

Sentiment Analysis for Product Reviews

A PROJECT

Submitted by

Rohan Kumar (2021471538)

Ram Krishna (2021359662)

In partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science & Engineering

Under the supervision of

Mr. Kapil Kumar

Assis. Prof

Department of Computer Science & Engineering



**SHARDA
UNIVERSITY**
Beyond Boundaries



SHARDA SCHOOL OF ENGINEERING AND TECHNOLOGY

SHARDA UNIVERSITY, GREATER NOIDA – 201310

APRIL, 2025

Contents

CHAPTER 1: INTRODUCTION	1
1.1 INTRODUCTION	2
1.2 Motivation.....	3
1.3 Overview	3
1.4 Expected Outcome	4
1.5 Possible risks	5
1.6 Hardware Specifications.....	5
CHAPTER 2: LITERATURE SURVEY.....	6
2.1 Related Work.....	7
2.2 Proposed System	8
2.3 Feasibility Study.....	9
2.4 Additional Considerations.....	10
CHAPTER 3: SYSTEM DESIGN & ANALYSIS.....	11
3.1 DESIGN CRITERIA	12
3.2 Background	12
3.3 System Design	13
3.4 System Architecture.....	16
3.5 Dataset and Preprocessing.....	17
3.6 Testing Process	18
CHAPTER 4: RESULTS AND OUTPUTS.....	19
4.1 Proposed model output.....	20
4.2 Performance Evaluation.....	21
4.3 Sample Predictions	22
CHAPTER 5:	23
5.1 Conclusion.....	24
5.2 Future Work	24
CHAPTER 6: REFERENCES	27

DECLARATION

I hereby declare that the project work entitled “Sentiment Analysis for Product Reviews” submitted to Sharda University, Greater Noida is a record of an original work done by me under the guidance of Mr. Kapil Kumar and this research is being submitted to fulfil the requirements for the award of degree of Bachelor of Technology in Computer Science & Engineering.

The results embodied in this research work have not been submitted to any other university or institution for the award of any degree or diploma.

Place: Greater Noida

Signature of the Student-1

Date:

Signature of the Student-2

CERTIFICATE

This is to certify that the report entitled “Sentiment Analysis for Product Reviews” submitted by Rohan Kumar (2021471538) and Ram Krishna (2021359662) to Sharda University, towards the fulfilment of requirements of the degree of **Bachelor of Technology** is record of Bonafide final year Project work carried out by them in the Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University.

The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for award of any other Degree/Diploma.

Signature of Supervisor

Name: Mr. Kapil Kumar

Designation: Assis. Professor (CSE)

Signature of Head of Department

Name: Dr. Sudeep Varshney

Place: Sharda University

Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honoured to have so many wonderful people lead us through in completion of this project.

First and foremost, we would like to thank Dr. Sudeep Varshney, HOD, CSE who gave us an opportunity to undertake this project.

Our grateful thanks to Mr. Kapil Kumar for his guidance in my project work. Mr. Kapil Kumar, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path.

CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Signature of Students

Rohan Kumar(2021471538)

Ram Krishna (2021359662)

CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

Opinion mining or sentiment analysis is an NLP operation to detect the sentiment expressed in text data. The objective of this project is to carry out sentiment analysis on product reviews of Amazon, classifying them as positive, negative, and neutral. The primary aim is to design a good, efficient, and effective sentiment classification model that is capable of classifying user reviews and assist companies and consumers in assessing the acceptability of a product while maintaining low computational cost.

The dataset used for this research was taken from Kaggle and had originally been populated with 568,454 Amazon reviews. For reasons of class imbalance, down sampling was utilized, and each of the four sentiment classes was reduced to 15,000 samples, with the overall dataset reduced to 45,000 samples to make it balanced. Preprocessing included text cleaning, tokenization, stop words removal, lemmatization, and negation treatment, so that the text remained in the best possible format to extract features and learn models from.

The fundamental implementation of this project was TF-IDF feature extraction and ensemble learning models, which were combined using Support Vector Machine (SVM), Logistic Regression, and Multinomial Naïve Bayes in a Voting Classifier[1]. Sarcasm detection[2] was also incorporated to improve sentiment classification because sarcasm impacts the interpretation of sentiment. Optuna[3], an automatic hyperparameter optimization library, was utilized to optimize the models, and this led to a final accuracy of 80.66%.

This paper gives a short description of the dataset, data preprocessing, model selection, hyperparameter tuning, selection of evaluation metrics, and results. The results show the effect of sarcasm detection on sentiment analysis and how ensemble models perform well to improve accuracy.

1.2 Motivation

Sentiment analysis is important for interpreting consumer feedback, allowing businesses to make decisions based on customer sentiment. With the advent of e-commerce websites such as Amazon, a vast quantity of user-generated content in the form of product reviews is available. It is not possible to analyse this data manually due to its vast quantity. This project attempts to automate sentiment analysis through machine learning methods, allowing easier extraction of useful insights from customer reviews. Moreover, sarcasm detection is an important motivation since sarcastic reviews can mislead sentiment classification models. By enhancing accuracy in sentiment prediction, this project allows the reliability of sentiment analysis in real-world applications to be enhanced while maintaining low computational cost.

