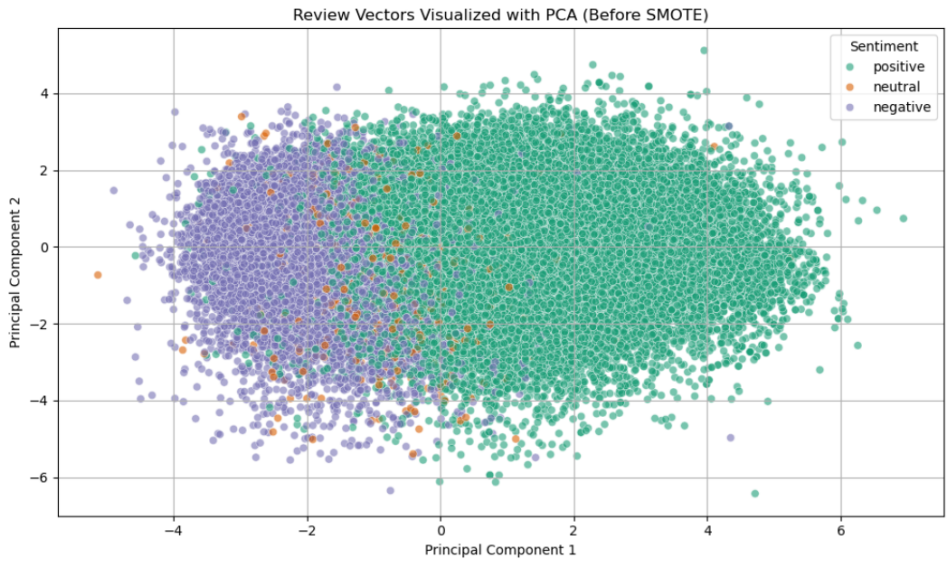


Figure 4: Before SMOTE

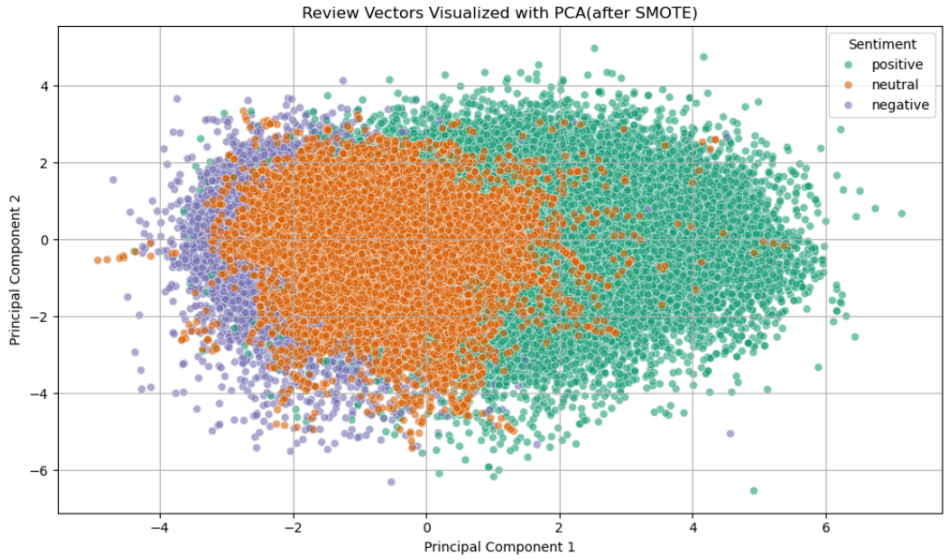


Figure 5: after SMOTE

6.2 Tables

Table 1: Model Performance Comparison

Model	Accuracy	Precision	Recall	F1-Score
RandomForestClassifier	0.94	0.93	0.93	0.93
LogisticRegression	0.8338	0.8090	0.8125	0.8106
LinearSVC	0.83	0.80	0.80	0.80

7 Conclusion

This project focused on sentiment analysis of Amazon product reviews using three machine learning models: Random Forest, Logistic Regression, and LinearSVC. The data was preprocessed by cleaning text, removing irrelevant columns, encoding sentiment labels, and addressing class imbalance using oversampling.

Random Forest delivered the best performance with 94 accuracy and strong precision, recall, and F1-scores across all classes. Logistic Regression and LinearSVC followed with accuracies of 83.38 and 83 respectively. Evaluation was done using confusion matrices and cross-validation.

While all models showed solid results, Random Forest proved most effective for this dataset. The findings align with existing research that supports the strength of classical models when combined with proper preprocessing. Future improvements could include testing deep learning models or applying the approach to other types of review data. Overall, this work supports findings in existing literature and demonstrates a practical

pipeline for sentiment classification. For future work, extending the model to multilingual datasets, integrating deep learning methods like LSTM or BERT, or deploying the model in real-time applications could further enhance its value and impact.

References

Liaw, A., & Wiener, M. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Mohammad, S. M. (2016). Sentiment analysis: Detecting valence, emotions, and other affectual states from text. *Emotion Measurement*, 201–237.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, *2*(1–2), 1–135.

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, 79–86.

Wankhade, M., Rao, A. C. S., & Kulkarni, C. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, *55*(7), 5731–5780.

Zhang, X., Zhao, J., & LeCun, Y. (2018). Random forest based sentiment analysis of online reviews. *2018 International Conference on Big Data and Smart Computing (BigComp)*, 435–438.

[Wankhade et al., 2022] [Mohammad, 2016] [Zhang et al., 2018] [Liaw and Wiener, 2002] [Pang and Lee, 2008] [Pang et al., 2002]