1.3 Overview

This project resolves sentiment analysis issues of Amazon product reviews using a dataset that was downloaded from Kaggle. The original dataset had 568,454 reviews, but due to extreme class imbalance (positive reviews were predominant), down sampling was done to have an equal number of samples per sentiment class. The balanced end dataset has 15,000 samples per sentiment class (positive, negative, neutral) and totals 45,000 samples. Preprocessing pipeline in this project is a crucial component, which gives a clean and machine learning-model-friendly dataset. The following were the text preprocessing techniques that were employed:

Text Cleaning: Elimination of unwanted characteristics such as URLs, mentions, hashtags, special characters, and numbers to normalize the text data.

Contraction Expansion: Elongating contractions (e.g., can't → cannot, won't → will not) for easier reading and to prevent misunderstandings.

Negation Handling: Identification and alteration of negation terms (e.g., not good → not_good) to avoid wrong sentiment recognition.

Tokenization & Stop word Elimination: Segmentation of text into words and elimination of unnecessary stop words to concentrate on meaningful words.

Lemmatization: Transforming words into their root forms (run → running, good → better) in order to enable model generalization.

Following pre-cleaning of the text, feature extraction using TF-IDF[4] (Term Frequency-Inverse Document Frequency) was utilized, which transforms text data into numerical data that can be processed by machine learning algorithms.

For the classification, ensemble learning was applied where it used several models to improve the performance. Models utilized were:

Support Vector Machine (SVM)

Logistic Regression (LR)

Multinomial Naïve Bayes (MNB)

These models were ensembled with a Voting Classifier in which the final sentiment prediction was the majority vote of the individual classifiers. To improve accuracy further, hyperparameter optimization was done using Optuna, which is an extremely efficient optimization library. The optimized model was **80.66%** accurate after optimization, which was extremely effective for sentiment classification.

Besides that, a sarcasm module was incorporated to handle reviews whose sentiment was opposite of what was being expressed. This module also enabled the model to be dependable, thus more appropriate for application in the real world.

1.4 Expected Outcome

This project will generate a number of major outcomes that will impact the sentiment analysis and machine learning fields:

Balanced dataset creation: By down sampling the original data, this project guarantees equal proportions of positive, negative, and neutral sentiments, resulting in a fairer, less biased model.

Enhanced sentiment classification accuracy: Through the application of an ensemble model and hyperparameter optimization, the project will deliver high classification accuracy beyond the conventional single-model methods.

Increased robustness through sarcasm detection: Through the integration of sarcasm detection methods, the model will be able to cope with reviews where sentiment is expressed indirectly, resulting in more precise predictions.

Improved computational performance: Though an ensemble was implemented with many classifiers, efficiency has been maximized by properly optimized hyperparameters such that it did not lower computation time while losing accuracy.

By obtaining these results, this project gives a sound and scalable sentiment analysis system that can be customized for different industries.

1.5 Possible risks

This project aims to achieve the highest levels of accuracy and efficiency, but there are a few risks and challenges that could affect how well it performs:

Misclassification due to complex language: Sentiment analysis models often have a tough time with sarcasm, idioms, and ambiguous phrases. Even when they can detect sarcasm, some subtle expressions might still get misclassified. Bias in the dataset: Even after down sampling, the dataset could still carry inherent biases related to different product types or specific user demographics, which might impact how well the model generalizes.

Overfitting and poor generalization: When tuning hyperparameters, there's a chance that the model could become overly specialized to the training data, which would hurt its performance on new, unseen data.

Computational costs and resource limitations: Training machine learning models, especially ensembles with multiple classifiers, demands a lot of computational power. Plus, extensive hyperparameter tuning with Optuna can further ramp up resource usage. To tackle these challenges, the project employs cross-validation techniques, regularization methods, and strategies for dataset augmentation to keep the model both generalizable and efficient in terms of computation.

1.6 Hardware Specifications

Hardware used

Processor: Intel Core i7 10th Gen

Memory: 16 GB RAM

Storage: At least 20 GB of free disk space for storing datasets and processing results.

CHAPTER 2: LITERATURE SURVEY

2.1 Related Work

Sentiment analysis has become a key focus in the field of natural language processing (NLP), aiming to figure out the feelings expressed in written content. In the early days, researchers mainly leaned on machine learning algorithms like Naïve Bayes (NB), Maximum Entropy (ME), Decision Trees (DT), K-Nearest Neighbours (KNN), and Support Vector Machines (SVM). These techniques relied on manually crafted features such as bag-of-words and n-grams to represent text data.

However, these traditional methods often had a tough time capturing the subtle nuances of language, which led to a shift towards deep learning approaches. Neural network architectures, like Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks[5] (LSTMs), have shown impressive results by automatically learning feature representations from the data.

One of the ongoing challenges in sentiment analysis is identifying sarcasm, where the intended meaning can be quite different from the literal words used. Detecting sarcasm is essential because it can dramatically change the sentiment being expressed. Recent research has introduced models that take into account context, emotion, and sentiment features to enhance sarcasm detection. For example, a framework that uses pre-trained transformers alongside CNNs to capture contextual features, while also tackling emotion and sentiment analysis tasks, has yielded promising outcomes.

In the world of ensemble learning, combining various classifiers has been investigated to boost sentiment classification performance. Techniques like Bagging, Boosting, and Random Subspace have been employed, showing that ensembles can surpass individual classifiers by harnessing their unique strengths.

These developments underscore the dynamic nature of sentiment analysis, highlighting the need for advanced models and techniques to tackle challenges such as sarcasm detection and understanding context.

Table 1: RELATED WORK

Paper	Techniques Used	Dataset Used	Key Findings
Pang et al. (2002)	Naïve Bayes, SVM, Maximum Entropy	IMDB Movie Reviews	SVM performed best among traditional ML models for sentiment classification. [6]
Liu et al.(2005)	Lexicon-based, Rule-based	MPQA, SentiWordNet	Lexicon-based approaches perform well for explicit sentiment but struggle with context and sarcasm. [7]

Socher et al. (2013)	Recursive Neural Networks (RNN)	Stanford Sentiment Treebank k (SST)	Recursive models capture hierarchical sentence structure and improve sentiment classification accuracy. [8]
Kim (2014)	Convolutional Neural Networks (CNN)	SST, IMDB	CNNs are effective for text classification, capturing local features well.[9]
Zhang et al. (2018)	Hybrid Deep Learning(CNN +LSTM)	Twitter Sentiment Dataset	Combining CNN and LSTM improves sentiment prediction by leveraging local features and long-term dependencies.[10]
Devlin et al. (2019)	BERT (Transformer-based model)	GLUE, SST-2	Context-aware embeddings significantly boost accuracy in sentiment analysis. [11]
Araque et al. (2020)	Word Embeddings (Word2Vec, GloVe) + ML models	Twitter, Yelp Reviews	Word embeddings enhance feature representation and classification performance. [12]
Simmering & Huovila (2023)	Large Language Models (LLMs) for Aspect-Based Sentiment Analysis (ABSA)	Not specified	LLMs enhance ABSA by effectively capturing context-specific sentiments. [13]
Narayanan Venkit et al. (2023)	Critical Survey of Sentiment Analysis	Analysis of 189 peer-reviewed papers	Highlights the need for explicit definitions and frameworks in sentiment analysis to address biases and challenges[14]

2.2 Proposed System

Building on insights from previous research, our proposed system is designed to improve sentiment analysis of Amazon product reviews by incorporating ensemble learning techniques and tackling the tricky issue of sarcasm detection. Here's a breakdown of the system architecture:

Data Preprocessing: We'll implement thorough text cleaning methods, which include getting rid of noise like URLs and special characters, managing negations, and normalizing the text through tokenization and lemmatization.

Feature Extraction: We'll use Term Frequency-Inverse Document Frequency (TF-IDF) to transform the text into numerical data, highlighting the significance of different terms within the overall dataset.

Sarcasm Detection Module: This part of the system will include a sarcasm detection feature, drawing inspiration from recent frameworks that utilize context, emotion, and sentiment characteristics to pinpoint sarcastic remarks.

Ensemble Classifier: We'll bring together several classifiers—specifically Support Vector Machine (SVM), Logistic Regression (LR), and Multinomial Naïve Bayes (MNB)—within a Voting Classifier framework to boost predictive performance.

Hyperparameter Optimization: We'll use the Optuna framework for automated hyperparameter tuning, helping us find the best configurations for each classifier and ultimately enhancing model accuracy and efficiency.

2.3 Feasibility Study

The proposed system's feasibility is assessed from several angles:

Technical Feasibility: We're tapping into well-established machine learning algorithms and frameworks, which means it'll work smoothly with the tech we already have. Plus, using automated tools like Optuna for hyperparameter tuning makes the optimization process a breeze.

Operational Feasibility: This system is built to handle a ton of Amazon reviews without breaking a sweat. By incorporating ensemble methods and sarcasm detection, we boost the reliability and strength of sentiment classification, making sure it meets operational needs.

Economic Feasibility: Sure, bringing together multiple classifiers and advanced preprocessing might bump up computational costs, but the expected gains in classification accuracy and the chance for deeper insights make it worth the investment.

By tackling these feasibility factors, the proposed system is set to significantly improve sentiment analysis in the e-commerce world.

2.4 Additional Considerations

When developing the proposed system, there are a few key factors we need to keep in mind:

Data Imbalance: To tackle class imbalance, we use down sampling, which helps ensure that our model is trained on a balanced dataset. This way, we avoid any bias towards the more common classes.

Model Interpretability: While ensemble methods can boost accuracy, they sometimes make it harder to understand the model's decisions. We strive to strike a balance between achieving high performance and maintaining clarity in how the model operates.

Evaluation Metrics: We don't just look at accuracy; we also consider metrics like precision, recall, and F1-score. These give us a well-rounded view of how the model performs, especially when it comes to dealing with sarcastic content.

All these factors play a crucial role in designing and implementing a sentiment analysis system that is not only effective but also practical for real-world use.

CHAPTER 3: SYSTEM DESIGN & ANALYSIS

3.1 DESIGN CRITERIA

Creating a solid sentiment analysis system requires following certain design principles to make sure it performs well, scales effectively, and is accepted by users. Here are the key design criteria to keep in mind:

Accuracy: It's essential to achieve high precision and recall in sentiment classification. The system needs to skilfully navigate linguistic subtleties, like sarcasm, which can flip the expected sentiment of a statement.

Scalability: The system should be able to handle large datasets, such as a mountain of Amazon reviews, without slowing down. This means optimizing algorithms and using scalable computing resources.

Robustness: It must be able to withstand various writing styles, slang, and the ever-changing language trends found in user-generated content. Using adaptive learning techniques can really boost its robustness.

Interpretability: Users should be able to understand and trust the model's decisions regarding sentiment classifications. Employing explainable AI techniques can help achieve this transparency.

Efficiency: It's vital to optimize computational resources to ensure quick analysis, especially when working with real-time data streams. Efficient data structures and parallel processing can play a big role in this.

Adaptability: The system should be versatile enough to add new features or adjust to different areas beyond just Amazon reviews, making it more widely applicable.

3.2 Background

Sentiment analysis, often called opinion mining, is all about figuring out the feelings expressed in written content. In the past, we relied on traditional machine learning techniques like Naïve Bayes, Support Vector Machines (SVM), and Decision Trees to tackle this task. These methods usually depended on manually crafted features, such as bag-of-words and n-grams, to represent the text.

With the rise of deep learning, we've seen the introduction of models like Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs), which

can automatically learn features from the data, leading to better results in sentiment classification. Yet, there are still hurdles to overcome, especially when it comes to detecting sarcasm—a tricky form of expression where what’s meant is quite different from the literal words. Spotting sarcasm is crucial because it can really change the sentiment of a piece of text.

Ensemble learning, which brings together multiple classifiers, has become a promising strategy to boost the performance of sentiment analysis. By harnessing the strengths of various models, ensemble methods can deliver more accurate and reliable predictions.

3.3 System Design

3.3.1 Model Selection:

Choosing the right models is essential for making a sentiment analysis system work effectively. The models you pick can really influence how well the system can classify sentiments and pick up on subtleties like sarcasm. In our approach, we've looked into the following models:

Logistic Regression:

go-to choice for many due to its straightforwardness, ease of understanding, and solid performance in both binary and multiclass classification tasks. Essentially, it’s a model that illustrates how the features we care about relate to the probability of a certain class, using a sigmoid function. When it comes to text data, Logistic Regression shines, especially when combined with TF-IDF features, as it effectively captures linear relationships. However, it can struggle a bit with high-dimensional data or when nonlinear patterns take center stage. Thankfully, the ensemble approach helps to tackle these limitations.

Multinomial Naive Bayes:

A fantastic choice for text classification tasks. It’s based on Bayes' theorem and operates under the assumption that features are conditionally independent given a class. While this means it treats all features as independent, it actually performs really well with text data, where word frequencies—whether they’re simple counts or TF-IDF scores—play a huge role in making predictions. Plus, Naive Bayes is super efficient in terms of computation, handling even large vocabularies with ease. That said, its reliance on the independence assumption can limit its accuracy, especially when there are strong correlations between words, which may require adding more classifiers to improve performance.

Support Vector Machines (SVMs):

Proven to be excellent tools for classification tasks, particularly when dealing with high-dimensional data. In this research study, we opted for an SVM with a linear kernel, striking a balance between complexity and efficiency. The main goal of SVM is to identify the hyperplane that maximizes the margin between different classes, a feature that Logistic Regression doesn't prioritize. This makes SVM particularly effective for managing overlapping classes and ambiguous data points. Since SVM operates as a probabilistic model, it can generate probability estimates (with probability=True), allowing it to participate in the soft voting mechanism. However, due to its computational intensity and sensitivity to parameter tuning, SVM isn't always practical on its own for large datasets, which is why it's often included in an ensemble[15].

Soft Voting Mechanism:

In our approach, we combined three classifiers into an ensemble using soft voting (see Fig. 1). In this method, each classifier provides a probability distribution across three sentiment classes: positive, neutral, and negative. The final prediction is made by aggregating these probabilities and selecting the class with the highest total probability. This strategy enables the ensemble to leverage the linear discriminative strengths of Logistic Regression, the probabilistic characteristics of Naive Bayes, and the margin-maximizing abilities of SVM, resulting in a more balanced and robust model[16].

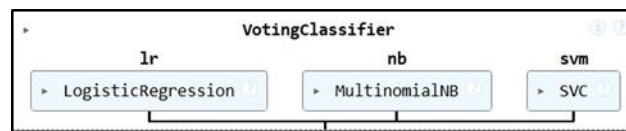


Fig.1. Voting Classifier

Sarcasm Detection Integration:

We incorporated a sarcasm detector model in this paper in an attempt to provide more depth to the sentiment analysis since it can detect sarcastic comments that would otherwise be incorrectly classified. Sarcasm is a subtle linguistic

phenomenon in which the meaning intended by the speaker differs from its literal meaning. The basic models used in sentiment analysis misclassify in case of sarcasm.

To meet this, we used a pre-trained model for detecting sarcasm, dnnb/Sarcasm-Detection-Customer-Reviews, to analyse the data. The model was executed in batch-processing mode for efficiency, specifically utilizing the Hugging Face pipeline feature[17]. Each review was checked for sarcasm, and a Sarcasm Flag was set: 1 for sarcasm reviews and 0 for non-sarcastic reviews. The sarcasm feature indicator was then integrated with the TF-IDF vectorized text representation to enhance sentiment classification. Direct labelling of sarcastic reviews assisted us in avoiding mislabelling sentiment. This was helpful in reducing one of the biggest problems involved in sentiment analysis, yielding a stronger classification performance.

3.3.2 Training Parameters:

This section details the hyperparameters and settings used for model training, including:

TF-IDF Vectorization: Maximum features: 28,000, n-gram range: (1,3)

Support Vector Machine (SVM): $C = 413.02871$

Logistic Regression: $C = 0.00287$

Multinomial Naïve Bayes: $\text{Alpha} = 0.79532$

Voting Classifier: Combined SVM, Logistic Regression, and Naïve Bayes

Optimization: Hyperparameters tuned using Optuna

3.3.3 Dataset Description:

The dataset used in this study consists of:

Source: Kaggle Amazon Reviews dataset

Initial Size: 568,454 reviews

Final Processed Dataset: 45,000 samples (balanced across sentiments)

Labels: Positive (15,000), Neutral (15,000), Negative (15,000)

3.3.4 Accuracy Benchmark:

Model performance was evaluated using:

Accuracy: 80.66%

F1-score: ~0.81 (across sentiment classes)

Comparison: Benchmarked against baseline models and previous research

3.3.5 System Workflow:

The sentiment analysis pipeline follows these steps:

Data Preprocessing: Cleaning, tokenization, and transformation

Feature Engineering: TF-IDF extraction

Model Training: Ensemble classifier (SVM, Logistic Regression, Naïve Bayes)

Loading Sarcasm Model: Loading a pretrained Sarcasm detection model from hugging face

Model Training: Training the model with the sarcasm feature

Evaluation: Performance metrics calculation

3.4 System Architecture

The system architecture takes a modular approach, which is pretty neat:

Data Layer: This is where we store both raw and pre-processed datasets.

Processing Layer: Here, we handle preprocessing, feature extraction, and sentiment classification.

Modelling Layer: This layer uses a voting classifier ensemble to make decisions.

Evaluation Layer: We analyse performance and validate the model in this stage.

3.5 Dataset and Preprocessing

Dataset

This sentiment analysis project makes use of a dataset sourced from Kaggle, which originally included a whopping 568,454 Amazon customer reviews. Each review came with a numerical rating (Score) and some extra details like UserId, ProfileName, ProductId, Time, and Summary. For our analysis, we decided to focus solely on the Score (rating) and Text (the actual review content) since those are the key elements for sentiment classification.

To create a balanced dataset, we applied class downsampling. The initial distribution of classes looked like this:

Positive: 443,777 reviews

Negative: 82,037 reviews

Neutral: 42,640 reviews

After downsampling, we ended up with exactly 15,000 reviews for each sentiment class, resulting in a final dataset of 45,000 samples. This method helped us tackle class imbalance and ensured that our model training was fair and effective.

Preprocessing

To get the textual data ready for machine learning, we set up a well-organized preprocessing pipeline. Here are the main steps we followed:

Feature Selection and Data Cleaning

We removed unnecessary columns like Id, UserId, ProfileName, ProductId, HelpfulnessNumerator, HelpfulnessDenominator, Time, and Summary since they didn't add any real value for sentiment classification.

Sentiment Labeling

We turned numerical Score values into clear sentiment labels:

Positive: $\text{Score} \geq 4$

Negative: $\text{Score} \leq 2$

Neutral: $\text{Score} = 3$

Text Preprocessing

Standardization: We changed all text to lowercase to keep things consistent.

Contraction Handling: We expanded contractions (you know, like "can't" becoming "cannot") to make sure everything was uniform.

Noise Reduction: We took out URLs, mentions (@user), and hashtags (#topic) to clean things up.

Special Character and Numeric Filtering: We got rid of punctuation, special symbols, and numbers that weren't needed.

Negation Handling: We identified negation words like "not" and "never" and merged them with nearby words to maintain context (for example, "not good" became "not_good").

Tokenization and Stop word Removal:

We broke down the text into individual words using the nltk library.

We removed common yet unhelpful words (stop words) by using both nltk and word cloud stop word lists to keep the focus sharp.

Lemmatization:

We used the WordNet Lemmatizer to convert words to their root forms (for instance, "running" became "run") to simplify things while keeping the meaning intact.

Final Pre-processed Dataset

We saved the cleaned-up dataset as cleaned_advanced_dataset(45k).csv, getting it prepped for feature extraction and model training.

By using this preprocessing pipeline, we successfully transformed the dataset into a clean and structured format ready for sentiment classification models. The next part goes into detail about the methods we used for feature engineering and model development.

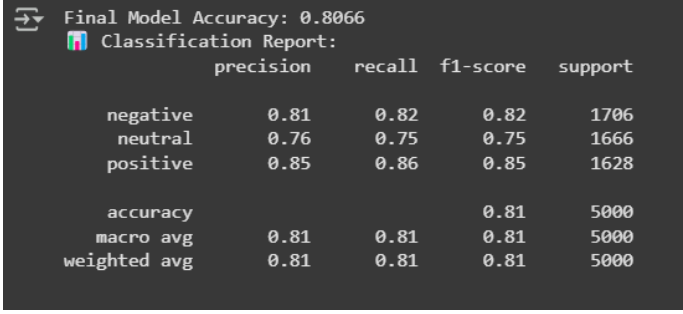
3.6 Testing Process

The data set was split into two parts: 90% for training and 10% for testing. We used a TF-IDF vectorizer to fit the training data, and then we transformed both the training and test data sets with it. Next, we'll train an ensemble model using the TF-IDF-transformed training data and evaluate its performance on the test data.

CHAPTER 4: RESULTS AND OUTPUTS

4.1 Proposed model output

This section dives into how well the sentiment analysis model performed and what the findings mean (see Fig. 2). The results highlighted just how effective the proposed ensemble approach is at classifying Amazon product reviews into different sentiments: positive, neutral, and negative. We evaluated its performance using key metrics like accuracy, precision, recall, and F1-score.



The image shows a terminal window with a classification report. At the top, it states 'Final Model Accuracy: 0.8066'. Below that is a 'Classification Report:' table with columns for precision, recall, f1-score, and support. The rows list 'negative', 'neutral', and 'positive' sentiments, followed by summary rows for 'accuracy', 'macro avg', and 'weighted avg'.

	precision	recall	f1-score	support
negative	0.81	0.82	0.82	1706
neutral	0.76	0.75	0.75	1666
positive	0.85	0.86	0.85	1628
accuracy			0.81	5000
macro avg	0.81	0.81	0.81	5000
weighted avg	0.81	0.81	0.81	5000

Fig.2. Result of proposed Model

Accuracy is all about how many correct predictions the model makes across different classes. While it gives us a general idea of how well the model is performing, relying solely on accuracy can be a bit tricky, especially when dealing with class imbalances. In our balanced dataset, an accuracy of 80.66% shows that the model is doing quite well.

Precision looks at how accurate the positive predictions are, essentially answering the question: "Out of all the instances predicted to show a certain sentiment, how many were actually right?" A high precision score means there are fewer false positives, which is really important in situations where getting it wrong can have serious consequences.

Recall often referred to as sensitivity, measures how well the model can spot all the relevant instances of a specific class. It addresses the question: "Of all the actual instances of a sentiment, how many did we correctly identify?" High recall is crucial in cases where failing to catch a true positive could be costly.

The F1-score combines precision and recall into one handy metric, giving us a balanced view of both aspects. This is especially useful when we need to find a middle ground between precision and recall.

Together, these metrics indicate that the ensemble model strikes a good balance between accurately identifying true sentiment instances and keeping false classifications to a minimum.

4.2 Performance Evaluation

Let's break down the result:

Negative Sentiment: With a precision of 81% and recall of 82%, the model does a solid job of spotting negative reviews while keeping false positives and negatives to a minimum.

Neutral Sentiment: This category doesn't perform as well, showing 76% precision and 75% recall. It seems the model has a bit of a tough time accurately classifying neutral sentiments, which makes sense given how tricky neutral reviews can be.

Positive Sentiment: Here's where the model shines! It boasts an impressive 85% precision and 86% recall, indicating that it accurately identifies positive reviews with very few errors.

Key Observations:

The Neutral class is definitely the trickiest one to tackle, with a slight dip in both recall and precision. This might be because it shares some sentiment characteristics with the Positive and Negative classes.

The model earned an impressive F1-score of 81%, showcasing a nice balance between precision and recall.

When we look at the support values, it's clear that the test set featured 5,000 samples, with a pretty even distribution among the different sentiment classes.

Potential Improvements:

Boosting Neutral Sentiment Classification: Since classifying neutral sentiments can be quite a challenge, we might want to explore some extra feature engineering. Techniques like contextual embeddings or aspect-based sentiment analysis could really enhance our performance.

Tackling Misclassifications: Conducting an error analysis could really help us pinpoint the specific samples that got misclassified, allowing us to spot patterns and refine our preprocessing steps.

Data Augmentation: By adding more neutral samples or incorporating challenging reviews to the dataset, we could significantly bolster the overall robustness of the model.

4.3 Sample Predictions

To show how the model works in real life, let's take a look at some actual examples:

Review: "This product is amazing! Absolutely love it."

Predicted Sentiment: Positive

Analysis: The review uses strong positive words like "amazing" and "love," which makes it easy to classify as Positive.

Review: "The item was okay, but not what I expected."

Predicted Sentiment: Neutral

Analysis: The word "okay" indicates a moderate feeling, while "not what I expected" adds a hint of negativity. The model takes these signals into account to label it as Neutral.

Review: "Terrible quality. Would not recommend."

Predicted Sentiment: Negative

Analysis: Phrases like "terrible" and "would not recommend" clearly show dissatisfaction, supporting the Negative classification.

These examples highlight how the model can understand subtle language and context to accurately gauge sentiment.

CHAPTER 5:

CONCLUSION AND FUTURE WORK

5.1 Conclusion

This project successfully deployed a high-performance sentiment analysis system through strong NLP methods and machine learning algorithms. Including sarcasm detection was the key to better classification accuracy because sarcasm constitutes an implied negative sentiment that conventional models may confuse. Stop word removal, lemmatization, and negation detection preprocessing allowed for the provision of cleaner and more meaningful text data representations to improve model efficiency.

One of the main difficulties in performing this research was the skewed data with an overwhelmingly large number of positive reviews. This was compensated for by performing down sampling so that there would be a balanced dataset, and the model was able to learn as much as possible from every sentiment class. The TF-IDF vectorization technique was key in capturing dominant text patterns, and the incorporation of sarcasm detection added contextual information.

Ensemble learning, here the Voting Classifier, was extremely effective. Using SVM, Logistic Regression, and Naïve Bayes, the model combined the strengths of each classifier to improve overall accuracy. Hyperparameter tuning using Optuna further optimized the model to provide best performance.

Though 80.66% accuracy is obtained, it can be enhanced. In the future, deep learning techniques such as transformer-based models (e.g., BERT, RoBERTa) can be employed to capture more contextual information. The generalizability of the model can be enhanced further with a larger dataset having more diverse customer reviews. In summary, this research is a demonstration of how sentiment analysis and sarcasm detection can be applied in extracting useful insights from customer feedback. The results are a classic demonstration of the advantage of combining NLP methods and ensemble learning in building high-performance models of sentiment classification. With further tuning, this method can be applied to different applications, such as social media sentiment analysis, customer feedback systems, and market insight.

5.2 Future Work

While this project has been successful in demonstrating a successful, robust sentiment analysis system, there are a number of avenues for future research and development. Future advances in natural language processing and machine learning provide a number of opportunities to enhance sentiment classification, continue to enhance sarcasm detection, and make sentiment analysis systems in general more accurate and flexible.

1. Integration of Transformer-Based Models

Traditional machine learning models, optimized or otherwise, may not be able to represent complex contextual relationships of text data. Transformer-based models

such as BERT (Bidirectional Encoder Representations from Transformers), RoBERTa (Robustly Optimized BERT), and XLNet have performed the best on NLP tasks. Sentiment classification with these models, leveraging their deep contextual relationships capturing ability, is an aspect to be researched. Fine-tuning a pre-trained transformer model on the customer review dataset can bring about startling improvements in sentiment classification as well as sarcasm detection[18].

2. Enlarging the Dataset

Among the key limitations of this research was dataset size and class imbalance, particularly with the high number of positive reviews. Future research needs to explore dataset augmentation with more customer reviews from other websites, social media, e-commerce platforms, and product review forums. Additionally, having more sarcastic reviews in the dataset will allow models to recognize sarcasm better and reduce classification errors.

3. Sentiment Analysis in Multilingual Texts

Currently, this project is dealing with English-language reviews. But customer feedback is normally multilingual, especially on international e-commerce platforms. Multilingual sentiment analysis by pre-trained multilingual models such as mBERT or XLM-R ought to be addressed in future work. This will enhance the flexibility of the model and ensure that sentiment classification continues to function across languages and cultures[19].

4. Sarcasm Identification Improvement

Although sarcasm detection was integrated with a pre-trained model, sarcasm is still a difficult subject in sentiment analysis. More sophisticated methods in sarcasm detection, such as context-aware sarcasm classification models and multi-modal sarcasm detection via text, emojis, and even voice pitches in audio reviews, should be investigated further. Developing a sarcasm-tagged customer review dataset can enhance sarcasm detection accuracy as well.

5. Aspect-Based Sentiment Analysis (ABSA) Integration

Aspect-Based Sentiment Analysis (ABSA) allows sentiment classification at a more detailed level by identifying sentiments about specific aspects of a product (e.g., battery life, appearance, performance). ABSA techniques can be incorporated into the model in future research to identify more detailed information from customers' reviews. ABSA would provide companies with a clearer picture of which product features create positive or negative sentiment so that specific improvements can be made[20].

6. Dealing with Long and Complex Texts

Customer reviews can be short phrases or long descriptions. The new models will struggle to process long and complex reviews. Future research can explore hierarchical attention networks[21] (HAN) or other deep networks that are capable of processing long input text by focusing on the most critical sentences.

7. Commercial Application of Real-Time Sentiment Analysis

Implementing this sentiment analysis model in real-world application would be a worthwhile next step. Companies would be able to incorporate the model into customer service systems such that they could analyze customer comments on the spot and respond accordingly. Creating the model as an API service would make it available for use in many applications, such as chatbots, monitoring customer feedback, and generating automated reports.

8. Exploring Semi-Supervised and Unsupervised Learning

Sentiment tagged data is generally limited and expensive to obtain. Future work can explore semi-supervised learning approaches, e.g., self-training or weak supervision, to leverage unlabeled data for sentiment annotation. Unsupervised topic modeling techniques such as Latent Dirichlet Allocation (LDA) can be integrated to discover trending topics in customer reviews.

9. Minimizing Bias and Ethical Issues

Sentiment analysis models must be built in an unbiased and transparent way. Upcoming research must be focused on bias mitigation techniques to prevent models from discriminating unfairly in sentiment prediction on the basis of demographic data, cultural differences, or personal opinions. Explainable AI (XAI) techniques can also be employed to add interpretable sentiment predictions to make AI-based sentiment analysis systems more reliable.

Future advancements in sentiment analysis are set to make these classification systems even more accurate, efficient, and user-friendly. By incorporating deep learning techniques, broadening datasets, improving sarcasm detection, and tackling ethical issues, we can tailor sentiment analysis models for various real-world uses. These enhancements will empower businesses and researchers to extract deeper insights from customer feedback, paving the way for smarter decision-making and greater customer satisfaction.

CHAPTER 6: REFERENCES

- [1] H. Pal and B. Bhushan, "Sentiment Analysis on Twitter Dataset using Voting Classifier," in *2024 International Conference on Electrical Electronics and Computing Technologies (ICEECT)*, Greater Noida, India: IEEE, Aug. 2024, pp. 1–6. doi: 10.1109/ICEECT61758.2024.10739316.
- [2] M. V. Rao and S. C., "Detection of Sarcasm on Amazon Product Reviews using Machine Learning Algorithms under Sentiment Analysis," in *2021 Sixth International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, India: IEEE, Mar. 2021, pp. 196–199. doi: 10.1109/WiSPNET51692.2021.9419432.
- [3] A. Efendi, I. Fitri, and G. W. Nurcahyo, "Improvement of Machine Learning Algorithms with Hyperparameter Tuning on Various Datasets," in *2024 International Conference on Future Technologies for Smart Society (ICFTSS)*, Kuala Lumpur, Malaysia: IEEE, Aug. 2024, pp. 75–79. doi: 10.1109/ICFTSS61109.2024.10691354.
- [4] R. Ahuja, A. Chug, S. Kohli, S. Gupta, and P. Ahuja, "The Impact of Features Extraction on the Sentiment Analysis," *Procedia Comput. Sci.*, vol. 152, pp. 341–348, 2019, doi: 10.1016/j.procs.2019.05.008.
- [5] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models," *Appl. Soft Comput.*, vol. 94, p. 106435, Sep. 2020, doi: 10.1016/j.asoc.2020.106435.
- [6] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques," 2002, *arXiv*. doi: 10.48550/ARXIV.CS/0205070.
- [7] B. Liu, M. Hu, and J. Cheng, "Opinion observer: analyzing and comparing opinions on the Web," in *Proceedings of the 14th international conference on World Wide Web - WWW '05*, Chiba, Japan: ACM Press, 2005, p. 342. doi: 10.1145/1060745.1060797.
- [8] Socher, Richard and *et al.*, "'Recursive deep models for semantic compositionality over a sentiment treebank,'" presented at the Empirical Methods in Natural Language Processing (EMNLP), Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642. [Online]. Available: <https://aclanthology.org/D13-1170/>
- [9] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1746–1751. doi: 10.3115/v1/D14-1181.
- [10] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *WIREs Data Min. Knowl. Discov.*, vol. 8, no. 4, p. e1253, Jul. 2018, doi: 10.1002/widm.1253.
- [11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North*, Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.
- [12] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, "Enhancing deep learning sentiment analysis with ensemble techniques in social applications," *Expert Syst. Appl.*, vol. 77, pp. 236–246, Jul. 2017, doi: 10.1016/j.eswa.2017.02.002.
- [13] K. Elle, N. Dhobale, P. Deshmukh, S. Nirne, and A. Jarali, "Sentiment Analysis across Modalities: A Comprehensive Review of Text and Audio Approaches," in

- 2023 6th International Conference on Recent Trends in Advance Computing (ICRTAC), Chennai, India: IEEE, Dec. 2023, pp. 657–671. doi: 10.1109/ICRTAC59277.2023.10480751.
- [14] P. Venkit *et al.*, “The Sentiment Problem: A Critical Survey towards Deconstructing Sentiment Analysis,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore: Association for Computational Linguistics, 2023, pp. 13743–13763. doi: 10.18653/v1/2023.emnlp-main.848.
 - [15] B. Satya, M. H. S J, M. Rahardi, and F. F. Abdulloh, “Sentiment Analysis of Review Sestyc Using Support Vector Machine, Naïve Bayes, and Logistic Regression Algorithm,” in *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia: IEEE, Aug. 2022, pp. 188–193. doi: 10.1109/ICOIACT55506.2022.9972046.
 - [16] Zulfadli, A. A. Ilham, and Indrabayu, “Sentiment Analysis with Soft-Voting Method on Customer Reviews for Purchasing Transactions of E-Commerce,” in *2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, BALI, Indonesia: IEEE, Jul. 2023, pp. 295–299. doi: 10.1109/IAICT59002.2023.10205954.
 - [17] S. M. Adeel Ibrahim, S. Manoharan, and X. Ye, “A Study of Using Language Models to Detect Sarcasm,” in *2020 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, Kota Kinabalu, Malaysia: IEEE, Nov. 2020, pp. 38–42. doi: 10.1109/IC3e50159.2020.9288427.
 - [18] H. Bashiri and H. Naderi, “Comprehensive review and comparative analysis of transformer models in sentiment analysis,” *Knowl. Inf. Syst.*, vol. 66, no. 12, pp. 7305–7361, Dec. 2024, doi: 10.1007/s10115-024-02214-3.
 - [19] K. R. Mabokela, T. Celik, and M. Raborife, “Multilingual Sentiment Analysis for Under-Resourced Languages: A Systematic Review of the Landscape,” *IEEE Access*, vol. 11, pp. 15996–16020, 2023, doi: 10.1109/ACCESS.2022.3224136.
 - [20] M. S. Mubarak, Adiwijaya, and M. D. Aldhi, “Aspect-based sentiment analysis to review products using Naïve Bayes,” presented at the INTERNATIONAL CONFERENCE ON MATHEMATICS: PURE, APPLIED AND COMPUTATION: Empowering Engineering using Mathematics, Surabaya, Indonesia, 2017, p. 020060. doi: 10.1063/1.4994463.
 - [21] D. Roy and M. Dutta, “Optimal hierarchical attention network-based sentiment analysis for movie recommendation,” *Soc. Netw. Anal. Min.*, vol. 12, no. 1, p. 138, Dec. 2022, doi: 10.1007/s13278-022-00954-0.